AVEVA InTouchHMI

aveva.com

© 2015-2026 by AVEVA Group Limited or its subsidiaries. All rights reserved.

本ガイドのいかなる部分も、AVEVA Group Limited の書面による事前の許可なく、いかなる形式、手段的、機械的、複写、記録、またはその他の手段を含むによっても、複製すること、検索システムに保存すること、および送信することはできません。ここに記載されている情報の使用に関しては、いかなる責任も負いません。

このマニュアルの準備には予防措置が取られていますが、AVEVA はエラーや省略に対して責任を負いません。このマニュアルに記載されている情報は予告なしに変更される場合があり、AVEVA 側の誓約を示すものではありません。このマニュアルに記載されているソフトウェアは、ライセンス契約に基づき提供されます。本ソフトウェアは、そのようなライセンス契約の条件に従ってのみ使用またはコピーできます。AVEVA、AVEVA ロゴおよびロゴタイプ、OSIsoft、OSIsoft ロゴおよびロゴタイプ、ArchestrA、Avantis、Citect、DYNSIM、eDNA、EYESIM、InBatch、InduSoft、InStep、IntelaTrac、InTouch、Managed PI、OASyS、OSIsoft Advanced Services、OSIsoft Cloud Services、OSIsoft Connected Services、OSIsoft EDS、PIPEPHASE、PI ACE、PI Advanced Computing Engine、PI AF SDK、PI API、PI Asset Framework、PI Audit Viewer、PI Builder、PI Cloud Connect、PI Connectors、PI Data Archive、PI DataLink、PI DataLink Server、PI Developers Club、PI Integrator for Business Analytics、PI Interfaces、PI JDBC Driver、PI Manual Logger、PI Notifications、PI ODBC Driver、PI OLEDB Enterprise、PI OLEDB Provider、PI OPC DA Server、PI OPC HDA Server、PI ProcessBook、PI SDK、PI Server、PI Square、PI System、PI System Access、PI Vision、PI Visualization Suite、PI Web API、PI WebParts、PI Web Services、PRiSM、PRO/II、PROVISION、ROMeo、RLINK、RtReports、SIM4ME、SimCentral、SimSci、Skelta、SmartGlance、Spiral Software、WindowMaker、WindowViewer、Wonderware は AVEVA および/または子会社の商標です。その他のすべてのブランドは、それぞれの所有者の商標である可能性があります。

米国政府の権利

米国政府による使用、複製、開示は、AVEVA Group Limited または子会社とのライセンス契約および DFARS 227.7202、DFARS 252.227-7013、FAR 12-212、FAR 52.227-19、またはこれらを継承するものに記載されている制限に準じるものとします。

発行日：Thursday, September 10, 2026

パブリケーション ID：1632353

Contact information（お問い合わせ先）

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

セールスおよびカスタマー トレーニングへの連絡方法については、https://sw.aveva.com/contact を参照してください。

テクニカル サポートへの連絡方法については、https://sw.aveva.com/support を参照してください。

AVEVA のナレッジ ベースとサポート センターには、https://softwaresupport.aveva.com からアクセスできます。

目次

Standard. 22
統合開発環境. 23
InTouch アプリケーション マネージャについて. 23
アプリケーション エクスプローラを使用する. 24
アプリケーション エクスプローラでナビゲートする. 27
アプリケーション エクスプローラにアプリケーションを追加する. 27
ツールバーを管理する. 28
WindowMaker 環境設定を設定する. 28
マウス ショートカット. 32
全画面モードを使用する. 33
［ファイル］メニューからウィンドウを管理する. 33
フォントのデフォルトを設定する. 34
矢印キーを使用してオブジェクトを移動する. 34
グラフィック キャンバスとグラフィック アンカーを使用する. 34
カラー パレットを使用する. 35
カラー パレットを開く. 36
カスタム カラーを作成する. 36
カスタム カラーのインポートとエクスポート. 37
パンとズーム. 38
サムネイル ウィンドウを使用してパンとズームする. 38
マウス ホイールを使用してズームとパンする. 39
パンとズームの制限. 39
画面グリッドとルーラーの使用. 40
グリッドにオブジェクトを配置する. 40
ルーラーを使用する. 40
WindowViewer について. 41
ランタイム環境をカスタマイズする. 41
リモート セッションで実行するアプリケーションへのユーザー アクセスを設定する. 46
WindowViewer のメモリ管理について. 47
WindowViewer ウィンドウを操作する. 56
産業用グラフィック. 57
産業用グラフィックの作成. 58
産業用グラフィック エディタ. 58
アプリケーション. 60
スタンドアロン InTouch アプリケーション. 60
マネージド InTouch アプリケーション. 61
パブリッシュ済み InTouch アプリケーション. 62
InTouchView アプリケーション. 63

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Application Server のアーキテクチャ 64
Galaxy との通信 65
スタンドアロン、マネージド、パブリッシュ済みの InTouch アプリケーションの比較 66
アプリケーションを構築する 67
アプリケーションを実行する 68
タグ 69
InTouch タグを操作する 69
InTouch タグのタイプ 70
メモリ型タグ 71
I/O タグ変数 I/O たぐへんすう 72
間接型タグ 73
その他のタグ 74
システム タグ 75
システム タグ参照 75
タグ プロパティ 80
メモリ型タグのプロパティ 80
I/O 型タグのプロパティ 82
リモート タグ参照 84
間接型タグを定義する 84
間接型タグとスクリプトを使用する 85
間接型タグとローカル タグを使用する 86
間接型タグとリモート参照を使用する 86
再使用可能なタグ構造を定義する 88
スーパータグ テンプレートとメンバー タグを管理する 90
スーパータグ インスタンス 91
スーパータグ属性 97
スーパータグ メンバーを参照する 97
Bulk Import Utility を使用してスーパータグをインポートする 98
InTouch アプリケーションの移行中にアセットをインポートする 98
他のアプリケーションから履歴タグ値にアクセスする 99
DDE アイテムを使用して履歴データを表示する 99
DDE でログ データにアクセスする 102
他のアプリケーションから履歴データにアクセスする 109
HistData エラーのトラブルシューティング 110
IEEE 小数ユニット 112
Historian HMI で浮動小数点数を表示する 112
InTouch OPC UA サーバーを設定して使用する 113
OPC UA の設定とチェックリスト 113
InTouch OPC UA サーバーを設定する 114
OPC UA サービス用のファイアウォールを設定する 116
サードパーティ製 OPC UA クライアント アプリケーション用のサーバーおよびクライアント証明書を設定する 120
OI Gateway を使用してクライアント セキュリティ証明書を設定する 124
アラーム 128
InTouch アラームの概要 128
アラーム優先度 129
アラームのサブステート 129
アラームの確認 130

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム グループ. 130
InTouch イベントの概要. 130
InTouch アラームのタイプ. 131
論理値アラーム. 132
アナログ アラーム. 132
InTouch 分散アラート システム. 133
アラーム プロバイダとコンシューマ. 135
分散アラーム グループ リスト. 136
サマリ アラームと履歴アラーム. 137
アラームの無効化、抑止、非表示. 137
ターミナル サービス、アラームのサポート. 138
分散アラーム システム データの保存. 138

Build. 139

InTouch HMI アプリケーションの管理. 140
アプリケーション マネージャを起動します。. 140
アプリケーション マネージャの使用. 141
InTouch アプリケーションの作成. 142
新規 InTouchView アプリケーションの作成. 144
InTouch アプリケーションから InTouchView アプリケーションへの変更. 144
WindowMaker と WindowViewer でアプリケーションを開く. 145
アプリケーション マネージャ ウィンドウのカスタマイズ. 146
アプリケーション タイルを使用する. 147
InTouch アプリケーションのロックおよびロックを解除する. 148
InTouch アプリケーションの変更. 149
アプリケーション マネージャからの InTouch アプリケーションの削除. 151
InTouch アプリケーションの検索. 152
アプリケーション テンプレートでの作業. 153
InTouch アプリケーション フォルダの表示. 155
テンプレートとして使用するための InTouch アプリケーションのエクスポート. 155
アプリケーション マネージャを使用した Web Client 設定の更新. 156
InTouch OMI ViewApp の起動. 157
AVEVA Identity Manager での登録. 157
資格情報マネージャの操作. 158
名前付き資格情報の管理. 159
名前付き資格情報のエクスポートとインポート. 160

IDE を使用した InTouch アプリケーションの管理. 162
マネージド InTouch アプリケーションの作成. 163
アプリケーション テンプレートからのマネージド アプリケーションの作成. 166
IDE からの WindowMaker の起動. 168
マネージド InTouch アプリケーションの言語の切り替え. 169
InTouch アプリケーションの変更の送信. 170
InTouch アプリケーションのインポート. 170
InTouchViewApp オブジェクトのインポートとエクスポート. 172
タグ データのエクスポートとインポート. 172
タグの値およびパラメータの保持. 173
マネージド InTouch アプリケーションを削除する. 174
InTouchViewApp 内のすべての Galaxy グラフィックの関連付け. 174

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウ 176
アプリケーション ウィンドウの作成 176
アプリケーション ウィンドウをテンプレート ウィンドウとして設定する 179
テンプレート ウィンドウからのアプリケーション ウィンドウの作成 180
枠ウィンドウの操作 181
プロパティ パネルの使用 182
枠ウィンドウに埋め込まれたグラフィックの操作 183
枠ウィンドウのリボンの使用 185
アプリケーション ウィンドウの変更 185
ウィンドウを開く、保存する、閉じる 186
ウィンドウのサムネイル プレビューを表示する 187
ウィンドウの複製 188
ウィンドウの削除 188
クイック アクセス ツールバーの使用 188
InTouch ウィンドウに関する情報の印刷 190
コマンド プロンプトからのウィンドウ情報の印刷 192
ウィンドウを印刷するための .CSV 形式 192
コマンド プロンプトから印刷するための構文 193
コンテキスト メニューの深さの設定 193
グラフィック要素 195
WindowMaker オブジェクト 195
WindowMaker オブジェクト 195
シンプル オブジェクト 196
複合オブジェクト 198
共通の操作 200
すべてのオブジェクトの特殊操作 207
特殊オブジェクトの特殊操作 209
オブジェクトのアニメーション化 213
産業用グラフィック エディタの操作（IT） 251
産業用グラフィックの管理 251
HMI/SCADA アプリケーションと共に使用する産業用グラフィック エディタ ワークフロー 251
シンボル変更の反映 255
シンボルのダイナミック サイズの反映 256
WindowMaker と産業用グラフィック エディタの比較 256
InTouch タグへのアニメーションの接続 266
WindowMaker での産業用グラフィックの使用 268
WindowMaker での産業用グラフィックの使用について 269
InTouch ウィンドウへの産業用グラフィックの埋め込み。 270
埋め込まれた産業用グラフィックのサイズ変更。 275
WindowMaker での産業用グラフィックの設定 276
産業用グラフィック エディタでの産業用グラフィックの編集 286
InTouch タグを使用したグラフィック要素と産業用グラフィックの作成 288
産業用グラフィック エディタの［ツールボックス］タブからのグラフィックの埋め込み. 290
WindowViewer での産業用グラフィックのテスト 291
グラフィック パフォーマンスの予測 293
新しいオートメーション インスタンスの作成 293
新しい Application Server オブジェクト インスタンスの自動作成 293
InTouch ウィンドウから産業用グラフィックへの変換 294

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウの変換の準備 294
ウィンドウの変換ルールと例外 294
アニメーション スクリプトの変換 295
ウィンドウ変換の既知の制限事項 295
ウィンドウを変換した後 296
ウィンドウの変換手順 297
ウィンドウ変換エラーの診断 298
シンボル ウィザード エディタでのシンボル ウィザードの作成 299
シンボル ウィザードの作成 300
シンボル ウィザードのデザイナ ワークフロー 300
シンボル ウィザードのコンシューマ ワークフロー 301
トレンド ペンの履歴データ検索について 301
ランタイム時のトレンド ペン プロパティの変更 303
MinValue プロパティ 303
MaxValue プロパティ 304
StartTime プロパティ StartTime ぷろぱてぃ 304
EndTime プロパティ 305
PlotType プロパティ 305
TimeMode プロパティ 305
FillTrend プロパティ 305
マルチ ペン トレンドを設定するには 306
ペン名と参照の設定 307
ペンの詳細の設定 308
ペン オプションの設定 309
ソースの設定 310
マルチ ペン トレンドのカスタマイズ 310
ランタイム時のマルチ ペン トレンドの使用 311
SmartSymbol について 320
SmartSymbol マネージャおよびライブラリ 320
InTouch SmartSymbol および産業用グラフィック SmartSymbol 321
SmartSymbol の制限 323
SmartSymbol テンプレートとインスタンスの作成 323
SmartSymbol の管理 331
SmartSymbol の編集 336
InTouch SmartSymbol の移行 342
タグ 349
タグ名ディクショナリを使用したタグの管理 349
タグ変数の使用法の計画 350
新しいタグの作成 351
タグ プロパティの設定 352
タグ変数の変更 359
OPC UA サーバーからの InTouch タグの作成 360
ﾀｸﾞ の削除 364
タグ変数リストと使用情報の印刷 364
タグ ドットフィールドを使用したタグ プロパティの表示または変更 364
利用可能なタグ タイプのドットフィールド 365
タグ変数のしきい値の変更 374
I/O を使用したデータ アクセス 382

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Gateway Communications Driver の操作 384
サポートされている InTouch 通信プロトコル 385
アクセス名の設定 394
I/O タグ変数による I/O データへのアクセス 397
タグのリモート 参照への変換 405
リモート 参照による I/O データ アクセス 409
InTouch から Application Server データへのアクセス 412
I/O タグ変数の時間スタンプと品質情報の表示 422
ランタイム時の I/O 接続の初期化と再設定らんたいむじの IO せつぞくのしょきかとさいせってい 433
アクセス名によるフェイルオーバー機能の使用 437
I/O 接続ステータスの監視 443
他のアプリケーションからの InTouch タグ変数データへのアクセス 446
タグ値の記録 447
履歴ログの設定 448
ランタイムでの履歴ログの開始と停止 455
タグ データのトレンド 456
InTouch トレンドのタイプ 456
保存済みタグの値の履歴トレンド表示 457
履歴トレンド オブジェクトの使用 458
履歴トレンド ウィザードの使用 475
スクリプトの使用による履歴トレンド ウィザードのコントロール 480
リアルタイム トレンド オブジェクトの使用 491
ランタイム中のトレンド印刷 494
他の InTouch ノードや Historian Server の履歴タグ値の表示 495
ユーザー定義データ型 500
UDT の概要 500
UDT の仕様 501
手動でのユーザー定義データ型の作成 502
新しいデータ型の作成 502
新しいメンバーの作成 503
新しい派生データ型の作成 504
新しいインスタンスの作成 505
別のデータ型の下でのメンバーの入れ子 506
データ型をインポート 508
データ型のエクスポート 512
ユーザー定義データ型の管理 513
ユーザー定義データ型の一括編集 514
ユーザー定義データ型の表示 515
［ユーザー定義型］ビュー 516
［モデル - タグ名］ビュー 516
プロパティ グリッドでのユーザー定義データ型の編集 517
変更の反映 520
既存の機能の UDT サポート 521
Intellisense 522
アニメーション 523
スクリプト 523
参照の変更 524

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

UDT メンバーの参照. 525
インスタンスの UDT メンバーの産業用グラフィック エディタ キャンバスへのドラッグ アンド ドロップ. 526
アニメーションとスクリプト. 527
アラームおよびアラーム クライアント コントロール. 527
履歴およびトレンド クライアント. 529
I/O タグとしての UDT メンバーの設定. 530
所有オブジェクトとしての UDT インスタンスの使用. 532
Web Client. 533
マップアプリケーション. 534
相互参照. 535
未使用タグの削除. 536
スーパータグとの共存. 537
ライセンス. 538
ランタイム Tag Viewer. 539
UDT の制限事項. 540
アラーム. 541
アラームの設定. 541
アラーム階層の定義. 542
アラーム状況を使用したタグ変数の設定. 545
個々のタグのイベント プロパティの設定. 551
アラームとイベントのグローバル設定. 551
アラーム グループ リスト ファイルの作成. 554
アラーム クエリー. 555
アラーム クエリーの例. 557
詳細な InTouch クエリー情報の取得. 557
アラーム冗長性による工場セキュリティの強化. 558
Hot Backup の理解. 559
Hot Backup ペアを設定する. 560
Hot Backup ペアの例. 570
Hot Backup ペアについての注記. 574
アラーム監査記録の作成. 575
認証を要求するアラームを確認する. 576
分散アラーム表示オブジェクトの操作. 578
分散アラーム表示オブジェクトの操作. 579
分散アラーム表示オブジェクトについて. 579
デザイン時の分散アラーム表示オブジェクトの設定. 579
ランタイムでの分散アラーム表示オブジェクトの使用. 590
関数とドットフィールドを使用した分散アラーム表示オブジェクトの制御. 595
アラーム階層の表示. 636
Alarm Viewer コントロールの設定. 636
ランタイムでの Alarm Tree Viewer コントロールの使用. 644
Alarm Tree Viewer コントロールの ActiveX プロパティの使用. 645
Alarm Tree Viewer コントロールの ActiveX メソッドの使用. 647
メソッドとプロパティを使用するときのエラー処理. 653
Alarm Tree Viewer コントロール ActiveX イベントを使用したスクリプトのトリガ. 653
アラームの印刷. 653
アラーム印刷とログの設定. 654

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Alarm Printer でのアラームの印刷の概要. 665
アラームのファイルへのログ. 666
特定の設定での Alarm Printer の起動. 666
スクリプトを使用した Alarm Printer の制御. 667
アラーム データベースへのアラームの記録. 683
Alarm DB Logger Manager 用の SQL Server アカウント. 684
Alarm DB Logger Manager の使用. 684
アラーム データベースのビュー. 691
アラーム データベースのストアド プロシージャ. 697
レコードされたアラームの表示. 699
Alarm DB View コントロールの設定. 699
ランタイムでの Alarm DB View コントロールの使用. 721
Alarm DB View ActiveX プロパティの使用. 722
Alarm DB View ActiveX メソッドの使用. 746
メソッドとプロパティを使用するときのエラー処理. 751
Alarm DB View ActiveX イベントを使用したスクリプトのトリガ. 751
複数のタグにわたるアラーム分散の分析. 752
Alarm Pareto ActiveX コントロールの設定. 752
ランタイムでの Alarm Pareto コントロールの使用. 763
Alarm Pareto ActiveX プロパティの使用. 764
Alarm Pareto ActiveX メソッドの使用. 766
メソッドとプロパティを使用する際のエラー処理. 770
Alarm Pareto ActiveX イベントを使用したスクリプトのトリガ. 770
スクリプト. 771
基本的なスクリプトの概念. 771
スクリプトのタイプ. 772
高度なスクリプトの概念. 772
OLE オブジェクト OLE おぶじぇくと. 772
ActiveX コントロールを使用したスクリプト. 773
スクリプトの作成と編集. 773
InTouch スクリプト エディタの操作. 773
編集用にスクリプトを開く. 778
スクリプトへの変更の保存または破棄. 779
テキストのコピー、切り取り、および貼り付け. 779
スクリプト内での検索. 779
コード要素の挿入. 786
スクリプト関数のヘルプへのアクセス. 786
正確な構文を確認するためのスクリプトの検証. 786
スクリプトの印刷. 787
スクリプトの削除. 788
スクリプトを表示するズーム オプションの調整. 788
スクリプト トリガ. 788
スクリプト トリガのタイプ. 789
複数のトリガの使用. 789
定期的なスクリプトの実行. 790
アプリケーション スクリプトの設定. 790
ウィンドウ スクリプトの設定. 792
キー スクリプトの設定. 794

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

条件スクリプトの設定 796
データ変化スクリプトの設定 798
アクション スクリプトの設定 799
ActiveX イベント スクリプトの設定 803
ランタイムでのスクリプト実行の一時停止 806
スクリプト言語 806
基本構文規則 807
標準関数の呼び出し 809
カスタム関数（クイック関数）の呼び出し 810
値の割り当てと演算子 811
条件付きプログラム分岐構造の使用 819
プログラム ループの使用 820
ローカル変数の使用 823
カスタム スクリプト関数 824
クイック関数について 824
クイック関数の設定 824
クイック関数の呼び出し 826
非同期クイック関数の作成 826
組み込み関数 827
アニメーション表示リンクでの強制更新 828
数学計算 828
文字列操作 835
データ タイプの変換 844
ランタイムでの InTouch ウィンドウの操作 848
日付と時刻情報の操作 861
他のアプリケーションとの対話 869
ファイルを操作する 876
システム関連情報を取得する 883
InTouch 関連情報を取得する 885
セキュリティ関連スクリプト 887
その他のスクリプト 889
OLE オブジェクトを使用したスクリプト 893
OLE オブジェクトの作成、確認および解放 894
OLE オブジェクトのプロパティとメソッドを使用する 895
同じ OLE オブジェクトに複数のポインタを割り当てる 897
OLE エラーをトラブルシューティングする 897
OLE を使用して行える操作 899
ActiveX コントロールをスクリプト化する 902
ActiveX コントロール メソッドを呼び出す 903
InTouch HMI から ActiveX コントロール プロパティにアクセスする 903
ActiveX イベント スクリプトの作成と再使用 905
ActiveX イベント スクリプトをインポートする 908
ActiveX コントロール 909
ActiveX コントロールを使用する 909
ActiveX コントロールを設定する 911
ActiveX コントロールに名前を付ける 911
ActiveX コントロールの標準操作 912
ActiveX コントロールのインストールとアンインストール 912

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィザード **914**
ウィザードを操作する 914
ウィザードのタイプ 914
ツールバーにウィザードを追加する 915
ウィザード インスタンスを貼り付ける 916
ウィザードを設定する 916
ウィザードの標準操作を実行す r 916
ウィザードのインストールとアンインストール 916
トレンド オブジェクト 918
ウィンドウ コントロール ウィザード 918
ウィンドウ コントロールの作成と設定 919
テキスト ボックスを作成する 919
リスト ボックスを作成する 920
コンボ ボックスを作成する 921
チェックボックスを作成する 922
ラジオ ボタン グループを作成する 923
ウィンドウ コントロールをスクリプト化する 924
コントロールの値の取得または設定 924
ユーザー入力用のコントロールの有効化または無効化 926
ウィンドウ コントロールを動的に非表示にする 927
コンボ ボックスのアイテムの追加と削除 927
ファイルに対するリスト アイテムのロードと保存 931
コンボ ボックスまたはリストのアイテムを検索する 933
コンボ ボックスまたはリストのアイテム インデックスを操作する 934
リスト ボックスまたはコンボ ボックス アイテムをカウントする 937
リスト アイテムの値の取得または設定 938
リスト アイテムの名前を取得する 939
テキスト ボックスの内容をロードする 940
テキスト ボックスが読み取り専用であるかどうかを確認する 942
チェック ボックスのラベルの取得または設定 943
ウィンドウ コントロール エラー メッセージを理解する 943
ウィジェット **945**
カルーセル ウィジェット 945
Web ブラウザ ウィジェット 946
QR コード スキャナ 947
MapApp ウィジェット 948
MapApp の設定: 概要 948
マップ設定 950
画像での MapApp の設定 961
例 962
DataGrid ウィジェット 964
DataGrid ウィジェットの設定 968
DataGrid ウィジェット用のソース データの準備 971
Teamwork ウィジェット 971
XML ファイルを使用してウィンドウをインポートする **973**
XML ファイルを準備する 973
コマンド ファイルを準備する 974
最小コマンド ファイルを作成する 974

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ファイルに印刷情報を送信する. 975
クロスリファレンスをファイルに送信する. 975
ログ ファイルを作成する. 975
コマンドの構文. 975
アプリケーションを作成する. 977
新しく作成されたアプリケーションにタグを追加する. 978
ウィンドウを削除します. 978
ウィンドウの名前を変更する. 978
ウィンドウをインポートします. 979
エラーを処理する. 979
SmartSymbol の欠落. 980
産業用グラフィックの欠落. 980
式、タグ名、およびスクリプト. 980
コマンド プロンプトから WindowMaker を実行する. 980
System Platform IDE 拡張機能. 981
マネージド InTouch アプリケーションから WindowMaker を実行する. 981
コマンド プロンプトから DBDump を実行する. 982
マネージド InTouch アプリケーションに対して DBDump を実行する. 983
コマンド プロンプトから DBLoad を実行する. 983
マネージド InTouch アプリケーションに対して DBLoad を実行する. 984
XML の形式. 984
一般的な XML ファイル形式. 984
共通の要素定義. 985
ウィンドウ要素. 992
サポートされていない InTouch 機能. 1056
CONNECT での産業用グラフィックの操作. 1057
CONNECT にログインする. 1057
共有ドライブ間のナビゲーション. 1058
グラフィックのアップロード/ダウンロード. 1058
CONNECT へのグラフィックのアップロード. 1058
ローカルの可視化フォルダーにグラフィックをダウンロードする. 1059
さまざまなクラウド コンテンツのアップロード/ダウンロードのサポート. 1060
CONNECT でグラフィックを管理する. 1060
複数のユーザーによるグラフィックの管理. 1061
クラウドグラフィックバージョンについて. 1061
クラウドでのグラフィック コンテンツの上書き. 1062
クラウド グラフィックのバージョン管理 - シナリオと例. 1063
ショートカットキー. 1064

Deploy. 1071
ターミナル サービスとリモート デスクトップ サービスの配置と操作. 1072
ターミナル サーバー アプリケーションの計画. 1072
ターミナル サービス環境への InTouch アプリケーションの配置. 1073
ターミナル サービス環境のアラーム. 1073
ターミナル サービス環境のセキュリティ. 1073
ターミナル サービス環境の I/O. 1074
ターミナル サービス環境でのスクリプトの実行. 1074

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch を実行するためのターミナル セッションへの適切なログオン. 1074
ターミナル サービス環境のアラーム クエリの構文. 1074
ターミナル サービス環境のその他の制限. 1075
スクリプトを使用した InTouch クライアント セッションの情報の取得. 1075
TseGetClientId() 関数. 1076
TseGetClientNodeName() 関数. 1076
TseQueryRunningOnConsole() 関数. 1076
TseQueryRunningOnClient() 関数. 1077
リモート デスクトップ サービスの概要. 1077
リモート デスクトップ サービスの役割サービス. 1077
リモート デスクトップ サービス（RDS）接続の保護. 1078
Windows Server 2016 のリモート デスクトップ サービスのベスト プラクティス. 1079
アプリケーションの分散. 1081
サポートされている InTouch アーキテクチャ. 1081
単一のコンピュータのアーキテクチャ. 1081
クライアント ベースのアーキテクチャ. 1081
サーバー ベースのアーキテクチャ. 1082
ネットワーク アプリケーション開発（NAD）. 1082
ネットワーク アプリケーションの計画. 1083
ネットワーク アプリケーションの I/O データ アクセス. 1083
共有ファイルへのアクセス. 1085
分散環境でのデータのログ. 1087
特殊なネットワークの考慮事項. 1091
NAD 用 InTouch アプリケーションの設定. 1091
自動 NAD 更新の実行. 1093
手動による NAD 更新の実行. 1094
アプリケーション編集のロック. 1096
NAD 更新中のアプリケーションの変更. 1096
ランタイム時のアプリケーションの解像度のスケーリング. 1097
アプリケーションの解像度のロック. 1099
リモート ノードへのアプリケーションのパブリッシュ. 1101
発行したファイルのコンテンツ. 1102
スタンドアロン InTouch アプリケーションのパブリッシュ. 1104
マネージド InTouch アプリケーションのパブリッシュ. 1107
マネージド InTouch アプリケーションのパブリッシュ. 1108
Insight にアプリケーションをパブリッシュする. 1109
ランタイム時のマネージド InTouch アプリケーションの使用. 1110
マネージド InTouch アプリケーションを配置する. 1110
初めての InTouchViewApp オブジェクトの配置. 1111
マネージド InTouch アプリケーションへの変更の配置. 1111
マネージド InTouch アプリケーションの起動. 1111
マネージド アプリケーションを配置するときの WindowViewer の再起動待機時間の制御. 1112
オペレータ ノードでの新しいアプリケーション バージョンの受け入れ. 1113
埋め込まれた産業用グラフィックでの ArchestrA スクリプトの実行. 1115
ターミナル サービス環境での InTouchViewApp オブジェクトの配置. 1116
AVEVA Edge での InTouch HMI アプリケーションの使用. 1118
AVEVA Edge zip ファイルのダウンロード. 1118
AVEVA Edge zip ファイルのアップロード. 1118

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

エッジデバイスのユーザーの設定. 1119

Operate. 1121

ランタイム時のアプリケーションの表示の概要. 1122
ランタイム時のアプリケーションの表示の概要. 1122
別のターゲット解像度サイズでのラインタイムでアプリケーションを表示. 1122
InTouch Web Client について. 1123
元のアプリケーション解像度. 1123
ランタイム時のキーボード、マウス、およびタッチジェスチャの操作によるパンとズーム. . . 1124
ランタイムのズーム. 1124
ランタイムに画面を移動. 1126
タッチジェスチャのアニメーションサポート. 1127
枠ウィンドウと ShowGraphic() 関数を使用. 1128
インターネット上での InTouch ウィンドウの実行. 1128
実行時に表示されるウィンドウを設定する. 1129

Web Client でのアプリケーショングラフィックを表示. 1131
Web Client を使用. 1131
InTouch Web Client アプリケーションを設計. 1132
セキュアな InTouch ウェブクライアントアクセス. 1133
InTouch Web Client でのユーザーアクセスの設定. 1137
InTouch Web Client の仮想アカウントの使用. 1138
InTouch Web Client 機能の有効化. 1138
InTouch Web Client のカスタマイズ. 1139
InTouch Web Client で表示するためのグラフィックを設定する. 1140
InTouch Web Client のホームシンボルを設定. 1141
InTouch Web Client ホームアイコンの動作の理解. 1141
Web ブラウザでのアプリケーションの表示. 1142
InTouch Web Client の高速切り替え. 1143
InTouch Web クライアントページの理解. 1143
アプリケーション グラフィックおよびブラウザのサイズ調整. 1146
パンとズームのサポート. 1146
外部 Web サイトでの InTouch アプリケーション グラフィックのホスティング. 1148
iFrame コードブロックを生成. 1149
サポートされているグラフィック要素と既知の制限事項. 1150
既知の制限事項. 1150
すべてのサポート対象グラフィック要素でサポートされるプロパティ. 1151
すべてのグラフィック要素でサポートされるプロパティ（いくつかの制限あり）. 1151
サポートされているグラフィック要素と追加のプロパティ. 1153
サポートされるアニメーション. 1157
Web Client モバイルアプリ. 1159
オペレーティングシステムの要件. 1160
モバイルアプリケーションにログインする. 1160
InTouch Web Client モバイルアプリケーションを使用する. 1160

ランタイム時の言語の切り替え. 1162
ランタイム時の言語切り替えについて. 1162
ランタイム時の言語の切り替えのための言語の設定. 1162
設定言語のフォント設定を変更. 1163

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイム時の言語の切り替え機能の追加. 1164
SwitchDisplayLanguage() 関数. 1166
$Language システム タグ変数. 1167
オフライン翻訳用にアプリケーションテキストをエクスポート. 1167
既存のディクショナリ ファイルへのテキストのエクスポート. 1169
エクスポートされたディクショナリファイルを翻訳. 1169
翻訳されたディクショナリファイルをインポート. 1170
アラームのためのランタイム時の言語切り替え. 1171
翻訳のためのアラームコメントのエクスポート. 1171
既存のアラーム コメント ファイルへのエクスポート. 1174
翻訳されたアラーム コメントをインポート. 1176
翻訳のためのアラーム グループ名のエクスポート. 1177
翻訳されたアラーム グループ名のインポート. 1178
ランタイム時の言語の切り替え機能のテスト. 1178
ネットワーク アプリケーション開発（NAD）クライアントへのローカライズされたファイルの配布 1179
タグ. 1180
Tag Viewer を開始. 1180
Tag Viewer の有効化. 1180
Tag Viewer の起動. 1182
Tag Viewer でのナビゲート. 1183
Tag Viewer を閉じる. 1184
Tag Viewer の使用. 1184
タグを検索. 1184
クイック検索を行う. 1186
タグの管理. 1186
監視ウィンドウの管理. 1189
タグの使用数の削減. 1191
InTouch と Historian. 1192
タグ変数の使用数の確認. 1192
未使用タグの削除. 1202
アラーム. 1204
現在のアラームの表示. 1204
Alarm Viewer コントロールの設定. 1205
ランタイムでの Alarm Viewer コントロールの使用. 1218
Alarm Viewer コントロールの ActiveX プロパティの使用. 1221
Alarm Viewer コントロールの ActiveX メソッドの使用. 1228
メソッドとプロパティを使用するときのエラー処理. 1242
ActiveX イベントを使用したスクリプトのトリガ. 1242
リアル タイムでのアラームの確認. 1243
アラーム確認モデルの理解. 1244
アラームを確認するためのドットフィールドの使用. 1246
アラームを確認するためのスクリプト関数の使用. 1251
タグ値が平常に復帰するときの自動確認の使用. 1252
アラームを確認するためのアラーム クライアントの使用. 1253
アラーム コメントと確認コメントの使用. 1254
ランタイム時のアラームの解除. 1254
ドットフィールドを使用したアラームの解除. 1255

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイム時のタグとグループのアラーム プロパティの制御. 1256
タグ変数またはアラーム グループがアラーム状況にあるかどうかの確認. 1261
InTouch 7.1 の動作にアラーム状況を戻す. 1271
タグ変数のアラームしきい値設定の確認. 1271
タグ変数またはアラームグループのアラームの有効化と無効化. 1276
タグ変数のアラームしきい値の変更. 1290
タグ変数のアラーム デッドバンドの変更. 1296
タグ変数に関連付けられたアラーム コメントの変更. 1297
ユーザー定義情報とアラーム インスタンスとの関連付け. 1298
タグ変数またはアラーム グループの抑止タグ変数の確認. 1300
アクティブまたは未確認のアラーム数の確認. 1306

Maintain. 1314

アプリケーションの移行およびアップグレード. 1315
従来のアプリケーションから新しいスタンドアロン アプリケーションへの移動. 1315
古いアプリケーションの移行およびアップグレード. 1315
以前の InTouch アプリケーションの現在のバージョンへの移行. 1316
古いアラーム表示の変換. 1317
アプリケーション設定の管理. 1317
InTouch アプリケーションのインポート. 1317

System Platform IDE を使用した InTouch アプリケーションの管理. 1319
InTouchViewApp オブジェクト InTouchViewApp おぶじぇくと. 1320
InTouchViewApp テンプレートと InTouch アプリケーションの関連付け. 1321
マネージド InTouch アプリケーションの編集. 1321
マネージド InTouch アプリケーションのテスト. 1321
InTouchViewApp オブジェクトの配置. 1321
InTouchViewApp オブジェクトのエクスポートとインポート. 1322
InTouchViewApp オブジェクトの属性. 1322
InTouchViewApp オブジェクトと他のオートメーション オブジェクトとの違い. 1323
ViewEngine オブジェクト ViewEngine おぶじぇくと. 1324

InTouch コンポーネントのエクスポートとインポート. 1325
マネージド InTouch アプリケーションに関連付けられているタグ変数データのエクスポートとインポート. 1325
タグ定義をエクスポートする. 1325
エクスポートされたタグ定義の表示. 1326
タグ定義のインポート. 1327
タグ名ディクショナリ インポート ファイルの形式. 1327
インポート ファイル テンプレートの作成. 1328
ディクショナリ インポート ファイルの操作モードの設定. 1329
アクセス名とアラーム グループの設定. 1331
タグ変数タイプのキーワードと属性の定義. 1336
インポート ファイルでの空白文字列の使用. 1358
フィールドに対するデフォルト値の使用. 1358
スーパータグ インスタンスの作成. 1359
DBLoad を使用したタグ定義のインポート. 1359
InTouch アプリケーション間での InTouch ウィンドウのエクスポートとインポート. 1361
ウィンドウのインポート. 1361

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

インポートされたウィンドウのプレースホルダ タグの変換. 1363
ウィンドウのエクスポート. 1364
スクリプトのインポート. 1365
インポートされたスクリプトでのプレースホルダ タグの変換. 1367
インポートされたウィンドウとスクリプト用のタグ変数プレースホルダ. 1368
アプリケーションからの産業用グラフィックのエクスポート. 1369
アプリケーションへの産業用グラフィックのインポート. 1370
産業用グラフィック ツールボックスで選択したシンボルのエクスポート. 1371
カスタム クライアント コントロールのインポートと埋め込み. 1372
重複するクライアント コントロールをインポートする際の競合の解決. 1372
産業用グラフィックへのクライアント コントロールの埋め込み. 1374
HTML5 ウィジェットのインポート. 1374
InTouch アプリケーションへのスクリプト関数ライブラリのインポート. 1375
.NET スクリプト ライブラリ内の競合するメソッドのインポートの解決. 1376
アプリケーションのアプリケーション スタイル ライブラリの設定. 1377
アプリケーション スタイル ライブラリのエクスポートとインポート. 1378
アプリケーションのアラーム優先度マッピングの設定. 1379
アプリケーションからの産業用グラフィック テキスト文字列のエクスポート. 1380
アプリケーションへの産業用グラフィックのテキスト文字列のインポート. 1381
シンボルからのローカリゼーション文字列のエクスポート. 1383
産業用グラフィック ライブラリのインポート. 1384
マルチ モニタ システムの設定. 1385
マルチ モニタの設定. 1385
単一のビデオ カードの設定. 1385
複数のビデオ カードの設定. 1386
マルチ モニタ アプリケーションの計画. 1387
マルチ モニタ ビデオ カードの選択. 1388
アプリケーションの画面解像度の決定. 1388
アプリケーションを表示するモニタの数の決定. 1388
アプリケーション ウィンドウの場所の決定. 1389
マルチ モニタ InTouch アプリケーションの開発. 1390
マルチ モニタ パラメータの設定. 1390
画面解像度の変換の設定. 1391
アプリケーションの配置およびマルチ モニタの設定の確認. 1391
ランタイム中のマルチ モニタ サポートの確認. 1392
タブレット PC での InTouch の使用. 1393
視覚化画面の注釈付けと電子メール メッセージによる送信. 1393
ウィンドウの注釈の作成. 1394
ウィンドウの注釈の選択、コピー、および削除. 1394
注釈の付いたウィンドウの保存、印刷、電子メールによる送信. 1394
AnnotateLayout() 関数. 1395
画面の向きの変更. 1396
InTouch サービスの管理. 1397
サービスとしての WindowViewer の実行. 1397
サービスとして起動するための WindowViewer の設定. 1398
WindowViewer でアプリケーションをサービスとして実行するための WIN.INI の編集. 1400
手動によるサービスの開始. 1400
サービスの停止. 1400

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch サービスのユーザー アカウントの設定 1401
InTouch サービスのトラブルシューティング 1401
サービスのエラー メッセージの表示 1402
サービス ユーザー アカウントの問題のトラブルシューティング 1402
有効 I/O アイテムの使用解除 1402
InTouch サービスのレジストリ キー 1403
アラーム 1404
従来のアラーム システムからの移行 1404
標準アラーム システムから分散アラーム システムへの移行 1404
アラーム データベースの維持 1405
アラーム データベースの維持の概要 1405
パージ設定またはアーカイブ設定 1406
アラーム データベースの復元 1417
INTOUCH.ini ファイルからのアプリケーション設定のカスタマイズ 1422
カスタム INTOUCH.ini パラメータ 1422
カスタム ログ プロパティの設定 1424
WindowMaker ショートカット メニューの無効化 1424
カスタム WindowViewer プロパティの設定 1425

Diagnose 1428
QuickScript のトラブルシューティング 1429
Log Viewer へのメッセージのログ記録 1429
LogMessage() 関数 LogMessage()かんすう 1429
Log Viewer メッセージの表示 1430
エラー メッセージの理解 1431
メイン ウィンドウ 1431
タグ リファレンスの追加 1431
タグ値の変更 1432
監視ウィンドウの名前の変更 1432
InTouch Web Client の管理 1434
InTouch Web Client でのエラー処理 1434
ブラウザとモバイルのサポート 1434
Web ブラウザでのグラフィックの表示に関するトラブルシューティング 1435
無効な証明書エラー 1435

Manage 1437
AVEVA Operations Control 接続エクスペリエンス 1438
Operations Control 接続エクスペリエンスについて 1438
Operations Control 接続エクスペリエンスの設定 1438
認証とエンタイトルメント 1444
AVEVA Identity Manager を使用した認証 1445
接続損失時の動作 1448
ライセンスとエンタイトルメント 1449
サブスクリプション情報の表示 1450
WindowMaker の起動と実行 1451
WindowViewer の起動と実行 1452

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Operations Control 接続エクスペリエンスでのアプリケーション セキュリティの設定 1454
マネージド アプリケーションでのセキュリティ設定 1455
スタンドアロン アプリケーションでのセキュリティ設定 1458
混合モードでの操作 1459
Application Manager の起動と実行 1460
アプリケーション マネージャからのアプリケーションの選択と実行 1461
InTouch Access Anywhere の起動と実行 1462
ネットワーク アプリケーション開発（NAD）アプリケーションの起動 1462
セキュリティ書き込みおよび確認書き込みの実行 1463
Web Client（オンプレミス）の起動と実行 1465
トレンドコントロールの起動と実行 1467
AVEVA アプリケーションからのサインアウト 1471
サポートされている InTouch HMI 機能 1472
サポートされていない InTouch HMI 機能 1474
Operation Control モード以外での InTouch の使用 1475
InTouch HMI でのライセンス 1475
ライセンスタグの数について 1475
InTouch HMI で使用できるライセンス 1478
RDS と非 RDS 環境の InTouch ライセンス 1478
InTouchView アプリケーション ライセンスについて 1479
WindowMaker ライセンス 1480
WindowViewer ライセンス 1480
サーバー上の InTouch ライセンスの管理に関するベスト プラクティス 1481
ライセンス情報を表示する 1482
起動後における別のライセンスの使用の管理 1484
フリーモードとデモモードでの操作 1485
猶予期間での操作 1486
リモートタグを数える関数 1488
InTouch Web Client のライセンス付与 1494
InTouch アプリケーショングラフィックスのライセンスを取得する 1494
ライセンス機能 1495
シングル セッション モード 1497
猶予期間 1497
定期的な更新 1498
ライセンスの解放 1498
補足コンポーネントについて 1498
レシピ マネージャの使用 1499
InTouch からの SQL データベースの操作 1521
16 ペン トレンド ウィザードの使用 1570
Symbol FactorySymbol Factory 1584
InTouch のセキュリティ保護 1595
InTouch セキュリティ機能 1596
認証ベースおよび承認ベースのセキュリティ 1603
ユーザーの管理と承認レベルの設定 1613
ログオンとログオフ 1621
オペレータまたはアクセス レベルに基づく機能の有効化と無効化 1628
現在ログオンしているオペレータに関する情報の取得 1630
セキュリティ システム タグと関数の概要 1634

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

管理者以外のユーザーに許可されたアプリケーション マネージャの操作. 1635
アプリケーション フォルダへの役割ベースのセキュリティの適用. 1637
マネージド InTouch アプリケーションおよび NAD InTouch アプリケーションのローカル作業ディレクトリの完全性チェック. 1642
InTouch HMI のセキュリティの管理. 1644
InTouch HMI のセキュリティ設定. 1644

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Standard

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

章 1 統合開発環境

WindowMaker は、InTouch HMI アプリケーションを作成するために使用する開発環境です。製造プロセスを表示および管理するためにオペレータが使用するビジュアル インターフェイスを作成するには、WindowMaker を使用します。InTouch HMI のインターフェイスでは、運用環境のデータへの表示、および運用環境へのデータ書き込みが行われます。

WindowMaker を使用すると、InTouch アプリケーションの以下のビジュアル インターフェイス要素を設定できます。

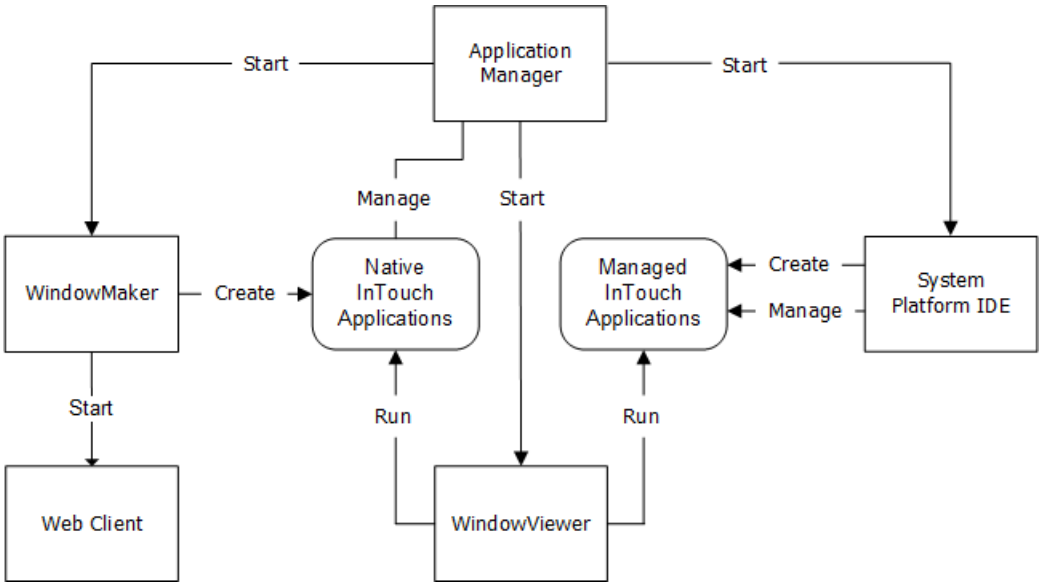
- ウィンドウには、工場のオペレータが製造プロセスを管理するための対話型の視覚要素が含まれています。
- 基本的なオブジェクトとは、長方形、円、線、およびテキストなどのシンプルなグラフィック要素です。
- ユーザー定義の複合オブジェクトは、バルブおよびタンクなどの、運用環境の要素を表す 1 つまたは複数の基本的なオブジェクトで構成されます。
- 事前定義済み複合オブジェクトを使用すると、アラーム、履歴トレンド、シンボル ウィザードなどの特定の機能を実行できます。
- アニメーション リンクは、外観をアニメーション化しユーザーの入力をタスクおよび製造データの変更に送信するための、シンプル オブジェクトおよび複合オブジェクトのプロパティです。
- ウィザードは、特定の関数を実行する、またはスライダーやメーターなどの特定の外観を持つ定義済み複合オブジェクトです。
- ActiveX コントロールとは、現在のアラームを表示するなどの特定の関数を実行するために InTouch ウィンドウに配置できるコントロールです。

詳細については、「WindowMaker のインターフェイス」というタイトルのビデオをご覧ください。

InTouch アプリケーション マネージャについて

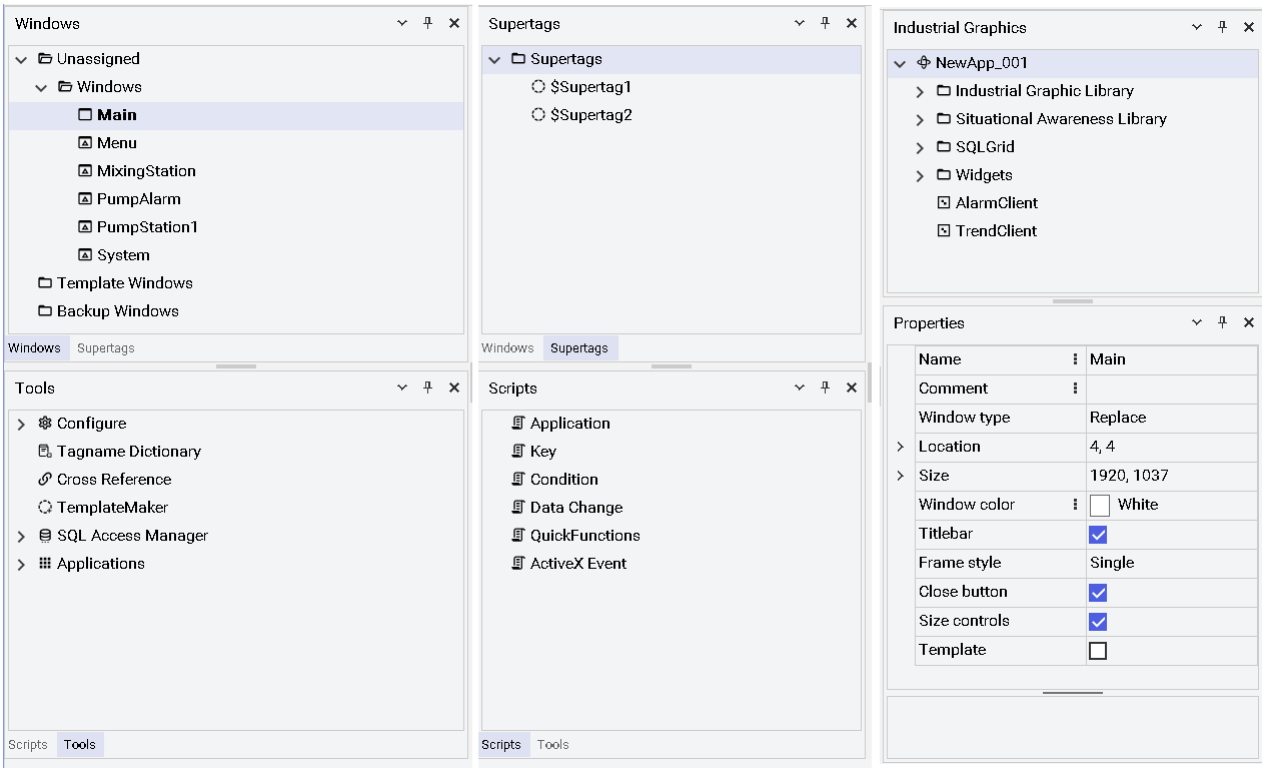
InTouch アプリケーションの作成、削除、および変更などの大部分のグローバル タスクを管理するには、InTouch アプリケーション マネージャを使用します。アプリケーション マネージャには、現在の InTouch アプリケーションのリストが表示されます。WindowMaker または WindowViewer を開くには、リストからアプリケーションを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



アプリケーション エクスプローラを使用する

アプリケーション エクスプローラには、ウィンドウ/スーパータグ、スクリプト/ツール、および産業用グラフィック ツールボックスのタブ形式セクションがあります。他のツールバーと同様に、開くこと、閉じること、ピン留めすること、およびドッキングすることができます。デフォルトでは、各ペインは以下の図に示す場所に表示されます。



これらのビューからは、すべてのアプリケーション ウィンドウ、スクリプト、設定メニュー、タグ名ディクショナリ、およびウィザードにアクセスできます。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ペインをドッキングする

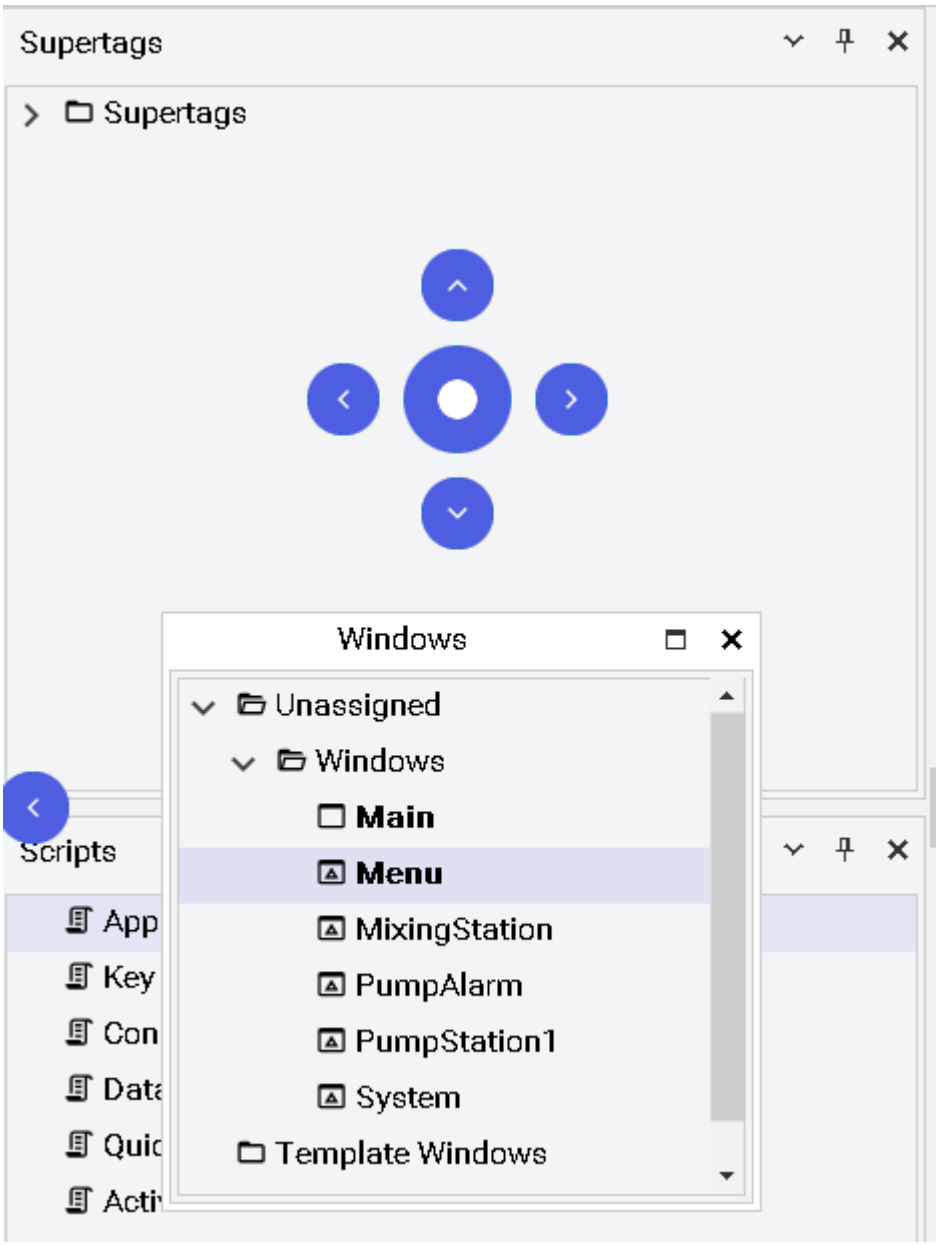
これらの各ペインは移動して、画面の上下左右の位置にドッキングすることができます。ペインをドッキングするには、ペインの見出しを選択します。カーソルが二股の矢印に変わります。ペインをドラッグして画面上の目的の場所にドッキングします。

ナビゲーション エイドを使用してフローティング ペインをドッキングすることもできます。画面の特定の場所で以下の操作を実行します。

- 上向きの矢印を選択すると、画面の上部にペインがドッキングします。
- 下向きの選択を選択すると、画面の下部にペインがドッキングします。
- 右向きの選択を選択すると、画面の右側にペインがドッキングします。
- 左向きの選択を選択すると、画面の左側にペインがドッキングします。

ペインは、垂直および水平方向にドッキングできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



ペインの位置を変更した場合、WindowMaker を閉じてから再度開いた後でもペインのカスタム設定は維持されます。

ペインの位置変更とドッキングに関する情報は DockSettings.xml に保存されます。 ドッキング設定を変更した後にナビゲーションの問題が発生した場合は、以下の手順を実行します。

1. WindowMaker を閉じます。
2. **C:\Users\<ユーザー名>\AppData\Local\Wonderware** に移動します。
3. DockSettings.xml ファイルを見つけて削除します。
4. WindowMaker を再度開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

アプリケーション エクスプローラでナビゲートする

いずれかのアプリケーション エクスプローラ ツールバーにあるフォルダは、表示または非表示にすることができます。

［アプリケーション］ビューには、インストールされているその他のアプリケーションが表示されます。

アプリケーション エクスプローラ フォルダを展開または非表示にする

1. フォルダまたはアイコンをダブル選択して展開し、グループ メンバーを表示します。
2. メンバーをダブルクリックして、そのメンバーを開きます。

アプリケーション エクスプローラにアプリケーションを追加する

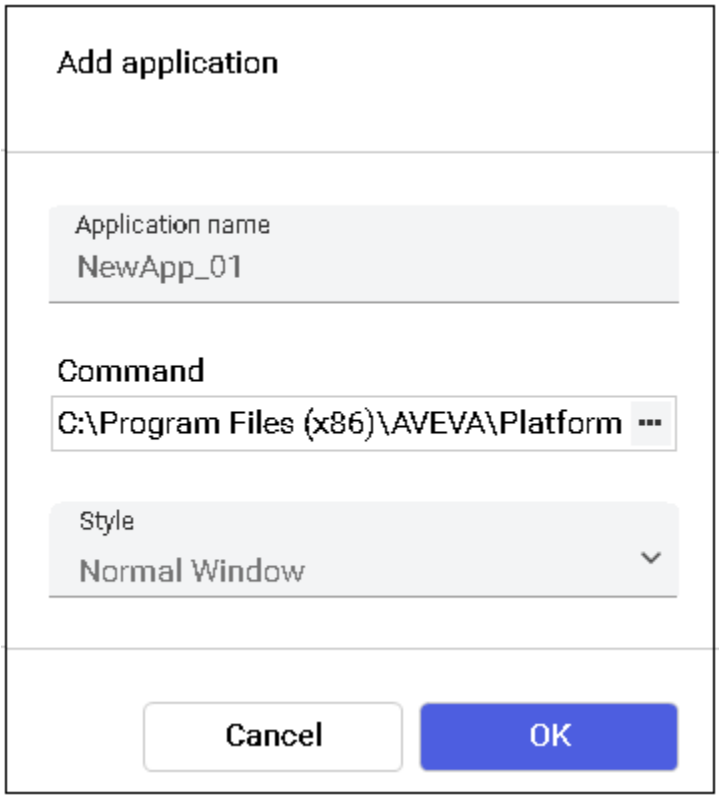
WindowMaker アプリケーション エクスプローラは、WindowMaker 内から他のアプリケーションを起動できます。たとえば、アプリケーションの開発と同時に I/O サーバーを実行して、設定することができます。Windows のメモ帳、ワードパッド、Microsoft Excel、Microsoft Word、Microsoft ペイントなどのサード パーティ プログラムを起動できます。

アプリケーション エクスプローラは、文書やスプレッドシートなどの特定のファイルを開くように設定することもできます。

新しいアプリケーションをアプリケーション エクスプローラに追加する

1. ［ツール］ペインで、［アプリケーション］を右クリックし、［新規］を選択します。

   ［アプリケーションの追加］ウィンドウが開きます。



2. ［アプリケーション名］ボックスに、アプリケーションの名前を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［**コマンド**］ボックスにアプリケーションのフル パスを入力します。アプリケーションを参照するには［...］ボタンを選択します。

   ［**コマンド ライン**］ボックスには、アプリケーションに対するコマンド ライン パラメータを追加できます。

4. ［**スタイル**］リストから、アプリケーションの起動時の表示方法を選択します。使用可能なオプションは、［狭いウィンドウ］、［最小化］、および［最大化］です。

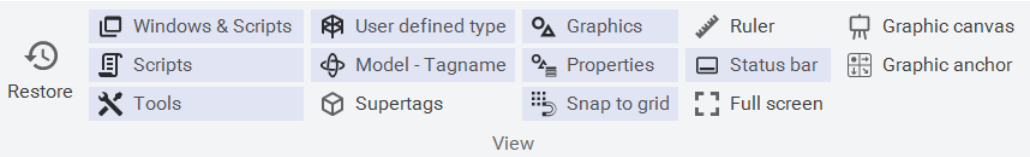
5. ［**OK**］を選択します。

   ［アプリケーション］の下の［アプリケーション エクスプローラ］にアプリケーションが追加されます。これで、アプリケーションを WindowMaker から直接起動できるようになりました。

注意：WindowViewer （view.exe）は**アプリケーション エクスプローラ**に追加しないでください。WindowViewer を正しく起動させるには、［**ファイル**］メニューの［**WindowViewer**］を選択するか、ツールバーの［**実行**］高速切り替えを選択します。

ツールバーを管理する

WindowMaker のツールバーを使用すると、頻繁に使用するコマンドにすばやくアクセスできます。WindowMaker を起動すると、デフォルトですべてのツールバーが表示されます。ツールバーの管理、およびツールバーの表示と非表示の切り替えは、［表示］メニューを使用して行うことができます。



ツールバーをドラッグして、フローティングさせたりドッキングさせたりすることができます。非表示されていたドッキング ツールバーを表示すると、ツールバーはウィンドウで最後にドッキングされていた位置で再び表示されます。ツールバーはデフォルトの位置から開発ウィンドウ内の任意の位置に移動できます。フローティング ツールバーのタイトル バーを使用すると、サイズを変更できます。

ツールバーを表示または非表示にする

- ［**表示**］メニューで、ツールバー アイコンを選択します。

フローティング ツールバーのサイズを変更する

1. カーソルをツールバーの端に移動します。カーソルが両向き矢印に変わります。
2. ツールバーの端をドラッグして、ツールバーを移動したり、サイズ変更したりします。

   カーソルを移動すると、マウス ボタンを離した際のツールバーのサイズを示すボックスが表示されます。

WindowMaker 環境設定を設定する

WindowMaker の設定画面で、WindowMaker の動作に影響を与える環境設定およびオプションを設定できます。以下の操作を実行できます。

- タイトル バー テキストを変更する
- グリッドを表示または非表示にする
- グリッドのピクセル間のスペースを変更する

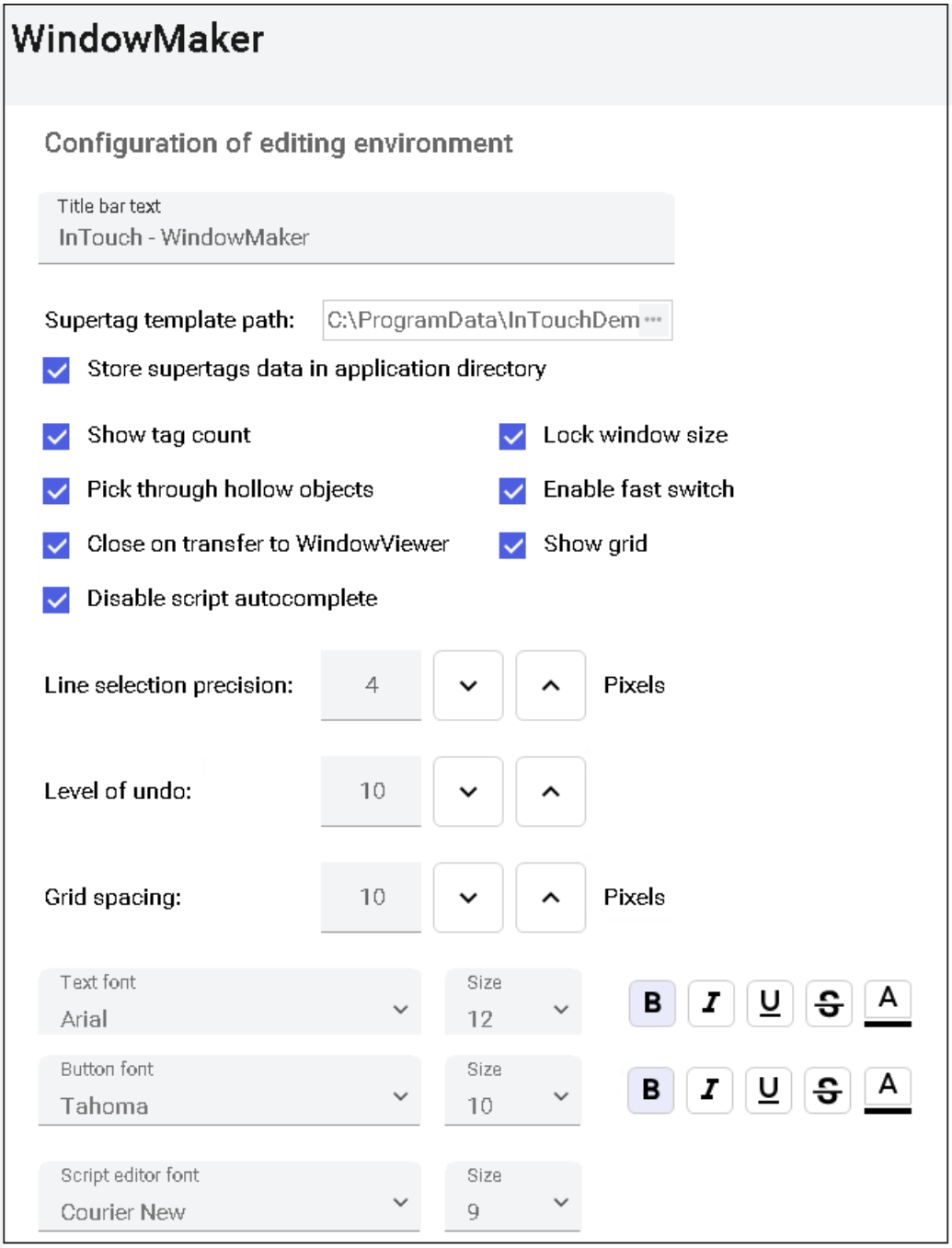
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- タグの数を表示する
- テキストおよびボタンのデフォルト フォントを変更する
- マウス カーソルの反応範囲を設定する
- WindowViewer に切り替えるときに WindowMaker を閉じるオプションを設定する
- 枠線だけのシンボルやセルの中のオブジェクトを選択する
- WindowMaker から WindowViewer への高速切り替えを有効にする
- 元に戻すレベルの数を設定する

WindowMaker のプロパティを設定する

1. ［ファイル］メニューの［**設定**］を選択し、［**WindowMaker**］を選択します。
   WindowMaker の設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. 環境編集領域の WindowMaker の設定で、タイトル バーの外観を設定します。以下のいずれかを実行します。
   - [タイトル バーのテキスト] ボックスに、デザイン時にタイトル バーに表示するテキストを入力します。
   - [スーパータグ テンプレート パス] で、スーパータグ テンプレートを保存するパスを指定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. スーパータグ データをアプリケーション ディレクトリに保存するには、［**アプリケーション ディレクトリにスーパータグ データを保存**］チェック ボックスをオンにします。このオプションを選択すると、［**スーパータグ テンプレート パス**］フィールドで指定したパスが上書きされます。
4. その他のウィンドウ プロパティを設定します。以下のいずれかを実行します。
   - タグ名ディクショナリのタグ名の数をメニュー バーに表示するには、［**タグの数の表示**］チェック ボックスをオンにします。大量のタグがある場合にタグの数を表示すると、タグ名ディクショナリのパフォーマンスに影響が生じることがあります。

     これは、制限されたタグ名ディクショナリ サイズでアプリケーションを作成する場合に役立ちます。タグ名の数には、リモート タグ名参照またはシステム タグは含まれていません。リモート タグ名参照の使用を確認するには、［**ホーム**］タブの［**使用カウントを更新**］を選択します。
   - InTouch アプリケーション ウィンドウのサイズをロックするには、［**ウィンドウ サイズをロック**］チェック ボックスをオンにします。

     このオプションを選択すると、ウィンドウとグラフィックのスケーリングなしでアプリケーションを別の解像度に変換できます。
   - ［**枠線だけのシンボルやセルの中のオブジェクトを選択**］チェック ボックスをオンにすると、枠線だけのシンボルやセルの背面にあるオブジェクトを選択できます。

     これにより、フレームを背景に移動する必要なく、フレーム内のオブジェクトを選択することができます。
   - WindowMaker と WindowViewer を切り替える「高速切り替え」を使用するには、［**高速切り替えを有効にする**］チェック ボックスをオンにします。

     高速切り替えは、WindowMaker の右上隅に表示される［**実行**］という単語です。WindowViewer では、［**開発**］が表示されます。

     高速切り替えを有効にすると、WindowViewer に切り替える前に WindowMaker で開いているすべてのウィンドウに対する変更が自動的に保存されます。
   - WindowViewer の起動時に WindowMaker を自動的に閉じるには、［**WindowViewer への移行時に自動的に終了**］チェック ボックスをオンにします。

     このオプションの目的は、制限されたメモリを節約することです。メモリが問題ではなく、WindowViewer と WindowMaker を頻繁に切り替える場合は、このオプションを選択しないでください。

     ［**WindowViewer への移行時に自動的に終了**］をオンにすると、［**WindowViewer のプロパティ**］画面の［**全般**］タブの対応コマンドである［**WindowViewer を自動的に終了する**］もオンになります。
   - グリッドを表示するには、［**グリッドの表示**］チェック ボックスをオンにします。
   - オートコンプリートの推奨リスト ボックスの表示を無効にするには、［**スクリプトのオートコンプリートを無効化**］を選択します。
   - ［**マウス カーソルの反応範囲**］ボックスに、カーソルがラインから離れていても選択できる範囲をピクセル単位で入力します。

     ほとんどの場合、デフォルト設定の 4 が適当です。
   - ［**元に戻すレベル**］ボックスに、［**元に戻す**］および［**やり直す**］コマンドで維持するレベルの数を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

最大レベルは 25 です。0 と入力すると、「元に戻す/やり直す」機能がオフになります。

1 つのレベルは 1 回のアクションを表しています。新しいウィンドウを作成するか、既存のウィンドウを開くと、［**元に戻す**］および［**やり直す**］のスタックは空になります。ウィンドウを閉じると、両方のスタックが空になります。

- ［**グリッド間隔**］ボックスに、グリッド間隔をピクセル数で入力します。

5. テキスト、ボタン、およびスクリプト エディタのフォント プロパティも［**テキストフォント**］、［**ボタン フォント**］、および［**スクリプトエディタ フォント**］フィールドを使用して設定できます。
6. 任意のウィンドウでのこれらのデフォルト値は、該当する画面の［**フォント**］設定を使用して上書きできます。
7. ［**OK**］を選択します。
8. 変更内容を適用するには、WindowMaker を再起動します。

マウス ショートカット

以下のショートカットを使用して、ダイアログ ボックスを開き、その他の一般タスクを実行します。

WindowMaker のアイテム用のメニュー コマンドにアクセスする

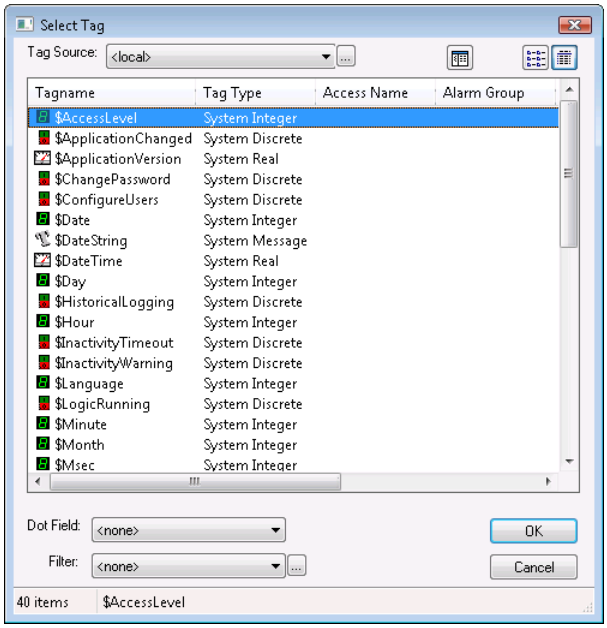
- アイテムを右クリックします。アイテムには、グラフィック オブジェクトやビュー内のフォルダ名などが含まれます。

［アニメーション リンク］ダイアログ ボックスを開く

- オブジェクトまたはシンボルをダブルクリックします。

［タグの選択］ダイアログ ボックス（タグ ブラウザ）を開く

- リンク定義ダイアログ ボックス内の空白の式入力フィールドをダブルクリックします。［**タグの選択**］ダイアログ ボックスが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ ドットフィールドにアクセスする

- ［タグ変数］または［タグ変数／式］入力ボックスに、タグ変数の後にピリオドを入力してから、ピリオドの右側をダブルクリックします。ピリオドのみを入力してその右側をダブルクリックすることもできます。［**フィールド名の選択**］ダイアログ ボックスが開き、すべてのタグ変数ドットフィールドが表示されます。

タグ名ディクショナリでタグ名定義を開く

- タグ変数をダブルクリックします。

全画面モードを使用する

全画面モードでは、開いているウィンドウとフローティング ツールバーを除くすべてのプログラム要素が非表示となります。

全画面モードのオン/オフを切り替える

- ［**表示**］ツールバーで、［**全画面**］を選択して、標準モードから全画面モードに切り替えます。

  ［**表示**］ツールバーが自動的に［**元のサイズに戻す**］ツールバーに変化して最上部にフロートされます。

［ファイル］メニューからウィンドウを管理する

［**ファイル**］メニューを使用して、ウィンドウを開く、保存、閉じる、または削除などの操作を行う場合、選択したコマンドを使用できるすべてのウィンドウの名前が一覧表示されます。

ウィンドウ リストの表示

ウィンドウは、アイコン ビュー、一覧ビュー、または詳細ビューで表示できます。

- アイコン ビュー：［**アイコン**］を選択すると、ウィンドウのリストがグリッドに配置されます。
- リスト ビュー：［**リスト**］を選択すると、ウィンドウが複数列のリストに配置されます。
- 詳細ビュー：［**詳細**］を選択すると、ウィンドウの説明を含むウィンドウのリストが表示されます。説明は、ウィンドウのプロパティ パネルの［コメント］フィールドから取得されます。

ウィンドウの選択

- ウィンドウを選択するには、ウィンドウ アイコンまたはエントリを選択します。選択したウィンドウが強調表示されます。
- 複数のウィンドウを選択するには、目的のウィンドウ アイコンまたはエントリを選択します。選択したウィンドウが強調表示されます。
- すべてのウィンドウを選択するには、［**すべて選択**］チェック ボックスをオンにします。

ウィンドウの選択解除

- 選択したウィンドウの選択を解除するには、ウィンドウ アイコンまたはエントリを再度選択します。
- すべてのウィンドウの選択を解除するには、［**すべて選択**］チェック ボックスを選択してクリアします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウの検索

- リストからウィンドウを検索するには、検索バーを使用します。

フォントのデフォルトを設定する

テキスト オブジェクトおよびテキストを持つボタン オブジェクトのデフォルト フォントを設定できます。これらのデフォルト設定は、フォント セクションを使用してウィンドウまたはボタン テキストをカスタマイズすることによって上書きできます。

フォントのデフォルトを設定する

1. ［**ファイル**］メニューの［**設定**］を選択し、［**WindowMaker**］を選択します。

   WindowMaker の設定画面が表示されます。

2. テキスト、ボタン、およびスクリプト エディタのフォント プロパティは、［**テキスト フォント**］、［**ボタン フォント**］、および［**スクリプト エディタ フォント**］フィールドを使用して設定できます。

   これらのデフォルト設定は［**フォント**］セクションを使用して上書きできます。

3. ［**OK**］を選択します。

矢印キーを使用してオブジェクトを移動する

WindowMaker で、**矢印キー**を使用して、選択したオブジェクトまたはオブジェクトのグループを移動できます。

矢印キーを使用してオブジェクトを移動する場合、オブジェクトの移動距離はグリッドが表示されているかどうかによって異なります。

グリッドが表示されている場合、オブジェクトの移動ピクセル数は WindowMaker の設定画面で設定したグリッド間隔に依存します。デフォルト設定のグリッド間隔は 10 ピクセルです。

グリッドが表示されている場合：

- 矢印キーを押すと、オブジェクトが 1 グリッド ポイント移動します。
- Shift キーと矢印キーを押すと、オブジェクトが 2 グリッド ポイント移動します。
- Ctrl キーと矢印キーを押すと、オブジェクトが 4 グリッド ポイント移動します。

グリッドが表示されていない場合：

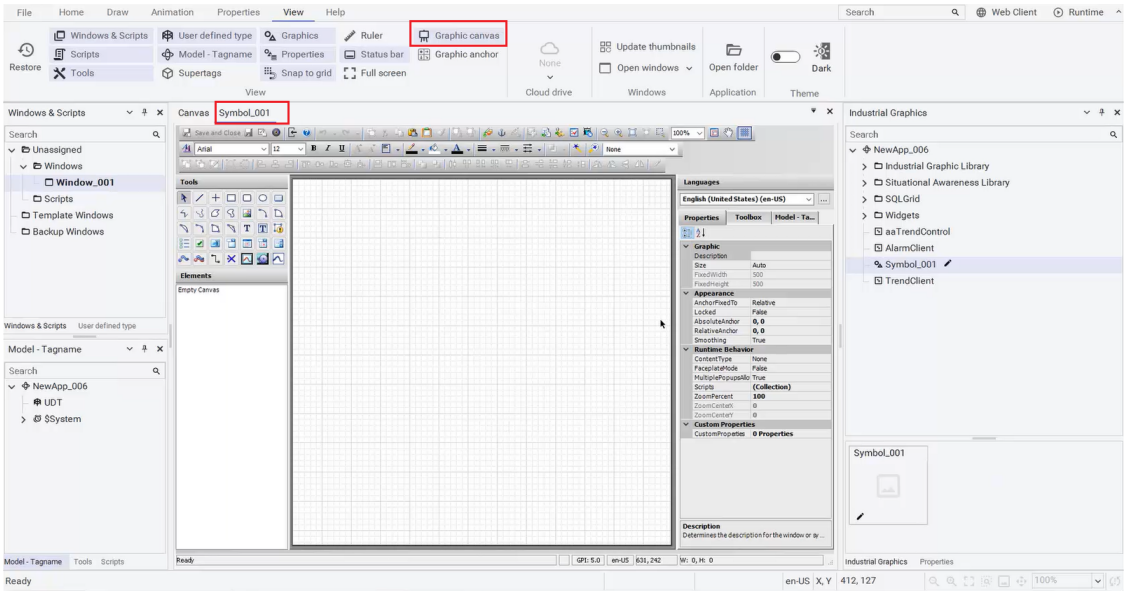
- 矢印キーを押すと、オブジェクトが 1 ピクセル移動します。
- Shift キーと矢印キーを押すと、オブジェクトが 10 ピクセル移動します。
- Ctrl キーと矢印キーを押すと、オブジェクトが 50 ピクセル移動します。

グラフィック キャンバスとグラフィック アンカーを使用する

ｸﾞﾗﾌｨｯｸ ｷｬﾝﾊﾞｽ

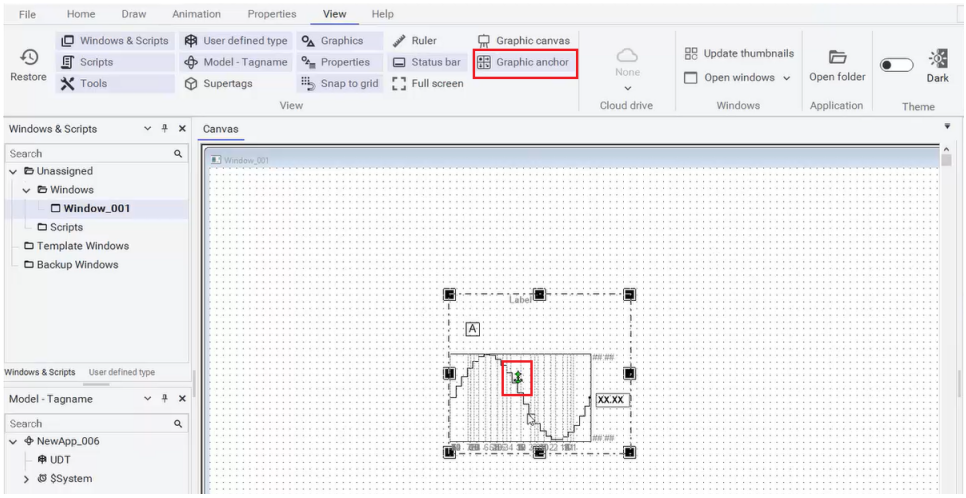
WindowMaker の［**表示**］メニューの［**グラフィック キャンバス**］オプションを使用して、Industrial Graphic Editor ウィンドウを WindowMaker のタブとして表示します。つまり、［グラフィック キャンバス］オプションを選択している時にグラフィックを開くと、WindowMaker 内のタブの Industrial Graphic Editor で開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



グラフィック アンカー

WindowMaker の［表示］メニューの**グラフィック** アンカー オプションを使用して、アンカー ポイントをグラフィックに表示します。アンカー ポイントの詳細については、AVEVA Industrial Graphic Editor ヘルプの「サイズの反映とアンカー ポイント」セクションを参照してください。



カラー パレットを使用する

カラー パレットを使用すると、線、長方形、角丸長方形、楕円形、折れ線、多角形、およびテキストの静的プロパティおよび動的プロパティに対して、色を適用できます。ウィンドウの背景色およびビットマップの透明色を選択できるため、ビットマップの背後にあるオブジェクトを表示できます。

パレットには、最大 1670 万色という幅広い選択範囲の色が提供されています。使用できる色の数は、ご使用のビデオ カードの性能によって制限される可能性があります。

カスタム カラーを定義して追加することもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

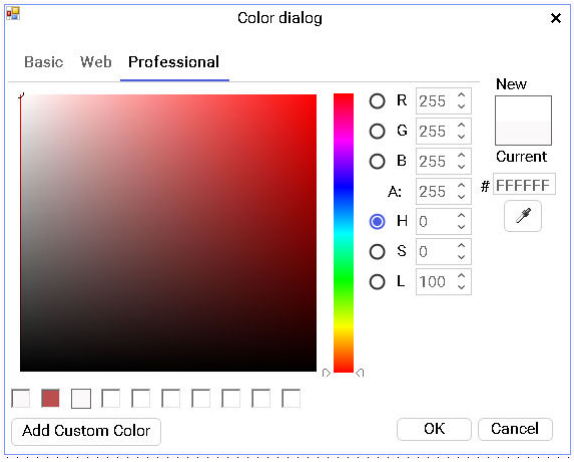
カラー パレットを開く

カラー パレットは、［プロパティ］パネルにある色付きの正方形を選択するか、選択したオブジェクトに線、塗りつぶし色、またはテキストの色を適用する色ツールを選択すると表示されます。

カラー パレットを開く

1. 設定画面の［色］フィールドの横にある色付きの正方形を選択します。

   ［色］ダイアログが表示されます。



1. 他のクラシック カラー パレットにアクセスするには、［**基本**］タブと［**Web**］タブを選択します。
2. 以下のいずれかを実行します。
   - カラー表示の任意の場所を選択し、スライダを使用してカラーを調整します。白と黒を含まない 100% の色を指定するには、ALT + O を押します。
   - ［**赤**］、［**緑**］、および［**青**］のボックスに値を入力して色を定義します。これらの値はカラー マトリックスを確認しながら調整できます。輝度、彩度、および明度の値も変わることに注意してください。
   - ［**輝度**］、［**彩度**］、および［**明度**］のボックスに値を入力して、色を定義します。これらの値を変更すると、赤、緑、青のスケールがそれに応じて変化します。

     輝度は論理色の値です。0 は赤、60 は黄、120 は緑、180 はシアン、200 はマゼンタ、240 は青です。

     彩度は特定の輝度での色の度合で、最大 240 です。

     明度は色の明るさです。
3. ［**OK**］を選択します。

   ［色］ダイアログが閉じて、選択した色が適用されます。

カスタム カラーを作成する

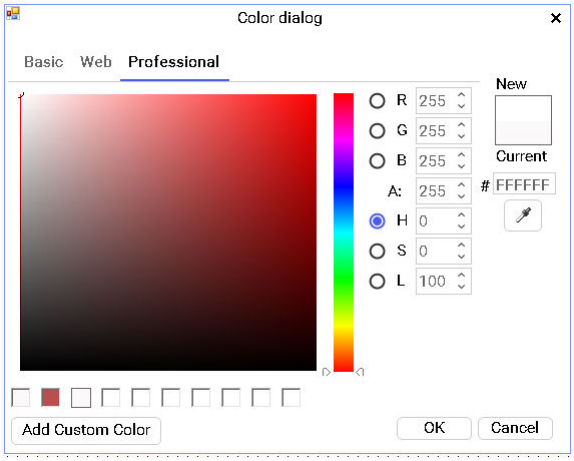
カスタム カラーのパレットを作成することができます。

カスタム カラーを作成する

1. カラー パレットを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**カスタム パレット**］領域で色を選択し、［**カスタム カラーを追加**］を選択します。



3. ［**色 | 純色**］ボックスで表示される色を確認します。

   使用しているモニタが 256 色表示に設定されている場合は、［**色 | 純色**］ボックスには 2 つの色が表示されます。右側には選択した色が純色としてどのように表示されるかを示します。左側にはディザ色または 256 色の 2 色を使用した特定の色の類似色が表示されます。

4. ［**OK**］を選択します。

スポイト ツールを使用してカスタム カラーを作成することもできます。

注意：この機能は、透過ビットマップを作成するときに使用します。

スポイト ツールを使用してカスタム カラーを選択する

1. **カラー パレット**を開きます。
2. スポイト ツールを選択して、追加する色を選択します。

カスタム カラーのインポートとエクスポート

カスタム カラー パレットを定義した場合、InTouch アプリケーションからエクスポートして別の InTouch アプリケーションにインポートできます。

カスタム パレットをインポートするには

1. カラー パレットを開きます。
2. **カスタム パレット**の下向き矢印をクリックします。
3. ［**パレットの読み込み**］をクリックします。Windows 標準の［**ファイルを開く**］ダイアログ ボックスが表示されます。
4. 希望の色定義を持つ .pal パレット ファイルを探し、選択します。
5. ［**開く**］をクリックします。パレット ファイルに含まれている色が**カスタム パレット**にロードされます。

カスタム パレットをエクスポートするには

1. カラー パレットを開きます。
2. **カスタム パレット**の下向き矢印をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［**パレットのエクスポート**］をクリックします。Windows 標準の［**名前を付けて保存**］ダイアログ ボックスが表示されます。
4. パレット ファイルの名前を指定して、［**保存**］をクリックします。

パンとズーム

編集している要素をズーム インまたはズーム アウトして表示を調節し、オブジェクトが正確に整列しているか、または正しく配置されているかを確認することができます。

［パンおよびズーム］ツールバーがデフォルトで画面の右下に表示されます。他のツールバーと同じ様に、他の場所にフローティングさせたりドッキングさせたりすることができます。

画面解像度と異なるターゲット解像度を指定した場合、境界キャンバスは、パンとズームに応じて調整されます。

以下の操作を実行できます。

- 100% ～ 500% のズーム インおよびズーム アウト
- ラバー バンド ツールを使用した特定の領域へのズーム
- 特定のパーセンテージでウィンドウをズームする
- 選択してドラッグし、ウィンドウをパンします。
- 標準デフォルト ビューに戻す

［パンおよびズーム］ツールバーを表示または非表示にする

- ［**表示**］メニューで、［**ステータス バー**］を選択します。

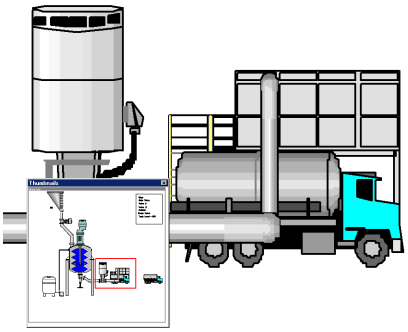
サムネイル ウィンドウを使用してパンとズームする

アプリケーション ウィンドウの詳細にズーム インすると、サムネイル ウィンドウにアプリケーション全体に対する詳細の関係が表示されます。

［**サムネイル**］ウィンドウには開発領域の概要と詳細の両方が表示されます。

［**サムネイル**］ウィンドウを表示または非表示にするには、［**サムネイル**］ボタンを選択します。

赤い長方形がウィンドウ内のズームされた領域の境界です。

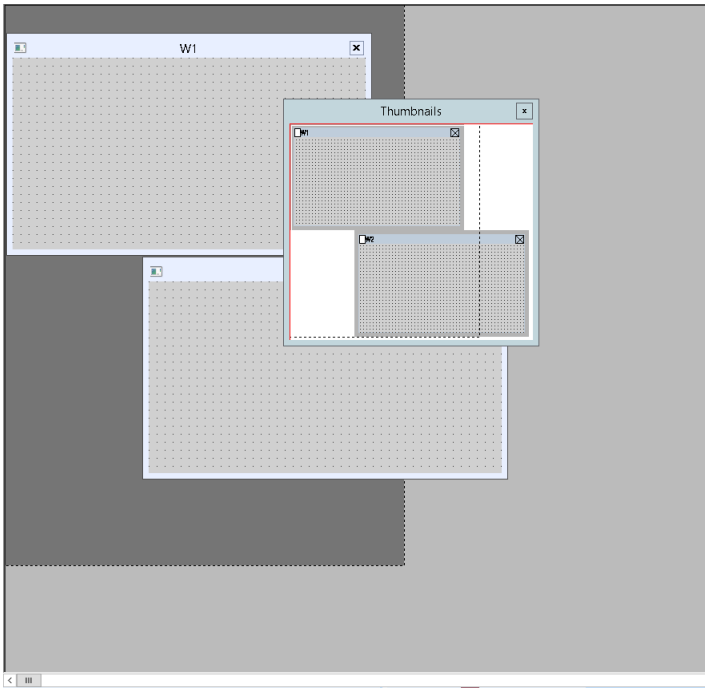


- 赤い長方形をドラッグすると、ウィンドウの別の部分が表示されます。
- ウィンドウの別の領域を選択すると、長方形がその領域に移動します。
- 長方形のサイズを変更すると、表示領域のズーム レベルが変わります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

サムネイル ビューでは、ActiveX コントロールなど、ズームできないオブジェクトには白い長方形が表示されます。

画面解像度と異なるアプリケーション ターゲット解像度を指定した場合、指定したターゲット解像度の寸法を示す境界キャンバスもサムネイル ビューに表示されます。



注意 : ターゲット解像度の境界の寸法を超えるアプリケーション ウィンドウは、上の例に示すようにサムネイル ビューに引き続き表示されます。

マウス ホイールを使用してズームとパンする

マウスにホイールが付いている場合は、Ctrl キーを押しながらホイールを回転させると画像のズーム レベルを変更できます。

- マウス ホイールを回転させると、選択の度にズーム レベルが 20% ずつ変化します。
- カーソルを InTouch ウィンドウに置いてマウス ホイールを押すことによって、ウィンドウ内をナビゲートすることもできます。マウス ホイールを押すと、4 つの矢印が付いたアイコンが表示されます。ウィンドウをナビゲートするには、マウスを移動します。

パンとズームの制限

パンとズームは以下のコントロールには適用されません。

- ActiveX コントロール
- 分散アラーム オブジェクト
- 16 ペン トレンド
- テキスト ボックス
- チェック ボックス
- リスト ボックス

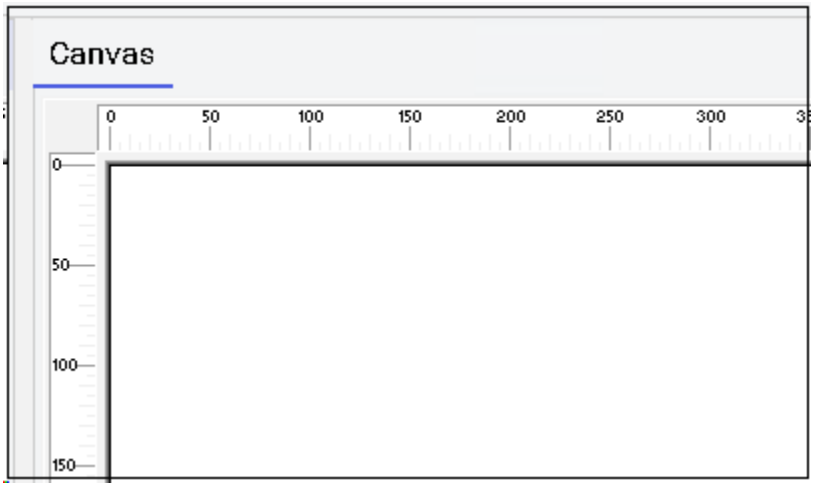
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- コンボ ボックス
- ラジオ グループ オブジェクト

ビューが 100% を超えるズームになっているときに、これらのコントロールの 1 つが表示領域にある場合は、コントロールの名前が付いた長方形がコントロールが使われている領域に表示されます。

画面グリッドとルーラーの使用

開発領域のグリッドおよびルーラーを表示して、オブジェクトの配置および整列に使用できます。



グリッドにオブジェクトを配置する

オブジェクトを定義済みグリッド ポイントに配置するには、［**グリッドに配置**］を選択します。

WindowMaker を起動したときデフォルトでは、グリッドは 10 ピクセルで表示されるように設定されています。グリッドのピクセル間隔は、WindowMaker の設定画面で設定します。

グリッドを表示するには、WindowMaker の設定画面の［**グリッドの表示**］を選択して、［**表示**］メニューの［**グリッドに配置**］を選択します。

グリッドを設定する

1. ［**ファイル**］メニューの［**設定**］を選択し、［**WindowMaker**］を選択します。WindowMaker の設定画面が表示されます。
2. ［**グリッド幅**］ボックスで、グリッド座標間の間隔をピクセル数で入力します。
3. ［**グリッドに配置**］が選択されているときにグリッドを表示する場合は、［**グリッドの表示**］チェック ボックスをオンにします。

   ［**グリッドの表示**］をオンにしないと、［**グリッドに配置**］を選択してもウィンドウにグリッドは表示されません。

ルーラーを使用する

ルーラーを使用すると、ウィンドウでオブジェクトを正確に整列させることができます。ルーラーは開発環境ウィンドウの最上部から一方のサイドに沿って表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ルーラーを表示または非表示にする

1. ［表示］メニューの［表示］グループで［ルーラー］を選択します。
2. ルーラーを非表示にするには、手順を繰り返します。

WindowViewer について

WindowViewer は、InTouch アプリケーションのランタイム環境を提供します。アプリケーションの動作要件に基づいて、WindowViewer がアプリケーションをサポートする方法を設定できます。たとえば、アプリケーションのセキュリティ要件に応じて、WindowViewer でオペレータが使用できるメニューおよびコマンドを設定できます。

ランタイム環境をカスタマイズする

プロパティを設定して、ランタイムの WindowViewer 環境をカスタマイズすることができます。

- 一般プロパティにより、InTouch アプリケーションを実行するための環境条件が決まります。
- ウィンドウ設定プロパティにより、WindowViewer で実行されている InTouch アプリケーションとユーザーがデータをやり取りする際にアクセスできるメニュー、コマンド、およびウィンドウ コンポーネントが決まります。

一般的な WindowViewer プロパティを設定する

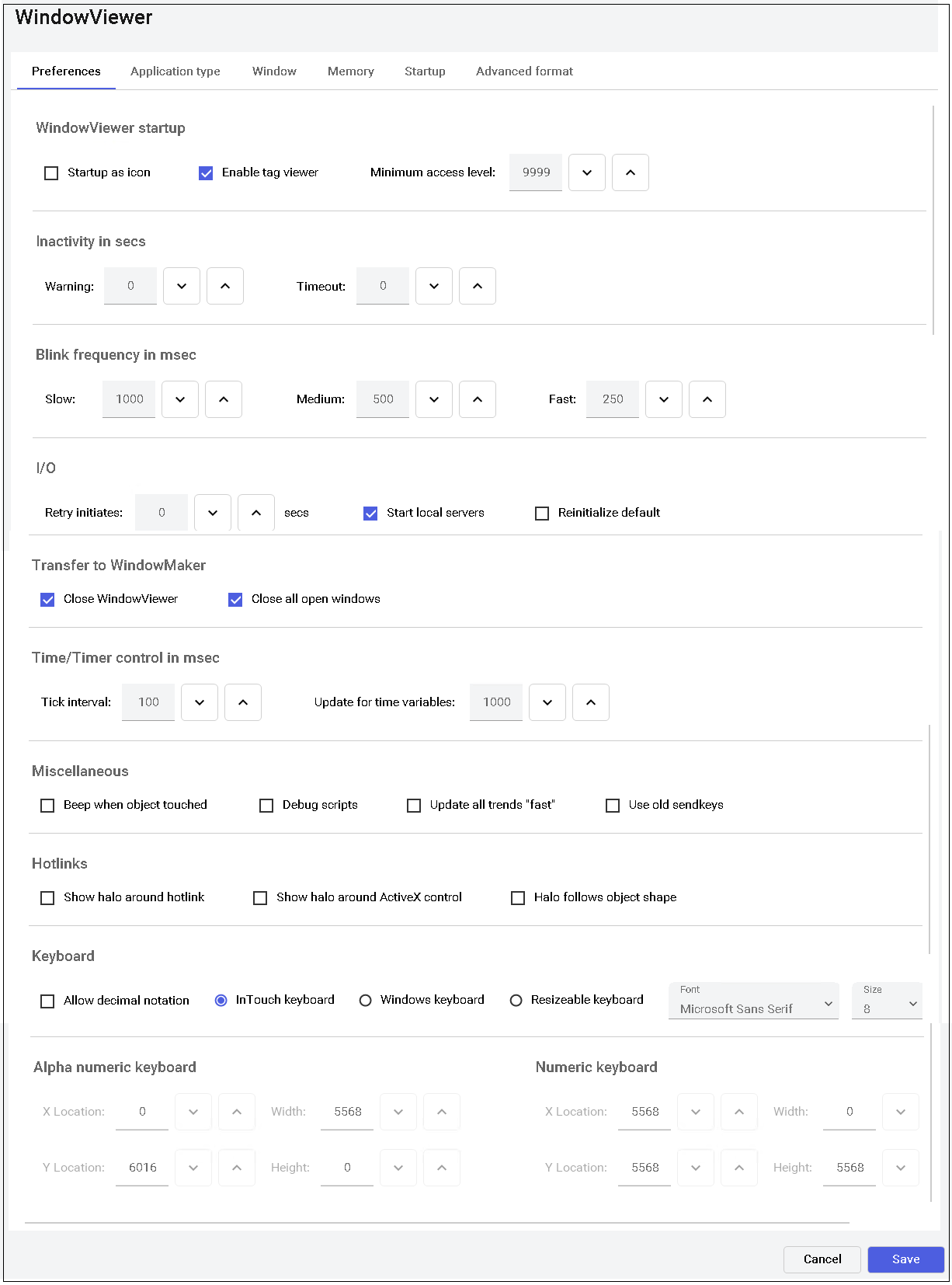
InTouch アプリケーションを実行するときの WindowViewer の特性を決定する一連の一般プロパティを設定できます。これらの一般プロパティを変更した後に変更を有効にするには、WindowViewer を再起動する必要があります。

注意：オペレーティング システムの地域設定を変更する前に WindowViewer を閉じてください。

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］を選択し、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［環境設定］タブの［**WindowViewer 起動時の設定**］セクションで、以下の手順を実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- WindowViewer を最小化した状態で起動するには、［**起動時アイコンにする**］チェック ボックスをオンにします。
- Tag Viewer がランタイム時に起動するようにするには、［**Tag Viewer の有効化**］チェック ボックスをオンにします。Tag Viewer では、ランタイム時にタグの監視やタグ値の変更を行うことができます。詳細については、『AVEVA™ InTouch HMI アプリケーション ランタイム 』の「Tag Viewer の有効化」を参照してください。
- ［**最小アクセス レベル**］ボックスに、Tag Viewer アプリケーションを実行するために必要な最小のセキュリティ アクセス レベルを入力します。値は 0 から 9999 の間の数値である必要があります。詳細については、『AVEVA™ InTouch HMI アプリケーション ランタイム 』の「Tag Viewer の有効化」を参照してください。

4. ［**無操作時間の設定（秒）**］セクションで、オペレータの無操作時間に対する警告およびタイムアウト期間を設定します。

   警告の設定およびタイムアウト時間の詳細については、『AVEVA™ InTouch HMI 管理』の「無操作タイムアウトの設定」を参照してください。

5. ［**点滅速度の設定（ミリ秒）**］セクションに、［**低速**］、［**中速**］、および［**高速**］の点滅アニメーション リンクの間隔の長さをミリ秒で入力します。

6. ［**I/O**］領域で、以下の手順を実行します。
   - ［**I/O 通信リトライ時間**］ボックスに、I/O サーバーへの接続の試行に失敗した後に InTouch アプリケーションが接続を再試行するまで待機する秒数を入力します。最初の接続試行で InTouch が I/O サーバーに正常に接続された場合、［**I/O 通信リトライ時間**］ボックスの設定は何の影響も及ぼしません。
   - WindowViewer の起動時に、通信しようとしている I/O サーバーが実行中でない場合にダイアログ ボックスを表示するには、［**ローカルのサーバーを起動**］チェック ボックスをオンにします。
   - デフォルト設定でアクセス名を再初期化するには、［**デフォルトの再初期化**］チェック ボックスをオンにします。アクセス名に割り当てられた現在の値は無視され、元の設定で再初期化されます。

7. ［**WindowMaker 移行時の設定**］領域で、以下の手順を実行します。
   - WindowMaker を起動するときに WindowViewer を自動的に終了するには、［**WindowViewer を自動的に終了する**］チェック ボックスをオンにします。

     このオプションを選択すると、WindowMaker の設定画面の［全般］タブの［WindowViewer への移行時に自動的に終了］オプションも自動的に選択されます。メモリの容量に問題があり、WindowViewer と WindowMaker 間の高速切り替えを使用する場合は、このオプションは選択しないでください。
   - WindowViewer から WindowMaker に切り替えるする際に、開いているすべてのウィンドウを自動的に閉じるには、［**全ウィンドウを自動的に閉じる**］チェック ボックスをオンにします。

8. ［**データ更新時間の設定（ミリ秒）**］領域で、以下の手順を実行します。
   - ［**内部イベント チェック周期**］ボックスに、InTouch HMI が内蔵タイマーをチェックする間隔を入力します。

     この間隔に基づいて、アプリケーションの［実行中］、ウィンドウの［表示中］、条件の［True の時］または［False の時］、キーとタッチ押しボタン アクションの［押している間］などの QuickScript が実行されるときが決定されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

このオプションにより、INTOUCH.ini ファイル内の TimerTickInterval パラメータの値が設定されます。100 ミリ秒ごとに実行するようスケジュールされたスクリプトでは、［内部イベント チェック周期］は 50 ミリ秒を超えないように設定する必要があります。Windows XP または Windows 2003 を実行するコンピュータでは、最も低い内部イベント チェック周期は 10 ミリ秒です。

- ［**時間システム変数の更新周期**］ボックスに、システム タグ（$Msec、$Second、$Minute など）の時間が更新される間隔をミリ秒単位で入力します。

  デフォルト設定（1000 ミリ秒）を使用することが推奨されます。このオプションを 0 に設定すると、時間変数は更新されません。

9. ［**その他の設定**］領域で、以下の手順を実行します。
   - ウィンドウ上のタッチ対応オブジェクトをオペレータが選択したときに InTouch アプリケーションがビープ音を発するようにするには、［**オブジェクト クリック時ビープする**］を選択します。
   - QuickScript が実行されるたびにロガーにメッセージを書き込む場合、［**スクリプトのデバッグ**］を選択します。

     ［**メニュー/操作設定**］プロパティ ページで［**デバッグ**］を選択した場合、WindowViewer の［**システム**］メニューから QuickScript のログ記録の有効化と無効化を切り替えることができます。
   - トレンド オブジェクトの更新スピードを速くする場合、［**トレンドの更新を速くする**］を選択します。

     このオプションは、アプリケーション ウィンドウ上のランタイム トレンドにオブジェクトが重なっていない場合にのみ選択してください。オブジェクトがトレンドに重なっている場合にこのオプションを選択すると、オブジェクトを正しく描画できない場合があります。
   - InTouch バージョン 3.26 以前で開発された国際アプリケーションを使用している場合、［**旧バージョンの使用**］チェック ボックスをオンにします。
10. ［**Hotlink**］領域で、以下の手順を実行します。
    - ランタイム画面上のオブジェクトにユーザーがカーソルを移動したときにオブジェクトを強調表示するには、［**HotLink の周りにハローを表示する**］チェック ボックスをオンにします。
    - ActiveX コントロールの周りにハローを表示するには、［**ActiveX コントロールの周りにハローを表示する**］チェック ボックスをオンにします。
    - オブジェクトの輪郭の周りのハローにユーザーがカーソルを移動したときに強調表示するには、［**オブジェクトの形に従いハローを表示する**］チェック ボックスをオンにします。
11. ［**キーボード**］領域で、使用するキーボードのタイプを必要に応じて選択します。

    WindowViewer でのキーボード オプションの設定の詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「オブジェクトのアニメーション化」を参照してください。
12. ランタイム時に 10 進値だけを許可して数値以外の文字を制限するには、［**10 進法**］チェック ボックスをオンにします。（［**10 進法を許可**］オプションは英語以外のオペレーティング システムではテストされていません）。
13. ［**英数字キーボード**］および［**数値キーボード**］領域で、X, Y の位置、および［幅］と［高さ］の値を設定します。
14. ［**保存**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowViewer の表示特性を設定する

InTouch アプリケーションを実行するときの WindowViewer の表示特性を決めるプロパティを設定できます。これらのプロパティにより、WindowViewer ウィンドウに表示されるメニュー、コマンド、および標準的なコントロールが決まります。

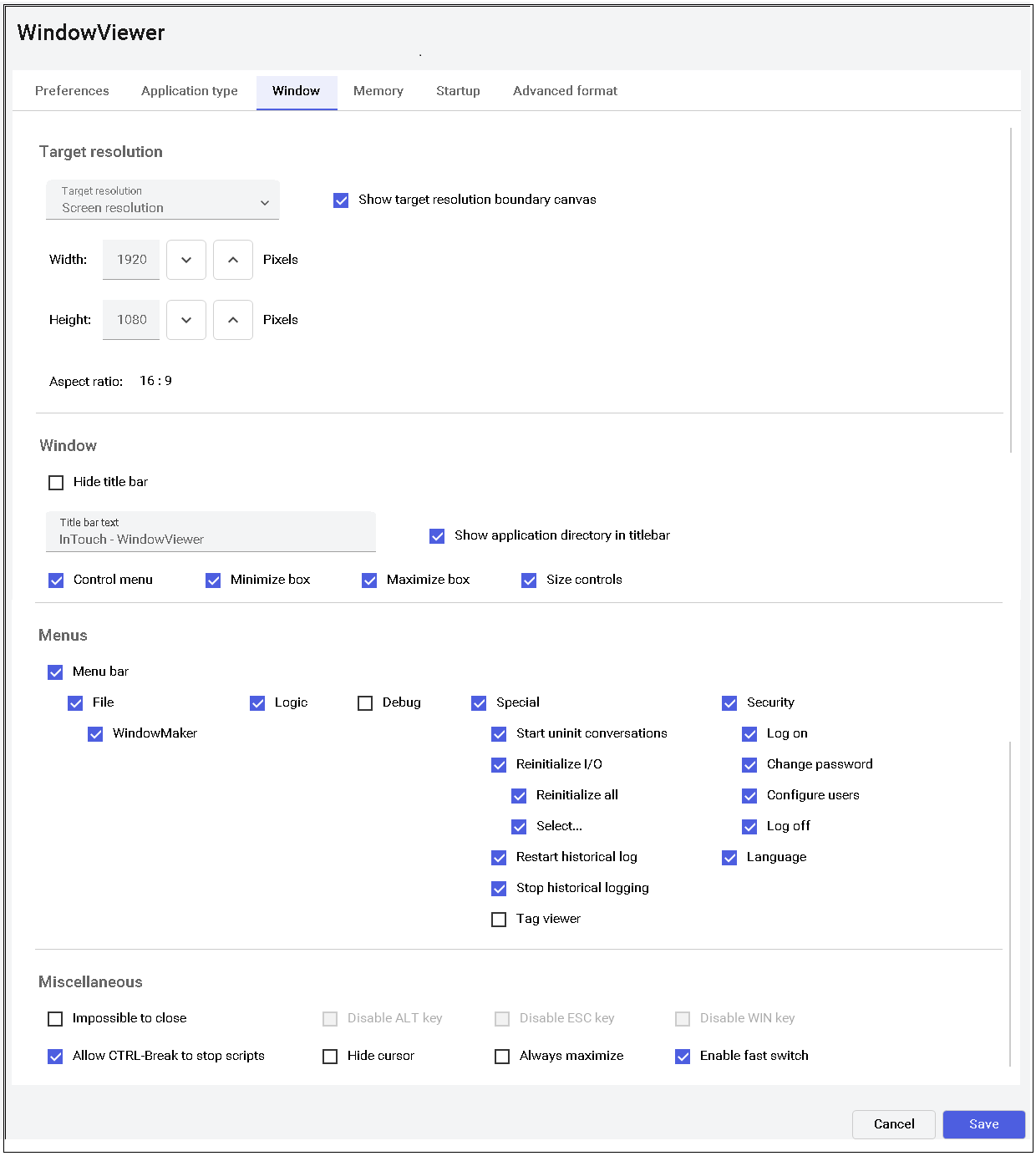
WindowViewer の表示特性を設定する

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］を選択し、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［ウィンドウ］タブを選択します。

   表示特性のチェック ボックスをオンにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. WindowViewer を再起動します。

リモート セッションで実行するアプリケーションへのユーザー アクセスを設定する

リモート デスクトップ接続またはターミナル サービス セッションで実行する分散 InTouch アプリケーションへの読み取り専用アクセスを割り当てることができます。読み取り専用アクセスは、アプリケーションを表示する必要があっても書き込み許可を与えるべきでない非オペレータに適切です。たとえ

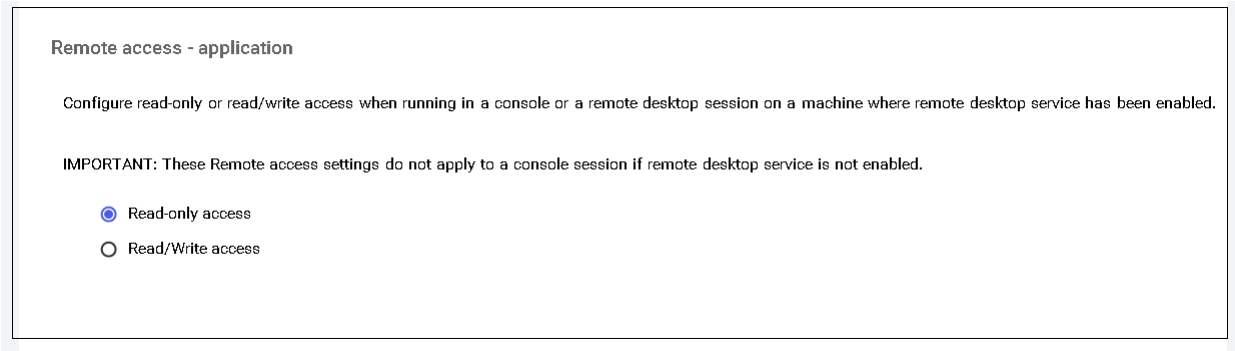
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ば、マネージャは、保護されていない公共ネットワークで InTouch Access Anywhere を使用してモバイルデバイス上のアプリケーションを表示する必要があります。読み取り専用アクセスを使用すると、公共ネットワークからアクセスできる分散 InTouch アプリケーションの保護を向上させることができます。

リモート ノードで実行するアプリケーションにユーザー アクセスを設定する

1. WindowMaker でアプリケーションを開きます。
2. ［ファイル］メニューの［設定］を選択し、［**WindowViewer**］を選択します。
3. WindowViewer 設定画面で［アプリケーション タイプ］タブを選択します。

   ［リモート アクセス］セクションが表示され、WindowMaker で開いているアプリケーションへの［読み取り/書き込み］または［読み取り専用］アクセス権限を付与するオプションが表示されます。［読み取り/書き込み］アクセスがデフォルト設定です。



4. 選択を行い、［**OK**］を選択してダイアログを閉じます。

初期化中、WindowViewer は、アプリケーションがリモート セッションで実行しているかどうか、および読み取り専用として指定されているかどうかを検証します。また、リモート ノードに読み取り専用ライセンスがあることのチェックも行われます。これらの条件がすべて満たされた場合、アプリケーションの InTouch リンクおよびユーザー入力アニメーションは読み取り専用モードでのみ表示できます。

WindowViewer のメモリ管理について

ランタイム時のパフォーマンスを改善するために、WindowViewer がウィンドウとポップアップ シンボルにメモリを使用する方法を設定することができます。特定の条件下では、ウィンドウ、および ShowSymbol アニメーションの使用によって表示されるポップアップ シンボルと ShowGraphic スクリプト関数は、データ取得の高速化を可能にするため、ランタイム時にメモリ上に保存できます。

また、定期的なメモリのヘルス チェックの周期、およびヒープ メモリ セグメントの設定を指定することもできます。新しいポップアップ シンボルが開くたびに、事前に設定されている間隔に関わらず、メモリのヘルス チェックがトリガされます。

WindowViewer への高速切り替え、または WindowViewer から WindowMaker への高速切り替えは、グラフィック キャッシュとウィンドウ メモリ キャッシュの両方を消去します。

ShowGraphic 関数によって表示されるシンボルのメモリ管理の詳細については、『産業用グラフィック エディタ ユーザー ガイド』を参照してください。

WindowViewer ウィンドウのメモリ使用量の設定

ランタイム時のパフォーマンスを改善するために、WindowViewer アプリケーション ウィンドウのメモリ使用量を設定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

特定の条件下では、キャッシュされた閉じたウィンドウを再び開くと、ディスクからではなく、メモリから読み取ります。

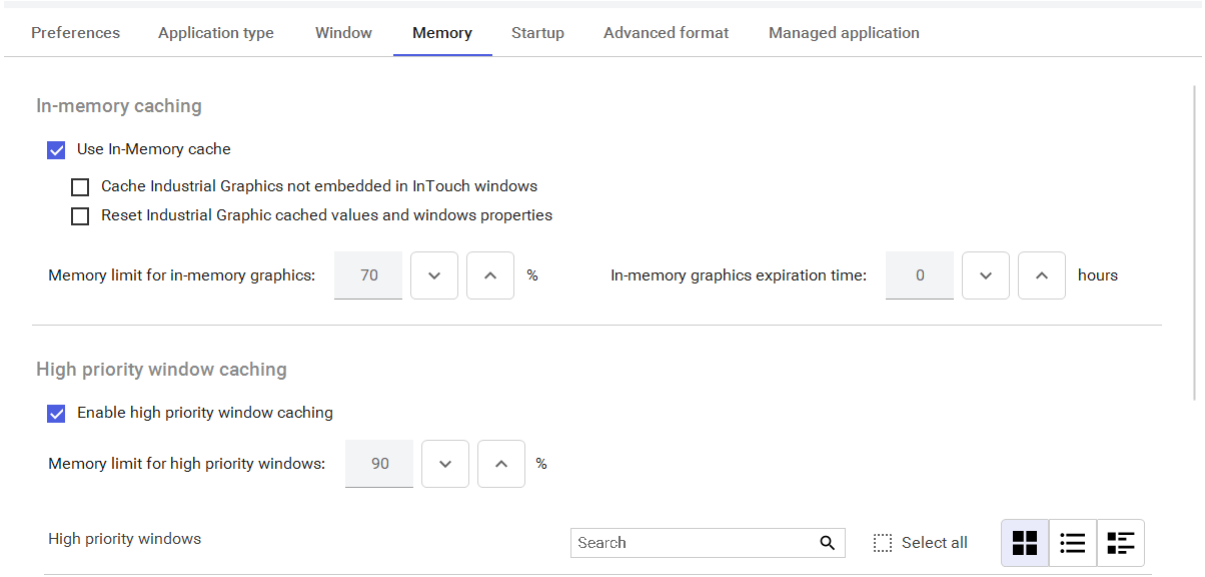
特定のウィンドウに高い優先度のメモリ使用量を指定し、それらのウィンドウだけに個別のメモリ設定を構成することができます。

WindowViewer のメモリ オプションのいずれかを変更した後、WindowViewer が再起動するまで変更は反映されません。

メモリのプロパティを設定するには

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］をクリックし、［**WindowViewer**］をクリックします。

   WindowViewer の設定画面が表示されます。
3. ［メモリ］タブをクリックします。



4. ［メモリ内キャッシュ］領域で、以下の手順を実行します。
   a. ランタイム時にすべての閉じたウィンドウをメモリにキャッシュする場合、［**メモリ内キャッシュを使用する**］チェック ボックスをオンにします。
   b. 産業用グラフィック シンボル キャッシュを有効にする場合、［**InTouch ウィンドウに埋め込まれていない産業用グラフィックをキャッシュ**］チェック ボックスをオンにします。
   c. ウィンドウを閉じて再度開いたときにカスタム プロパティとウィンドウ プロパティをリセットするには、［**産業用グラフィックのキャッシュ値とウィンドウ プロパティをリセット**］チェック ボックスをオンにします。
   d. ［**メモリ内ウィンドウのメモリ制限**］ボックスに、ランタイム時にキャッシュ メモリに保持される閉じたメモリ内ウィンドウの制限値を入力します。デフォルトのメモリ制限は、プロセス メモリの 70% です。

      このメモリ制限を超えると、高優先度ウィンドウとしてマークされていない限り、ランタイム時にキャッシュから最も古い閉じたメモリ内ウィンドウが自動的に削除されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メモリ内ウィンドウのメモリ制限は、高優先度ウィンドウのメモリ制限よりも常に少なくなります。

e. ［**メモリ内ウィンドウのキャッシュ期限**］ボックスに、閉じたメモリ内ウィンドウをランタイム時にキャッシュ メモリに保持しておく期限を入力します。0 ～ 8760 時間の値を入力することができます。デフォルト値は 0 時間（時間制限なし）です。

注意：InTouch ウィンドウと産業用グラフィックは、メモリ内キャッシュを共有します。設定されたメモリ制限を超えると、キャッシュから最も古いメモリ内グラフィックが自動的に削除されます。

メモリ制限またはキャッシュ期限は、どちらの制限に最初に達したかに応じて適用されます。

注意： 実行時に、ウィンドウを再度開くと、それぞれのアニメーションで設定されたデフォルトのテキストが、以前にキャッシュされた値の代わりに表示され、混乱を招きます。この動作でサポートされているアニメーションは、次のとおりです。
値の表示
ユーザーの表示
塗りつぶし色の変更
罫線の変更
テキストの色の変更
ツールチップ
名前付きスタイル

5. ［**高優先度ウィンドウのキャッシュ**］領域で、以下の手順を実行します。

a. ［**高優先度ウィンドウのキャッシュを有効にする**］チェック ボックスをオンにすると、一部のウィンドウを高優先度としてマークできます。これらのウィンドウはランタイム時に閉じられると、常にキャッシュ メモリに保持されます。

b. ［**高優先度ウィンドウのメモリ制限**］ボックスに、ランタイム時にキャッシュ メモリに保持される閉じた高優先度ウィンドウの制限値を入力します。デフォルトのメモリ制限は 90% です。ランタイム時にメモリの使用パーセンテージがこの制限を超えると、最も古いメモリ内ウィンドウが最初に削除され、次に最も古い高優先度ウィンドウが削除されます。

c. ［**高優先度ウィンドウ**］ボックスで、高優先度ウィンドウとしてマークするウィンドウを選択します。

注意：［**メモリ内キャッシュを使用する**］または［**高優先度ウィンドウのキャッシュを有効にする**］チェック ボックスがオンになっている場合、［**産業用グラフィックのキャッシュ値とウィンドウ プロパティをリセット**］チェック ボックスもオンになります。［**産業用グラフィックのキャッシュ値とウィンドウ プロパティをリセット**］チェック ボックスがオンになっている場合、リセット アクションは、［**メモリ内キャッシュを使用する**］または［**高優先度ウィンドウのキャッシュを有効にする**］オプションにも影響します。

6. ［**保存**］をクリックします。

メモリのヘルス チェックの周期を設定する

システムは、メモリとグラフィカル デバイス インターフェイス（GDI）の数を一定の周期でチェックします。メモリまたは GDI の数の制限が超えている場合、システムはウィンドウを削除し始めます。デフォルトでは、5 分周期に設定されています。

周期を変更するには、InTouch.ini ファイルに新しいエントリを追加して、新しい周期の値を指定します。

チェックを行わない場合には、新しいエントリを追加して、その値を 0 に設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

周期を変更した後で、変更を適用するには、WindowViewer を再起動する必要があります。

ウィンドウが削除される方法の詳細については、「WindowViewer ウィンドウのメモリ使用量の設定」を参照してください。

メモリのヘルス チェックの周期を設定する

1. アプリケーションのフォルダの InTouch.ini ファイルを開きます。
2. ［Intouch］セクションに、次のスクリプトを追加します。ここで *<interval>* は、希望する周期を分単位で表したものです。

```
MemoryHealthCheckInterval = <interval>
```

新しいポップアップ シンボルまたは新しいウィンドウを開くと、事前に設定されている間隔に関わらず、それはメモリのヘルス チェックをトリガします。

wwHeap メモリを設定する

wwHeap はヒープ メモリ セグメントを管理するメモリ マネージャです。このメモリ マネージャは、1 つまたは複数のプログラムが仮想メモリを共有するためのメカニズムを提供します。

警告：wwHeap のメモリ設定は、ロガーでメモリの競合が報告された場合にのみ、変更してください。

wwHeap のサイズと場所を指定することで、wwHeap メモリを設定することができます。デフォルトのサイズ、デフォルトの開始場所、および可能な場所の範囲は、オペレーティング システムごとに異なります。

デフォルトのサイズについては、次の表を参照してください。

オペレーティング システム	デフォルトのサイズ
32 ビット	1519 MB
32 ビット（/3GB スイッチ有効）	2048 MB
64 ビット	2048 MB

デフォルトの場所と可能な場所の範囲については、次の表を参照してください。

オペレーティング システム	デフォルトの開始場所	可能な範囲
32 ビット	0x21000000	0x00010000 ～ 0x7FFEFFFF
32 ビット（/3GB スイッチ有効）	0x40000000	0x00010000 ～ 0xBFFEFFFF
64 ビット	0x80000000	0x00010000 ～ 0xFFFEFFFF

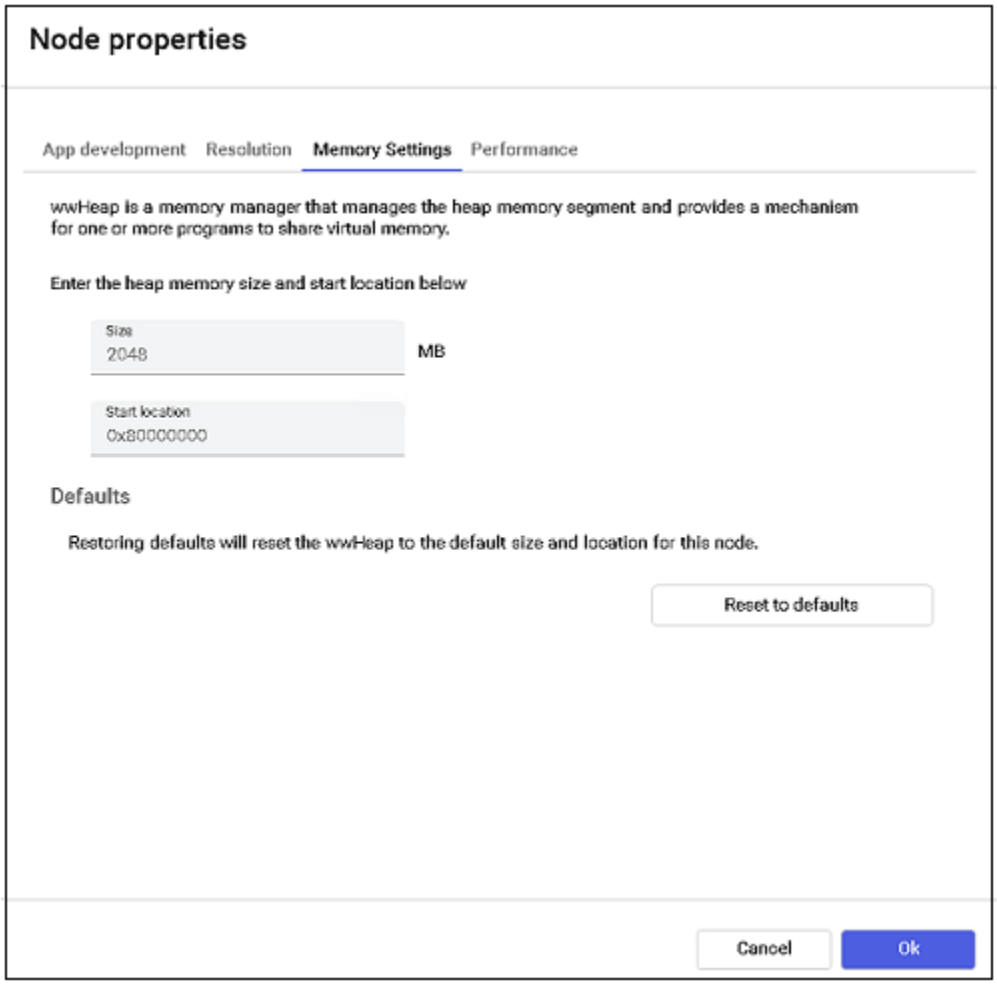
wwHeap メモリを設定する

1. **アプリケーション マネージャ**を起動します。
2. ［**ツール**］メニューで、［**ノードのプロパティ**］を選択します。

   ［**ノードのプロパティ**］画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［メモリ設定］タブを選択します。



4. 以下の手順を実行します。
   - ［サイズ］ボックスに、wwHeap メモリのサイズを入力します。1 ～ 2048 MB の値を入力することができます。
   - ［開始場所］ボックスに、開始場所のアドレスを入力します。
5. デフォルト値に戻す場合は、［デフォルトの復元］を選択します。
6. ［OK］を選択します。

WindowViewer オプションの［地域の設定］を選択する

InTouch WindowMaker には、WindowViewer の詳細な書式設定［地域の設定］設定オプションが含まれます。

地域ロケールによる数値形式を有効にするには、デザイン時に［地域の設定］オプションを選択して、産業用グラフィックの数値を［地域］設定で選択されている国に設定する必要があります。デフォルトでは、［地域の設定］オプションは無効になっています。

OS の地域の設定がチェックされるのは、WindowViewer の起動時だけです。したがって、次のいずれかの操作を行った場合、WindowViewer を再起動する必要があることがあります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- WindowViewer が実行している間に［**地域の設定**］オプションを選択する。
- ［**地域の設定**］オプションが選択されている状態で WindowViewer が実行しているときに OS の地域の設定を変更する。

詳細な書式設定のプロパティを設定する

WindowViewer では、値の大きさに基づいて、ランタイム時に表示する小数点以下の桁数を設定することができます。この機能は、［値の表示］または［ユーザー入力］アニメーションで［実数］を選択した場合にのみ使用可能になります。

フィールド デバイスから収集したデータの品質が低い場合、または大きすぎて表示できない場合に、ランタイム時に表示される特殊文字を指定できます。

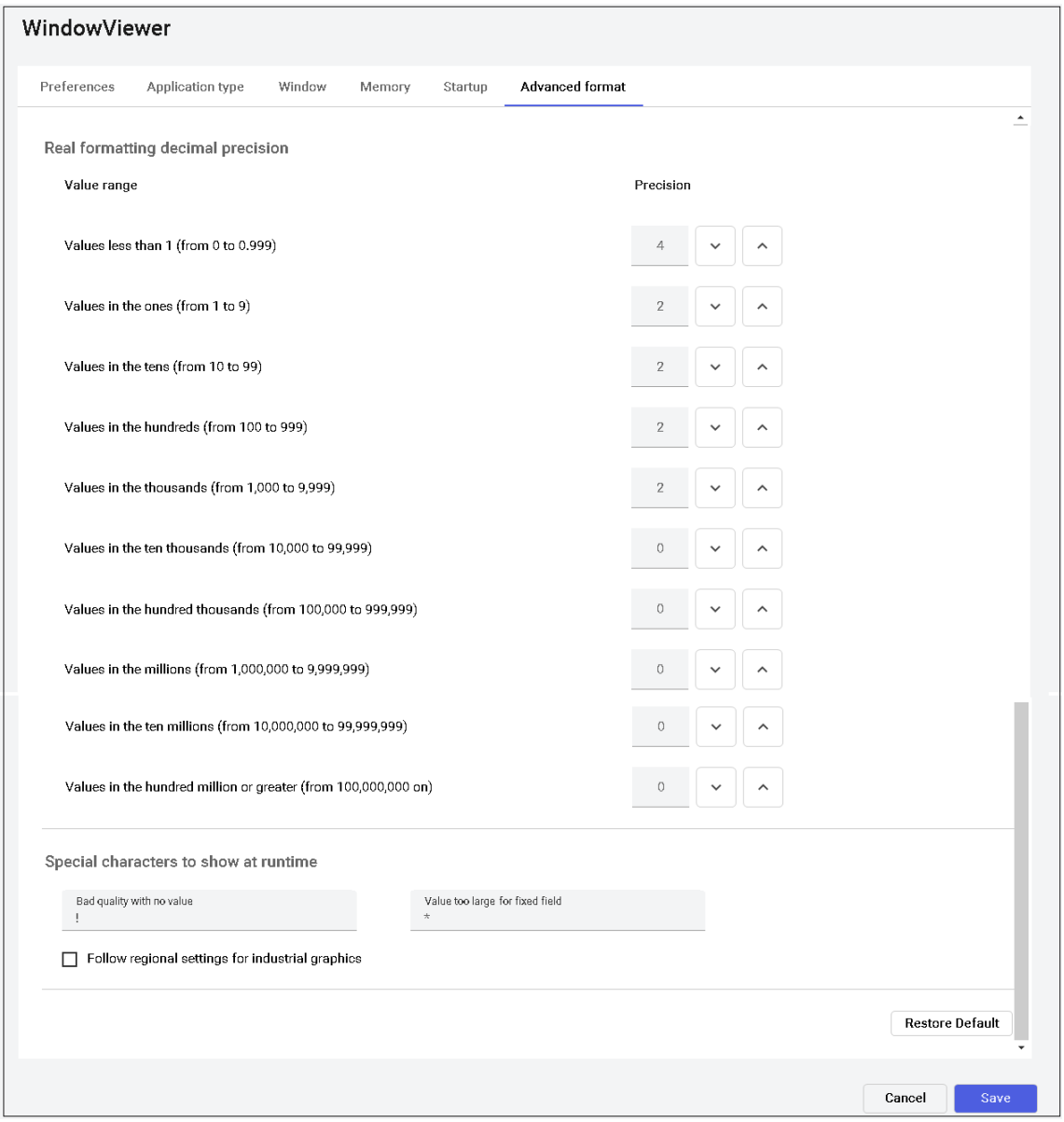
産業用グラフィックに表示される数値は、Windows のコントロール パネルの［**地域**］設定の主な使用場所に設定された国の形式で表示されます。ランタイム時、産業用グラフィックの数値は、OS の地域の設定で指定された国の数値形式に一致する 1000 桁区切り文字を使用して表示できます。

詳細な書式設定のプロパティを設定する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］を選択し、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［**詳細な書式設定**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**小数点精度の書式設定**］セクションに、実数タイプの値の範囲ごとに、ランタイム時に表示される数値の小数点以下の桁数を入力します。
5. ［**ランタイム時に表示する特殊文字**］セクションで、以下の手順を実行します。
   - ［**値を持たない低クォリティ**］ボックスに、データ ポイントの品質が低すぎるか、または良い品質の最後の既知の値がない場合に、ランタイム時に表示される文字を入力します。デフォルトの文字は **!** ですが、スペース以外の ASCII 文字を入力することができます。
   - ［**固定幅よりも大きい値**］ボックスに、値が大きすぎて表示できない場合に、ランタイム時に表示される文字を入力します。デフォルトの文字は ***** ですが、スペース以外の ASCII 文字を入力することができます。
6. コンピュータの［**地域**］設定で指定されている国の形式で産業用グラフィック内に数値を表示するには、［**産業用グラフィックの地域の設定に従う**］チェック ボックスをオンにします。

   デフォルトでは、［**地域の設定**］オプションは無効になっています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意： OS の地域の設定がチェックされるのは、WindowViewer の起動時だけです。したがって、
1) WindowViewer が実行しているときに［**地域の設定**］オプションを選択した場合、または
2) ［**地域の設定**］オプションが選択されている状態で WindowViewer が実行しているときに中に OS の地域の設定を変更した場合、WindowViewer を再起動する必要があることがあります。

7. デフォルトの値に戻すには、［**デフォルトの復元**］を選択します。

HMI/SCADA アプリケーションをホストするコンピュータの地域ロケールを設定する

地域ロケールによる数値形式を有効にするには、HMI/SCADA アプリケーションを実行するコンピュータの地域を産業用グラフィックの数値の書式として使用する国に設定します。

地域の設定は、Windows の［コントロール パネル］からアクセスできます。産業用グラフィックの数値を米国以外の形式で表示するには、［**形式**］タブを選択して、［**形式**］フィールドで国を選択します。

ターミナル サーバー環境での WindowViewer のコア親和性を設定する

コンピュータに複数のプロセッサが搭載されている場合、ターミナル サービス クライアントの実行時に、WindowViewer を実行するために CPU 0 以外の CPU（コア）を使用できます。これは、ターミナル サーバー環境で動作する InTouch アプリケーションが、ターミナル サーバーのマルチコア機能を活用できるようにするためです。ただし、WindowViewer がターミナル サーバー コンソールで動作している場合、このオプションは使用できません。InTouch アプリケーションは、利用できるプロセッサ数には関わりなく、常に 1 つのプロセッサ上で動作します。

WindowViewer が起動すると、システムは、コンピュータに複数のプロセッサが搭載されているかということと、どのプロセッサで WindowViewer のインスタンスを実行することが許されているかをチェックします。WindowViewer は次にプロセッサを順にチェックして、実行されている View インスタンスの最も少ないプロセッサで実行を開始します。

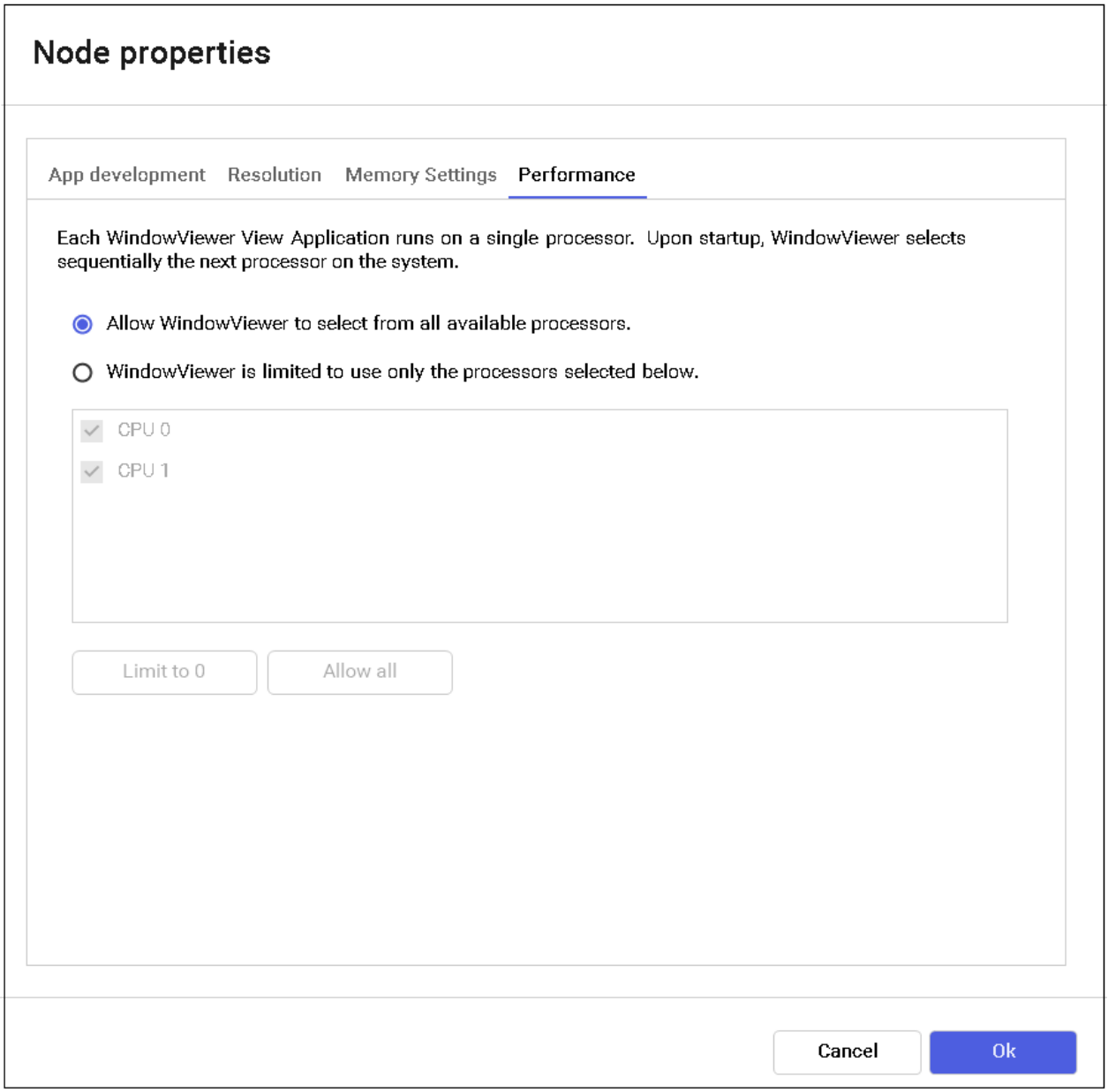
オペレータが管理特権を持っていて［**パフォーマンス**］タブにアクセスできる場合には、プロセッサの「プール」を設定して、そこから WindowViewer が動作するプロセッサを選択しようにすることができます。

WindowViewer のコア親和性は、アプリケーション マネージャで設定します。タスク マネージャで WindowViewer のコア親和性を手動で調整しなくても済むように、WindowViewer のコア選択プロセスは、タスク マネージャのコア親和性設定を考慮しないようになっています。

プロセッサ「プール」を設定する

1. **アプリケーション マネージャ**を起動します。
2. ［**ツール**］メニューで、［**ノードのプロパティ**］を選択します。［**ノードのプロパティ**］ダイアログ ボックスが表示されます。
3. ［**パフォーマンス**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. WindowViewer が利用可能なプロセッサをすべて使用できるようにするには、[**WindowViewer がすべての利用可能なプロセッサから選択することを可能にします**] を選択します。
5. WindowViewer が使用するプロセッサを制限するには、[**WindowViewer は、以下の選択されたプロセッサのみの使用に制限されます**] を選択して、以下のいずれかを実行します。
   - WindowViewer が動作できる各 CPU に対応する [**CPU**] チェック ボックスをオンにします。
   - WindowViewer がプロセッサ 0 でのみ動作するようにする [**0 に制限**] を選択します。このボタンを選択すると、[**CPU 0**] チェック ボックスは自動的にオンになります。
   - すべてのチェック ボックスをオンにするには、[**すべてを許可**] を選択します。

   選択したプロセッサは、いつでもオフにすることができます。また、リスト内の新しいプロセッサをオンにすることもできます。また、一度に複数のプロセッサをオンにすることもできます。ある

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロセッサのチェック ボックスをオフにしても、WindowViewer のインスタンスはそのプロセッサでの動作を続けます。

6. ［**OK**］を選択します。WindowViewer は、他の View セッションに基づき、次の CPU で動作を開始します。

WindowViewer ウィンドウを操作する

通常、InTouch アプリケーションには、生産プロセスを管理するためにオペレータが操作するウィンドウがいくつかあります。WindowViewer の設定画面の［**ウィンドウ**］タブで設定したプロパティに基づいて、オペレータは WindowViewer の［**ファイル**］メニューから標準的なコマンドを実行してウィンドウを開くことや閉じることができます。

さまざまなモードで WindowViewer を表示する

［**詳細**］を選択して、一覧表示を詳細表示に切り替えます。ウィンドウのタイプ、ウィンドウが最後に変更された日時などの詳細が表示されます。

詳細表示では、チェック ボックスだけではなく、行の部分を選択しても、現在開いていないウィンドウを選択したり選択解除したりできます。選択すると行全体がハイライトされます。

- 選択したウィンドウを開くには、［**OK**］を選択します。
- 選択をキャンセルしてダイアログ ボックスを閉じるには、［**キャンセル**］を選択します。
- ダイアログ ボックスを元の一覧表示に戻すには、［**リスト**］を選択します。
- 表示されたウィンドウをすべて選択するには、［**すべて選択**］を選択します。
- すべての選択ウィンドウを解除するには、［**すべて消去**］を選択します。
- 昇順または降順でリストをソートするには、カラム ヘッダーを選択します。

WindowViewer からウィンドウを開く

［**ファイル**］メニューを表示するように WindowViewer を設定すると、オペレータは InTouch アプリケーション ウィンドウを開いたり閉じたりできます。［ファイル］メニューでオペレータが［**開く**］コマンドまたは［**閉じる**］コマンドのどちらかを選択すると、選択したコマンドに対応するダイアログ ボックスが表示されます。

選択したコマンドに適用可能なすべてのウィンドウの名前が、リストに表示されます。たとえば、［**ウィンドウを開く**］コマンドを選択した後に、［**表示するウィンドウ**］画面が表示されます。

［**ファイル**］メニューを表示するように WindowViewer が設定されている場合、オペレータは InTouch アプリケーション ウィンドウを開くことができます。

WindowViewer からウィンドウを開く

1. ［**ファイル**］メニューで、［**開く**］を選択します。

   ［**ウィンドウを開く**］ダイアログ ボックスが表示されます。

2. 開く各ウィンドウの名前の横にあるチェック ボックスをオンにします。
1. ［**OK**］を選択してダイアログ ボックスを閉じ、選択したウィンドウを開きます。

注意：「リプレース」タイプのウィンドウを選択すると、このウィンドウと交差しているウィンドウはすべて閉じます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowViewer からウィンドウを閉じる

［ファイル］メニューを表示するように WindowViewer が設定されている場合、オペレータは InTouch アプリケーション ウィンドウを閉じることができます。

開いているウィンドウを閉じる

1. ［ファイル］メニューで、［ウィンドウを閉じる］を選択します。

   ［閉じるウィンドウを選択］画面が表示されます。

2. 閉じようとしている 1 つ以上のウィンドウの名前の横にあるチェック ボックスをオンにします。
3. ［**OK**］を選択して、選択したウィンドウを閉じます。

WindowViewer から WindowMaker に切り替える

InTouch アプリケーションを開発するとき、［ファイル］メニューの［WindowMaker］コマンドまたはツールバーの［開発］コマンドを選択することにより、WindowMaker と WindowViewer を簡単に切り替えることができます。これを高速切り替えと言います。

高速切り替えは、短期間での開発テスト時専用です。運環境では使用しないでください。WindowMaker に切り替えるためのコマンドを非表示にすることができます。

WindowViewer から WindowMaker に切り替える

1. WindowViewer を起動します。
2. ［ファイル］メニューで、［**WindowMaker**］を選択します。

   ［ウィンドウの編集］ダイアログ ボックスが表示されます。

3. WindowMaker に切り替えるときに開く各ウィンドウの名前の横にあるチェック ボックスをオンにします。
4. ［**OK**］を選択してダイアログ ボックスを閉じ、WindowMaker に切り替えます。

注意：開発中に WindowViewer のプロパティを設定したときに、アプリケーション開発者が［**WindowViewer を自動的に終了する**］オプションを選択した場合、WindowMaker に切り替えるときに WindowViewer が自動的に終了します。

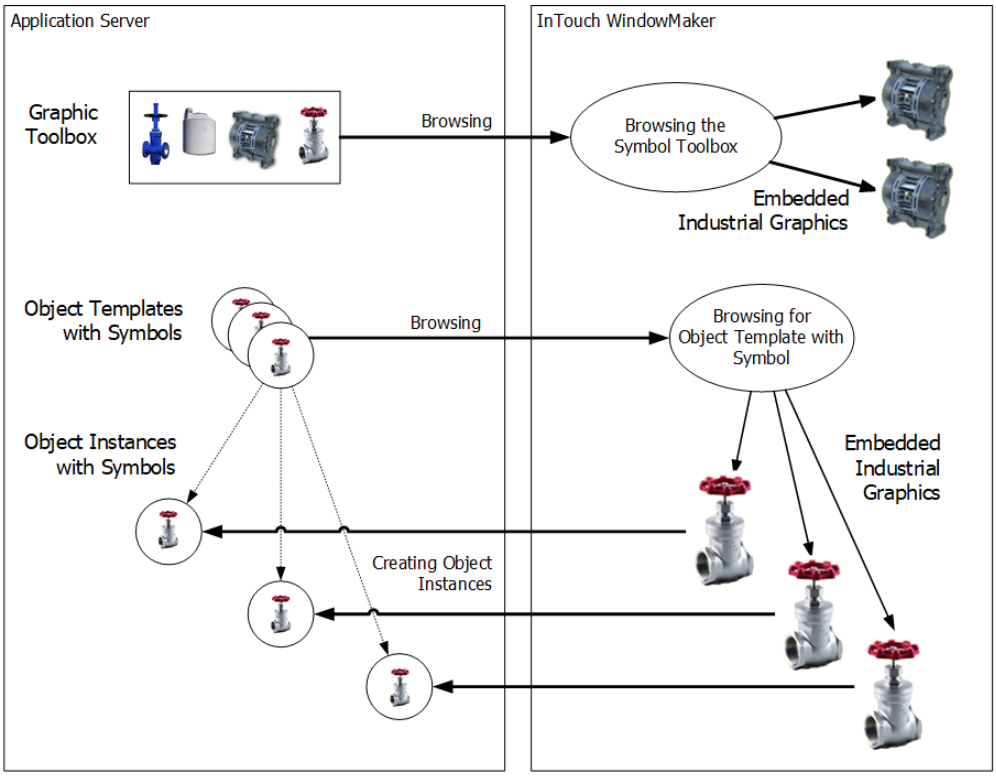
産業用グラフィック

System Platform IDE には、製造プロセスの表現やオートメーション オブジェクトへの HMI インターフェイスの提供を行うシンボルを作成する産業用グラフィック エディタが含まれています。

以下の図は、産業用グラフィック エディタで作成したシンボルが InTouch アプリケーションでどのように使用されるのかを示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Using Industrial Graphics



詳細については、「産業用グラフィックの概要」というタイトルのビデオをご覧ください。

産業用グラフィックの作成

産業用グラフィックは IDE の産業用グラフィック エディタで作成します。

以下を作成できます。

- 産業用グラフィック（産業用グラフィック ツールボックスを使用します）。これらは、特定のオブジェクト テンプレートやオブジェクト インスタンスには関連付けられていません。
- 特定のオブジェクト テンプレートまたはインスタンスに含まれる産業用グラフィック。

産業用グラフィック エディタ

System Platform IDE 内で InTouch アプリケーションを管理する利点に加え、産業用グラフィック エディタでグラフィックを作成することにより製造環境をモデル化できます。産業用グラフィック エディタは System Platform IDE に完全に統合されており、強力な編集機能をサポートします。

産業用グラフィック エディタでは、シンボル要素に要素スタイルを適用できます。要素スタイルは、グラフィック要素に適用されるテキスト、要素の塗りつぶし、要素の輪郭、および線の表示プロパティの定義済みセットです。要素スタイルを使用すると、標準の表示スタイルをグラフィック要素に追加して、一貫したシンボル標準を容易に確立できます。

産業用グラフィック エディタにはシンボル ウィザード エディタも含まれているので、さまざまな表示設定および機能設定を含むシンボル ウィザードを作成できます。シンボル ウィザード エディタを使用すると、左右の弁を個々に設定できるポンプ シンボルを作成することなどが可能です。Situational Awareness Library シンボルは、構築済みの設定を含むシンボル ウィザードのサンプルのコレクション

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

で、シンボルのウィザード オプションから選択できます。単一の Situational Awareness Library シンボルは、容易に設定して温度計、風量系、または圧力計を表すことができます。シンボル ウィザードがマネージド InTouch アプリケーションに埋め込まれている場合、必要な設定を選択します。

WindowMaker には、産業用グラフィック エディタを開くことなく InTouch アプリケーションの産業用シンボルと Situational Awareness Library シンボルを選択できる産業用グラフィック ツールボックスが組み込まれています。シンボルは、WindowMaker から InTouch ウィンドウに直接ドラッグ アンド ドロップすることによって追加できます。産業用シンボルと Situational Awareness Library シンボルの操作に関する詳細については、『産業用グラフィック エディタ マニュアル』または WindowMaker ヘルプを参照してください。

注意 : Microsoft のレンダリング技術の最近のバージョンにおいては、特定のグラデーション機能が廃止されました。将来的な対応が考慮されたグラフィックを使用できるよう、影響を受ける機能が設定環境から削除されました。廃止された機能を使用して以前に設定されたグラフィックについては、引き続き期待通りにレンダリングされます。『産業用グラフィック エディタ ユーザー マニュアル』の「廃止された機能を含むグラフィックのロード」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 2 アプリケーション

InTouch アプリケーション マネージャまたは IDE を使用して、InTouch アプリケーションを管理できます。

InTouch アプリケーションは、管理方法、サポートされるシンボルの種類、およびパブリッシュ元によって分類されます。

- スタンドアロン InTouch アプリケーション: スタンドアロン アプリケーションはアプリケーション マネージャで作成されるデフォルトのアプリケーションで、InTouch シンボルと産業用グラフィックの柔軟性を提供します。
- マネージド InTouch アプリケーション: マネージド アプリケーションは、IDE を使用して作成および管理されます。
- パブリッシュ済み InTouch アプリケーション: 派生した InTouchViewApp テンプレートからマネージド アプリケーションをエクスポートすることにより、パブリッシュ済みアプリケーションを作成できます。
- InTouchView アプリケーション: InTouchView アプリケーションは InTouch アプリケーション マネージャまたは IDE から作成できます。

InTouch HMI は、古いモダン アプリケーションの移行およびアップグレードをサポートします。

スタンドアロン InTouch アプリケーション

スタンドアロン InTouch アプリケーションは、InTouch アプリケーション マネージャによって管理されます。これらは、InTouch アプリケーション マネージャに［**スタンドアロン**］として表示されます。

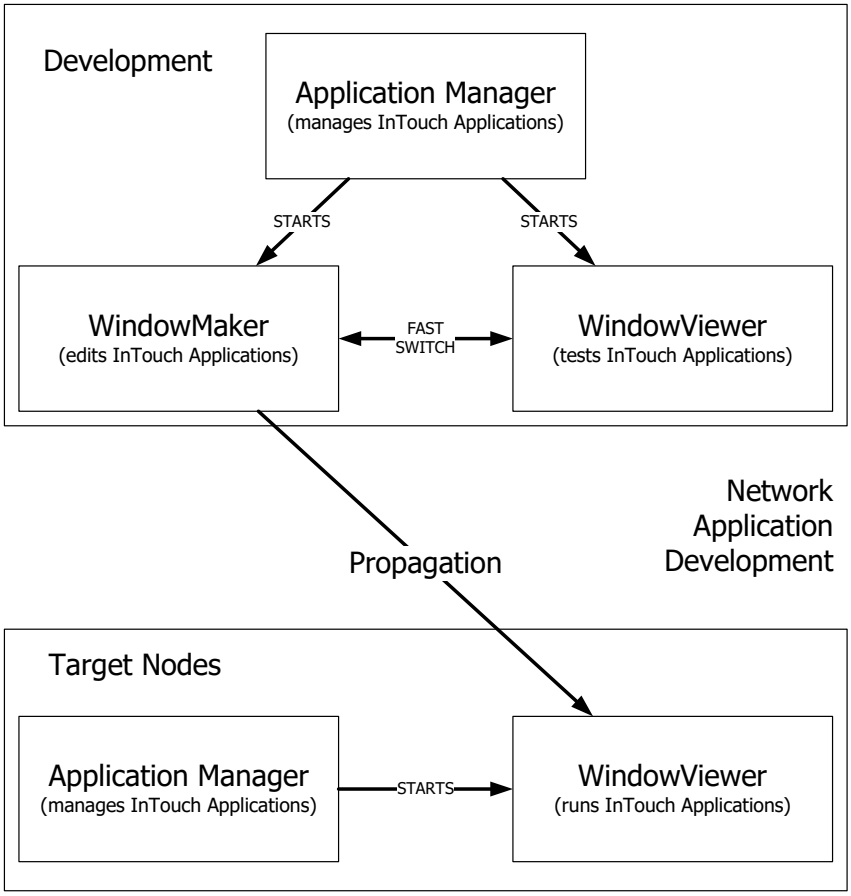
アプリケーション マネージャを使用すると、以下を行うことができます。

- スタンドアロン InTouch アプリケーションの作成と管理。
- WindowMaker の起動と InTouch アプリケーションの編集
- WindowViewer の起動と InTouch アプリケーションの実行

また、アプリケーションをテストまたは実行したり、アプリケーションに変更を加えるために切り替える目的で、WindowMaker と WindowViewer 間で直接切り替えることもできます。スタンドアロン アプリケーションは、ディレクトリ ファイル システムにある InTouch HMI によって保持されている一連のファイルで構成されます。これには産業用グラフィックを含めることもできます。スタンドアロン アプリケーションは複数のネットワーク ノード上に配置でき、単一のノードに制限されていません。IDE にインポートすることやマネージド アプリケーションに変換することができます。

開発ノード上の InTouch アプリケーションから対象ノードで実行中の InTouch アプリケーションへの変更の反映は、ネットワーク アプリケーション開発によって管理されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



マネージド InTouch アプリケーション

System Platform IDE（統合開発環境）を使用して、InTouch アプリケーションを管理できます。これらのアプリケーションは、マネージド InTouch アプリケーションと呼ばれます。

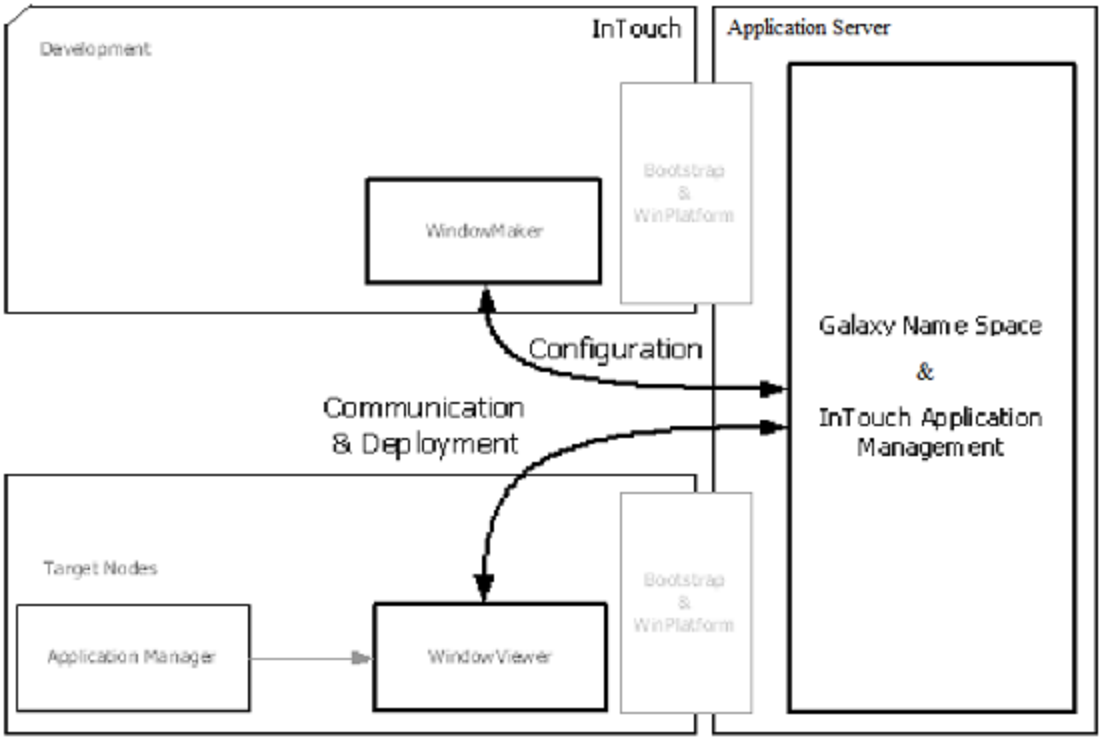
これらのアプリケーションは、InTouch アプリケーション マネージャに［**マネージド**］として表示されます。マネージド InTouch アプリケーションは ArchestrA 環境と密接に統合され、高度なグラフィックをサポートします。

InTouch アプリケーションは、Galaxy の 1 つのノード上で WindowMaker を使用して開発します。その後、WindowViewer が実行されている 1 つまたは複数の対象ノードに配置します。

IDE のシステム プラットフォーム機能を使用して InTouch アプリケーションを管理すると、以下を行うことができます。

- どのノードでどの InTouch アプリケーションが実行されているのかの参照。
- InTouch アプリケーションの中央リポジトリの使用。
- リモート ノードで実行している WindowViewer への変更の配置。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



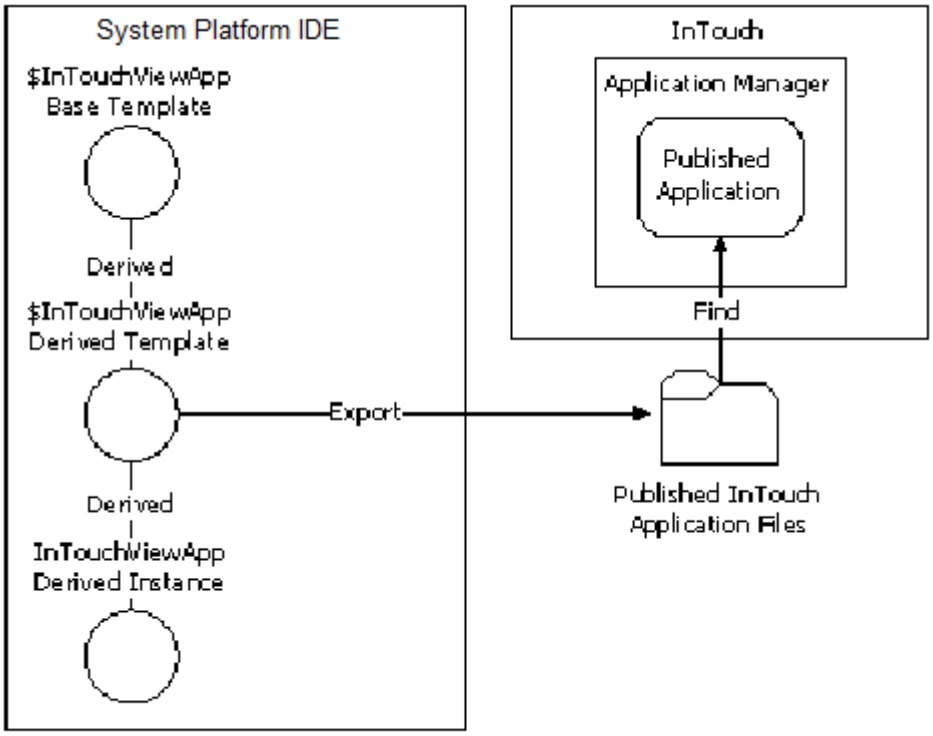
WindowViewer でマネージド アプリケーションを起動できるのは、アプリケーション マネージャからだけです。WindowViewer を起動すると、ランタイム中にマネージド アプリケーションのファイルがフォルダにコピーされます。

各マネージド アプリケーションは、ベース テンプレートから派生した InTouchViewApp オブジェクトに関連付けられています。InTouchViewApp オブジェクトには、マネージド InTouch アプリケーション フォルダおよびマネージド InTouch アプリケーションのその他の動作固有の情報への参照のみが含まれています。アプリケーション ファイルは、Galaxy ファイル リポジトリ内の個別のフォルダに保存されます。1 つのフォルダには最新のチェックイン バージョンの InTouch アプリケーションが含まれ、その他のフォルダには最新のチェックアウト バージョンが含まれます。IDE には産業用グラフィック エディタが含まれており、InTouch HMI アプリケーション内で製造プロセスを表すシンボルの作成に使用できます。IDE から WindowMaker を開いた場合のみ、WindowMaker と WindowViewer をすばやく切り替えてマネージド アプリケーションをテストできます。

パブリッシュ済み InTouch アプリケーション

マネージド InTouch アプリケーションを編集した後、パブリッシュすることができます。派生した InTouchViewApp テンプレートからマネージド アプリケーションをパブリッシュできます。マネージド アプリケーションをパブリッシュすると、InTouch アプリケーション ファイルおよびアプリケーションに埋め込まれた産業用グラフィックが含まれるユーザー定義フォルダが作成されます。フォルダを検索するには、アプリケーション マネージャの［検索］ユーティリティを使用します。その後、変換されたアプリケーションが、パブリッシュ済みアプリケーションとしてアプリケーション マネージャに表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



パブリッシュ済み InTouch アプリケーションの利点は、スタンドアロン アプリケーションのように分散しつつ、産業用グラフィックの機能を引き続きサポートする点です。

ただし、以下を行うことはできなくなります。

- System Platform IDE の使用による InTouch アプリケーションの配置。
- InTouch アプリケーションの産業用グラフィックの編集または追加。
- パブリッシュ済み InTouch アプリケーションの編集。

WindowMaker を使用して、パブリッシュ済みアプリケーションをバージョン 11.1（2014 R2 Patch 01）以前のバージョンからバージョン 11.1 Patch 01 の InTouch に移行できます。その後、アプリケーション ソース ファイルを更新して再度パブリッシュすることにより、パブリッシュ済みアプリケーションを変更できます。マネージド アプリケーションをパブリッシュした後でも、埋め込まれた産業用グラフィックを引き続き使用して、Galaxy にデータを書き込むことやデータを視覚化することができます。パブリッシュ済みアプリケーションを Galaxy に再度インポートすることはできません。パブリッシュ済みマネージド アプリケーションを移行した後、アプリケーションを再度パブリッシュする必要があります。再パブリッシュの際、埋め込まれたアラーム コントロールが新しいバージョンにアップグレードされます。

InTouchView アプリケーション

InTouchView アプリケーションには、Application Server 環境で使用するための視覚的なインターフェイスが表示されます。InTouchView アプリケーションは WindowViewer で動作し、HMI のほとんどの機能は Application Server から提供されます。InTouch タグ サーバーとして機能するようアプリケーションを設定して、その他のノードにセキュアなタグ情報を提供できます。InTouchView アプリケーションは、ク

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ライアント ノードで必要な機能がデータへのアクセスだけで、開発ノードの完全な機能は必要ない場合に便利です。

InTouchView アプリケーションでは、フル機能の InTouch アプリケーションで使用できる標準的な関数の一部のみが提供されています。InTouchView アプリケーションの特徴を以下に示します。

- Application Server Galaxy や InTouch タグ サーバー以外の I/O ソースには接続できません。
- アラームを生成できません。しかし、Application Server オブジェクトや InTouch アラームなどのリモート アラーム プロバイダからアラームを表示および確認できます。
- アプリケーション データやイベントをログに記録しません。InTouchView アプリケーションでは、SYS およびユーザー関連イベントのみが生成されます。
- アプリケーションで消費される Galaxy のデータを保護できるのは、ArchestrA セキュリティだけです。
- 埋め込まれた産業用グラフィック内のタグを参照できません。

InTouch アプリケーションは WindowMaker で開発し、WindowViewer で実行します。その後、InTouch アプリケーションを InTouchView アプリケーションに変更します。InTouchView では、Application Server を介して InTouch アプリケーションを管理できます。同様に、InTouchView アプリケーションを InTouch アプリケーションに変更できます。

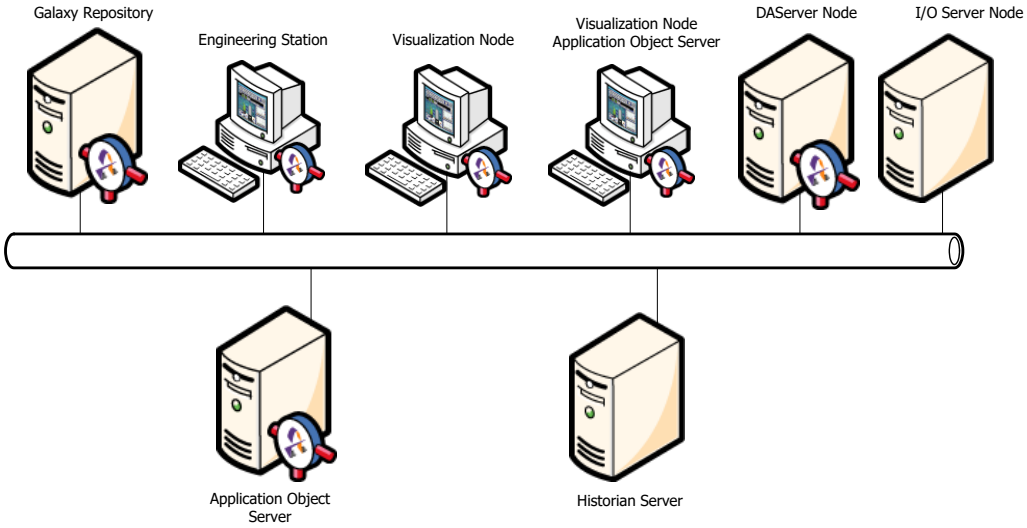
InTouchView アプリケーションを作成するときに使用できない WindowMaker コマンドおよび［タグ名ディクショナリ］オプションを次のリストに示します。

- 使用できないコマンド:
  - アクセス名
  - アラーム グループ
  - 設定...アラーム
  - 設定...履歴ログ
  - 設定...分散名前マネージャ
- 使用できないタグ名ディクショナリ オプション:
  - アラーム
  - 両方
  - データ ログ
  - イベント ログ
  - 優先度

Application Server のアーキテクチャ

Application Server は、分散 InTouch HMI アプリケーションにサービスを提供するために ArchestrA 技術を使用します。Application Server のサービスは、アプリケーション インターフェイスを提供するために、ビジュアル化ノードに InTouch HMI がインストールされている一連のノード上に分散されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



Application Server Galaxy は、複数のノードに分散されたさまざまなコンポーネントを組み込んだ、単一の論理ネームスペースに基づいたシステムの設定を表します。

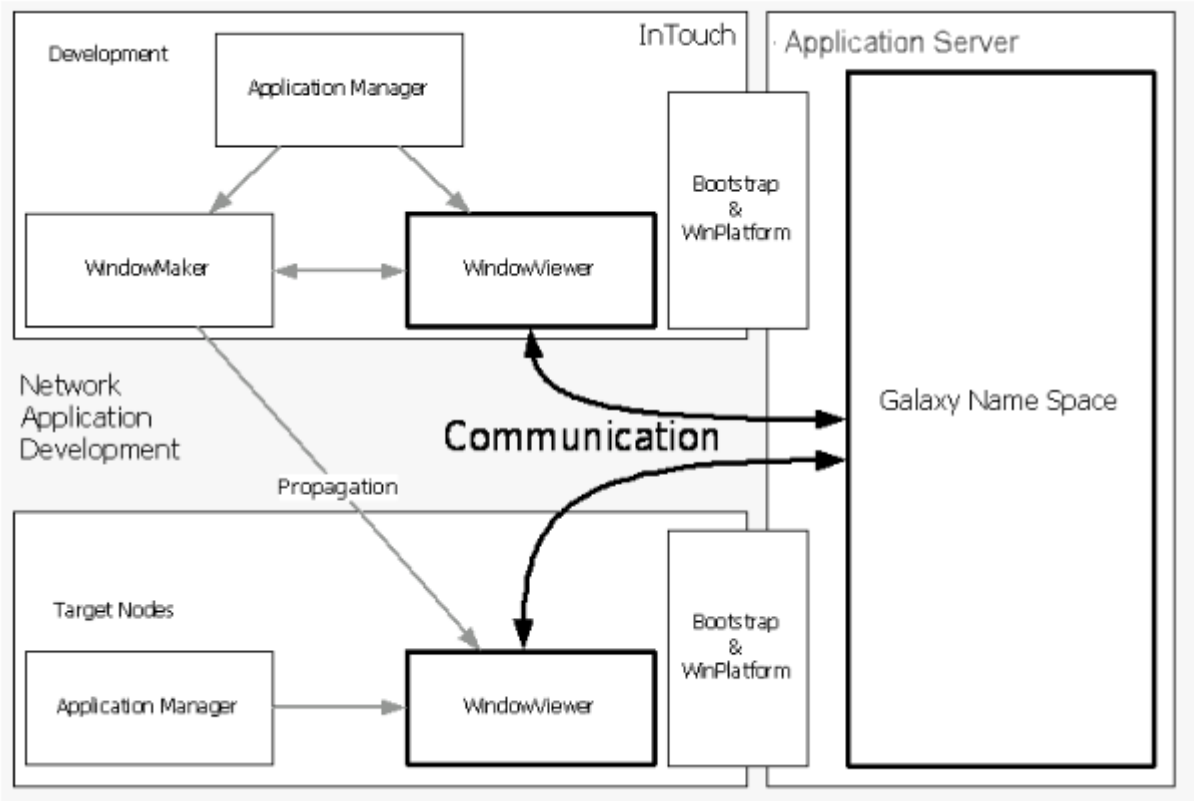
Application Server には InTouch タグ変数に類似した製造データを保存できますが、より多くのデータ タイプおよび配列がサポートされています。Application Server では、InTouch アラーム システムと互換性のあるアラーム機能が提供されています。Application Server では、.NET 関数の有効な機能強化がサポートされた、InTouch HMI と互換性のあるスクリプト言語が提供されています。Galaxy データの読み取りおよび書き込みを可能にするには、InTouch ノードに Application Server Bootstrap コンポーネントをインストールする必要があります。Galaxy の参照を可能にするには、Application Server IDE コンポーネントをインストールする必要があります。

詳細については、Application Server のマニュアルを参照してください。

Galaxy との通信

ArchestrA を使用すると、Galaxy 全体で名前空間を使用する際に、生産関連のデータを含めたり処理することが可能になります。また、ハイ レベルなビジュアル化、および製造環境で InTouch が実行されているさまざまなノードからのデータ アクセスの管理も可能になります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



スタンドアロン、マネージド、パブリッシュ済みの InTouch アプリケーションの比較

以下の表で説明するように、スタンドアロン、マネージド、およびパブリッシュ済みの InTouch アプリケーションにはいくつかの相違点と類似点があります。

	スタンドアロン InTouch アプリケーション	**マネージド InTouch アプリケーション**	**パブリッシュ済み InTouch アプリケーション**
アプリケーションの作成	アプリケーション マネージャ	System Platform IDE • 新しいアプリケーション • スタンドアロン アプリケーションのインポート • SmartSymbol のインポート	非サポート
アプリケーションの編集	アプリケーション マネージャから起動した WindowMaker	IDE 内から起動した WindowMaker	非サポート
アプリケーションの削除	アプリケーション マネージャからの削除	InTouchViewApp テンプレートの削除	アプリケーション マネージャからの削除

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	スタンドアロン InTouch アプリケーション	マネージド InTouch アプリケーション	パブリッシュ済み InTouch アプリケーション
産業用グラフィックのサポート（シンボル ウィザードを含む）	可	すべての操作でサポート	表示のみサポート、作成および編集は非サポート
DB ダンプと DB ロードのサポート	サポート（アプリケーション マネージャ内の関数）	IDE 内の関数でサポート	サポート（アプリケーション マネージャ内の関数）
元の解像度でのアプリケーションの編集における変換	可	不可	可
分散アプリケーションの管理	ネットワーク アプリケーション開発（NAD）	System Platform IDE	ネットワーク アプリケーション開発（NAD）
新しい InTouch アプリケーション バージョンが受け入れられる方法の設定	アプリケーション マネージャで設定（ネットワーク アプリケーション開発）	WindowMaker で設定	アプリケーション マネージャで設定（ネットワーク アプリケーション開発）
アプリケーションのテストで高速切り替えの使用	可	Yes	可
タグ変数の値およびタグ変数パラメータの保持の使用	可	サポート（ローカル ディレクトリの設定も必要）	可

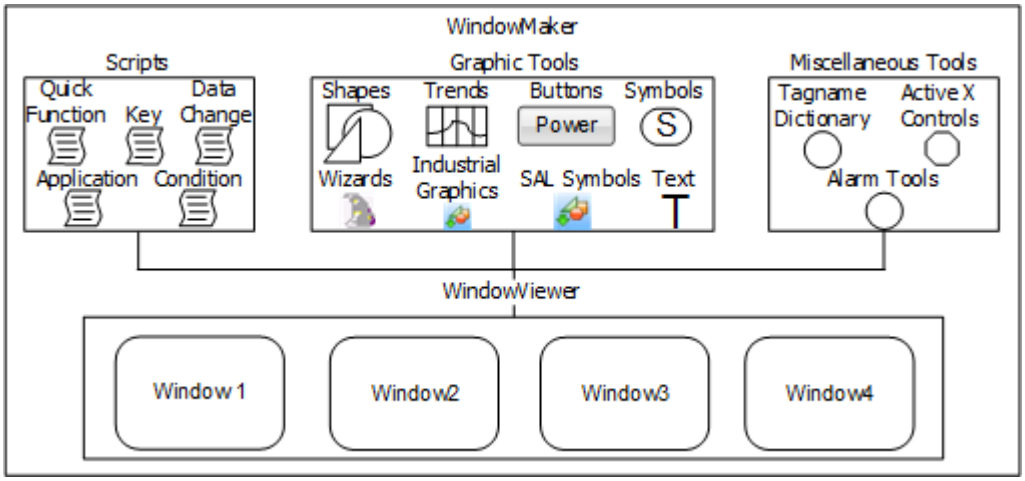
アプリケーションを構築する

WindowMaker では、アプリケーションのウィンドウに表示されるオブジェクトの動作を定義するグラフィック ツール、スクリプト言語、およびタグ変数管理ユーティリティが提供されます。WindowMaker を使用すると、ウィンドウ オブジェクトに関連付けられているデータ ポイントを表すタグ変数を作成できます。製造工程のデータは、最終的にはタグ変数の値として関連付けられます。このタグ変数デー

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タは、アラームの監視、トレンドの作成、およびランタイム中のアプリケーションの動作を決定するためにアプリケーションで使用できます。

以下の図は、InTouch HMI アプリケーションを作成するためのいくつかの WindowMaker のツールを示しています。



より複雑なオブジェクトを作成するために組み合わせることができるシンプルな図形から、事前定義されたプロパティを持つ標準の産業用グラフィックまで、豊富な種類のグラフィック ツールを使用できます。トリガのメカニズムに基づいて、異なるタイプのスクリプトを作成できます。また、事前定義された InTouch 関数をスクリプトに挿入することもできます。タグ変数がいつ通常の状態またはアラーム状況になるのかを決定するタグ変数ディクショナリを使用すると、さまざまなタグ変数の値のしきい値を定義できます。

アプリケーションを実行する

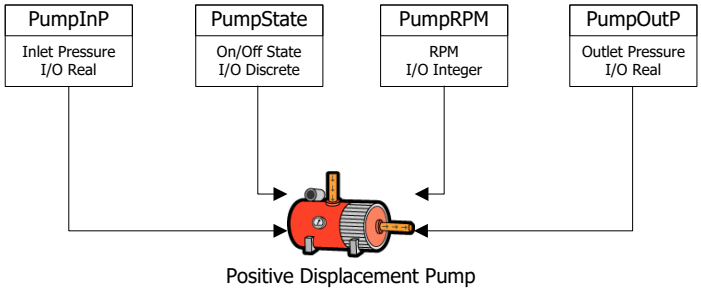
WindowViewer を使用して、すべてのタイプの InTouch アプリケーションを実行します。System Platform IDE からマネージド アプリケーションを配置した後、アプリケーション マネージャを使用して WindowViewer で開きます。豊富な種類のランタイム トリガを使用して、アプリケーションの実行中にスクリプトを開始できます。アプリケーション データおよびアラームをファイルまたは SQL Server データベースに保存するように、WindowViewer を設定できます。オペレータに WindowViewer へのログオンを要求し、WindowViewer を実行するコンピュータへの変更を行うことを禁止することでセキュリティを強化できます。WindowViewer のメニュー コマンドを選択することにより、オペレータはアプリケーションの履歴のロギングを開始および停止できます。タグ変数データの保存方法および分散方法に基づいて、WindowViewer が実行されているコンピュータを、クライアントまたはサーバーとして機能するよう設定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 3 タグ

InTouch ヒューマン マシン インターフェイス（HMI）は、製造環境におけるコンポーネントをグラフィック表示するアプリケーションです。このグラフィカル インターフェイスを使用して、工場のオペレータは製造工程を監視および管理することができます。

以下の図は、製造工程のコンポーネントであるポンプの例を示しています。ポンプには、値に関連付けられているプロパティがあります。圧力、RPM、およびステータスはポンプのプロパティであり、その値は HMI から監視されます。



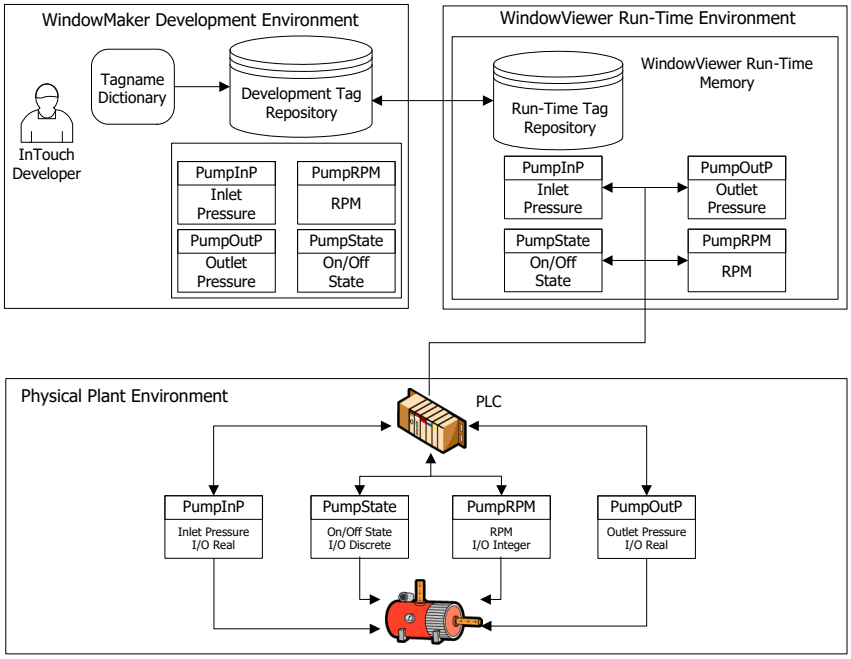
タグは InTouch HMI アプリケーションのデータ アイテムを表します。タグを使用すると、製造環境からの特定のデータ アイテムとして、特定のコンポーネント プロパティをアクセス可能にできます。上の図では、PumpState タグはポンプがオンであるかオフであるかを示しています。製造環境において、InTouch アプリケーションでプロパティを監視または制御するコンポーネントに対してタグを作成します。

製造コンポーネントから収集された異なるタイプのデータに対しては異なるタイプのタグを使用できます。たとえば、PumpState タグは、ポンプが実行中であるか停止中であるかを示すブール型のオン/オフ値を返します。アプリケーションの一部とするデータ型に対しては、適切なタイプの InTouch タグを割り当てます。

InTouch タグを操作する

InTouch アプリケーションを作成することによって開始します。WindowMaker のツールであるタグ変数ディクショナリを使用して、アプリケーションのタグ変数を定義します。以下の図は、InTouch の開発環境およびランタイム環境を示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



タグ変数ディクショナリを使用して、タグ変数の名前とタイプを割り当てます。タグ変数のタイプによっては、タグ変数ディクショナリのその他のオプションを使用して、タグ変数の追加のプロパティを指定できます。たとえば、I/O タイプのタグ変数には、リモート データ ソースへの接続を指定するための追加オプションが含まれています。

InTouch アプリケーションは、WindowViewer 環境で動作します。WindowViewer がアプリケーションを起動すると、開発リポジトリからタグ変数を読み取り、ランタイム メモリに配置します。

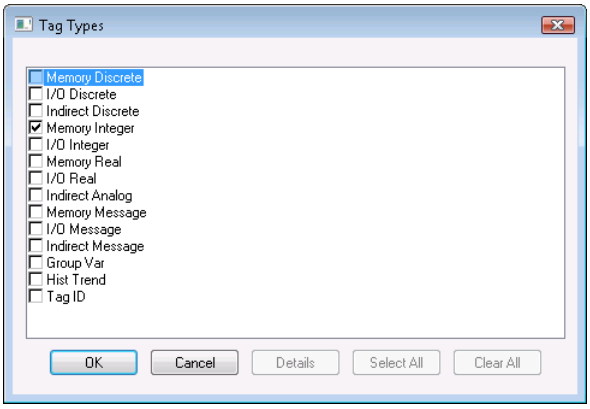
InTouch アプリケーションは、アニメーション リンクやスクリプトを使用して、ランタイム メモリに配置されたタグ変数と通信します。InTouch アプリケーションは、タグ変数に割り当てられたコンポーネント プロパティから現在値とその他のステータス情報を追跡します。

InTouch タグのタイプ

タグ変数を定義する場合、そのタグ変数に関連付けられた処理データに応じて、特定のタイプを割り当てます。たとえば、タグ変数がポンプの RPM を表示する場合、このタグ変数を整数型として割り当てます。

タグ変数ディクショナリで、［**タグ変数タイプ**］ダイアログ ボックスを使用して、作成したタグ変数にタグ変数タイプを割り当てます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



タグ変数タイプを割り当てると、タグ変数ディクショナリでは、選択したタグ変数のタイプに対して、特定のオプションが一覧表示されます。

メモリ型タグ

メモリ型タグは、InTouch アプリケーション内の内部システム定数や変数を定義します。たとえば、実数 3.414 を内部定数として定義できます。プロセス シミュレーションでは、カウンタとして動作することによって、メモリ型タグは QuickScript のバックグラウンドのアクションを制御できます。タグ変数に関連付けられているカウントに基づき、QuickScript はさまざまなアニメーション効果を起動できます。また、メモリ型タグは、他のプログラムによってアクセスされる計算済みの変数としても動作できます。

メモリ型タグのタイプは、タグに関連付けられた処理データに基づいて、次の 4 種類から選択します。

- **メモリ論理型**

  メモリ論理型タグ変数は、プロセス コンポーネントの状態プロパティに関連付けられています。メモリ論理型タグ変数に割り当てられた値は、考えられる次のような 2 つのブール型の状態で示されます。

  - 0 または 1。
  - False または True
  - On または Off
  - High または Low

- **メモリ整数型（アナログ）**

  メモリ整数型タグ変数には、-2,147,483,648 ～ 2,147,483,647 の範囲の 32 ビット符号付き整数値を割り当てることができます。

- **メモリ実数型（アナログ）**

  メモリ整数型タグ変数には、-3.4 x 1038 ～ 3.4 x 1038 の範囲の浮動小数点数を指定できます。2 つの実数型タグが比較されると、2 つの実数型タグの差分は FLT_EPSILON（値 1.19209290E-07F、10 進数では 0.0000001192092896）よりも大きくなる必要があります。浮動小数点の計算はすべて 64 ビットの解像度で行われますが、その結果は 32 ビットの 10 進数で保存されます。実数の最大の精度については、「IEEE 小数ユニット」を参照してください。

- **メモリ メッセージ型**

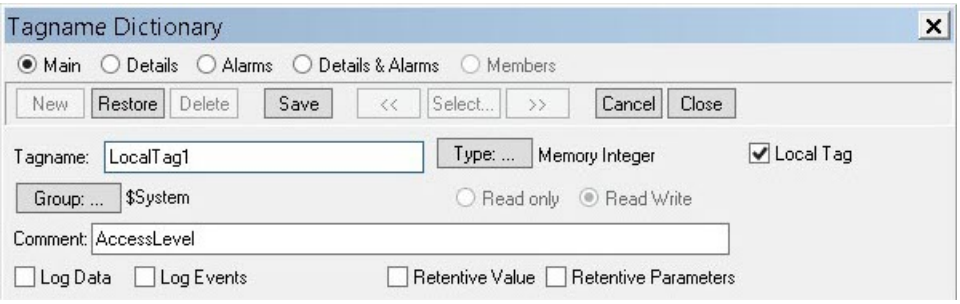
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メモリ メッセージ型タグには、半角で最大 131 文字のテキスト文字列を割り当てることができます。

ローカル タグ

ローカル タグを使用すると、Web Client で使用するセッションごとのメモリ タグを作成できます。たとえば、ナビゲーションに使用するローカル メモリ タグを設定できます。ランタイム時、ユーザーがそのタグにリンクされているナビゲーション シンボルを選択すると、各セッションは独立して動作します。

注意：ドット フィールドはローカル タグでサポートされていません。



ローカル タグを作成する

1. メモリ タグを作成し、［**ローカル タグ**］チェック ボックスをオンにします。その他のタグ プロパティを設定します。新しいタグの作成の詳細については、「新しいタグの作成」を参照してください。
2. ［**保存**］を選択します。

I/O タグ変数 I/O たぐへんすう

I/O 型タグ変数は、外部ソースに対して、InTouch アプリケーションのデータの読み取りまたは書き込みを行います。外部データには、プログラマブル コントローラ、プロセス コンピュータ、および各ネットワーク ノードからの入力および出力が含まれています。I/O 型タグ変数のデータ値は、以下のプロトコルを介してリモートでアクセスされます。

- Microsoft Dynamic Data Exchange（DDE）
- SuiteLink™

I/O 型タグ変数の値がランタイム メモリで変更されると、InTouch HMI によってリモート アプリケーションが更新されます。反対に、対応するデータ アイテムの値がリモート アプリケーションで変更されるたびに、InTouch の I/O 型タグ変数の値は更新されます。

InTouch HMI には、タグ変数に関連付けられているプロセス データに基づいて、次の 4 種類の I/O 型タグ変数が用意されています。これら 4 種類の I/O 型タグは、メモリ型タグに類似しています。

- **I/O 論理型**

  I/O 論理型タグ変数は、コンポーネント プロセスのプロパティに関連付けられており、値は考えられる次のような 2 つの状態で表されます。

  - 0 または 1。
  - False または True
  - On または Off

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- High または Low

- **I/O 整数型**（アナログ）

  I/O 整数型タグ変数には、-2,147,483,648 ～ 2,147,483,647 の範囲の 32 ビット符号付き整数値を割り当てることができます。

- **I/O 実数型**（アナログ）

  I/O 実数型タグ変数には、-3.4 x 1038 ～ 3.4 x 1038 の範囲の浮動小数点数を割り当てることができます。2 つの実数型タグが比較されると、2 つの実数型タグの差分は FLT_EPSILON（値 1.19209290E-07F、10 進数では 0.0000001192092896）よりも大きくなる必要があります。I/O 実数型タグの浮動小数点の計算はすべて 64 ビットの解像度で行われますが、その結果は 32 ビットの数字で保存されます。I/O 実数の最大の精度については、「IEEE 小数ユニット」を参照してください。

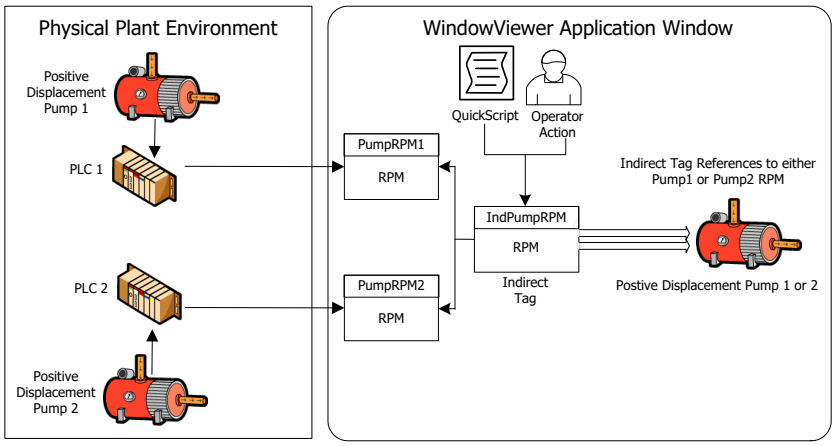
- **I/O メッセージ型**

  I/O メッセージ型タグには、半角で最大 131 文字のテキスト文字列を割り当てることができます。

間接型タグ

間接型タグ変数は、他のタグ変数に対して「ポインタ」として動作します。たとえば、単一の InTouch ウィンドウを作成し、複数の異なる一連のタグ変数の間接型タグ変数を使用して、データを表示することができます。

以下の図は、複数のポンプを表示できるアプリケーション ウィンドウの例を示しています。ポンプごとに個別のウィンドウを作成する代わりに、1 つのウィンドウで間接型タグ変数を使用して、個々のポンプに関連付けられた異なるソース タグ変数の値を表示できます。



QuickScript またはオペレータのアクションによって、間接型タグ変数がソース タグ変数に指定されます。たとえば、次のスクリプト ステートメントは、PumpNo タグ変数の値に基づいて、2 つの PumpRPM タグ変数を **IndPumpRPM** と呼ばれる間接アナログ型タグ変数に割り当てます。

```
IF PumpNo == 1 THEN
   IndPumpRPM.Name = "PumpRPM1";
   ELSE
   IndPumpRPM.Name = "PumpRPM2";
ENDIF;
```

間接型タグ変数を別のソース タグ変数と等式化すると、間接型タグ変数はソース タグ変数のように動作します。元のソースに関連付けられた値と間接型の重複タグ変数は、共に同期しています。ソース タ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グ変数の値が変化すると、間接型タグ変数の値はその変化を反映します。また、間接型タグ変数の値が変化すると、それに対応してソース タグ変数の値も変化します。

論理型、アナログ型、メッセージ型の間接型タグ変数を使用できます。この 3 種類の間接型タグ変数は、メモリ型タグ変数や I/O 型タグ変数に類似しています。

間接型タグ変数の詳細については、「間接型タグを定義する」を参照してください。

その他のタグ

特定の限られた用途に設計された、その他のタイプの InTouch タグ変数を使用できます。これらのタグを使用すると、動的なアラーム表示の作成、履歴トレンドの作成、および履歴トレンド ペンに割り当てられたタグの変更を行うことができます。

履歴トレンド タグ

履歴トレンド タグ変数は、履歴トレンド グラフを参照するために使用できます。履歴トレンドに関連付けられたすべての**ドットフィールド**は、**履歴トレンド**型タグ変数に適用できます。

履歴トレンド タグ変数の定義および使用方法については、「タグ データのトレンド」を参照してください。

タグ ID タグ

タグ変数 ID タグ変数は、InTouch 履歴トレンド グラフに値がプロットされているタグ変数からの情報を取得します。一般的に、タグ変数 ID タグ変数を使用して、特定のトレンド ペンに割り当てられたタグ変数の名前を表示したり、トレンド ペンに割り当てられたタグ変数を変更したりします。

QuickScript にステートメントを挿入し、タグ変数 ID 型タグ変数を使用して、新しいタグ変数を任意の履歴トレンドのペンに割り当てることができます。たとえば、次の QuickScript ステートメントは、履歴トレンド ペンに関連付けられたタグ変数を変更します。

```
HistTrend.Pen1=MyLoggedTag.TagID;
```

この QuickScript を実行すると、履歴トレンドの Pen1 は MyLoggedTag に対してログ記録されたデータの履歴トレンドの作成を開始します。

履歴トレンド タグ変数の定義および使用方法については、「履歴トレンド ウィザードの使用」を参照してください。

スーパータグ

スーパータグは、関連するタグのセットを含むテンプレートです。たとえば、ポンプのすべてのプロパティに割り当てられたタグ セットを含むスーパータグ テンプレートを作成できます。

生産工程で同一の装置がある場合は、スーパータグを使用します。設備ごとにタグ セットを作成するのではなく、スーパータグ テンプレートのインスタンスを同じ工程の各アイテムに割り当てます。

スーパータグの詳細については、「再使用可能なタグ構造を定義する」を参照してください。

旧版のタグ

グループ変数タグ変数を使用すると、InTouch の標準アラーム システムを使用して、動的なアラーム表示、ディスクへの動的ログ記録、および動的印刷を作成できます。グループ変数タグ変数は、InTouch バージョン 7.11 以前のソフトウェアで開発されたアプリケーションとの下位互換性用のためだけに含まれています。InTouch バージョン 7.11 以降で開発されたアプリケーションでは、グループ変数タグ変数を使用しないでください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

システム タグ

システム タグ変数の使用によって、システム関連情報や InTouch スクリプト用の日付や時間などの標準関数が提供されます。システム タグ変数はすべてのアプリケーションの一部です。また、実行中のアプリケーションの以下のような動作を管理するために、スクリプトでシステム タグ変数を使用することもできます。

- アプリケーション セキュリティの監視と管理。
- リモート ノードへの I/O 通信の確立。
- アラームが発生した場合の検出。
- 履歴のロギングの開始と終了。
- NAD を使用した、新しいバージョンへのアプリケーションの更新。

タグ名ディクショナリでは、システム タグはドル記号（$）がタグ名の最初の文字として識別されます。システム タグ変数は削除できません。システム タグ変数に関連付けられたコメントのみが変更できます。

システム タグ参照

以下の表は、InTouch システム タグ変数を説明しています。

システム タグ変数	説明	関連情報
$AccessLevel	現在ログオンしているオペレータに関連付けられているアクセス レベルを指定する読み取り専用の整数型タグ変数。 この情報は、オペレータによる特定の InTouch 関数へのアクセスを制御するアニメーション リンクまたはスクリプトで使用できます。	「InTouch のセキュリティ保護」を参照してください。
$ApplicationChanged	プライマリ アプリケーションが NAD 環境で変更されたかどうかを示す読み取り専用の論理型タグ変数。	「アプリケーションの分散」を参照してください。
$ApplicationVersion	WindowViewer で実行中のアプリケーションの現在のバージョンを示す読み取り専用の実数型タグ変数。	「アプリケーションの分散」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

システム タグ変数	説明	関連情報
$ChangePassword	1 に設定されると［**パスワードの変更**］ダイアログ ボックスを示す書き込み専用の論理型タグ変数。	「InTouch のセキュリティ保護」を参照してください。
$ConfigureUsers	セキュリティ ユーザー名リストを編集するための［**ユーザーの設定**］ダイアログ ボックスを表示する書き込み専用の論理型タグ変数。	「InTouch のセキュリティ保護」を参照してください。
$Date	1970 年 1 月 1 日から経過した日数の整数を表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$DateString	Windows の［**地域と言語のオプション**］ダイアログ ボックスで指定されているものと同じ形式で日付を表る読み取り専用のメッセージ型タグ。	「組み込み関数」を参照してください。
$DateTime	1970 年 1 月 1 日から経過した日数の少数を表示する読み取り専用の実数型タグ変数。	「組み込み関数」を参照してください。
$Day	現在の日付（1 ～ 31）を表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$False	式内で FALSE 値を返す論理型読み取り専用タグ変数。 $False システム タグ変数は、アプリケーションを以前のバージョンの InTouch を現在のバージョンに更新する際に旧版のタグ変数のインスタンスを置き換えるために使用します。	追加情報はありません。
$HistoricalLogging	InTouch アプリケーションの実行中の履歴ログ記録を開始または停止するために使用される読み取りまたは書き込みの論理型タグ変数。	「タグ値の記録」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

システム タグ変数	説明	関連情報
$Hour	現在の時間を 0 ～ 23 の値で表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$InactivityTimeout	ユーザーの無操作時間が経過したことを示す読み取り専用の論理型タグ変数。1 に設定すると、非アクティブ期間が経過するとユーザーが WindowViewer から自動的にログオフされます。	「InTouch のセキュリティ保護」を参照してください。
$InactivityWarning	無操作警告時間が経過したことを示す読み取り専用の論理型タグ変数。$InactivityWarning の値は、オペレータへの無操作警告を発行するために使用できます。	「InTouch のセキュリティ保護」を参照してください。
$LogicRunning	アプリケーション スクリプトの開始および停止に使用できる読み取りまたは書き込みの論理型タグ変数。	「InTouch のセキュリティ保護」を参照してください。
$Minute	現在の分（0 ～ 59）を表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$Month	現在の月（1 ～ 12）を表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$Msec	現在のミリ秒（0 ～ 999）を表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$NewAlarm	新しいローカル アラームが発生したことを示す読み取りまたは書き込みの論理型タグ変数。	「ランタイム時のタグとグループのアラームプロパティの制御」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

システム タグ変数	説明	関連情報
$ObjHor	画面上で選択したオブジェクトの中心の水平ピクセル位置を表示する読み取り専用の整数型タグ変数。	『AVEVA™ InTouch HMI アプリケーション開発』の「オブジェクトのアニメーション化」を参照してください。
$ObjVer	画面上で選択したオブジェクトの中心の垂直ピクセル位置を表示する読み取り専用の整数型タグ変数。	『AVEVA™ InTouch HMI アプリケーション開発』の「オブジェクトのアニメーション化」を参照してください。
$Operator	InTouch アプリケーションにログオンしたオペレータの名前を表示する読み取り専用のメッセージ型タグ。	「InTouch のセキュリティ保護」を参照してください。
$OperatorDomain	アプリケーションがオペレーティング システム ベースのセキュリティで保護されている場合に、ログオン時に指定されたドメイン名またはマシン名を含む読み取り専用のメッセージ型タグ。	「InTouch のセキュリティ保護」を参照してください。
$OperatorDomainEntered	InTouch アプリケーションへのログイン試行を行うオペレータのドメインが割り当てられた書き込み専用のメッセージ型タグ。 ログイン試行は、$PasswordEntered システムタグ変数に値を割り当てるまで開始されません。	「InTouch のセキュリティ保護」を参照してください。
$OperatorEntered	InTouch アプリケーションへのログイン試行を行うオペレータのユーザー アカウント名が割り当てられた読み取り/書き込みメッセージ型タグ。 ログイン試行は、$PasswordEntered システムタグ変数に値を割り当てるまで開始されません。	「InTouch のセキュリティ保護」を参照してください。
$OperatorName	オペレーティング システム ベースまたは ArchestrA® 認証が使用される場合に、オペレータのフル ネームを表示する読み取り専用のメッセージ型タグ。	「InTouch のセキュリティ保護」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

システム タグ変数	説明	関連情報
$PasswordEntered	InTouch アプリケーションへのログイン試行を行うオペレータのパスワードが割り当てられた書き込みのメッセージ型タグ。 このタグ変数に値を書き込むと、$OperatorDomainEntered、$OperatorEntered、および $PasswordEntered システム タグ変数の値を使用してログイン試行が開始されます。	「InTouch のセキュリティ保護」を参照してください。
$Second	現在の秒（0 ～ 59）を表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$StartDdeConversations	初期化されていない通信をランタイムで開始するために使用される読み取りまたは書き込みの論理型タグ変数	「I/O を使用したデータアクセス」を参照してください。
$System	ルート アラーム グループを識別する読み取り専用のタグ変数。	「*アラームとイベントの概要*」を参照してください。
$Time	現在の日付の午前 0 時からの経過時間をミリ秒単位で表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。
$TimeString	Windows の［**地域と言語のオプション**］ダイアログ ボックスで指定されているものと同じ形式で現在の時刻を表示する読み取り専用のメッセージ型タグ。	「組み込み関数」を参照してください。
$VerifiedUserName	確認されたユーザーのフル ネームまたは Null が含まれる読み取り専用のメッセージ型タグ。	「InTouch のセキュリティ保護」を参照してください。
$Year	現在の年を 4 桁の数字で表示する読み取り専用の整数型タグ変数。	「組み込み関数」を参照してください。

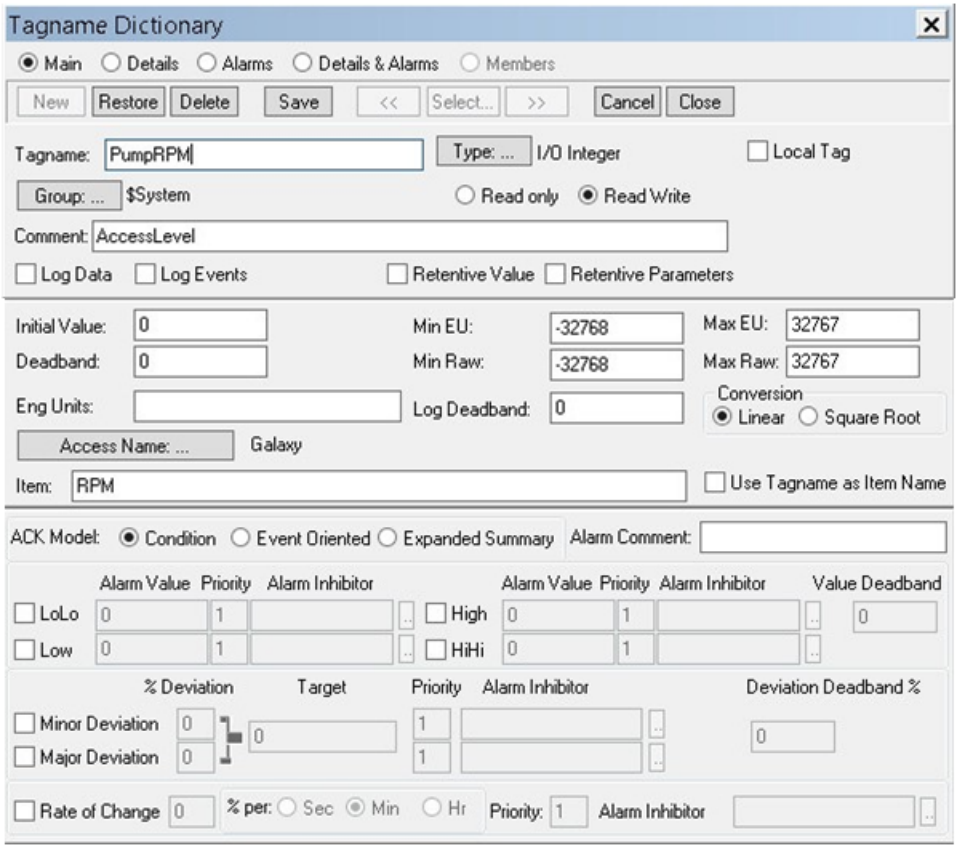
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ プロパティ

InTouch タグの各タイプには、そのタグに関連付けられたデータの特性を説明するプロパティ セットがあります。InTouch タグに関連付けられた 4 種類の主要データ型は、以下のとおりです。

- 論理値
- 整数
- 実数
- テキスト メッセージ

タグ名ディクショナリでタグを作成すると、タグのすべてのプロパティには初期値が設定されます。以下の図は、I/O 整数型タグのプロパティを示しています。



タグ名ディクショナリからタグ プロパティの初期値を設定すると、ほとんどのタグのプロパティはドットフィールドを使用して動的に変更できます。ドットフィールドは、InTouch アプリケーションの稼動中に、スクリプトで監視または変更できるタグ プロパティを識別します。ドットフィールドは、スクリプトのタグの名前に追加します。

ドットフィールドを使用したタグ プロパティの動的な変更の詳細については、「タグ ドットフィールドを使用したタグ プロパティの表示または変更」を参照してください。

メモリ型タグのプロパティ

次の表は、メモリ型タグ変数の 4 種類のタイプのプロパティを一覧表示しています。各プロパティは、タグ変数ディクショナリのオプションとして選択または変更できます。詳細については、「新しいタグの作成」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ プロパティ	論理型	整数型	メッセージ型	実数型
% 偏差		•		•
% /		•		•
確認モデル	•	•		•
アラーム コメント	•	•	•	•
アラーム グループ	•	•	•	•
アラーム抑止タグ変数	•	•		•
アラーム状態	•			
アラーム値		•		•
コメント	•	•	•	•
デッドバンド		•		•
工学単位		•		•
偏差デッドバンド %		•		•
Hi		•		•
HiHi		•		•
初期値	•	•	•	•
データ ログ	•	•		•
ログ デッドバンド		•		•
イベント ログ	•	•	•	•
Lo		•		•
LoLo		•		•
最大文字数			•	
大偏差		•		•
最大値		•		•
最小値		•		•
小偏差		•		•
オフ メッセージ	•			
オン メッセージ	•			
優先度	•	•	•	•

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ プロパティ	論理型	整数型	メッセージ型	実数型
変化率		•		•
読み取り専用	•	•	•	•
読み書き	•	•	•	•
パラメータ値保持		•		•
変数値保持	•	•	•	•
ターゲット		•		•
ディッドバンド値		•		•

I/O 型タグのプロパティ

メモリ型タグ変数と同じように、I/O 型タグ変数のプロパティもタグ変数ディクショナリのオプションとして選択または変更できます。詳細については、「新しいタグの作成」を参照してください。

タグ プロパティ	論理型	整数型	メッセージ型	実数型
% 偏差		•		•
% /		•		•
アクセス名	•	•	•	•
確認モデル	•	•		•
アラーム コメント	•	•	•	•
アラーム グループ	•	•	•	•
アラーム抑止タグ変数	•	•		•
アラーム状態	•			
アラーム値		•		•
コメント	•	•	•	•
変換方法		•		•
デッドバンド		•		•
偏差デッドバンド %		•		•
工学単位		•		•
Hi		•		•

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

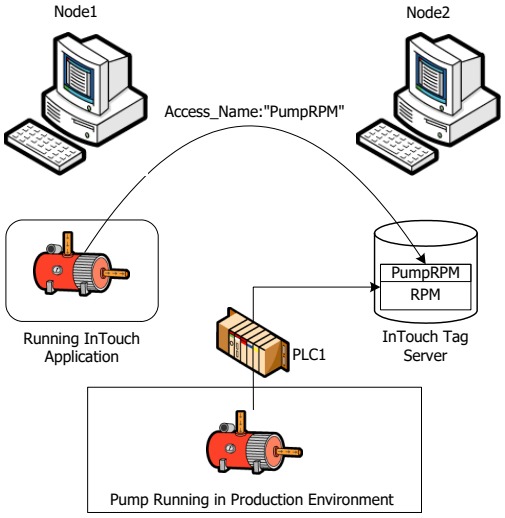
タグ プロパティ	論理型	整数型	メッセージ型	実数型
HiHi		•		•
初期値	•	•	•	•
入力データの変換	•			
アイテム	•	•	•	•
データ ログ	•	•		•
ログ デッドバンド		•		•
イベント ログ	•	•	•	•
Lo		•		•
LoLo		•		•
最大文字数			•	
大偏差		•		•
工学値最大値		•		•
生データ最大値		•		•
最大値		•		•
工学値最小値		•		•
生データ最小値		•		•
最小値		•		•
小偏差		•		•
オフ メッセージ	•			
オン メッセージ	•			
優先度	•	•	•	•
変化率		•		•
読み取り専用	•	•	•	•
読み書き	•	•	•	•
パラメータ値保持		•		•
変数値保持	•	•	•	•
二乗根変換		•		•

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

タグ プロパティ	論理型	整数型	メッセージ型	実数型
ターゲット		•		•
タグ変数名をアイテム名として使用	•	•	•	•
ディッドバンド値		•		•

リモート タグ参照

InTouch アプリケーションが稼動しているノードとは別のノード上で稼動しているタグ変数サーバーで、分散 InTouch アプリケーションを作成できます。以下の図は、別のノードで稼動しているタグ変数サーバーから PumpRPM タグ変数へのリモート リファレンスを作成する InTouch アプリケーションを示しています。



リモート ノードにあるタグ変数を参照する InTouch アプリケーションを作成するには、次の 2 つの方法があります。

- I/O 型タグ変数に、リモート サーバーをタグ変数ソースとして識別するアクセス名を関連付ける。I/O 型タグ変数のアクセス名の定義については、「アクセス名の設定」を参照してください。
- **タグ変数**を直接参照するリモート リファレンスを使用する。たとえば、PLC1:PumpRPM です。

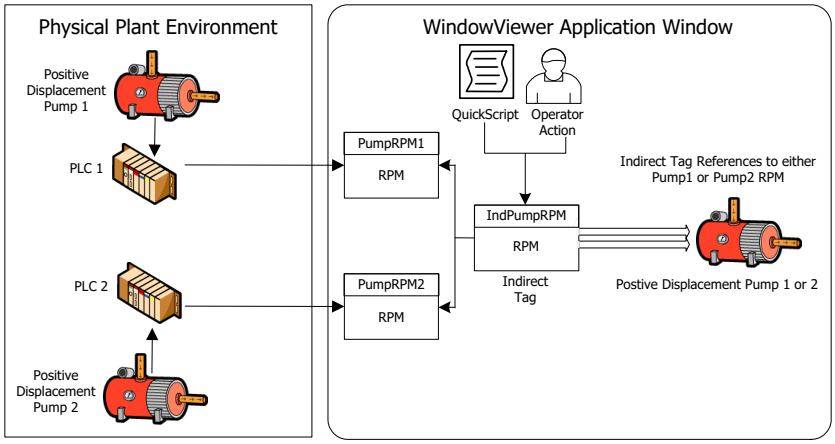
詳細については、「リモート 参照による I/O データ アクセス」を参照してください。

間接型タグを定義する

間接型タグ変数を使用して、複数のタグ変数からの値を表示するウィンドウ オブジェクトを持つアプリケーションを作成できます。

以下の図は、アプリケーション ウィンドウ内のポンプ オブジェクトを示しています。ポンプ オブジェクトは、間接型タグ変数から設定された値に基づいて可能な 2 つのプロセス ポンプを表示します。QuickScript またはオペレータの操作によって、間接型タグ変数に関連付けられたソース タグ変数が選択されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



間接型タグ変数によって、開発時間が最小化されます。運用環境で実行中の複数のプロセスを単一のウィンドウ オブジェクトで表すことができるため、作成するアプリケーション ウィンドウの数を少なくできます。

間接型タグとスクリプトを使用する

スクリプトを使用すると、入力ソース タグ変数を間接型タグ変数に割り当てることができます。入力ソース タグを間接型タグに割り当てるには、ソース タグの名前を間接型タグ数の **.Name** ドットフィールドに割り当てます。

たとえば、**IndPumpRPM** という名前の間接アナログ型タグ変数を作成する場合、以下の例のようなスクリプト ステートメントを使用して、2 つのソース **PumpRPM** タグ変数が IndPumpRPM に割り当てられます。

```
IF PumpNo == 1 THEN
   IndPumpRPM.Name = "PumpRPM1";
   ELSE
   IndPumpRPM.Name = "PumpRPM2";
ENDIF;
```

間接型タグの割り当てスクリプトは、アプリケーション イベントまたはウィンドウ ボタンを選択するなどのオペレータによる操作によってトリガできます。

間接型タグ変数を別のソース タグ変数と等式化すると、間接型タグ変数はソース タグ変数のように動作します。ソース タグ変数の値が変化すると、間接型タグ変数の値はその変化を反映します。また、間接型タグ変数の値が変化すると、それに対応してソース タグ変数の値も変化します。

間接型タグ変数の .Name ドットフィールドは簡単な文字列であるため、ランタイムで間接型タグ変数を動的に割り当てることができます。たとえば、**Number** タグ変数の値が変わるたびに実行されるデータ変化 QuickScript を作成すると、その変化に応じて間接型タグ変数 **IndPumpRPM** に割り当てられるソース タグ変数が変化します。

```
IndPumpRPM.Name = "PumpRPM" + Text(数値、 "#" );
```

このスクリプトが実行されると、アナログ型タグ **Number** の値がテキストに変換されて、**PumpRPM 文字列に追加されます。Number が 1 の場合には、間接型タグ IndPumpRPM タグ**の名前は **PumpRPM1** に設定されます。

間接アナログ型タグ変数は、整数型と実数型のタグ変数の両方に使用できます。タグ変数のタイプが同じであれば、他のどのタグ変数にも間接型タグ変数をマップできます。

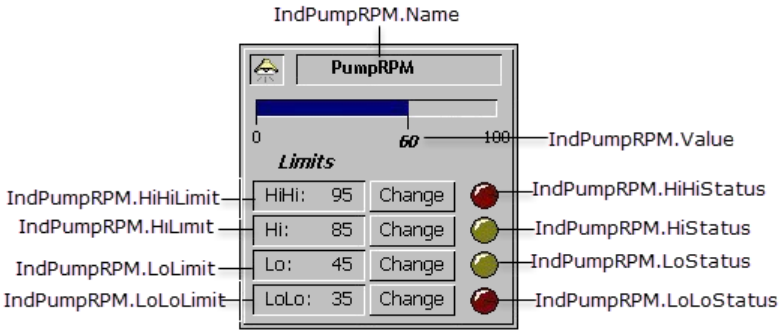
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

また、間接型タグ変数に保持属性を割り当てることもできます。保持属性を使用すると、アプリケーションが再起動されたときに、間接型タグ変数では最後のタグ変数割り当てが保持された状態となります。

間接型タグとローカル タグを使用する

一般的に、間接型タグ変数はローカルのタグ変数ディクショナリで定義されたタグ変数と共に使用されます。間接型ローカル タグ変数を使用すると、ローカル タグ変数の複数の属性を表示する視覚的なオブジェクトを作成できます。たとえば、アプリケーション ウィンドウにフェイスプレートを作成できます。フェイスプレートには、異なるドットフィールドに割り当てられた間接型ローカル タグ変数にリンクする、選択可能なアイテムが含まれています。以下の例では、オペレータは間接型ローカル タグ変数 **PumpRPM** にリンクしているドットフィールドのアラームしきい値を変更できます。

以下の図は、ポンプ RPM のアラームしきい値へのアニメーション リンクを持つフェイスプレートを示しています。間接型タグ変数は、アラームしきい値フェイスプレートに対して異なるローカル タグ変数の属性を割り当てます。



フェイスプレートを適切なタグ変数にリダイレクトするには、QuickScript 内にステートメントを含めます。

```
Indirect_tag_name.Name = "tag_name";
```

このスクリプトの例では、tag_name はローカルのタグ変数ディクショナリで定義された実際のタグ変数の名前です。このスクリプトが実行されると、このローカル タグ変数に関連付けられているすべてのドットフィールド値は、間接型タグ変数を介してアプリケーション オブジェクトにアクセスできるようになります。

間接型タグとリモート参照を使用する

リモート間接型タグ参照は、ローカル タグ参照とは異なります。リモート参照の構文は以下のとおりです。

AccessName:Item

構文の要素を以下に示します。

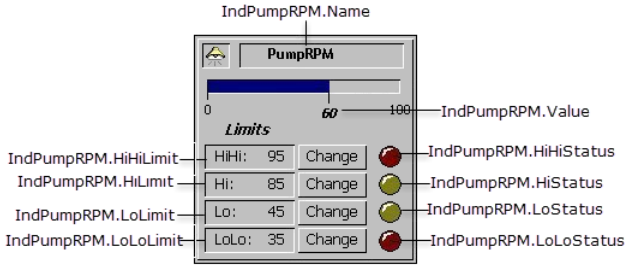
- *AccessName* は、任意の有効な InTouch アクセス名です。
- *Item* は、アクセス名の定義で指定した、I/O サーバーによってサポートされる任意の有効なアイテム名です。

リモート参照を使用すると、サーバーはタグ構造ではなく値をクライアントに返します。この値には、時刻スタンプと品質スタンプが含まれています。そのため、リモート参照に割り当てられている間接型タグは、値、時刻、品質に関するもの以外、タグ ドットフィールドにアクセスすることはできません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

たとえば、間接型タグは、アラームしきい値を指定するために、リモート参照を通じてタグ属性にアクセスすることはできません。

考えられる解決策の 1 つは、間接型タグのセットを使用してフェイスプレートを作成することです。以下の図は、ポンプのアラームしきい値を変更するフェイスプレートを示しています。



この例では、黙示的な **.Value ドットフィールド**に関連付けられた 10 個の間接型タグがフェイスプレートで使用されています。アラーム フェイスプレートは、TagServer1 というリモート InTouch ノード上のリモート参照タグ IndPumpRPM にリダイレクトされています。InTouch アクセス名は以下のように設定されます。

アクセス名：	TagSource1
ノード名：	TagServer1
アプリケーション名:	表示
トピック名：	タグ名

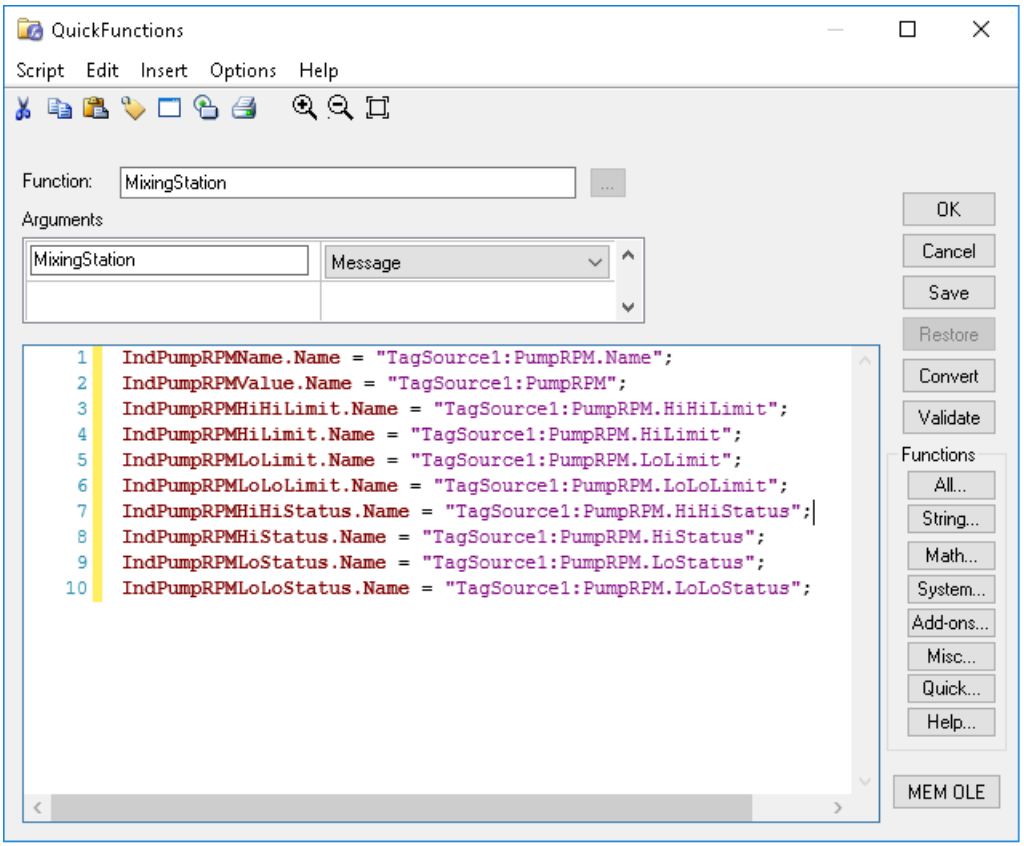
フェイスプレートをリモート参照タグ PumpRPM にリダイレクトするには、以下の QuickScript を実行します。

```
IndPumpRPMName.Name = "TagSource1:PumpRPM.Name";
IndPumpRPMValue.Name = "TagSource1:PumpRPM";
IndPumpRPMHiHiLimit.Name = "TagSource1:PumpRPM.HiHiLimit";
IndPumpRPMHiLimit.Name = "TagSource1:PumpRPM.HiLimit";
IndPumpRPMLoLimit.Name = "TagSource1:PumpRPM.LoLimit";
IndPumpRPMLoLoLimit.Name = "TagSource1:PumpRPM.LoLoLimit";
IndPumpRPMHiHiStatus.Name = "TagSource1:PumpRPM.HiHiStatus";
IndPumpRPMHiStatus.Name = "TagSource1:PumpRPM.HiStatus";
IndPumpRPMLoStatus.Name = "TagSource1:PumpRPM.LoStatus";
IndPumpRPMLoLoStatus.Name = "TagSource1:PumpRPM.LoLoStatus";
```

このスクリプトは、フェイスプレートをリダイレクトするたびに実行する必要があります。別の解決策は、単独のスクリプトを記述して、そのスクリプトにリモート参照の名前を渡す InTouch クイック関数を作成することです。同じクイック関数を呼び出す複数のフェイスプレートを使用することによって、スクリプトのコーディング作業を軽減できます。

たとえば、類似したスクリプト コマンドのセットを使用して、**RedirectAlarmFacePlate** というクイック関数を定義できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



RedirectAlarmFacePlate 関数を呼び出すと、リダイレクト処理全体を行えます。この処理を実行するには、この関数が別の InTouch QuickScript によって呼び出される必要があります。以下に例を示します。

```
CALL RedirectAlarmFacePlate ("TagSource1:PumpRPM");
```

再使用可能なタグ構造を定義する

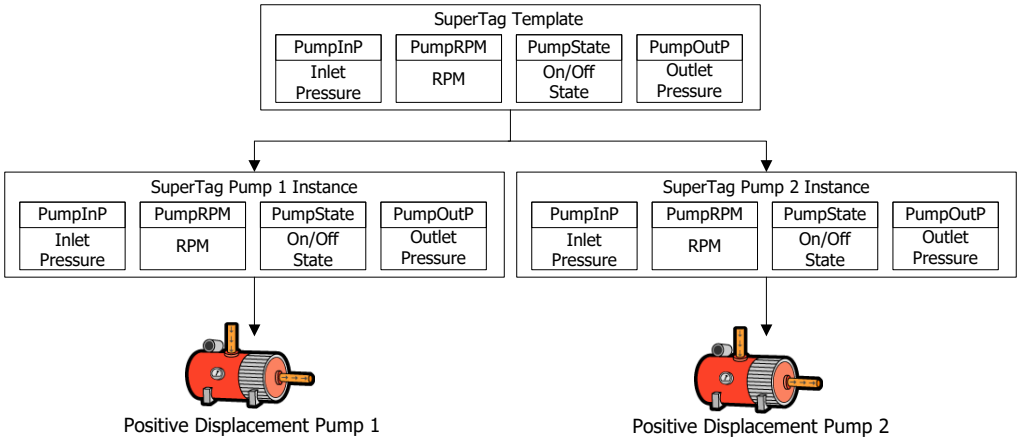
スーパータグは、関連するタグのセットのテンプレートです。スーパータグ テンプレートに属しているタグは、生産工程で 1 つのコンポーネントの共通のプロパティに関連付けられます。

スーパータグを利用することで、開発期間が短縮されます。生産工程でコンポーネントごとにタグのセットを作成する代わりに、1 つのスーパータグ テンプレートを複製し、同じプロパティを持つ工程内のすべてのコンポーネントに個別のインスタンスを作成することができます。

スーパータグ テンプレート

下図に、ポンプのデータに関連付けられた関連タグのセットで構成されるスーパータグ テンプレートを示します。このテンプレートを複製して、プロセスの中ですべての同じポンプに対してスーパータグのインスタンスを作成することができます。

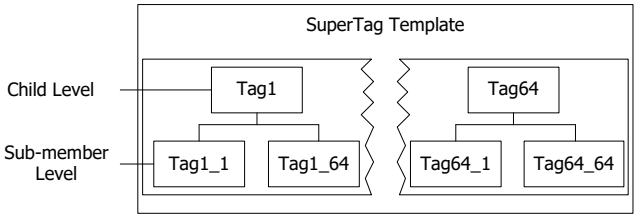
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



スーパータグのインスタンスに含まれるタグはすべて、普通のタグとまったく同じように機能します。メンバー タグのデータ型には、InTouch の論理型、整数型、実数型、メッセージ型を指定することが可能です。タグは、トレンド、アラーム、およびすべてのタグ ドットフィールドをサポートしています。

スーパータグ テンプレートは、メンバーのタグを 2 階層の入れ子構造にまとめることができます。スーパータグ テンプレートには、最大 64 個の子タグを組み込むことができます。各子タグも最大 64 個のサブメンバー タグを持つことができます。つまり、スーパータグ テンプレートには合計 4095 個のタグが入ります。

以下の図は、スーパータグ テンプレート内のタグの構造を示しています。



1 つのスーパータグ テンプレートの親が別のスーパータグ テンプレートに埋め込まれると、子メンバーになります。

スーパータグの親テンプレートを作成すると、タグ名ディクショナリでは［*タグ タイプ*］ダイアログ ボックスにタグ タイプとして一覧表示されます。新しいタグを作成した場合、タグ タイプとして直ちにそのテンプレートを選択できるようになります。新しく作成したスーパータグ タイプを使用するタグを定義するために WindowMaker を再起動する必要はありません。

スーパータグ テンプレートの保存

デフォルトでは、すべてのスーパータグ テンプレートは、**C:\ProgramData\Wonderware\InTouch** フォルダの Supertag.dat ファイルに保存されます。スーパータグ テンプレートは別の場所に保存することもできます。

スーパータグ テンプレートを保存する場所を指定する

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］を選択し、［**WindowMaker**］を選択します。

   WindowMaker の設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［**スーパータグのパス**］テキスト ボックスで、スーパータグ テンプレートを保存するパスを指定します。
4. スーパータグ データをアプリケーション ディレクトリに保存するには、［**アプリケーション ディレクトリにスーパータグ データを保存**］チェック ボックスをオンにします。このオプションを選択すると、［スーパータグ テンプレート パス］フィールドで指定したパスが上書きされます。

   注： スーパータグ テンプレートをエクスポートする場合、このチェック ボックスをオンにして、それぞれの InTouch アプリケーションをエクスポートできます。詳細については「テンプレートとして使用するための InTouch アプリケーションのエクスポート」を参照してください。

スーパータグをインポートする

InTouch アプリケーション移行の際にスーパータグをインポートするには、「InTouch アプリケーションの移行中にアセットをインポートする」を参照してください。

スーパータグ テンプレートとメンバー タグを管理する

InTouch HMI 2023 のリリースでは、テンプレート メーカー ユーティリティが廃止されました。スーパータグは、キャンバスのスーパータグ ウィンドウを使用して管理できます。スーパータグ テンプレートおよびそのメンバー タグの作成、編集、および削除を行うことができます。スーパータグ テンプレートを作成した後、タグ名ディクショナリの［**タグ タイプ**］ダイアログ ボックスには、スーパータグ テンプレートとタグ タイプとしてのテンプレートの間接型が一覧表示されます。

スーパータグ テンプレートまたはスーパータグ メンバーのタグは、いつでも修正できます。あるテンプレートから派生した既存のスーパータグ インスタンスには、このテンプレートへの変更内容は継承されません。ただし、新しいインスタンスはすべて、修正されたテンプレートの変更内容を継承します。

スーパータグ テンプレートは、テンプレート全体を一括して削除することも、テンプレートに含まれるメンバーのうち選択したタグだけを削除することもできます。削除したテンプレートは、タグ名ディクショナリの［**タグ タイプ**］ダイアログ ボックスに一覧表示されなくなります。

スーパータグ テンプレートを作成する

1. ［**スーパータグ**］ペインを開きます。
2. **Supertags** フォルダを右クリックし、［**New Supertag（新規スーパータグ）**］を選択します。

   スーパータグ テンプレートが作成されます。スーパータグ テンプレートの名前は、必要に応じて変更できます。

既存のスーパータグのテンプレートまたはメンバーのタグを編集する

注意： スーパータグ ウィンドウ ペインを使用してスーパータグを管理することが推奨されます。

1. ［**スーパータグ**］ペインを開きます。
2. 編集するスーパータグ テンプレート名またはメンバー タグに移動します。
3. スーパータグ テンプレートを右クリックし、［**スーパータグを編集**］を選択します。

   または、スーパータグ テンプレートをダブルクリックします。

   キャンバスにスーパータグ ウィンドウが表示されます。
4. スーパータグ テンプレートの名前や説明を変更することができます。
5. 必要な変更を行います。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

6. ［**OK**］を選択します。

スーパータグ テンプレートまたはメンバー タグを削除する

1. ［**スーパータグ**］ペインを開きます。
2. 削除するスーパータグ テンプレート名またはメンバー タグに移動します。
3. スーパータグ テンプレートを右クリックし、［**削除**］を選択します。

   選択した項目を削除するか確認を要求するメッセージが表示されます。
4. ［**はい**］を選択して、選択したテンプレートを削除します。

重要：スーパータグ テンプレートのインスタンスのメンバー タグは削除できません。たとえば、PumpRPM が TankPump スーパータグ テンプレートのメンバー タグである場合、PumpRPM は TankPump のいずれのインスタンスからも削除することはできません。スーパータグ テンプレートから削除できるのはタグだけです。

スーパータグ インスタンス

スーパータグ テンプレートとテンプレート インスタンスは異なります。インスタンスは、InTouch HMI アプリケーションにスーパータグ テンプレートを実際に実装したものです。

テンプレートとインスタンスの最も大きな違いは、親のテンプレート名が各インスタンスのタグ名で置き換えられる点です。子テンプレートの名前とサブメンバー タグは変更されません。

スーパータグ インスタンスを作成する

タグ名ディクショナリを使用して、スーパータグ インスタンスを作成することができます。新しいスーパータグ インスタンスを定義すると、タグ名ディクショナリによってすべてのメンバー タグとサブメンバー タグが自動的に作成されます。

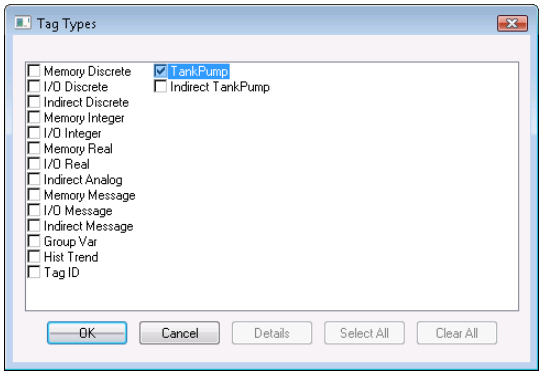
テンプレートからスーパータグ インスタンスを作成する

1. ［**スーパータグ**］ペインで、インスタンスを作成するスーパータグ テンプレートを選択します。
2. スーパータグ テンプレートを右クリックし、［**新しいインスタンス**］を選択します。

   新しいスーパータグ インスタンスが作成されます。
3. 新しいインスタンスをダブルクリックしてタグ名ディクショナリを開きます。
4. ［**タグ名**］ボックスに新しいスーパータグ インスタンスの名前を入力します。

   スーパータグ インスタンスの名前には、最大 32 文字まで指定できます。インスタンス名の命名規則は、通常の InTouch のタグと同じです。
5. ［**タイプ**］を選択して、［**タグ名タイプ**］ダイアログ ボックスを開きます。
6. リストからスーパータグ テンプレートの名前を選択します。スーパータグの作成方法の詳細については、「スーパータグ テンプレートとメンバー タグを管理する」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



7. ［**OK**］を選択します。［**タグ名ディクショナリ**］ダイアログ ボックスが展開され、詳細なオプションが表示されます。

   ［**タグ名**］ボックスに入力した新しいタグは、選択したスーパータグ テンプレートに含まれるすべてのメンバー タグの親になります。

8. タグのプロパティを設定します。以下の手順を実行します。
   a. ［**メンバー リスト**］ボックスで、スーパータグ テンプレート リストからタグを選択します。
   b. ［**データ アクセス**］から、［**メモリ**］または［**I/O**］を選択して、それぞれ［メモリ］ダイアログ ボックスまたは［I/O 型詳細］ダイアログ ボックスを開きます。
   c. 通常の InTouch タグと同じように、詳細設定を入力します。
   d. リストから残りのメンバー タグも選択し、それらについても設定します。
9. スーパータグ インスタンスに属しているすべてのメンバー タグの詳細設定を指定したら、［**閉じる**］を選択します。

スーパータグ インスタンスを複製する

タグ名ディクショナリを使用して、既存のインスタンスを複製することができます。インスタンスを複製すると、タグは直ちにアニメーション リンクや InTouch QuickScript で使用できるようになります。

タグ名ディクショナリからスーパータグ インスタンスを複製する

1. タグ名ディクショナリを開きます。
2. ［**選択**］を選択して、アプリケーションに定義されているタグが一覧表示された［**タグを選択**］ダイアログ ボックスを開きます。
3. 新しいインスタンスのテンプレートとして使用するスーパータグ インスタンスをリストから選択します。
4. ［**OK**］を選択します。

   ［**タグ名**］ボックスに選択したテンプレートの名前が表示されます。

5. ［**新規作成**］を選択します。スーパータグ インスタンスを複製するか確認を要求するメッセージが表示されます。
6. ［**はい**］を選択します。［**名前を入力してください**］ダイアログ ボックスが開き、新しいスーパータグの名前を入力するように求められます。

   通常のタグの命名規則に従い、最大 32 文字の名前を入力します。

7. ［**OK**］を選択します。タグ名ディクショナリに新しいスーパータグ インスタンスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

8. 必要に応じて、通常の InTouch タグと同じようにメンバー タグを編集します。
9. ［閉じる］を選択します。

スーパータグ インスタンスにタグを追加する

タグ名ディクショナリを使用して、既存のスーパータグ インスタンスのメンバーとしてタグを追加することができます。

タグを追加するには、スーパータグ インスタンスの正確な名前に続いて円記号（\）の区切り文字と新しいメンバー タグの名前を入力します。

以下に例を示します。

```
Pump_8\PumpSTS
```

注意 : Bulk Import Utility を使用して InTouch HMI アプリケーションから Application Server にタグを移行することを計画している場合、標準の円記号（\）区切り文字の置換の詳細については、「Bulk Import Utility を使用してスーパータグをインポートする」を参照してください。

タグをスーパータグ インスタンスに追加する

1. タグ名ディクショナリを開きます。
2. タグをスーパータグ インスタンスに追加するには、以下の手順を実行します。
   a. ［新規作成］を選択します。
   b. ［タグ名］ボックスに、正確なスーパータグのインスタンス名に続いて円記号（\）の区切り文字と新しいメンバー タグの名前を入力します。
   c. ［タイプ］を選択します。
   d. インスタンスに追加しようとしている新しいメンバー タグのタグ タイプを選択します。
   e. ［OK］を選択します。メンバー タグのタイプに応じて、詳細ダイアログ ボックスが表示されます。
   f. 通常の InTouch タグと同じように、必要な詳細設定を入力します。
3. ［保存］を選択します。
4. ［選択］を選択します。
5. メンバー タグを追加したスーパータグ インスタンスを選択します。
6. ［OK］を選択します。

   ［メンバー リスト］ボックスに、このスーパータグ テンプレートに含まれるすべてのメンバー タグが一覧表示されます。

   リストには、新しいメンバー タグも表示されます。

スーパータグのその他の作成方法

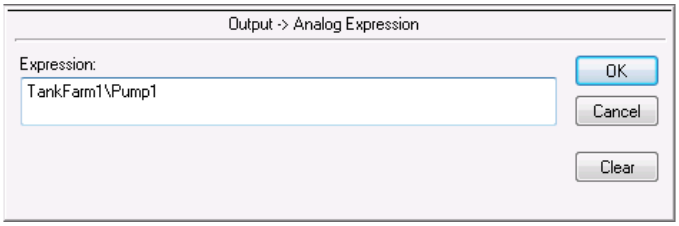
スーパータグは次の方法を使用して作成することもできます。

- アニメーション リンクの式の入力ボックス
- InTouch QuickScript 内
- InTouch DBLoad ユーティリティを使用してアプリケーションにロードした外部ファイル

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意 : 代替方法を使用してメンバーを作成すると、スーパータグ設定画面では、スーパータグ テンプレート定義にメンバーが表示されません。

スーパータグをアニメーション リンクの式または InTouch QuickScript で作成する場合には、有効なスーパータグの形式を使用する必要があります。以下に例を示します。



注意 : アニメーション リンクの式または QuickScript で指定したスーパータグ インスタンスおよびメンバー タグが現在定義されていない場合には、タグを定義するように求められます。［OK］を選択します。タグ名ディクショナリが開き、スーパータグ インスタンスと作成したメンバー タグが表示されます。

次の例は有効な構文を示しています。

```
ParentInstance\ChildMember ParentInstance\ChildMember\Submember
```

次の例は無効な構文を示しています。

```
ParentInstance\
ParentInstance\ChildMember\
```

無効な書式を使用した場合、スーパータグの構文にエラーがあることを示すエラー メッセージ ボックスが表示されます。

スーパータグ インスタンスを所有オブジェクトとして設定する

所有オブジェクトは、相対参照に使用できます。OwningObject プロパティは、1 つのグラフィックを別のグラフィックに埋め込んでいる場合に使用できます。所有オブジェクト プロパティは、産業用グラフィック エディタの設定［**プロパティ**］タブの［**ランタイム動作**］セクションで設定できます。マネージド InTouch アプリケーションとスタンドアロン InTouch アプリケーションの両方の `InTouch:` 接頭辞付きのインスタンス名を入力します。例えば、「InTouch:Tank01」と入力します。所有オブジェクトは、スクリプトを使用して設定することもできます。OwningObject プロパティは、ShowGraphic() スクリプト関数によって表示されるグラフィックの所有オブジェクトを設定します。これは定数文字列および参照文字列の連結である場合があります。例: graphicInfo.OwningObject = "UserDefined_001";

注意 : OwningObject プロパティはグラフィックの参照を設定しますが、シンボルがオブジェクト ウィザードの一部である場合、GraphicName プロパティに関連付けられません。したがって、所有オブジェクトを使用してシンボルのスクリプトを作成する場合、所有オブジェクトの名前を GraphicName プロパティの一部として指定します（UserDefined_001.Pump_001 など）。

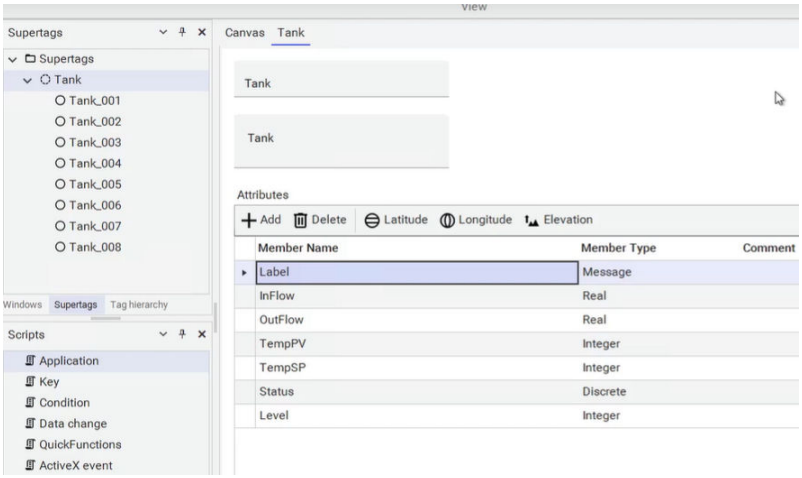
スーパータグ内の所有オブジェクト

スーパータグで相対参照を使用できます。スーパータグに基づいて、シンボルを作成し、所有オブジェクト プロパティを使用できます。この機能を使用すると、スーパータグ インスタンスをグラフィックの所有オブジェクトとして設定できます。個々のシンボルを設定する必要がなくなるので、アプリケーションを迅速に開発できます。

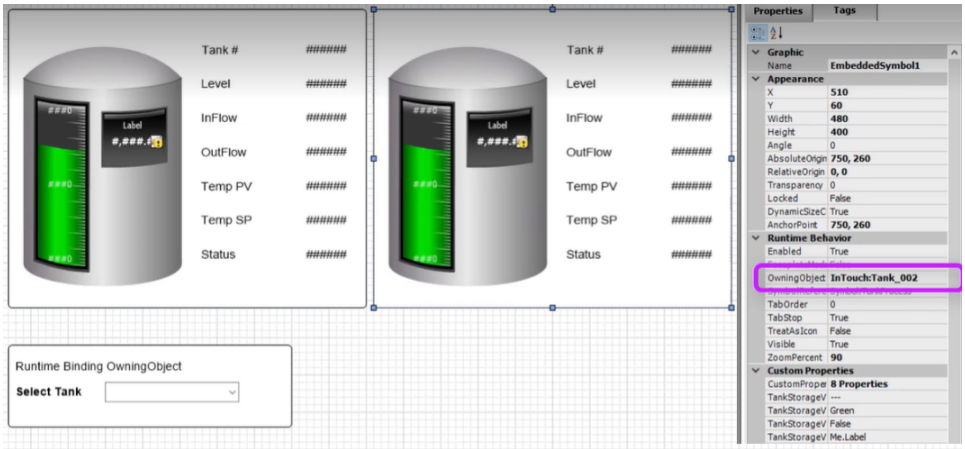
以下に例を示します。

7 つの属性が追加され、次に示すような 8 つのインスタンス（Tank_001、Tank_002 ... Tank_008）がある Tank という名前のスーパータグを考えてみます。

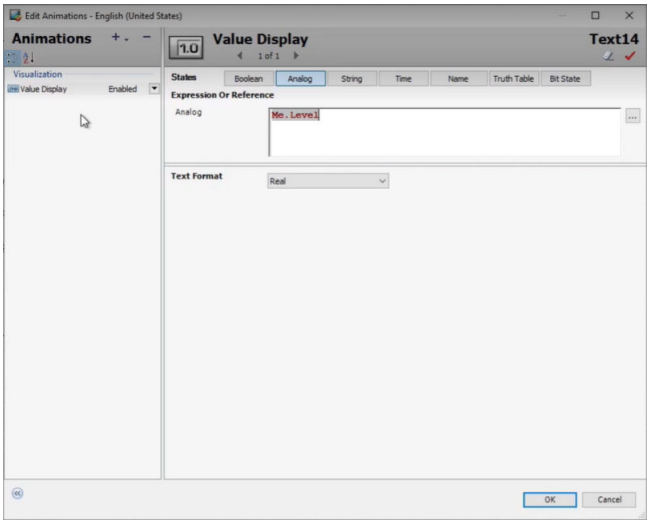
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



個々の Tank スーパータグ インスタンスは、所有オブジェクトとして設定されています。これらの Tank インスタンスを表すために、これらのタンク シンボルが産業用グラフィック エディタに追加されます。

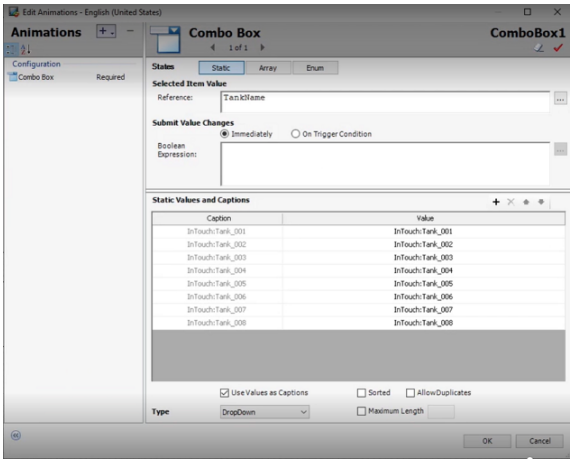


各タンク シンボルに対して、属性の値表示が相対参照として設定されます (埋め込まれたグラフィックには、相対参照が機能するための "me.Attribute" が含まれます。参照される "me.Level" などの属性は、そのスーパータグの属性にランタイム時にバインドされます)。

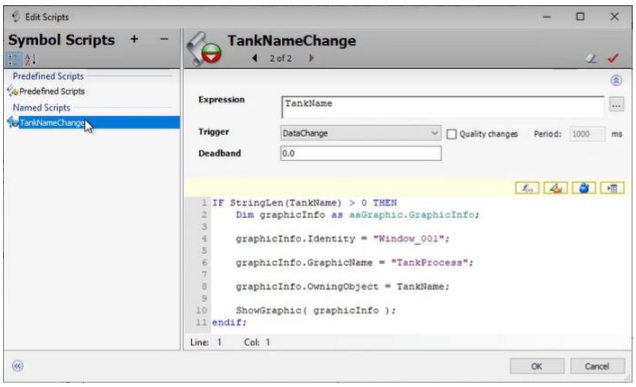


© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

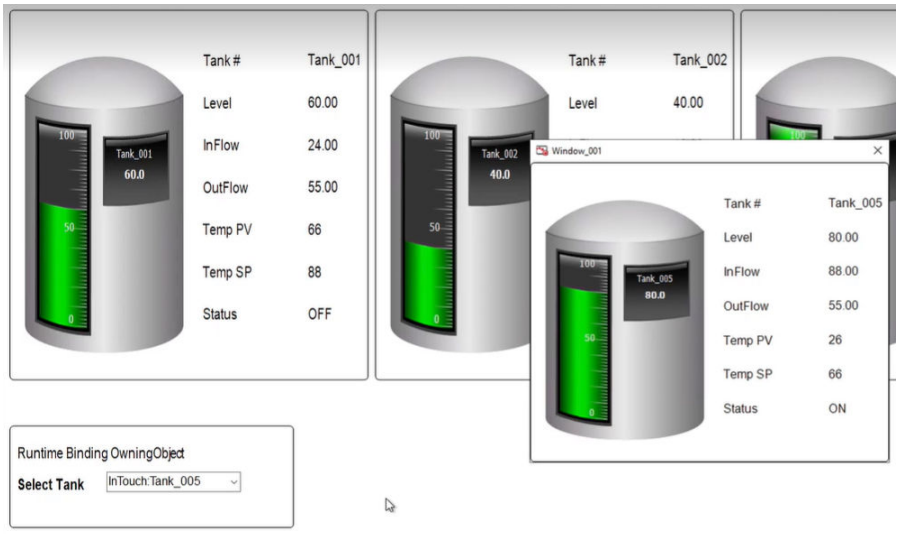
TankName カスタム プロパティを参照するコンボ ボックスがグラフィックに追加され、コンボ ボックスの値がスーパータグ インスタンスの名前として設定されます。



このグラフィックのスクリプトで、TankName カスタム プロパティが変更されたときに実行するようデータ変更スクリプトが設定されます。このスクリプトは、グラフィック表示オプションを使用してタンクを表示し、所有オブジェクトを TankName カスタム プロパティの値に設定します。



ランタイムに切り替えた後、コンボ ボックスでタンクを選択し、対応するタンクとその属性値を表示することができます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スーパータグ属性

キャンバスのスーパータグ ウィンドウを使用して、スーパータグ テンプレートに属性を追加できます。各属性に対して、メンバー名やメンバー タイプなどのプロパティも設定できます。

メンバー属性を追加または変更する

1. キャンバスでスーパータグ ウィンドウを開きます。
2. ［**属性**］セクションで［**追加**］を選択します。

   新しい属性がグリッドに追加されます。
3. ［**メンバー名**］セルをダブルクリックして名前を変更します。
4. ［**メンバー タイプ**］セルをダブルクリックして使用可能なオプション（整数、実数、メッセージ、および論理）から選択します。デフォルトでは、［メンバー タイプ］フィールドは［整数］に設定されています。
5. ［**コメント**］セルをダブルクリックして説明を変更します。

場所属性を追加または変更する

1. キャンバスでスーパータグ ウィンドウを開きます。
2. ［**属性**］セクションで［**緯度**］、［**経度**］、または［**標高**］を選択します。

   緯度、経度、および標高の新しい属性がグリッドに追加されます。
3. メンバー名には、選択した場所属性の**緯度、経度、または標高**がそれぞれ表示されます。

   注意：場所属性のメンバー名は変更できません。
4. ［**メンバー タイプ**］セルをダブルクリックして使用可能なオプション（整数、実数、メッセージ、および論理）から選択します。デフォルトでは、場所の［メンバー タイプ］属性は［実数］に設定されています。
5. ［**コメント**］セルをダブルクリックして説明を変更します。

属性を削除する

1. キャンバスでスーパータグ ウィンドウを開きます。
2. ［**属性**］セクションで、削除する属性を選択します。
3. ［**削除**］を選択します

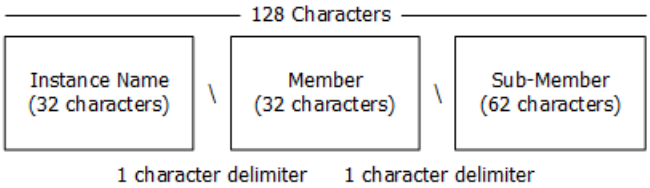
   選択した属性が削除されます。

スーパータグ メンバーを参照する

InTouch タグ名には、最大 128 文字まで指定できます。各スーパータグ ParentInstance\ChildMember\Submember の長さは最大 128 文字です。タグ名に最大長が設定されていることにより、スーパータグの参照に制限が適用されます。

スーパータグの参照には、次の図に示すように最高 2 つのテンプレート（ParentInstance\ChildMember）および 1 つの階層のメンバーだけを含むことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



注意：属性タイプが既存のスーパータグに設定されていない場合、スーパータグのサブメンバーは最大 62 文字の英数字です。スーパータグ タイプに基づくサブメンバーは、最大 32 の英数字の長さです。

スーパータグ テンプレートの各メンバーにアクセスする際は、通常の InTouch タグ タイプのドットフィールドにアクセスするために現在使用している標準の形式を使用できます。スーパータグ参照の構文は、通常のタグを使用できる InTouch アプリケーション全体で使用できます。たとえば、有効なスーパータグ ドットフィールド参照は以下のとおりです。

```
TankFarm\Tank1\Pump1RPM.RawValue
```

リモート タグ参照にスーパータグを使用することもできます。たとえば、有効なスーパータグ リモート参照は以下のとおりです。

```
PLC1:"TankFarm\Tank1\Pump1RPM.RawValue"
```

Bulk Import Utility を使用してスーパータグをインポートする

Bulk Import Utility を使用して、タグの定義を InTouch から Application Server のオブジェクト構造に変換することができます。Bulk Import Utility を使用すると、InTouch タグを Application Server に効率的に移行できます。さらに、Bulk Import Utility では、通常の InTouch タグだけでなくスーパータグも移行することができます。

スーパータグを InTouch アプリケーションから Application Server に移行する場合は、スーパータグ リファレンスで使用されている円記号（\）をサポートされているアンダースコア（_）などの文字に置き換えてください。

例：

```
TankFarm_Tank1_Pump1RPM.RawValue
```

注意：このリリースの時点では、DBDump および DBLoad プロセスに既知の制限があります。.CSV ファイルからのインポートによって作成されたスーパータグ インスタンスはタグ ディクショナリにのみ表示され、［スーパータグ］ペインには表示されません。

InTouch アプリケーションの移行中にアセットをインポートする

InTouch アプリケーションを InTouch SP2020 R2 以前から移行する場合、アセットは自動的にはインポートされません。

旧バージョンの InTouch アプリケーションで作成されたアセットをインポートするには、旧バージョンの supertag.dat ファイルをコピーして、新しい InTouch アプリケーションの同じ場所に貼り付けます。Supertag.dat ファイルには、アプリケーションで作成されたすべてのアセットが含まれています。

supertag.dat ファイルのファイルパスを以下に示します。

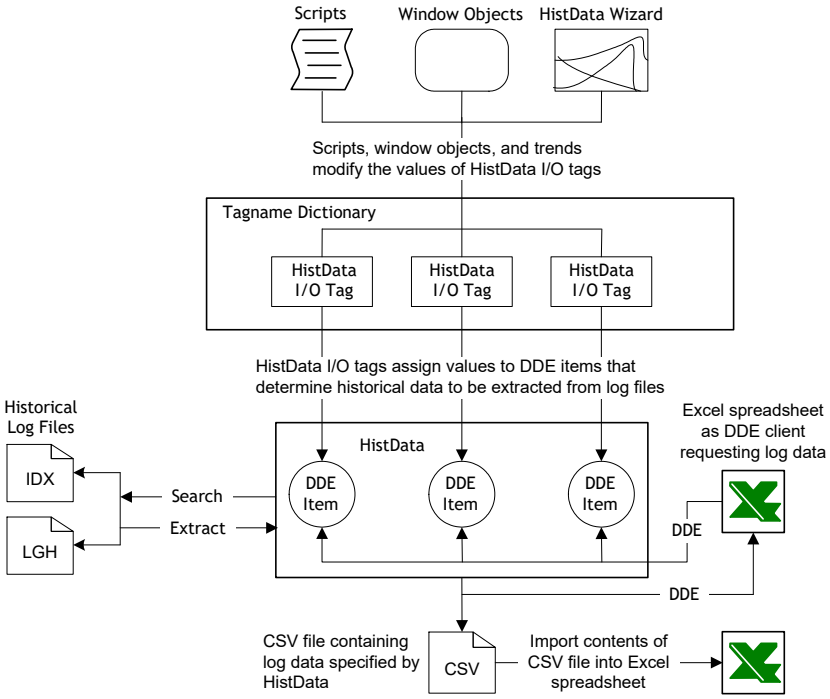
C:\ProgramData\Wonderware\InTouch\supertag.dat

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

他のアプリケーションから履歴タグ値にアクセスする

InTouch HistData ユーティリティを使用すると、履歴ログ ファイルからカンマ区切り形式（.csv）ファイルにデータを抽出することができます。Excel などのアプリケーションは、DDE クライアントとして HistData から直接 InTouch ログ データを抽出したり、または HistData ユーティリティで作成された出力ファイルからログ データをインポートすることが可能です。

下図は、選択した履歴ログ データをファイルまたは DDE クライアント アプリケーションに保存するプロセスを示しています。



DDE アイテムを使用して履歴データを表示する

HistData プログラムには、ログ ファイルから履歴データを抽出する方法を指定した DDE アイテムのセットが含まれています。これらのアイテムは、HistData 内部データベースの一部です。各アイテムに値を割り当てます。

次の表は、HistData プログラムで定義されている HistData アイテムをまとめたものです。

アイテム	データ型	説明
DATADIR	メッセージ型	履歴ログ ファイルの含まれるフォルダのパス
DBDIR	メッセージ型	InTouch タグ変数ディクショナリの内容が含まれるフォルダのパス
STARTDATE	メッセージ型	ログ ファイルからデータを抽出する開始日。開始日の形式は MM/DD/YY です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アイテム	データ型	説明
STARTTIME	メッセージ型	ログ ファイルからデータを抽出する開始時間。開始時間の形式は、24 時間表示で HH:MM:SS です。
DURATION	メッセージ型	ログ ファイルからのデータ収集期間の長さ。 **DURATION** は、次のように表記することができます。 • 週（w） • 日（d） • 時間（h） • 分（m） • 秒（s） **DURATION** には、秒単位の間隔も指定可能です。たとえば、`DURATION=0.5m` は 30 秒を意味します。サンプリングを 1 回だけ要求する場合は、**DURATION** を 0 に設定します。
INTERVAL	メッセージ型	データ収集期間の長さ。**INTERVAL** は週、日、時間、分、秒で指定します。**INTERVAL** 期間を表わす時間の単位は、**DURATION** 期間と同じです。 秒単位の間隔も指定可能です。たとえば、`INTERVAL=0.25d` は 6 時間を意味します。 **DURATION** または **INTERVAL** の最大期間は 6 週間です。最大期間の 6 週間は、**DURATION** または **INTERVAL** のすべての時間単位に適用されます。たとえば、**DURATION** または **INTERVAL** で指定できる最大日数は 42 日です。
FILENAME	メッセージ型	履歴ログ ファイルから抽出されたデータを含むファイルの名前およびフォルダの場所。
WRITEFILE	整数型	HistData の出力ファイルへの書き込み操作のステータスを示すフラグ。1 に設定すると、HistData は要求データを **FILENAME** アイテム名で指定したファイルに書き込みます。ファイルの更新が完了すると、**WRITEFILE** は自動的に 0 にリセットされます。
ERROR	メッセージ型	ログ ファイルからのデータの抽出中に発生した最後のエラーを説明する文字列。**STATUS** が 1 にセットされると、**ERROR** 文字列は None にセットされます。**STATUS** が 0 にセットされると、**ERROR** 文字列にはエラー メッセージが入ります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アイテム	データ型	説明

TAGS1、**TAGS2** など	メッセージ型	ログ ファイルから抽出されたデータを含む 1 つまたは複数のタグ変数の名前を含む文字列。
		
		TAGS 文字列には、WindowViewer では 131 文字、Excel で 255 文字を指定することができます。
		
		これ以上に長い文字列が必要な場合は、Tagsn という名前のタグ変数アイテムを追加することにより、文字列を追加することができます。ここで、*n* は追加する文字数を表す整数値です。
		
		タグ変数にタグ テキストを追加する必要がある場合は、文字列の最後にプラス記号（**+**）を付けます。
		
		以下に例を示します。
		`TAGS="$Date,ProdLevel,ProdTemp,+"`
		`TAGS1="ReactLevel,Temp,GasLevel,+"`
		`TAGS2="MotorStatus"`
		
		タグ変数を重複して付けることはできません。また、各タグ文字列の最大長は 512 バイトです。

PRINTTAGNAMES	論理型	タグ変数の名前を関連する値の列の上に印刷するかどうかを示すフラグ。1 にセットされると、タグ変数名が印刷されます。0 にセットされると、タグ変数名は印刷されません。
DATA	メッセージ型	要求データを .CSV 形式で HistData プログラムに保存します。他のアプリケーションから DDE 経由でデータを **ADVISE** または **REQUEST** する際に使用します。
ステータス	論理型	HistData の操作の最新のステータス。 1 は、HistData がログ ファイルから履歴データを正常に抽出したことを示します。0 は、エラーが発生したことを示します。
SENDDATA	整数型	HistData の更新操作のステータスを示すフラグ。1 に設定すると、HistData は **DATA** アイテムを要求データで更新します。更新が完了すると、**SENDDATA** は自動的に 0 にリセットされます。 **SENDDATA** で要求されたデータ量が多すぎるというエラー メッセージを受信した場合は、**DURATION** の期間を短くするか、または要求タグ変数の数を減らします。タグ変数を重複して付けることはできません。また、各タグ文字列の最大長は 512 バイトです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

DDE でログ データにアクセスする

ログ データを出力ファイルに抽出するには、2 つの方法があります。

- 8 個以上のタグ変数を含む履歴ログを出力ファイルに保存する場合は、手動の方法で行います。
- 現在履歴トレンドに割り当てられたペンにマップされているログ データを保存するだけなら、［履歴データ］ウィザードを使用します。

HistData を使用してログ データを手動抽出する

ログ データを出力ファイルに手動で抽出することができます。以下の操作を順番に実行します。

1. HistData アクセス名の作成
2. HistData 用 I/O タグの作成
3. HistData ウィンドウの作成
4. HistData の実行

HistData アクセス名の作成

InTouch で HistData プログラムからデータを要求する場合は、以下の手順でアクセス名を定義する必要があります。

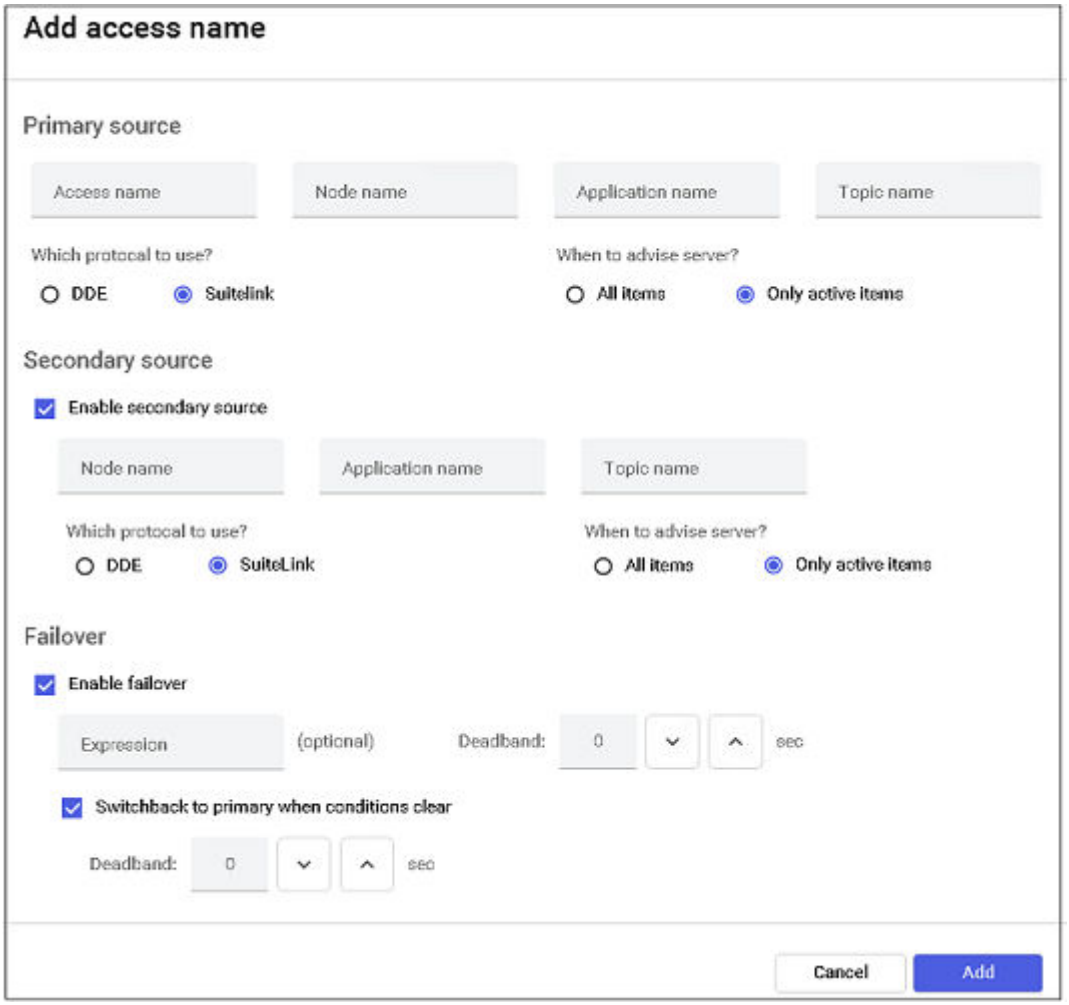
アクセス名を定義する

1. ［**ホーム**］メニューの［**タグ**］グループで［**アクセス名**］を選択します。

   ［**アクセス名**］ダイアログ ボックスが表示されます。
2. ［**追加**］を選択します。

   ［**アクセス名の追加**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［アクセス名］ボックスに、最大 32 文字の半角英数字で名前を入力します。［アクセス名］と［トピック名］には、同じ値を入れてください。
4. ［ノード名］ボックスに、ログ ファイルが現在置かれているノード名を入力します。
5. ［アプリケーション名］ボックスに、ファイル拡張子 .exe を付けずに HistData と入力します。
6. ［トピック名］ボックスに、［アクセス名］ボックスで指定した名前を入力します。［アクセス名］と［トピック名］には、同じ値を入力してください。
7. 適切な通信プロトコルを使用可能なオプションを **DDE** と **Suitelink** から選択します。
8. ［サーバーへの通知］領域で、HistData を使用する場合は常に［すべてのアイテムを要求］を選択します。
9. ［追加］を選択します。

HistData タグを作成する

アクセス名を定義したら、以下の I/O タイプのタグ変数を作成して、ログ ファイルからのデータを含む出力ファイルを 1 つ生成します。前の手順で作成したアクセス名をタグ変数に割り当てます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ	I/O 型タグ タイプ	アイテム
HDWDATADIR	メッセージ型	DataDir
HDWDBDIR	メッセージ型	DbDir
HDWDURATION	メッセージ型	間隔
HDWERROR	メッセージ型	エラー
HDWFILENAME	メッセージ型	FileName
HDWINTERVAL	メッセージ型	間隔
HDWSTARTDATE	メッセージ型	StartDate
HDWSTARTTIME	メッセージ型	StartTime
HDWSTATUS	メッセージ型	ステータス
HDWTAGS、**HDWTAGS1**、**HDWTAGS2**	メッセージ型	タグ
PRINTTAGNAMES	論理型	PrintTagNames
HDWWRITEFILE	整数型	WriteFile

注意：［HistData ウィザード］では、これらのタグを自動的に作成できます（ただし、PRINTTAGNAMES タグは除きます）。

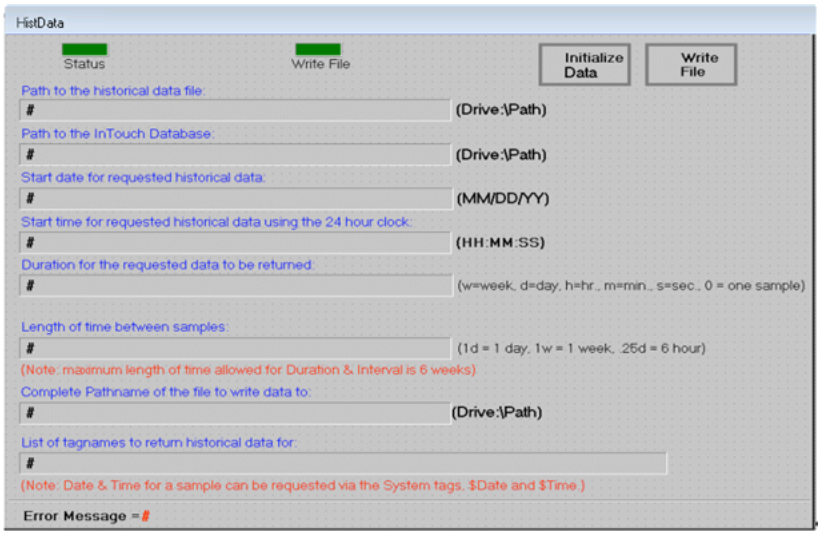
ログ データをデータ アイテムに送信して、他のアプリケーションからもアクセスできるようにするには、さらに 2 つのタグ変数を作成します。また、［HistData ウィザード］では、HDWSendData と HDWData タグは自動的に作成されません。

タグ	I/O 型タグ タイプ	アイテム
HDWSendData	論理型	**SendData**
HDWData	メッセージ型	データ

HistData ウィンドウの作成

I/O タイプのタグを作成してから、［**HistData**］と呼ばれる新しいウィンドウを作成します。これは、以下の例に似ています。

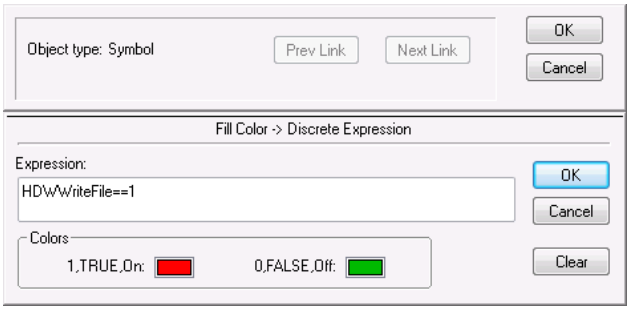
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



シンボルは、ユーザー入力リンクにリンクします。たとえば、# シンボルには、HDWDataDir タグへのユーザー入力/文字列リンクがあります。ユーザー入力リンクでは、ランタイム中にタグの値を変更できます。

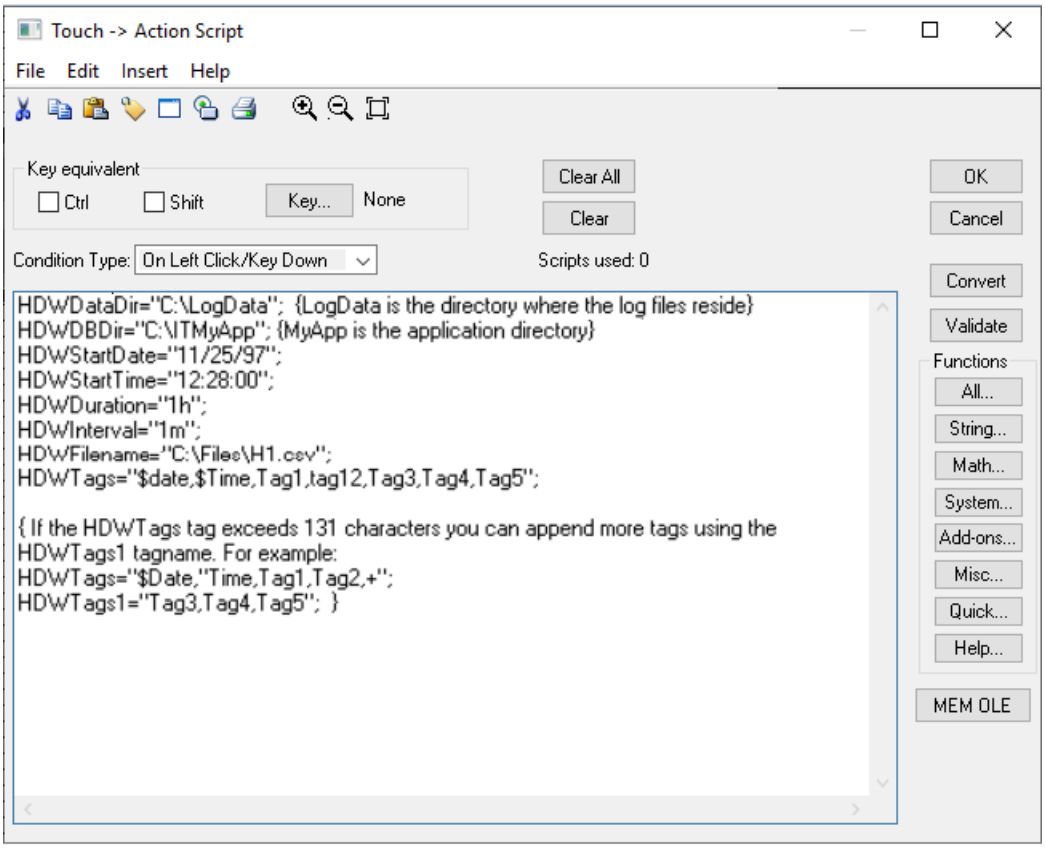
［ステータス］ボタンは、HDWStatus タグに基づいた塗りつぶし色-論理型式にリンクしています。

［ファイル書き込み］ボタンは、HDWWriteFile タグに基づいた塗りつぶし色-論理型式にリンクしています。



［データ初期化］ボタンは、［タッチ押しボタン］アクション スクリプトにリンクしています。以下に例を示します。

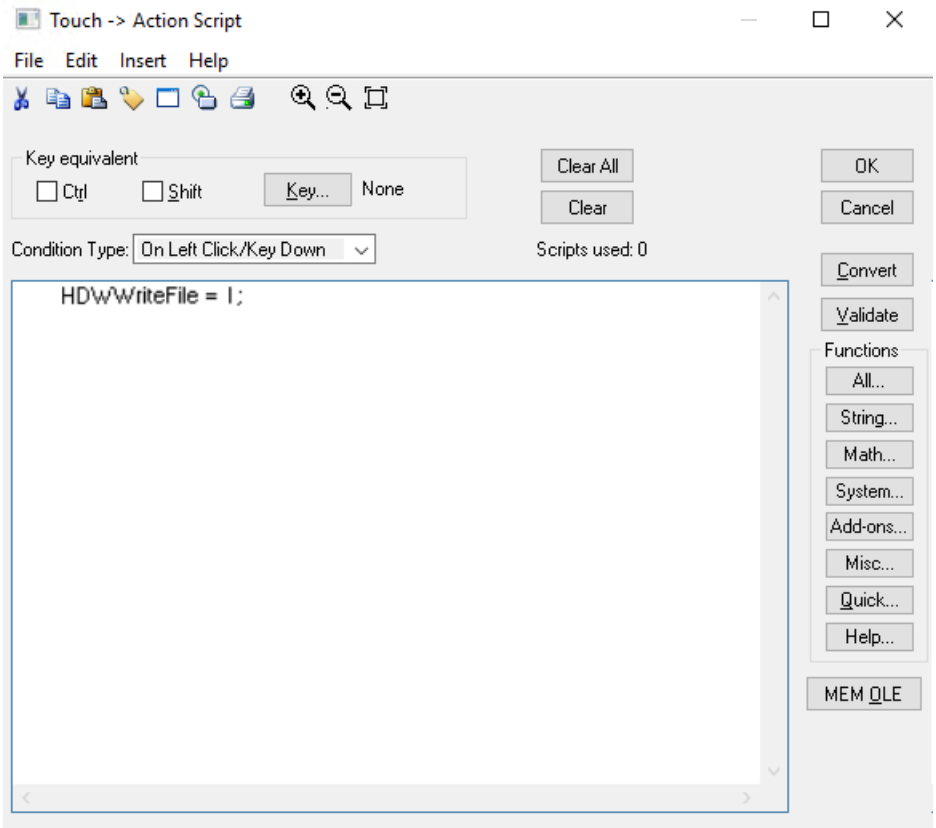
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



［データ初期化］ボタンを選択すると、HistData アイテムは目的の値に初期化されます。ユーザー入力リンクを使用すると、必要に応じてランタイム中にアイテムの値を変更できます。

［ファイル書き込み］ボタンは、［タッチ押しボタン］アクション スクリプトにリンクしています。［**WriteFile**］ボタンを選択すると、出力ファイルが生成されます。［ファイル書き込み］ ボタンの設定は次のとおりです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



HistData の実行

HistData ウィンドウを作成してから、以下の手順を完了させて、WindowViewer で実行します。

HistData ウィンドウを実行する

1. ［HistData］を開始して最小化します。
2. WindowViewer を起動して、［HistData］ウィンドウを開きます。
3. ［初期化］ボタンを選択し、HistData アイテムを変更します。
4. ［ファイル書き込み］ボタンを選択します。

   操作が成功すると、ステータスの値はオンになり、オン ステータスに関連付けられている色が表示されます。操作が失敗すると、ステータスの値はオフになり、失敗の原因を示すエラー メッセージが表示されます。

詳細については、「HistData ウィンドウの作成」を参照してください。

［HistData ウィザード］を使用してログ データを抽出する

履歴トレンドに表示されるログ データの含まれる出力ファイルを作成することができます。InTouch には［HistData ウィザード］が用意されており、ログ ファイルから自動的にデータを抽出します。

HistData の出力ファイルには履歴トレンドに表示されるログ データが書き込まれるため、このファイルには履歴トレンドのペンに現在割り当てられているタグからのデータだけが入ります。

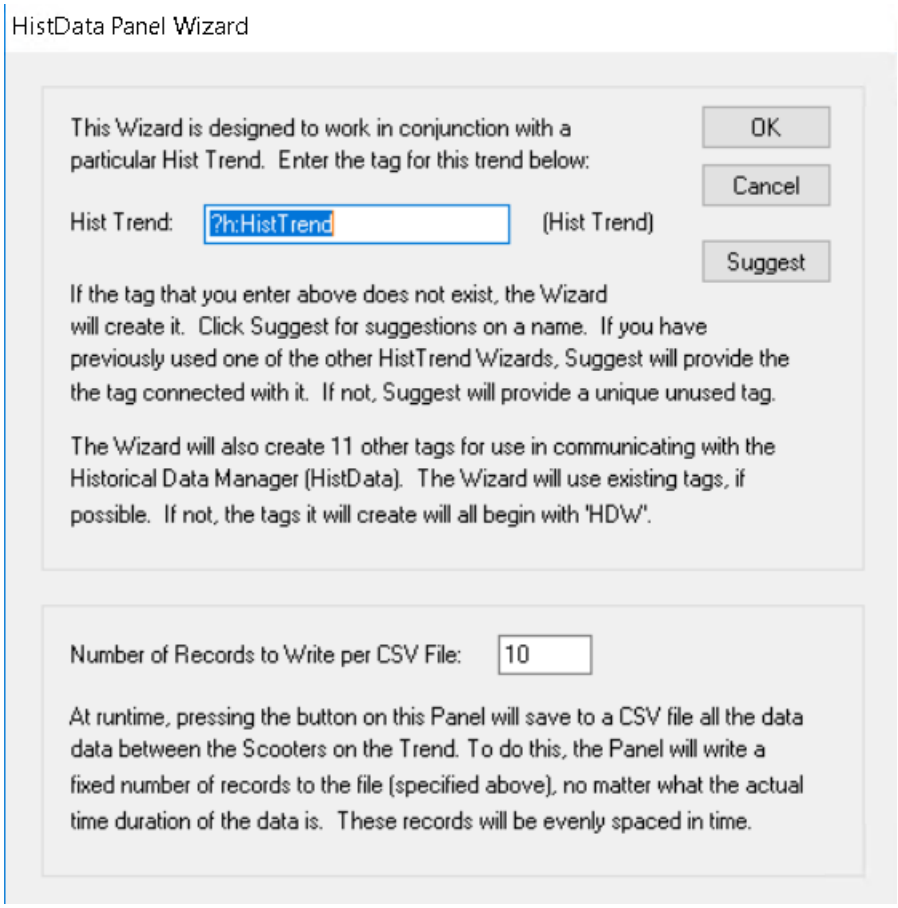
［HistData ウィザード］を使用してログ データを抽出する

1. WindowMaker を起動する。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. 履歴トレンドが表示されたウィンドウを開きます。
3. ［**描画**］メニューの［**挿入**］グループで［**ウィザード**］を選択します。

   ［**ウィザードの選択**］ダイアログ ボックスが表示されます。
4. 左ペインから、［**トレンド**］グループを選択します。
5. 右ウィンドウから、**HistData ウィザード**アイコンを選択し、［**OK**］を選択します。
6. トレンド ウィンドウ上の HistData オブジェクトを配置する場所にマウス ポインタを移動させます。
7. トレンド ウィンドウで、HistData オブジェクトを配置する場所を選択します。出力ファイルが作成されるパスが表示された［**ファイル**］ボックスとボタンで構成されたウィンドウ オブジェクトが開きます。
8. トレンド ウィンドウで［HistData ウィザード］オブジェクトをダブルクリックします。［**履歴データ パネル ウィザード**］ダイアログ ボックスが表示されます。



9. ［**履歴トレンド**］ボックスに HistTrend タグの名前を入力します。
10. ［**CSV ファイルごとに書き込むレコード数**］ボックスに、出力ファイルに書き込むレコード数を入力します。
11. ［**OK**］を選択します。HDW 接頭辞で識別されるタグのセットが作成されます。

    「HistData タグを作成する」に記載されているタグが作成されます。HisDataViewSt のアクセス名にタグが割り当てられます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

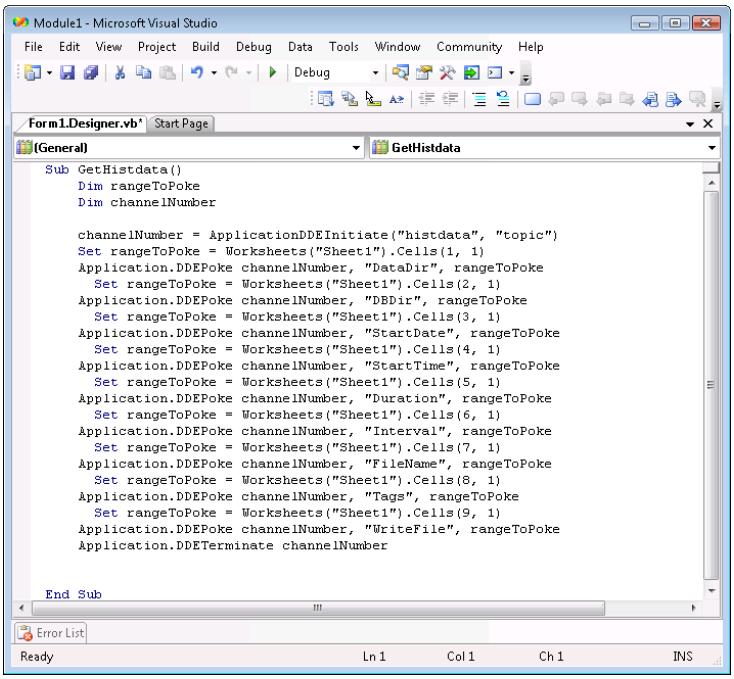
12. WindowViewer で履歴トレンド ウィンドウを実行します。
13. HistData ウィンドウ オブジェクトの中にある［**ファイルへ保存**］を選択します。ウィンドウ オブジェクトに表示されているフォルダの場所に、出力ファイルが作成されます。

注意：新しく作成された *.idx ファイルおよび *.lgh ファイルで InTouch を起動後 1 時間以内にタグの値が更新されない場合、.csv ファイルは出力されません。
HistData データには、履歴トレンド スクータと同じ解像度がありません。HistData の解像度は、時間範囲内で要求された値の数に基づきます。これは *.idx ファイルと *.lgh ファイルの値が正確に反映されたものではありません。

他のアプリケーションから履歴データにアクセスする

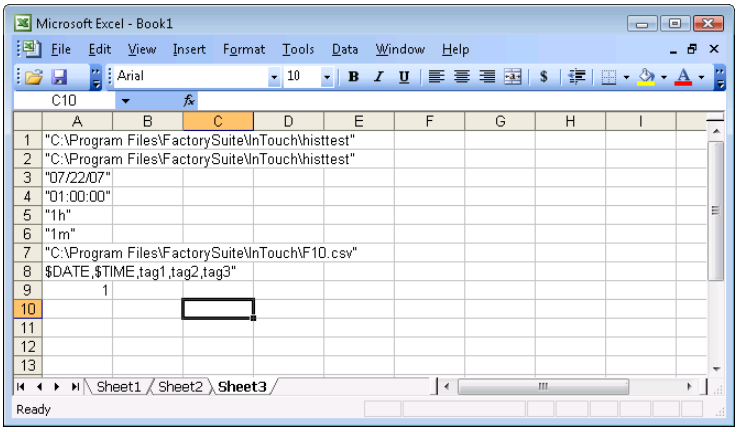
Excel マクロを記述して、HistData ファイルからデータを取得できます。HistData プログラムは、マクロの INITIATE、POKE、および TERMINATE 関数に対応しています。キーワード（内部データベース アイテム）を持つ POKE 関数は、クエリーを定義するパラメータの設定に使用します。クエリーを正しく設定すると、マクロが実行され、HistData ファイルから指定された履歴データを要求します。

以下に、VBA で書いたマクロの例を示します。



上の例では、POKE したデータは Sheet1 に入っています。次に、POKE したデータの例を示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



HistData エラーのトラブルシューティング

HistData でログ データを抽出しているときに発生する可能性のあるエラーを確認することができます。以下の表には、左のカラムに標準的な HistData の問題またはエラー メッセージが一覧表示されています。右のカラムは、その問題の考えられる原因と解決方法です。

エラー メッセージまたは状況	原因/解決方法
エラー メッセージ: 要求するデータが多すぎます。期間を短くするか、または要求するタグ変数の数を減らしてください。	このエラーは、SendData アイテムに要求されるデータ量が多すぎる場合に発生します。 **SendData** アイテムは、ログ ファイルからのデータが入った出力ファイルを作成するためだけに使用しないでください。
エラー メッセージ: ファイル `C: \FILES1\HISTDATA.CSV` を開くことができません	フォルダ パスが存在しないか、またはフォルダ パスのスペルが間違っています。
エラー メッセージ: ファイル `C:\FILES\`を開くことができません	出力ファイルが定義されていません。
エラー メッセージ: `DATADIR` アイテムが無効です	**DataDir** アイテムで指定された宛先フォルダ パスが存在しません。フォルダ パスのスペルを確認してください。
エラー メッセージ: `STARTDATE` アイテムが無効です	**StartDate** アイテムに無効な形式の開始日が含まれています。Windows の場合は、コンピュータの日付形式を mm/dd/yy に変更してください。
エラー メッセージ: ログ ファイルが見つかりません	**DataDir** アイテムで指定されたパスに要求された日付のログ ファイルが存在しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

エラー メッセージまたは状況	原因/解決方法
エラー メッセージ: タグ名 `TAG` は· データベースにありません	要求されたタグ変数が、アプリケーションのタグ変数ディクショナリに存在しません。タグ変数名のスペルが正しいことを確認してください。
エラー メッセージ: `C:\IT6.0B\HISTEST` に `tagname.x` が見つかりません。	ファイル **tagname.x** が存在しないか、または破損しています。
.csv ファイルが出力されず、エラーは表示されません。	• HistData が稼動していません。 • **Tags** アイテムの一覧に、ログ用に指定されているタグ変数が含まれていません。 • HDWWritefile の定義が正しくありません。タグ変数が整数型であること、DDE のアクセス名が正しいこと、アイテムが **WriteFile** であることを確認してください。また、**MinEU=MinRaw** および **MaxEU=MaxRaw** となるスケーリングがないことも確認してください。
出力された .csv ファイルには、日付と時刻は含まれていますが、要求されたタグ変数の記録日が入っていません。	要求された時間中、タグ変数の履歴ログにエントリがありません。履歴トレンドを表示して、要求された時間内にログ ファイルにデータが入るかどうかを確認してください。
WWLogger に以下のメッセージが表示されています。 `DDE HistData|Viewstream1` にエラーがあります `WriteFile: Poke` コマンドはサーバーから拒否されました。	HistData アイテムへの値の割り当てエラーのために、.csv ファイルの作成が失敗するたびに、WWLogger にエラー メッセージが書き込まれます。このエラーは、**WriteFile** アイテムを 0 にセットしようとした場合、またはエラー アイテムに書き込みを行おうとした場合にも発生する可能性があります。
ログ ファイルに多数のレコードがあると、.csv ファイルには 1 つのレコードしか書き込まれません。	**Interval** アイテムが間違った値に指定されて、収集期間が非常に短くなっている可能性があります。 または、**Duration** アイテムの形式が、`HDWDuration=1-` などのように無効である可能性があります（増分を指定することはできません）。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

IEEE 小数ユニット

Historian HMI では、32 ビットの 2 進数の値を浮動小数点数に変換するために、IEEE（Institute of Electrical and Electronics Engineers）754 規格を使用しています。

IEEE 754 形式の 32 ビット数は、16 ビットの 2 ワードとして 16 ビット PLC（Programming Logic Controllers）に保存されます。一般的に、PLC の浮動小数点レジスタの番号付け体系では、16 進数の下位ワードと上位ワードが連続しています。現在の 32 ビット パーソナル コンピュータでは、32 ビット単一レジスタが使用されています。レジスタのビット番号付け体系は、2 つの連続する 16 ビット レジスタと同じ形式です。

Historian アプリケーションで浮動小数点を使用するには、Historian が 2 つの 16 ビット PLC レジスタに保存された値を変換できなければなりません。Historian は常に PLC レジスタの生の値を個々の整数とみなすため、ビット変換を行う必要があります。2 つのレジスタ整数でブール値 **AND** 演算を行い、それを実数に変換することはできません。Historian は 2 つの整数レジスタのタイプ変換を行うことはできません。

Historian HMI で浮動小数点数を表示する

Historian HMI では、アプリケーション内の実数を表示するために IEEE 32 ビット浮動小数点形式を使用しています。IEEE 浮動小数点形式は、実際の実数の近似に過ぎません。IEEE 32 ビット浮動小数点形式では、実数が 2 の累乗でない限り、正確に表現することはできません。IEEE 32 ビット浮動小数点数の精度は、小数点以下約 8 桁です。

実数を Historian アプリケーションで表示する場合は、数値が 8 桁を超えないよう注意してください。以下に、Historian アプリケーションで有効な実数を表記する浮動小数点数形式を示します。

#

#,###

#.##

0#

###.##

#.#######

###.#####

######.##

ABCDEF

###.####

浮動小数点数が 8 桁を超えると、切り上げの誤差が生じることがあります。

テキストのフォーマットに小数点を含めない場合、数値は小数点と共に表示されます。実際のフォーマットの小数点の位置は WindowViewer の［詳細な書式設定］プロパティで設定されます。

注意：小数点の左側に「#」を追加した場合、または小数点がない場合、表示する桁数を制限しないでください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例 1

Historian アプリケーションで実数 2.3 を表示します。ただし、2.3 は 2 の累乗ではないため、IEEE 32 ビット浮動小数点形式で 8 桁以上の数を正確に表記することはできません。

アプリケーションが値 2.3 を ASCII 文字の 2.3 と認識できるようにするには、この数が 8 桁以下である必要があります。上限の 8 桁を超えた場合、この数は 2.29999999 または 2.30000001 と表記されます。

例 2

2 つの実数型タグが比較されると、2 つの実数型タグの差分は FLT_EPSILON（値 1.19209290E-07F、10 進数では 0.0000001192092896）よりも大きくなる必要があります。しかし、数値が 8 桁を超えると、結果の値は正しくなくなることがあります。この問題を修正するには、値を 1000 で乗算するか、それより大きい 10 の倍数で乗算します。この処理を行うことにより、値が 1e-7 よりも大きくなります。必要な比較演算を実行し、1000（または乗算した数）で除算します。

InTouch OPC UA サーバーを設定して使用する

InTouch HMI は、マシン ツー マシン通信用の OPC UA（Unified Architecture）プロトコルをサポートします。

この OPC UA サーバー機能を使用すると、サード パーティ クライアントが InTouch HMI と対話して、OPC UA などの業界標準を活用できます。InTouch HMI で OPC UA サーバー機能を有効にすると、クライアントは組み込みの Gateway 通信ドライバまたはサード パーティ クライアントを利用して InTouch HMI に接続し、Gateway 通信ドライバ またはクライアントを使用して OPC UA ネームスペースを安全に参照して InTouch HMI と対話できます。

OPC UA の設定とチェックリスト

OPC UA サーバーおよび OPC UA クライアントのエンド ツー エンドの設定に必要なタスク

設定がタスクは、完了する必要のある順序で示されています。

1. **システム管理サーバーの設定 :** System Management Server は、マシン間での信頼関係を構築するために使用するため、ノード間の安全な通信を保証するように設定する必要があります。System Management Server は、通常、最初の System Platform をインストールするときに設定します。詳細については、『*System Platform のインストール*』の「System Management Server の設定」を参照してください。

   注意 : セキュアな通信を可能にする暗号化証明書にアクセスするために、InTouch HMI は管理特権を持つユーザーのコンテキストで実行する必要があります。

2. **OPC UA サーバーの設定 :** 構成オプションの設定、OPC UA サーバー接続のテスト、Gateway 通信ドライバのアクティブ化を行います。

3. **IT コンプライアンス/ファイアウォールの検証 :** 構成のこの時点でファイアウォールの設定と検証を完了する必要があります。OPC UA サーバーが配置されているノードには、ファイアウォールの受信規則が構成され、検証されている必要があります。

   重要! 残りの設定タスクを行う前に、ファイアウォール テストを正常に完了する必要があります。

4. **OPC UA クライアントの設定 :** クライアント設定には以下が含まれることがあります。
   - OPC UA サーバー アドレスの定義（形式は `opc.tcp://<ServerName>:<PortNumber>`）。
   - 正しい OPC UA サーバー セキュリティ ポリシーの選択（`Basic256Sha256`）。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ユーザーを InTouchHMIOPCUAWriteUsers ユーザー グループに追加します。
- 設定済みの OPC UA ユーザー資格情報（ユーザー名およびパスワード）を入力します。
- 匿名接続は、InTouch タグの読み取りでのみサポートされます。セキュリティ リスクを回避するために、認証済みの資格情報を使用してデータにアクセスすることが推奨されます。

5. **セキュリティ証明書：**ランタイム ノードに OPC UA セキュリティ証明書をダウンロードして設定します。
6. **接続の検証：**OPC UA クライアントを開き、OPC UA サーバーに接続できること、およびネームスペースのアイテムを表示できることを確認します。

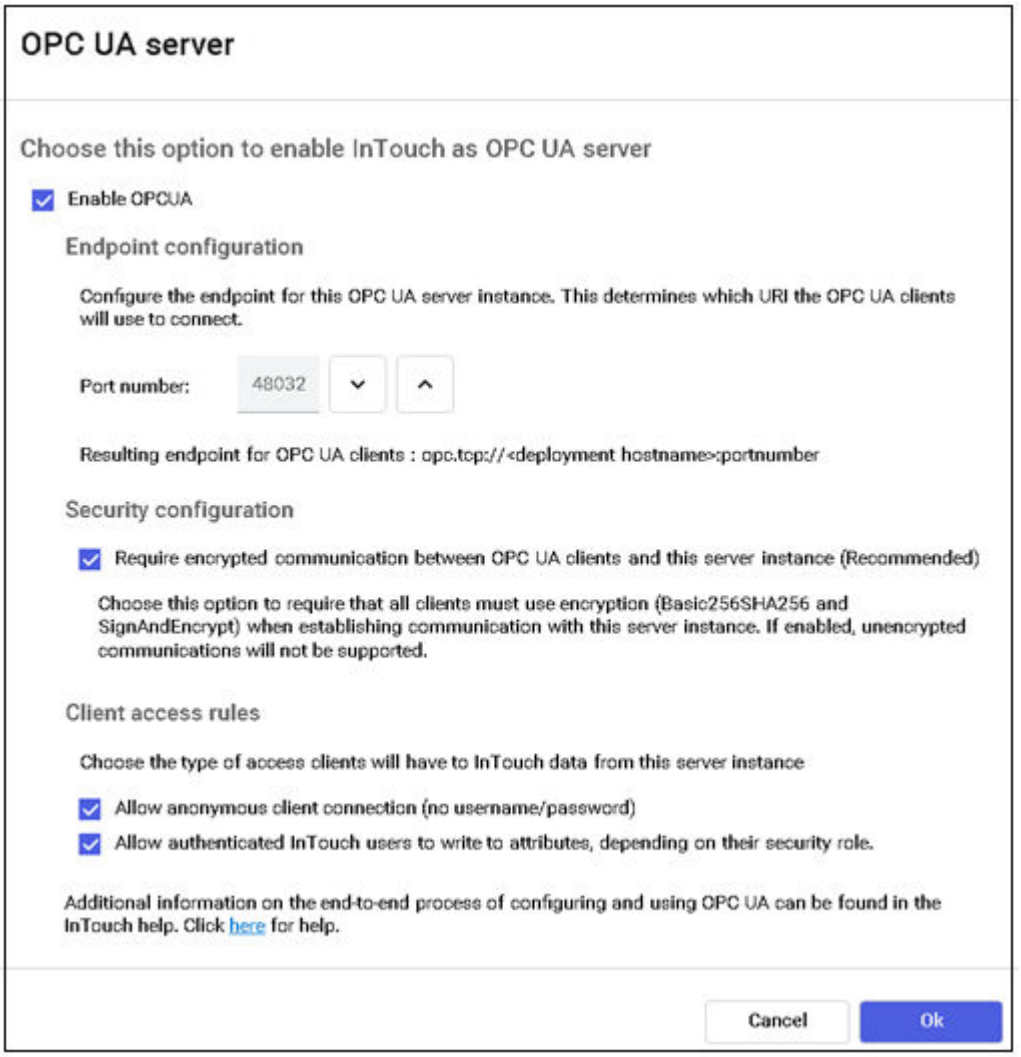
InTouch OPC UA サーバーを設定する

InTouch OPC UA サーバーは、ゲートウェイやその他のプロトコル変換メカニズムなしで OPC UA クライアントから InTouch HMI データへのアクセスを提供します。

1. **AVEVA InTouch HMI アプリケーション マネージャ**を起動します。
2. ［ツール］タブで［**OPC UA 設定**］を選択します。

   ［OPC UA サーバー］ダイアログ ボックスが表示されます。
3. OPC UA サーバーはデフォルトで無効になります。サーバーを設定するには、［**OPC UA を有効にする**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. **ポート番号**を編集します。デフォルトで、ポート番号は 48032 に設定されます。

   ポート番号は、ユーザーおよび RDS セッションに対して一意です。追加の RDS セッションには代替のポート番号を割り当てます。

5. **セキュリティ設定：**通信でペイロードが暗号化されるため、このオプションを有効化することが強く推奨されます。クライアント側も、この設定に一致する必要があります。
6. ［クライアント アクセス ルール］オプションを使用すると、InTouch データに対するクライアントのアクセスのタイプを決定できます。
   a. 匿名アクセスを許可するには、［**匿名クライアント接続を許可（ユーザー名とパスワードの入力は不要です）**］チェック ボックスをオンにします。
   b. 認証されたユーザーだけを許可するには、［**セキュリティの役割に応じて、認証済みの InTouch ユーザーによる属性への書き込みを許可する**］チェック ボックスをオンにします。
7. ［**OK**］を選択します。
8. 設定後、WindowViewer を起動して OPC UA サーバーを起動します。これで OPC UA クライアントから InTouch HMI データにアクセスできるようになります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

OPC UA サービス用のファイアウォールを設定する

InTouch HMI の OPC UA 通信では、ランタイム ノードのファイアウォールで OPC UA クライアント ノードとの通信を許可する必要があります。しかし、ファイアウォール設定を変更する前にファイアウォールをテストするを実行することが推奨されます。

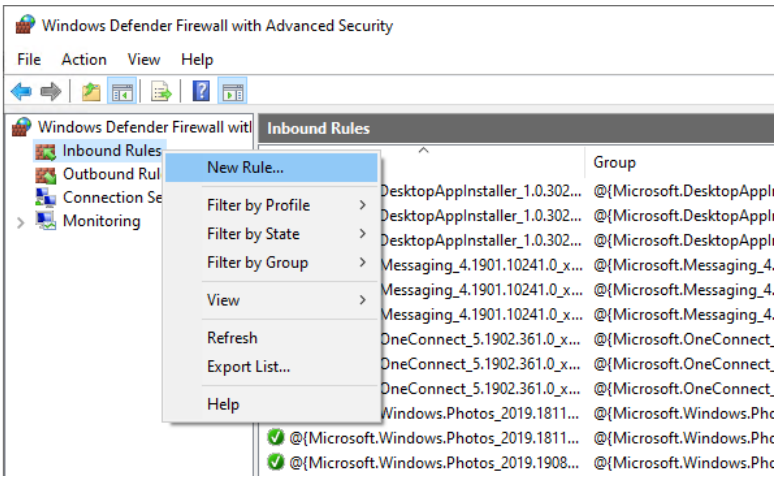
ランタイム ノードのファイアウォールを設定する

重要：OPC UA サーバー サービスを配置するノードにファイアウォール ルールを追加する必要があります。

ランタイム ノードのファイアウォールを設定する

OPC UA サーバー サービスを配置したランタイム ノードで Windows ファイウォールを開き、次のように設定します。

1. Windows の検索バーを使用して **Windows ファイアウォール**を開きます。
2. ［**詳細設定**］を選択し、**受信規則**を作成します。

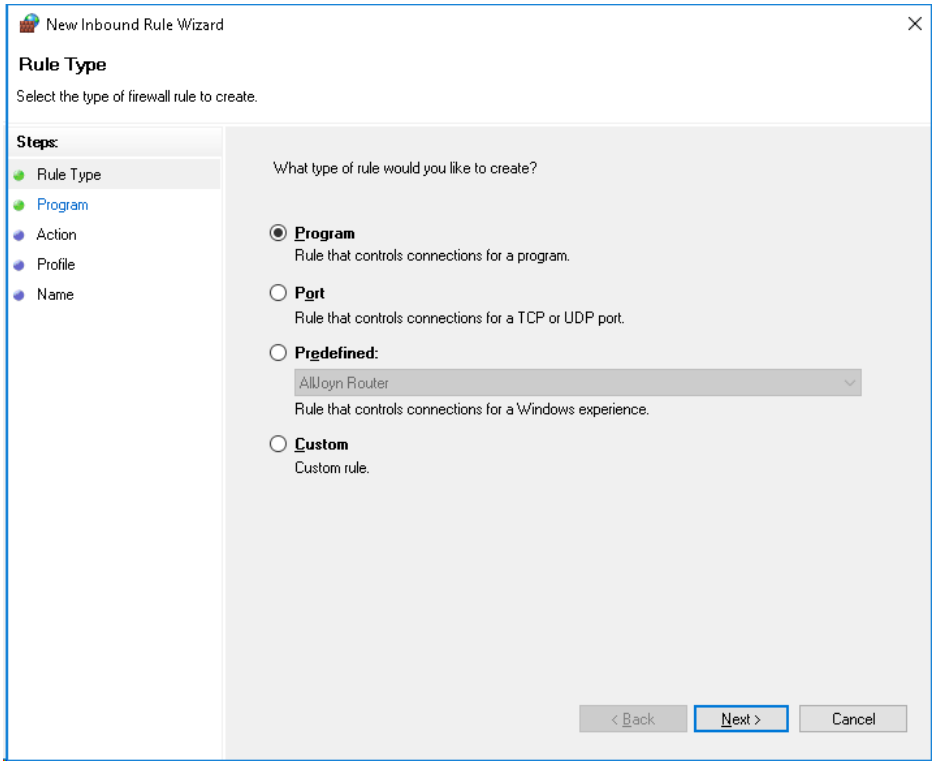


3. ［**新しい規則**］ を選択します。

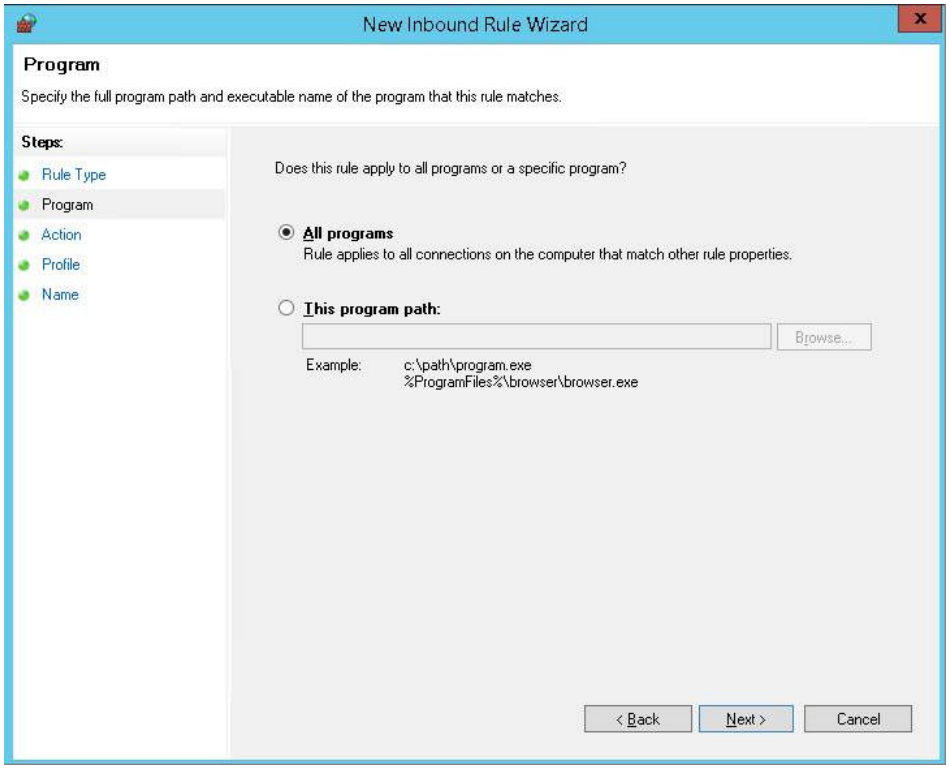
   新規の受信の規則ウィザードが開きます。

4. 規則のタイプに［**プログラム**］を選択して［**次へ**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

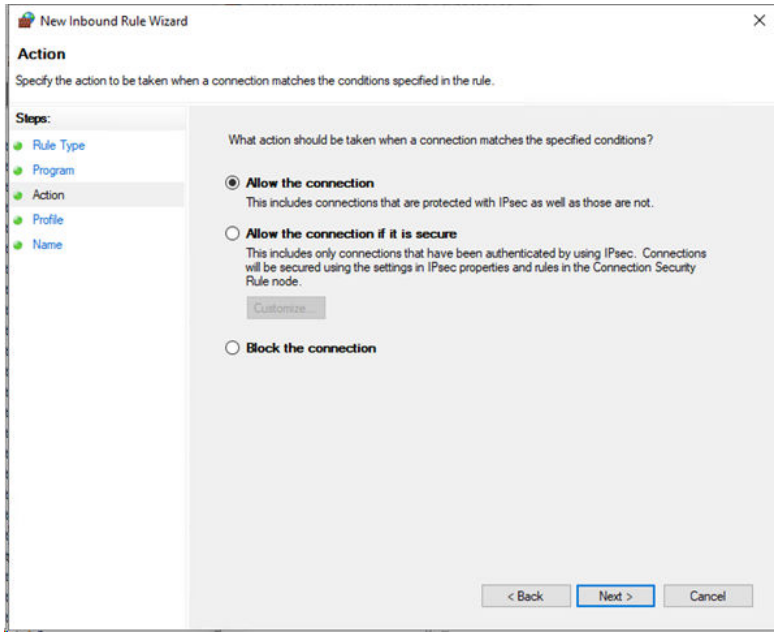


5. 選択したルールのプログラム パスを指定します。

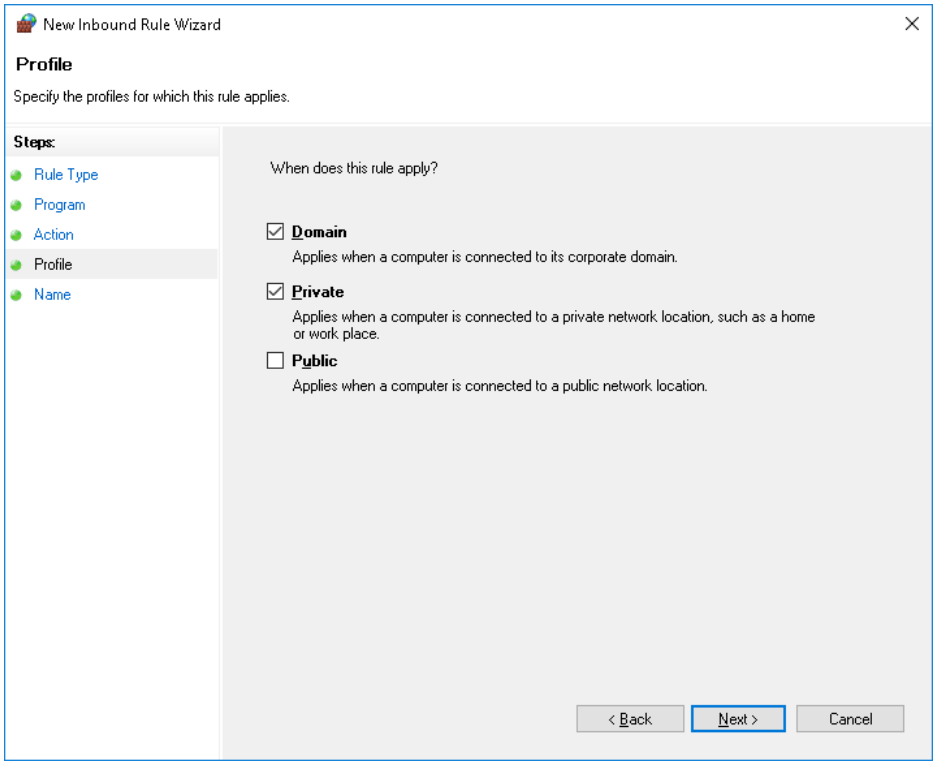


6. 次の画面で［**接続を許可する**］オプションを選択します。**[次へ]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

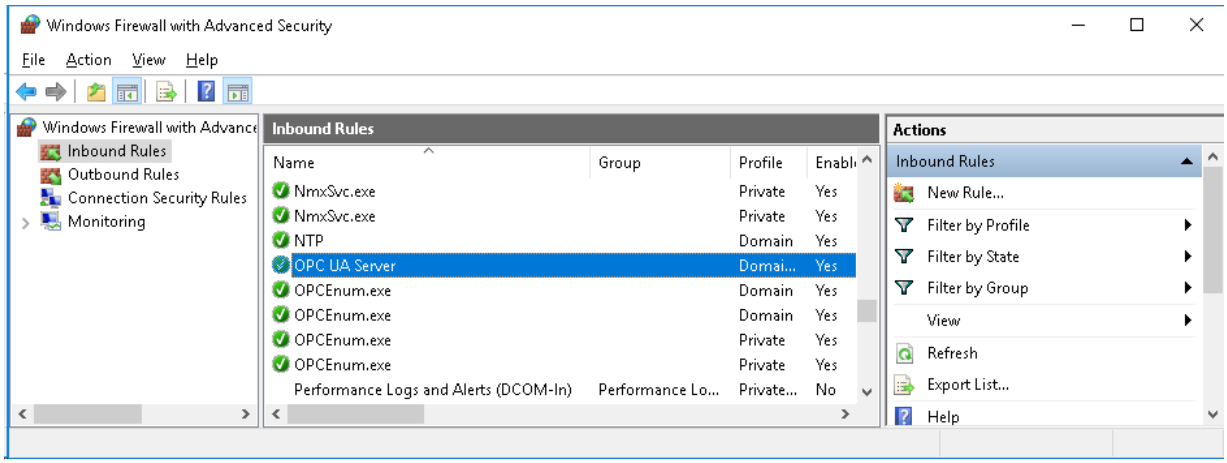


7. 規則を適用するときを尋ねる画面が表示されます。
   - ドメイン環境の場合：[ドメインとプライベート] を選択します。[パブリック] の選択を解除することをお勧めします。
   - ワークグループ環境の場合：[パブリック] を選択します。[ドメイン] および [プライベート] の設定は、ワークグループ環境では何の影響も及ぼしません。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

8. 最後に、このルールの名前を入力します（「OPC UA サーバー」など）。複数の OPC UA サービスを設定する場合、各サービスを区別できるような名前を使用してください。
9. 新しい規則が Windows ファイアウォールの受信規則のリストに追加され、有効になっていることを確認します。



10. ファイアウォールのテストを繰り返して OPC UA クライアント ノードからランタイム ノードに接続できることを確認します。

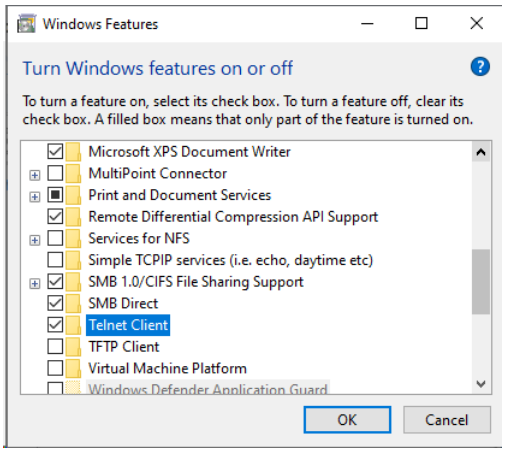
ファイアウォールをテストする

Telnet でファイアウォールのテストを実行する

1. OPC UA サーバーが実行している別のノードから、Windows の **Telnet Client** 機能を有効化して **Telnet** を有効にします。

   注意：このテストは、OPC UA クライアント ノードから実行します。

   a. Windows の**コントロール パネル**を開きます。
   b. ［**プログラムと機能**］を開き、［**Windows の機能の有効化または無効化**］を選択します。
   c. 機能のリストを下にスクロールして ［**Telnet Client**］を有効にします。Telnet はデフォルトで無効化されています。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. このテストを実行する前に、WindowViewer が実行していることを確認してください。OPC UA サービス ホストが設定されている場合、WindowViewer で InTouch OPC UA サービスが起動します。詳細については、「InTouch OPC UA サーバーを設定する」を参照してください。
3. OPC UA クライアント ノードのコマンド ウィンドウで以下を入力して Telnet を実行します。

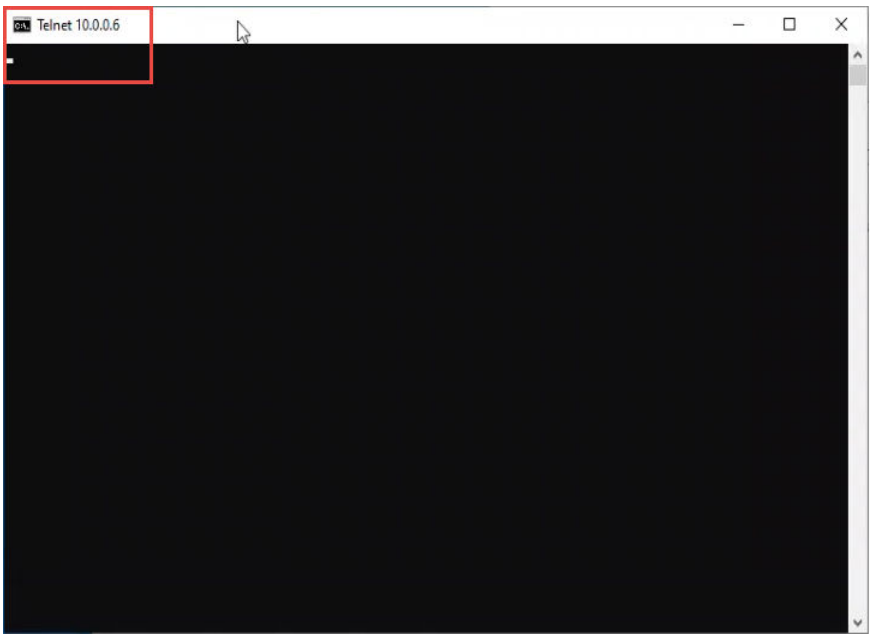
   telnet <nodeName または ipAddress> <portNumber>

   構文の要素を以下に示します。

   - **nodeName** は、InTouch HMI ランタイム ノードのマシン名です。nodeName か ipAddress のいずれかを使用してください（両方使用することはできません）。
   - **IPAddress** は、InTouch HMI ランタイム ノードの IP アドレスです。nodeName か ipAddress のいずれかを使用してください（両方使用することはできません）。
   - **portNumber** は OPC UA サービス用に InTouch HMI で設定したポート番号です。デフォルトのポート番号は 48032 です。

   例: `telnet 10.10.10.06 48031`

   - コマンドが正常に処理されない場合はタイムアウトが発生して、接続に失敗したことを示すメッセージが表示されます。その場合、「ランタイム ノードのファイアウォールを設定する」に進んでください。
   - Telnet コマンドが正常に処理された場合、コマンドプロンプトが Telnet プロンプトに変わります。



   - ファイアウォールのテストが正常に完了したら、OPC UA クライアントおよび OPC UA サーバー証明書を設定します。OPC UA 接続のセットアップの次のステップは、サードパーティ製 OPC UA クライアント アプリケーションを使用するか、Gateway 通信ドライバで使用可能な OPC UA 接続を使用するかに応じて異なります。

サードパーティ製 OPC UA クライアント アプリケーション用のサーバーおよびクライアント証明書を設定する

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

重要! これらの手順は、サードパーティ製 OPC UA クライアントを使用する場合にのみ適用されます。OPC UA クライアントとして Gateway 通信ドライバを使用する場合は、「OI Gateway を使用してクライアント セキュリティ証明書を設定する」を参照してください。

InTouch OPC UA サーバーとサードパーティ製 OPC UA クライアントの間の通信を暗号化するには、両方のコンピュータに以下の証明書へのアクセスが必要です。

- <コンピュータ名> ASB OPC UA サーバー
- OPC UA クライアントからのクライアント証明書。

このセットアップを完了するには、3 つの手順を実行する必要があります。

1. OPC UA サーバー ノードから証明書を OPC UA クライアント ノードにコピーします。この手順には以下の操作が含まれます。
   - OPC UA サーバー証明書のエクスポート。
   - 証明書のクライアントノードへのインストール。
2. OPC UA クライアント ノードの証明書の OPC UA サーバー ノードへのコピー。
3. ファイアウォールが InTouch.OPCUA.ServiceHost.exe アプリケーションを許可するよう設定されていることを確認してください。

OPC UA クライアント ノードへの OPC UA サーバー証明書のエクスポート

OPC UA サーバー証明書をエクスポートする

1. OPC UA サーバー ノードで Windows の証明書マネージャを開きます。

   証明書マネージャを開くには、Windows の検索ボックスに「コンピュータ証明書の管理」と入力するか、コマンド プロンプトを開いて `certlm.msc` を実行します。
2. ツリー ビューで［**個人**］ノードを展開して、［**証明書**］を選択します。
3. 「**<コンピュータ名> ASB OPC UA サーバー**」証明書を選択します。
4. 証明書を右クリックして、［すべてのタスク］>［エクスポート］を選択します。

   ［**証明書のエクスポート ウィザード**］が開きます。
5. クライアント アプリケーションが使用する証明書のタイプに応じて、証明書を 1 つまたは 2 つの証明書ファイル タイプとしてエクスポートする必要があります。
   - **.cer ファイル**: クライアントが Windows 証明書ストアを使用する場合
   - **.der ファイル**: クライアントがファイル ベースの証明書を使用する場合

   ファイル拡張子が異なっていてもファイル形式は同じです。
6. ［証明書のエクスポート ウィザード］で以下のオプションを選択します。
   - **秘密キーのエクスポート**：［いいえ、秘密キーをエクスポートしません］（デフォルト）を選択し、［**次へ**］を選択します。
   - **エクスポート ファイルの形式:** クライアント アプリケーションの要件に応じて［DER エンコードされたバイナリ X.509 (.CER)］または［Base-64 でエンコードされた X.509 (.CER)］のいずれかを選択します。選択してから、［**次へ**］を選択します。
   - **エクスポートするファイル:** ルート CA をエクスポートするファイル名（`c:\temp\<マシン名> OPC UA Server.cer` など）を入力し、［**次へ**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

7. ［証明書のエクスポート ウィザード］の処理が完了した後、クライアント アプリケーションの要件に応じて、証明書のファイル拡張子を ".cer" から ".der" に変更する必要がある場合があります。この操作は、OPC UA クライアント アプリケーションで証明書が Windows 証明書ストアではなく、特定のフォルダに格納される場合に行う必要があります。

クライアント コンピュータに OPC UA サーバー証明書をインポートする

OPC UA サーバー証明書のコピーと OPC UA クライアント ノードへのインストールの手順を完了するには、OPC UA サーバー証明書を OPC UA クライアント コンピュータにインポートする必要があります。

各 OPC UA クライアント アプリケーションには、証明書を管理するための独自のメカニズムがあります。一般的に、OPC UA クライアントは、次の 2 つのメカニズムのいずれかを使用して証明書を管理します。

- Windows 証明書ストアの利用
- OPC UA クライアント アプリケーションで定義された特定のフォルダへの証明書の格納。

サーバー証明書をインポートする方法の詳細については、使用する OPC UA クライアント アプリケーションのドキュメントを参照してください。

証明書を OPC UA クライアントの Windows 証明書ストアにインポートする

1. 証明書ファイル（<マシン名> OPC UA Server.cer）を OPC UA クライアント ノードにコピーします。ファイルをコピーする場所は重要ではありません。
2. 証明書ファイルを右クリックし、コンテキスト メニューから［**証明書のインストール**］を選択します。

   証明書のインポート ウィザードが開きます。

   注意： 証明書をインストールするには管理者特権が必要です。

3. 証明書のインポート ウィザードで以下のオプションを選択します。
   - **保存場所：**［ローカル マシン］を選択し、［**次へ**］を選択します。
   - **証明書ストア：**［個人用］を参照し、［**次へ**］を選択します。
   - **証明書のインポートウィザードを完了する：** 設定を確認し、［**終了**］を選択します。

証明書を OPC UA クライアント上の特定の場所にコピーする

1. 証明書ファイル（<マシン名> OPC UA Server.cer）を OPC UA クライアント アプリケーションで指定されているフォルダにコピーします。以下の表には、一般的な OPC UA クライアントの証明書フォルダの場所が示されています。

OPC UA クライアント	**製造業者**	**フォルダ**
dataFEED OPC UA クライアント	Softing	C:\ProgramData\Softing\OpcClient\pki\trusted\certs
UaExpert	UnifiedAutomation	C:\Users\Admin\AppData\Roaming\unifiedautomation\uaexpert\PKI\trusted\certs
UA クライアントの使用を開始する	UnifiedAutomation	C:\ProgramData\unifiedautomation\UaSdkNetBundleEval\pkiclient\trusted\certs

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

OPC UA クライアント	製造業者	フォルダ
Matrikon	Matrikon	C:\Users\Admin\Appdata\Local\Matrikon\OPCUAExplorer\pki\DefaultApplicationGroup\trusted\certs
KEPServer	Kepware	C:\ProgramData\Kepware\KEPServerEX\V6\UA\Client Driver\cert
上位サーバー	ソフトウェア ツールボックス	C:\ProgramData\Software Toolbox\TOP Server\V6\UA\Client Driver\cert

証明書の管理の詳細については、該当する OPC UA クライアント アプリケーションのドキュメントを参照してください。

2. 以下に示すように OPC UA 証明書を設定します。

OPC UA サーバーで OPC UA クライアント証明書を設定する

OPC UA サーバーおよびクライアント証明書の設定の次のステップは、OPC UA サーバー ノードにインストールされている OPC UA クライアント証明書を信頼することです。

最も簡単な方法は、OPC UA クライアントをサーバーに接続してみることです。クライアント証明書の信頼はまだ確立されていないので、接続は失敗することが予期されます。

接続が失敗すると、OPC UA サーバーにインストールされていない OPC UA クライアント証明書は OPC UA サーバーの “Rejected Certificate”（拒否された証明書）フォルダに配置されます。

フォルダのデフォルトの場所を以下に示します。

C:\ProgramData\AVEVA\PCS\OPC UA Rejected Client Certificates\certs

注意 : このフォルダにアクセスするには管理者権限が必要です。このフォルダはデフォルトで非表示になっています。

Rejected Certificate フォルダに配置された証明書をインポートする

1. OPC UA クライアント証明書をインポートするには、Rejected Certificate フォルダを参照します。
2. 信頼する OPC UA クライアントの証明書を右クリックして、［証明書のインストール］を選択します。**証明書のインポート ウィザード**が開きます。
3. ウィザードで以下のオプションを選択します。
   - **保存場所 :**［ローカル マシン］を選択し、［**次へ**］を選択します。
   - **証明書ストア :**［信頼できるユーザー］を選択し、［**次へ**］を選択します。
   - 証明書のインポートウィザードを完了します。設定を確認し、［**完了**］を選択します。

ポートを使用する

OPC UA は単一の TCP ポートを介して通信します。これは、OPC UA サーバーの**エンドポイント接続**設定で指定されています。デフォルトでは、ポートは 48032 に設定されています。詳細については、「InTouch OPC UA サーバーを設定する」を参照してください。

RDS セッションを使用して同一のコンピューター上で複数の OPC UA サーバーを実行する場合、後続の各サーバーに対して別々のポート番号を指定する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

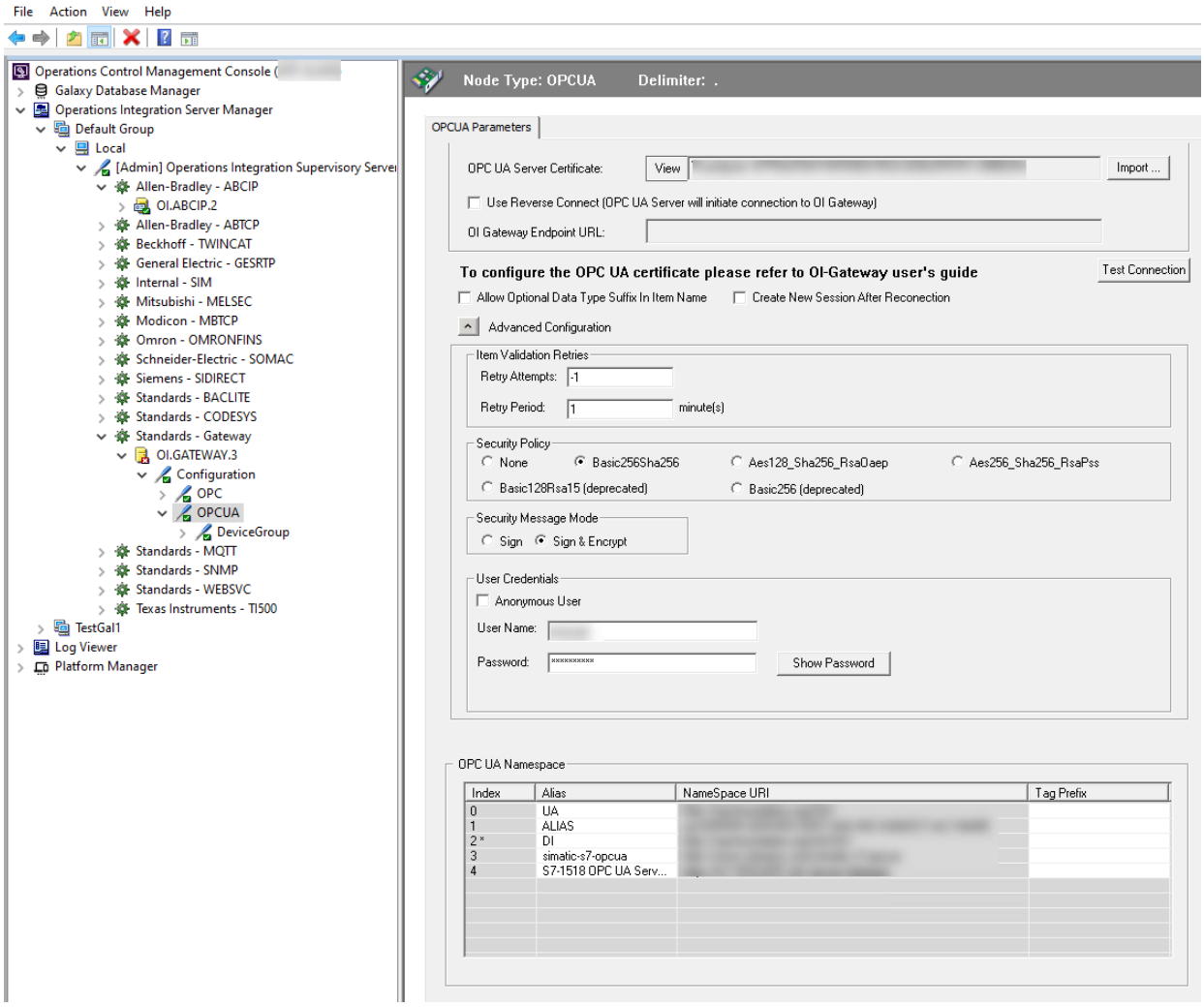
このポートがコンピュータにインストールされているファイアウォール ソフトウェアによってブロックされないことを確認してください。ファイアウォールを介した通信のテストと設定については、「OPC UA サービス用のファイアウォールを設定する」を参照してください。

OI Gateway を使用してクライアント セキュリティ証明書を設定する

Gateway 通信ドライバを使用すると、サーバーおよびクライアント OPC UA ノードでセキュリティ証明書を容易に設定できます。

Gateway 通信ドライバを介してセキュリティ証明書を設定する

1. ランタイム ノードの［スタート］メニューから Operations Control Management Console を開きます（**スタート> AVEVA > Operations Control Management Console**）。



2. コンソール ツリーで Operations Integration Supervisory Servers の下にある **OI.GATEWAY.3** ノードに移動します。
3. OPC UA 接続を作成します。
4. OPC UA サーバーの詳細を設定します。
   - **サーバー ノード：**ランタイム ノードのマシン名を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **OPC UA サーバー：**これは OPC UA サーバー（ランタイム ノード）の URI（Uniform Resource Identifier）です。アドレスは、この時点で探索できないので手動で入力する必要があります。「opc.tcp://<マシン名>:<OPC UA ポート番号>」の形式で入力します。

  InTouch OPC UA サーバーを設定するときに InTouch HMI アプリケーション マネージャで入力した OPC UA ポート番号を使用します。デフォルトのポート番号は 48032 です。

5. 承認および認証の資格情報を入力します。

   OPC UA サーバー ダイアログで設定した承認設定と一致する必要があります。［セキュリティ認証が必要］チェックボックスがオンになっている場合、以下の設定を選択する必要があります。

   - **セキュリティ ポリシー：**Basic256Sha256。
   - **セキュリティ メッセージ モード：** 署名し、暗号化します。
   - 匿名アクセスを許可するには、ユーザー資格情報の下で［**匿名ユーザー**］を選択します。OPC UA 設定時に該当するオプションを選択した場合は、認証済みユーザーのユーザー資格情報を入力することもできます。入力するユーザー資格情報は InTouchHMIOPCUAWriteUsers ユーザー グループのものである必要があります。

6. ［**テスト**］ボタンを選択します。テストは失敗しますが、OPC UA 証明書がダウンロードされます。

   重要! この最初のテストが失敗する理由は、クライアントおよびサーバー アプリケーションの間で証明書が信頼される必要があるからです。この問題は証明書をインストールすることで解決します。

7. 次のセクションに進みます。

証明書が信頼されたら、OPC UA クライアント設定を検証する必要があります。

詳細については、「OPC UA Data Source Object」（OPC UA データ ソース オブジェクトの設定）を参照してください。

OPC UA サーバーおよび OPC UA クライアントの間での証明書の信頼

この OPC UA サーバー サービスのリリースでは、OPC UA サーバーと OPC UA クライアントの間で信頼を作成する操作は手動で行う必要があります。**テスト**操作を実行すると、Gateway 通信ドライバによって独自の証明書が OPC UA サーバー ノードに送信されます。この証明書は信頼できます。以下の手順は、OPC UA サーバー ノードからクライアント証明書を信頼する方法を示します。Gateway 通信ドライバを使用する場合は、「サードパーティ製 OPC UA クライアント アプリケーション用のサーバーおよびクライアント証明書を設定する」を参照してください。

1. **C:\ProgramData\AVEVA\PCS\OPC UA Rejected Client Certificates** フォルダにアクセスします。;

   これは、OPC UA サーバーへの接続を試行する際にクライアントからの証明書がデフォルトで最初に配置される場所です。

   注意：ProgramData フォルダはデフォルトで非表示になっています。非表示の項目を表示するには、Windows エクスプローラーで非表示の項目のオプションを有効にする必要があります。

2. 証明書名（**OIGatewayOPC UA@OPCUA client node{long hex ID}.der** など）を右クリックします。
3. コンテキスト メニューから［**証明書のインストール**］を選択します。**証明書のインポート ウィザード**が開きます。
4. 保存場所に［**ローカル マシン**］を選択し、［**次へ**］を選択します。

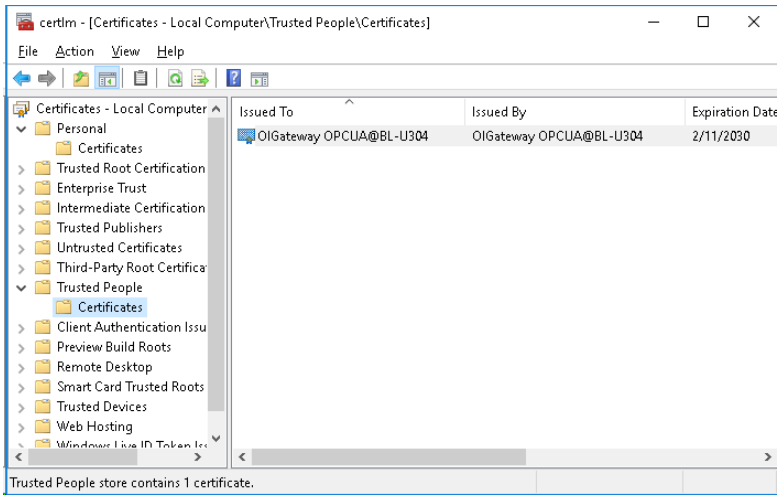
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［**証明書ストアの選択**］リストで証明書ストアとして［**信頼されたユーザー**］を選択します。これは OPC UA 証明書で機能する唯一のオプションです。
6. ウィザードを閉じてインストールを完了します。
7. 最後に、証明書がインストールされたので、**OPC UA Rejected Client Certificates** フォルダから証明書を削除します。

OPC UA 証明書のインストールを検証する

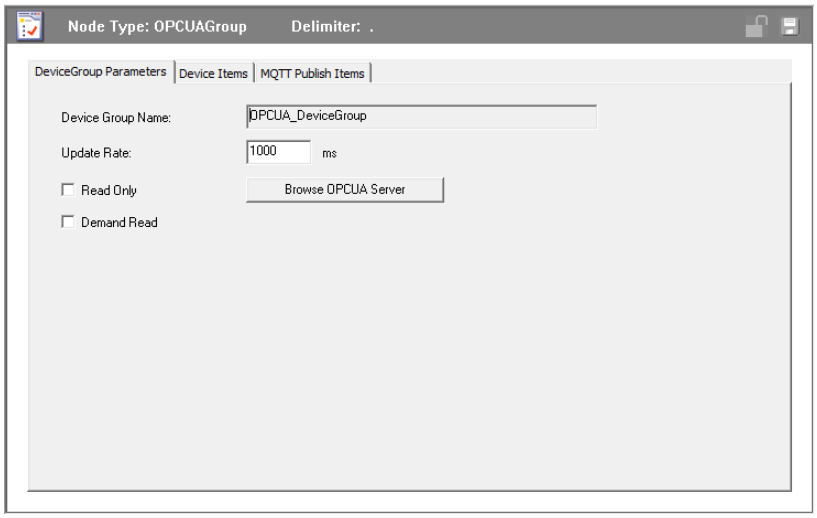
証明書のインストールを検証する

1. Windows の検索ウィンドウに「**証明書**」と入力し、検索結果から［**コンピュータ証明書の管理**］を選択して証明書マネージャを開きます。**証明書マネージャ**が開きます。
2. ［**信頼されたユーザー**］フォルダに移動して、OPC UA 証明書がインストールされていることを確認します。

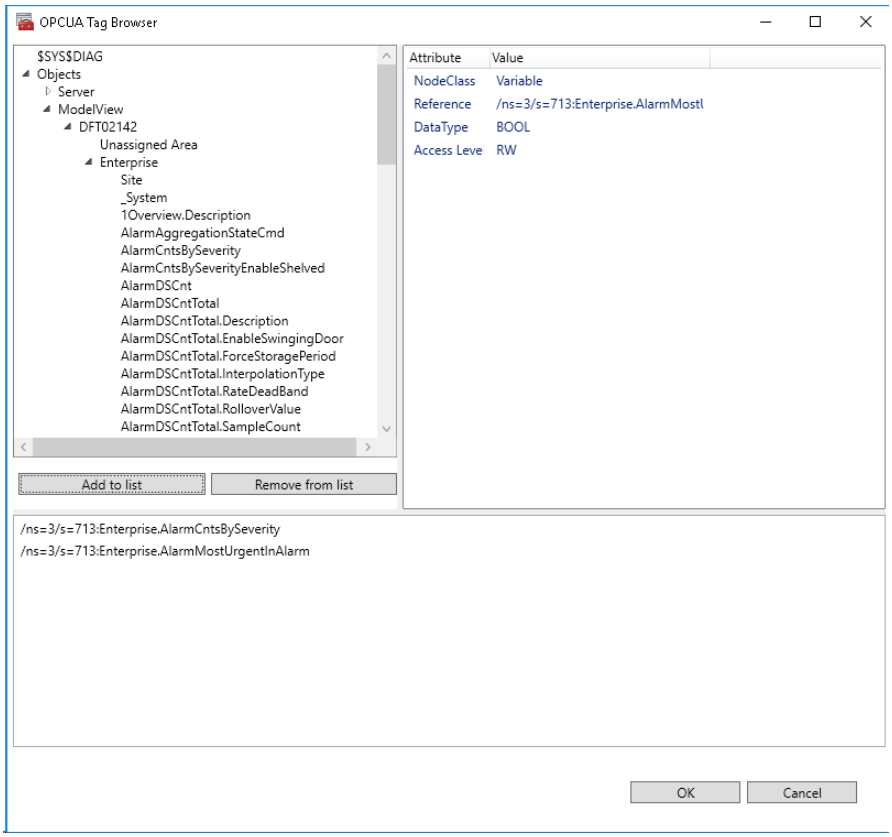


3. OPC UA クライアント ノードから **Operations Control Management Console** を再度開きます。OPC UA パラメータ ウィンドウで［**接続テスト**］ボタンを選択します。ウィンドウの下部にある **OPC UA ネームスペース**のエイリアス リストが自動的に更新され、接続が正常に確立されたことが示されます。
4. OPC UA 接続の下にグループを追加します。入力したグループ名にデバイス グループ名の接頭辞 **OPCUA_** が付けられます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. ［**OPC UA サーバーを参照**］ボタンを選択して OPC UA 階層を参照します。
6. 任意 : OPC UA タグを追加してエイリアス リストを監視すると、ネームスペースを確実に参照できます。



7. **Operations Control Management Console** で OI Gateway の設定に戻り、OPC UA タグが［**デバイス アイテム**］ウィンドウに一覧表示されていることを確認します。
8. デフォルトでは、リストのアイテム名は完全なアイテム参照パスと重複します。必要に応じてアイテムの名前を変更します。

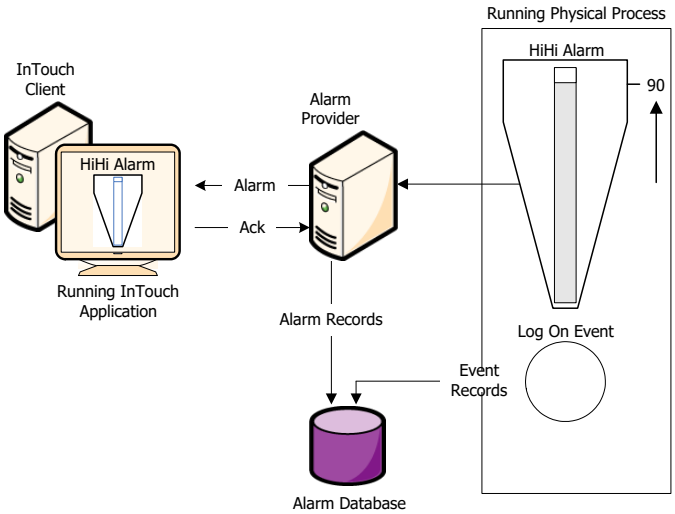
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 4 アラーム

アラームとイベントを生成してオペレータにプロセスの動作状況を通知する InTouch アプリケーションを作成できます。

- 問題を引き起こす可能性のあるプロセス状態に関する警告が、アラームによってランタイム オペレータに警告されます。通常は、プロセス値が定義済みのしきい値を超えた場合にトリガするようにアラームを設定します。オペレータは、アラームを確認する必要があります。
- イベントは、通常のシステム ステータス メッセージを表します。オペレータが InTouch アプリケーションにログオンするなど、システム条件が発生するときが典型的なイベントとなります。オペレータは、イベントを確認する必要はありません。

以下の図は、アプリケーションが実行されている間、InTouch HMI でアラームとイベントが処理される方法を示しています。アラームとイベントのデータは、アラーム データベースに保存されます。



イベント監視用のタグを設定できます。イベント メッセージは、タグの値が変更するたびにアラーム システムにログ記録されます。イベント メッセージには、値がどのように変更されたかに加えて、オペレータ、I/O、スクリプト、またはシステムのいずれによってその変更が開始されたのかという情報が含まれます。

InTouch アラームの概要

アラームは、問題の原因となるプロセス状態が発生したためにオペレータの応答を要求する警告です。アラームは通常、アナログ値がしきい値の上限を超えた場合など、プロセス値がユーザー定義しきい値の範囲を超えた場合に発生します。この状況によってアラームがトリガされ、オペレータに問題の発生が通知されます。オペレータがアラームを確認した後で、InTouch HMI はアラームが確認されたことを認識します。

アラームの原因となっている条件が解決された場合でも、InTouch HMI でアラームの確認を要求するように設定できます。これによって、一時的なアラーム状態の原因となったものの通常状態に戻ったイベントについてオペレータが認識していることを確認できます。

主なアラーム状態を以下の表で説明します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム状態	状況
確認	アラームが確認されました。
ALM	アラームが発生しました。
RTN	タグがアラーム状態から通常状態に戻りました。
ﾗｯﾁ済み	アラームが "UNACK_RTN" 状態から確認されたか、アラームが "ACK" 状態から通常状態に戻りました。

アラーム優先度

アラームに優先度（重要度）を割り当てます。たとえば、ボイラー温度のしきい値には高優先度のアラームが要求され、迅速な対応を必要とします。作業交代を通知するアラームの優先度はかなり低くなります。アラームの優先度は通常、工場アプリケーション、設備の性質、安全性、バックアップ システムの利用状況、考えられる損害費用、またはダウンタイムなど、さまざまな状況によって異なります。

アラームの優先度は、タグ変数を定義する際に割り当てます。優先度は 1 ～ 999 の範囲で設定でき、1 が最も重要度が高くなります。

アラーム優先度の範囲を指定して、アラームの分類を表すことができます。たとえば、プロセスで 4 つのレベルの重要度を必要としている場合、4 つの優先度の範囲を作成できます。

アラームの重要度	優先度の範囲
最高	1 ～ 249
高	250～499
低	500～749
情報	750～999

範囲を設定すると、アラームをフィルタする際に役立ちます。たとえば、重大アラーム以外のすべてのアラームをフィルタするようにアラーム表示を設定できます。すべてアラーム優先度の範囲に基づく、アニメーション リンク、確認スクリプト、およびフィルタされた表示や印刷を作成できます。

アラームのサブステート

マルチ状況アラームには、アラームのサブステートの範囲が含まれています。
たとえば、アナログ アラームには通常、いくつかのしきい値があります。

- Hi と Lo のしきい値で、正常な動作範囲の境界が設定されます。
- HiHi と LoLo のしきい値は、値の正常範囲からの極端な偏差をマーク付けします。

ボイラー温度レベルでは、これらのサブステートのいずれが発生した場合でもアラーム状況となります。ボイラー温度は、全体的なアラーム状況を維持したままで、2 つのサブステート間で切り替わることもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラームの確認

アラームの発生時には、ランタイム オペレータ（またはシステム）はアラームを確認する必要があります。確認とは、単にアラームを認識していることを意味します。修正操作を行うこととは別で、修正操作はすぐに行われるとは限りません。確認は、アラーム条件が通常状態に戻ることとも異なります。外部からの干渉がない場合でも自然にアラーム条件が通常状態に戻ることもあります。

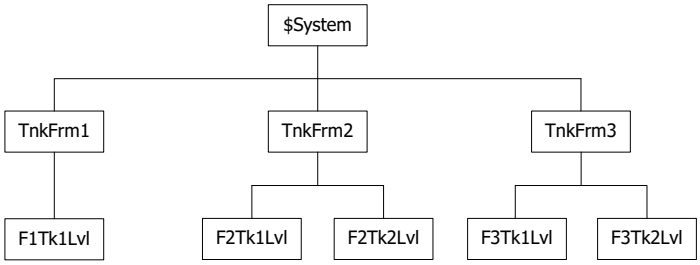
高い優先度または中間の優先度アラームでは、通常は迅速な確認が必要ですが、優先度の非常に低いアラームでは迅速な確認は必要とされません。アラームが生成される原因となった状況が自然に解消される場合（温度が高くなりすぎた後に温度が下がった場合など）であっても、確認されるまでそのアラームは解決されたとは見なされません。

アラームを確認すると、オペレータはアラームを解除できます。ラッチが有効な場合、通常状態に戻った確認済みアラームは、LATCHED アラーム状態になります。オペレータは、LATCHED アラームを解除して、アラーム クライアント コントロール グリッドの現在のモードから LATCHED アラームを削除できますが、アラームはアラーム クライアント コントロールの最近のモードに引き続き表示されます。

アラーム グループ

アラームをグループ化すると、追跡と管理を簡単に行えるようになります。アラーム グループは、工場の異なる領域、設備、オペレータの責任、または製造工程などを論理的に表したものです。

たとえば、以下の図はタンク ファーム アプリケーションの 3 層のアラーム グループ階層を示しています。



アラーム グループは、アラーム表示、アラーム プリンタ、および確認スクリプトでのフィルタに役立ちます。

各タグ変数は 1 つのアラーム グループに関連付けられています。デフォルトでは、タグ変数はメインの $System グループに割り当てられます。$System グループの下に、最大 32 レベルまでの追加のアラーム グループ階層を作成できます。

タグ変数ディクショナリでタグ変数を定義している間、アラーム グループを作成して、タグ変数をそのアラーム グループに関連付けることができます。

アラーム グループおよびグループ変数には SmartSymbol との互換性がありません。アラーム グループへの参照またはグループ変数を SmartSymbol 内で使用することはできません。

InTouch イベントの概要

イベントはシステム内で発生している検出可能な出来事で、必ずしもアラームに関連付けられているとは限りません。アラーム状況が発生するという移行やアラーム状況が解決されるという移行は、イベントの一種です。オペレータの操作、システム設定に対する変更、またはある種のシステム エラーなどもイベントと考えられます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

イベントは状況とは異なります。状況は数分、数時間、数日間、または数週間続くこともあります。イベントは一瞬だけのもので、発生した直後に終了します。アラームは状況であり、アラーム通知はイベントです。

イベントは正常のシステム ステータス メッセージであり、オペレータの応答は必要ありません。

イベント監視を行うようにタグを定義する場合、タグ値が変更されるたびにイベント メッセージがアラーム システムに印刷またはログ記録されるように選択できます。イベント メッセージには、値がどのように変更されたかに加えて、オペレータ、I/O、スクリプト、またはシステムのいずれによってその変更が開始されたのかという情報が含まれます。

イベントのタイプは以下のいずれかとなります。

イベント	状況
OPR	オペレータが値入力を使用してタグ値を修正しました。
LGC	QuickScript 関数でタグ値が修正されました。
DDE	DDE クライアントからタグ値がポークされました。
PROT	産業用グラフィック（カスタム プロパティまたは産業用グラフィックのプッシュボタン）からタグ値の変更が開始されました
SYS	システム イベントが発生しました。
USER	$Operator が変更されました。

SYS および USER イベントは、イベント ログが有効になっているかどうかに関係なく、システムによってすべてのタグに対して生成されます。DDE、OPR、および LGC イベントはタグ値に関係があり、イベント ログが有効になっているタグに対してのみ生成されます。

InTouch アラームのタイプ

InTouch HMI 内では、アラームはその特性に基づいて一般的なカテゴリに分類されます。「クラス」と「タイプ」という一般カテゴリに分類されます。分散アラーム システムでは、すべてのアラームは論理値、値、偏差、変化率、および SPC という 5 つの一般条件に分類されます。

アラーム状況	分散クラス	分散タイプ
論理型	DSC	DSC
値 - LoLo	値	LOLO
値 - Lo	値	Lo
値 - Hi	値	HI
値 - HiHi	値	HiHi
偏差 - 大偏差	DEV	MAJDEV
偏差 - 小偏差	DEV	MINDEV

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム状況	分散クラス	分散タイプ
変化率	ROC	ROC
SPC	SPC	SPC

各 InTouch タグ変数を定義する際、アラーム状況に関連付けます。タグ変数のタイプに基づいて、1 つまたは複数のアラーム クラスやタイプを定義できます。

論理値アラーム

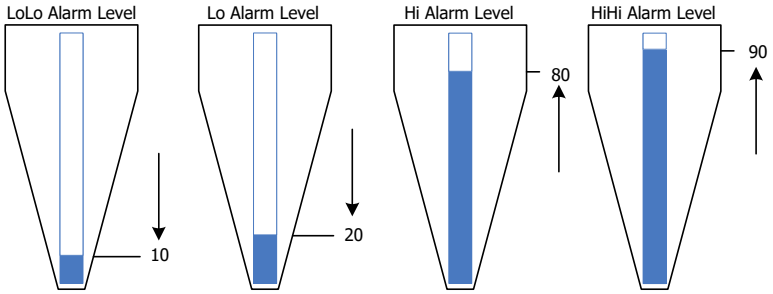
論理値アラームは、2 つのうちいずれかの状況となる論理型タグ変数に対応します。論理型タグ変数を作成する際、そのアラーム状況がタグ変数の true または false のどちらの状況に対応するのかを設定します。

アナログ アラーム

アナログ アラームは、整数値か実数値に関連付けられるアナログ型タグ変数に対応します。アナログ アラームのタイプの中で、値、偏差、および変化率といういくつかのサブタイプがあります。

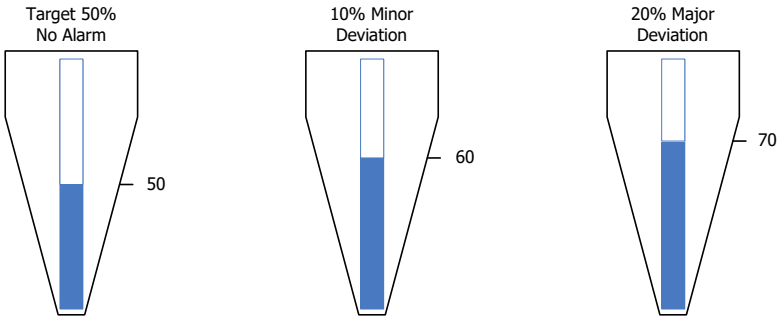
値アラーム

現在のタグ変数値は、事前に定義されている 1 つまたは複数のしきい値と比較されます。値がしきい値を超えると、アラーム状況が宣言されます。「LoLo」、「Lo」、「Hi」、および「HiHi」のしきい値に対して値および優先度を個別に設定し、各しきい値が使用されるどうかを指定できます。



偏差アラーム

現在のタグ変数値が目標値と比較され、差異の絶対値がタグ変数値の範囲のパーセントとして表される 1 つまたは複数のしきい値と比較されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

小偏差と大偏差のしきい値に対する値および優先度を個別に設定して、各しきい値が使用されるかどうかを指定できます。また、タグ変数の範囲のパーセントとして表される偏差デッドバンドの値を設定することもできます。これにより、アラーム状況であると評価される前にタグ変数値が変化する必要のある（合計範囲の）パーセントを制御できます。

たとえば、しきい値は以下のように設定します。

- 0 ～ 100 の範囲
- 基準値 50
- 小偏差 10 パーセント。小偏差のしきい値の範囲を 40 ～ 60 に設定します。
- 大偏差 20 パーセント。大偏差のしきい値の範囲を 30 ～ 70 に設定します。
- デッドバンド 10 パーセント

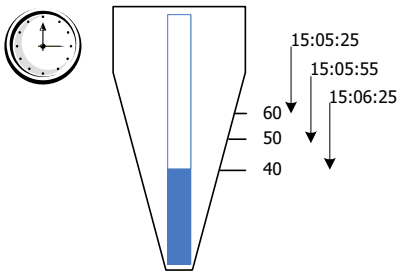
タグ変数の値が 39 の場合、小偏差アラームが発生します。ただし、アラームが評価されて消去されるには、値が少なくとも 50（40 とデッドバンド 10）まで変化する必要があります。

タグ変数の値が 72 の場合、大偏差アラームが発生します。アラームが消去されるには、この値が少なくとも 61 まで低下する必要があります。

変化率アラーム

タグ変数の現在の値と以前の値が、一定の測定期間で比較されます。タグの値の変化率は、以前のタグ値、以前の変更が行われた時間、現在のタグ値、および現在の時間を使用して計算されます。

タグの値が変更されると変化率アラームのテストが行われます。1 つの例外は、WindowViewer 内でアプリケーションの実行が開始するときの初期タグ値です。その場合、初期値は無視され、最初の変更率の比較はタグの値の 2 番目と 3 番目の変更の間で行われます。その後、変化率の測定は、連続するタグ値の変更の間で行われます。



連続するタグの変更の間における値の絶対的な差分が指定された制限を超えた場合、変化率アラームが発生します。変更率の制限は、一定の時間間隔（秒、分、または時間単位）におけるタグの値のパーセンテージとして表されます。変化率（ROC）のしきい値の値および優先度や、そのしきい値が使用されるかどうかも設定できます。

InTouch 分散アラート システム

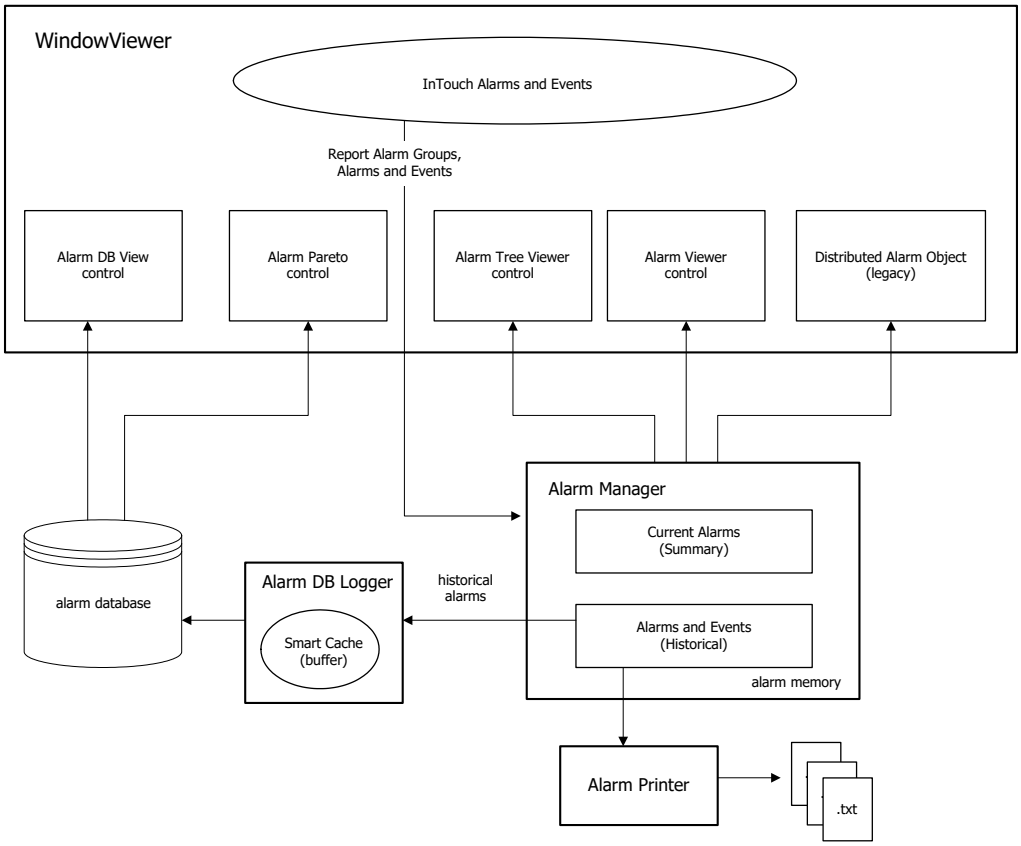
分散アラーム システムは、以下の要素で構成されています。

- アラーム マネージャ。現在アクティブなアラーム（サマリ アラーム）および履歴アラームとイベントを管理します。サマリ アラームおよび履歴アラームは、InTouch の内部アラーム メモリに保持されます。
- Alarm DB Logger。履歴アラームとイベントをアラーム データベースに保存します。アラーム データベースは、SQL Server データベースです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- Alarm Printer。履歴アラームとイベントを印刷します。
- 一組の ActiveX コントロール。ランタイムで内部アラーム メモリまたはアラーム データベースのいずれかからアラームとイベントを取得します。

以下の図は、システムの概要を示しています。



注意：分散アラーム システムは Windows サービスのセットとして実行します。管理者権限で分散アラーム システムを実行する場合にセキュリティの脅威にさらされる可能性を削減するために、ユーザーアカウントの許可は、これらのサービスに対して非対話型に設定されています。

ランタイム オペレータは、分散アラーム システムを使用して、以下の操作を実行できます。

- ローカルの InTouch アプリケーションとその他のネットワーク アプリケーションのアラーム システムによって生成されたアラームやイベントを表示、ログ記録、および印刷します。
- ローカルまたはリモート ネットワーク ノードから、アラームを確認します。
- InTouch アプリケーションでアラーム ActiveX コントロールを使用して、事前設定したアラーム表示を表示します。
- 個別のアラーム コメント フィールドを使用して、アラームに関する詳細なフィードバックを提供します。

アプリケーション開発者は、以下の操作を実行できます。

- ドットフィールドを使用して、アラームを制御します。
- アプリケーションのフルコントロール下で、アラームを直接または間接的に有効にしたり、無効にしたりできるように、アラームを設定します。特定の表示ノードでアラーム情報の表示を防ぐため

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

に、単一のアラーム クラス、タグ変数、またはグループにアラーム抑止を適用できます。システム全体で無効にすると、アラーム動作をソースでブロックできます。

- アラーム情報が履歴にログ記録されるように設定します。Alarm DB Logger は、Windows サービスとして実行することも、必要に応じて手動で起動することもできます。アラーム ロギングでは UCT（GMT）のタイムスタンプを使用しているため、DST やさまざまなタイムゾーンとの互換性を提供しています。
- フェイルオーバー アラーム プロバイダを設定します。プライマリ アラーム プロバイダが失敗した場合でも、分散アラーム システムはバックアップ システムからアラーム情報をシームレスに取得します。プライマリ ノードが再接続されると、分散アラーム システムは回復しているプライマリ システムがライブ状態になる前に、アラーム確認が再同期化されることを確認します。

分散アラーム システムでは、以下の処理が行われます。

- SuiteLink プロトコルを通じてデータを送信して、最小限の CPU リソースやネットワーク リソースを使用します。
- タイムスタンプは、コンシューマがアラームを受け取る時ではなく、アラームが発生した時に設定されます。タイムスタンプにはミリ秒も含まれます。

アラーム プロバイダとコンシューマ

どのノードでも、アラーム プロバイダ（発行側）とアラーム コンシューマ（要求側）を持つことができます。InTouch 分散アラーム システムでは、通信リンクを提供して、ノードやソフトウェア コンポーネント間でアラーム情報を渡すことができます。

アラーム プロバイダ

アラーム プロバイダは、以下の処理を実行します。

- アラーム可能なアイテム、つまりアラーム状況に移行する可能性のあるアイテムを追跡して、これらのアイテムの階層グループ化に関する情報を含む、アイテムのリストを分散アラーム システムに提供します。
- アラーム アイテムのステータスが変更されるとき、分散アラーム システムに通知します。ステータスの変更には、アイテムがアラーム状態に移行したかどうか、アラーム状態から戻ったかどうか、また最新のアラームが確認されたかどうかなどが含まれます。
- アラーム アイテムが無効になったかどうかを追跡します。

InTouch HMI は、QI Analyst、Application Server Galaxy、およびアラーム API ツールキットで作成されたその他のソフトウェアなどの外部アラーム プロバイダをサポートしています。これらのアラーム レコードの日付／時間スタンプはアラーム プロバイダによって提供され、分散アラーム システムによっては生成されません。

アラーム コンシューマ

アラーム コンシューマは、以下の処理を実行します。

- 通知を受け取ることを希望するアラーム可能アイテムを識別するクエリー セットを分散アラーム システムに提供します。クエリーはアラーム コンシューマによって変更または削除されるまでアクティブ状態を保ち、アラーム プロバイダやアラームのグループを指定します。これは、「ワイルドカード」を持つ SQL クエリーとよく似ています。アラーム プロバイダが変更の通知を発行するたびに、分散アラーム システムは登録されているすべてのクエリーとの一致についてアラームを確認し、対応するアラーム コンシューマにアップデートを渡します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アップデートを受け取ると、アイテムのステータスまたは移行に関連する情報を表示するか、ログ記録します。
- アラームを確認します。アラーム コンシューマは、アラームとアラーム プロバイダを識別する確認通知を分散アラーム システムに送信します。この通知はアラーム プロバイダに渡され、そのプロバイダによってアイテムのステータスが「確認」に更新され（適切な場合）、それを分散アラーム システムに通知することによって、その更新がそのアラームに関連を持つすべてのコンシューマに配布されることを確認します。

注意：分散アラーム システムの通信の大部分は、ノード間でのアラーム クエリーとアラーム レコードの送信により構成されています。ノード内では、ネットワーク トラフィックを最小限に抑えるために、内部アラーム メモリによってアラーム クエリーとアラーム レコードが追跡され、バッファされています。

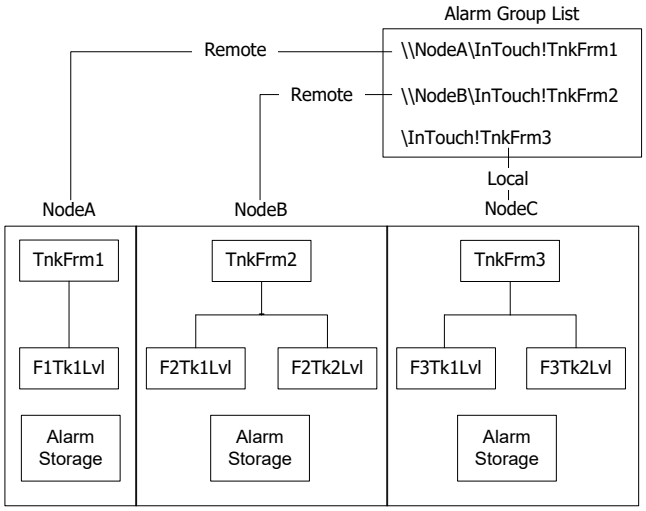
分散アラーム グループ リスト

分散アラーム システムでは、アラーム グループを使用して、ローカル ツリー ビューを構成します。分散アラーム表示では、このツリービューを使用して、アラームをフィルタします。これらのアラーム グループは、ネットワークの複数のノードから表示できます。

分散アラーム システムでは、アラーム グループ リストを使用して、ローカル ノードとリモートノードのアラーム グループを結合します。アラーム グループ リストは、InTouch ノードとこれらの各ノードで定義されるアラーム グループで構成される名前付きのリストです。このリストには、その他のアラーム グループ リストの名前やローカル アラーム グループも含めることができます。Alarm Viewer コントロールなどのアラーム コンシューマは、このリストを使用してアラームのクエリーを実行します。

クエリー エイリアスで、Alarm Viewer コントロールではリストに所属するグループからの組み合わされたアラームを表示できます。アラームはローカルの InTouch ノードまたはネットワークのリモート ノードから確認できます。

以下の図は、3 つのノードからのアラームグループを結合するアラーム グループ リストを示しています。アラーム グループ リストは NodeC でローカルに定義されます。残りのアラーム グループは、リモート ノードからのグループです。



分散アラーム表示では、このリストに所属するすべてのアラーム グループに対するクエリーから結果として生じするアラームが表示されます。たとえば、いくつかの InTouch ノードにあるタンク ファーム アラームをすべて表示する場合は、TankFarmAlarms という名前のリストを作成できます。このリストに、

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タンク ファームの InTouch アプリケーションを実行するすべてのノードからのアラーム グループを追加します。

アラーム グループの作成の詳細については、「アラーム グループの作成」を参照してください。

サマリ アラームと履歴アラーム

サマリ アラームは、現在アクティブなアラームです。履歴アラームは、現在アクティブではなく、通常はアラームデータベースに保存されているアラームです。

たとえば、確認を待機している現在のアラームすべてのサマリを表示して、その他のアラーム情報は緊急性が低いため履歴として記録することができます。

アラームの無効化、抑止、非表示

アラームは「オフ」にするか、アラーム設定を実際に削除することなく無視することができます。アラームは無効にするか、抑止するか、非表示にすることができます。

アラームの無効化と抑止は、アラーム プロバイダで制御されます。非表示は、アラーム コンシューマで制御されます。プロバイダとコンシューマの詳細については、「アラーム プロバイダとコンシューマ」を参照してください。

- **無効化**。アラームは、アラーム プロバイダで無効化をマーク付けするフラグを設定することによって無効にします。どのようなアラーム状況が発生しても、このアイテムはアラーム状況にはなりません。アラームを無効化するために使用するドットフィールドの詳細については、「タグ変数またはアラームグループのアラームの有効化と無効化」を参照してください。

  タグ変数のアラームをすべて一度に無効化または有効化することができます。また、サブステートを持つアラームの場合は、各サブステートを個別に無効にできます。

- **抑止**。アラームは、以下の方法で抑止します。

  a. WindowMaker のアラーム設定に「抑止」タグ変数を追加します。抑止タグ変数は、アラームを「抑止」とマーク付けするためにランタイムで使用されます。

  b. ランタイムで抑止タグ変数を True または False に設定します。抑止タグ変数が False の場合、アラームは通常通りに処理されます。抑止タグ変数が True の場合、アイテムはアラームにはなりません。

  各サブステートは異なるタグ変数によって抑止できます。また、一部のサブステートに抑止タグ変数を割り当てないまま残すこともできます。

  アラームのタグ変数を抑止タグ変数として割り当てると、そのクロスリファレンス使用数は増加します。

- **非表示**。非表示では、アラーム コンシューマでは特定のアラームが無視されます。アラームが例外基準に一致する場合、非表示となります。つまり、特定のアラーム コンシューマでそのアラームが表示、印刷、またはログ記録されなくなります。

  実際のアラーム生成は、非表示によってまったく影響されません。アラーム レコードは、アラーム履歴に引き続きログ記録されます。

アイテムがアラーム状況の間、アラームが無効となるかアクティブに抑止される場合、そのアイテムは強制的に別の（有効な）状態となります。どの状態となるべきかは、どの状態が使用可能であるかどうか、またそれらの状態が無効になっていないかどうかにより異なります。この処理は、アラームのタイプ、しきい値の値などに基づいて、アラーム プロバイダによって処理されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

無効であるかアクティブに抑止されているアラームは確認を待機していません。アラームがサブステートを持つ場合、引き続き使用可能であるサブステートでのみ確認を待機している状態となります。

ターミナル サービス、アラームのサポート

分散アラーム システムを Terminal Services for InTouch と併用することで、異なるターミナル セッションで実行しているアラーム クライアント上でどのアラーム データをどのように表示するかを選択できます。

アラーム プロバイダは、アプリケーションに固有な名前とアプリケーションのインスタンスで識別できます。この情報は、アラーム プロバイダまたはアラーム コンシューマを分散アラーム システムに登録するときに分散アラーム システムに対して利用可能になります。

アラーム プロバイダを実行しているノードは、システム内のコンピュータ ノードを識別する固有な名前で識別されます。ターミナル サービス エディション（TSE）クライアント セッションによって生成されるアラーム レコードには、「<ノード>:<IP アドレス>」の形式の拡張ノード名が含まれます（serverAlarm:192.168.1.23 など）。この情報は、そのコンピュータ ノードでそのインスタンスが起動したときに分散アラーム システムで利用可能になります。

アラーム イベントがログ記録されると、ノードと完全なアラーム プロバイダの名前によってアラームのソースが識別されます。

ターミナル サービス環境でアラームが確認されると、記録されるオペレータ ノードは、対応するオペレータがターミナル サービス セッションを確立しているクライアント マシンの名前になります。ノード名を取得できない場合、ノードの IP アドレスが代わりに使用されます。

ターミナル サーバー クライアント セッションが InTouch アラーム プロバイダとなることはできません。

分散アラーム システム データの保存

分散アラーム システムでは、いくつかの形式でデータが保存されます。

- **内部アラーム メモリ（バッファ）**

  現在のアラームと最近のアラームに関するほとんどの情報は、さまざまなコンピュータ ノードのメモリに保持されます。InTouch では 2 種類のメモリ ロケーションを使用しています。1 つは（現在の）サマリ アラーム用で、もう 1 つは履歴アラームとイベント用です。このモデルは分散アラーム システムでも使用されています。

  サマリ アラーム用のメモリは、現在のアラームをすべて格納できるように、必要に応じて使用可能なメモリサイズの限界まで拡大されます。履歴アラーム用のメモリは、事前定義されたしきい値までしか拡大することはできません。履歴メモリがこのしきい値に達すると、最も古いアラーム レコードが破棄され、新しいレコードが追加されます。マルチノード環境では、複数のノードのアラーム メモリが単一の集合アラーム メモリを構成します。

  制限の設定の詳細については、「アラーム バッファ サイズを設定する」を参照してください。

- **アラーム データベース**

  Alarm DB Logger は、データベースを作成し、アラームが発生したり、アラームがサブステート間の移行を行ったり、アラームが確認されたり、アラームが平常に戻る時間を追跡します。実質的に、これらのレコードはシステムのアラーム履歴を形成します。

分散アラーム システムはクエリーの使用に基づいているため、1 つのコンピュータ ノードを使用してその他の複数のノードに対してアラームをログ記録することがサポートされています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Build

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

章 5 InTouch HMI アプリケーションの管理

InTouch HMI アプリケーションの管理では、以下の作業を行います。

- InTouch アプリケーションを作成または削除する。「InTouch アプリケーションの作成」および「アプリケーション マネージャからの InTouch アプリケーションの削除」を参照してください。
- WindowMaker または WindowViewer のどちらかでアプリケーションを開く。「WindowMaker と WindowViewer でアプリケーションを開く」を参照してください。
- アプリケーションを検索する。「InTouch アプリケーションの検索」を参照してください。
- アプリケーションを別のコンピュータに移動する。「リモート ノードへのアプリケーションのパブリッシュ」を参照してください。
- 複数のコンピュータにアプリケーションを分散する。『*AVEVA™ InTouch HMI アプリケーション配置ガイド*』の「アプリケーションの分散」を参照してください。
- InTouch サービスを管理する。「InTouch サービスの管理の概要」を参照してください。
- タグの定義、ウィンドウ、およびスクリプトをインポートまたはエクスポートする。「InTouch コンポーネントのエクスポートとインポート」を参照してください。
- セキュリティを設定する。『*AVEVA™ InTouch HMI 管理ガイド*』の「InTouch のセキュリティ保護」を参照してください。

アプリケーションは以下の方法で拡張できます。

- テキスト文字列およびアラーム コメントを別の言語に翻訳する。『*AVEVA™ InTouch HMI アプリケーション ランタイム ガイド*』の「ランタイム時の言語の切り替え」を参照してください。
- Application Server とアプリケーションの統合。「マネージド InTouch アプリケーション」および「ランタイム時のアプリケーションの表示の概要」を参照してください。
- 複数のモニタにアプリケーションを表示する。「マルチ モニタ システムの設定の概要」を参照してください。
- タブレット PC でアプリケーションを使用する。「タブレット PC での InTouch の使用の概要」を参照してください。

Windows Vista、Windows 7、および Windows Server 2008 オペレーティング システムの場合、標準のユーザーは InTouch アプリケーション マネージャを使用してアプリケーションを検索し、WindowViewer を開くことができます。標準のユーザーの場合、アプリケーションのプロパティは読み取り専用になります。アプリケーション マネージャからすべての読み取り/書き込みまたは設定操作を行うには、管理特権が必要です。

アプリケーション マネージャを起動します。

アプリケーション マネージャは、［**スタート**］メニューから、またはコンピュータのデスクトップにあるショートカットから起動できます。

アプリケーション マネージャを初めて起動する

1. タスクバーで、［**スタート**］をクリックし、［**プログラム**］をポイントして、［**AVEVA InTouch HMI**］をポイントし、次に［**InTouch HMI アプリケーション マネージャ**］を選択します。

   AVEVA アプリケーション マネージャが開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

コンフィグレータで［**Operations Control 接続エクスペリエンス**］モードが有効化されている場合、アプリケーション マネージャの初回起動時に AVEVA Identity Manager での認証を受けるよう求めるメッセージが表示されます。

2. AVEVA Identity Manager を使用した認証詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

   ウィンドウには、アプリケーション マネージャを使用して作成または検索した、コンピュータ上にある InTouch アプリケーションが表示されます。

アプリケーションを最後に保存した InTouch のバージョンを確認する

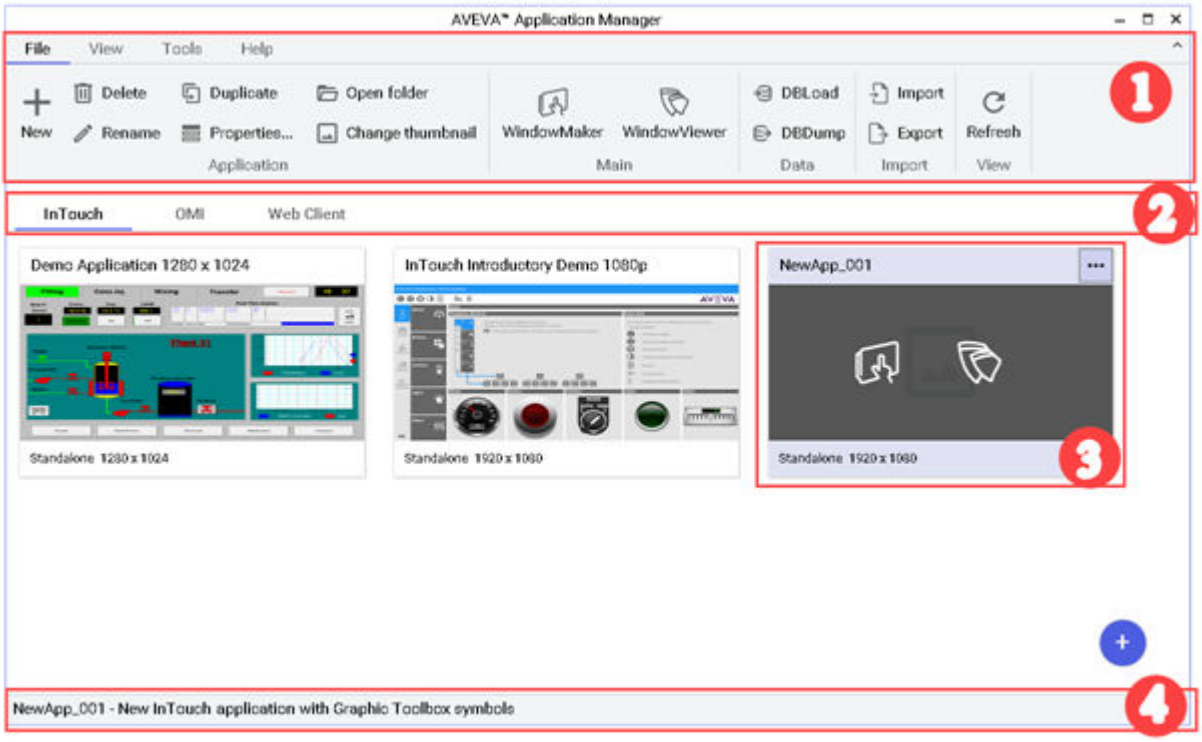
1. アプリケーション マネージャを開きます。
2. ［**表示**］メニューの［**表示**］グループで［**詳細**］をクリックします。

   ［詳細］グリッドビューにアプリケーション リストが表示されます。
3. アプリケーションの［**バージョン**］カラムを確認します。

アプリケーション マネージャの使用

InTouch HMI アプリケーションの作成に使用するアプリケーション マネージャは、アプリケーション管理に関する多くのオプションを提供します。

インターフェイスは、リボンと作業エリア/キャンバスに分割されます。



識別子	要素	説明
1	リボン バー	メニュー項目および各メニュー項目のコマンドが画面上部で機能に基づいてグループ化されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

識別子	要素	説明
2	アプリケーション バー	InTouch、OMI、および Web Client のタブを表示します。 リボン内の各メニュー項目のコマンドは、選択したアプリケーション タイプに応じて異なります。
3	アプリケーション タイル	各アプリケーションを表します
4	ステータス バー	選択した項目の説明メッセージを表示します。

リボンには、以下のメニュー項目が含まれます。

- ［**ファイル**］: **ファイル**メニューはデフォルトのリボン メニューです。このメニューには、5 つの関連コマンド（［**アプリケーション**］、［**メイン**］、［**データ**］、［**インポート**］、および［**表示**］）があります。アプリケーション設定は、作成、複製、削除、WindowMaker または WindowViewer で開く、インポート、エクスポート、およびその他の多くの一般的な機能などのコマンドを使用して変更できます。
- ［**表示**］: **表示**メニューには、キャンバスのリストの表示を変更するオプション（［**一覧表示**］、［**並べて表示**］、［**詳細**］など）があります。
- ［**ツール**］: **ツール**メニューには、重要なアプリケーション マネージャ ツール（［**ノードのプロパティ**］、［**OPC UA 設定**］、［**検索**］、［**Insight Publisher**］）が含まれています。
- ［**ヘルプ**］: **ヘルプ**メニューからは、InTouch HMI ヘルプ ドキュメント、アプリケーション マネージャのバージョン情報、および著作権情報にアクセスできます。

作業エリア/キャンバスはタブで分割され、各タブはアプリケーション タイプ（InTouch、OMI、および Web Client）を表します。

InTouch アプリケーションの作成

アプリケーション マネージャを使用して、新規 InTouch アプリケーションを作成できます。アプリケーション パスは、ネットワーク ドライブ文字、コロン、およびすべての円記号を含め、114 文字を超えないようにしてください。制限を超えると、WindowMaker でアプリケーションを開くことができません。アプリケーション名は 32 文字以内である必要があります。

アプリケーションを作成すると、INTOUCH.ini ファイルが作成されます。INTOUCH.ini ファイルには、アプリケーションのデフォルト設定内容が含まれています。アプリケーションの設定をユーザーが変更すると、新しい設定が INTOUCH.ini ファイルに書き込まれます。

アプリケーションを作成した後、既存の INTOUCH.ini ファイルをアプリケーション フォルダにコピーできます。このようにすると、新規アプリケーションを作成するたびに、カスタマイズされた InTouch パラメータを設定する必要がありません。

注意： スタンドアロン InTouch アプリケーションを作成するには、InTouchDevelopers OS ユーザー グループのメンバーである必要があります。

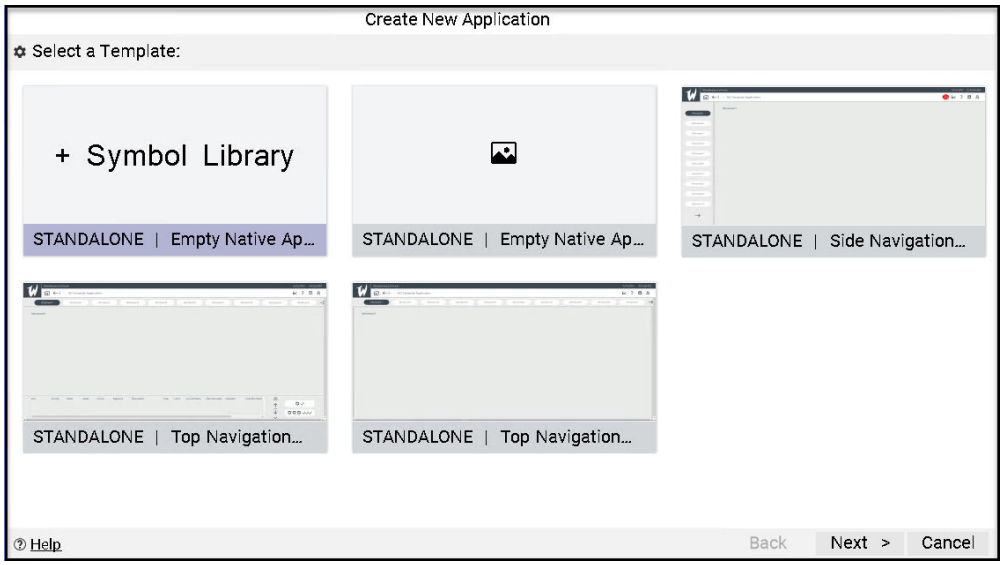
新しいアプリケーションの作成

1. アプリケーション マネージャで別のオプションを使用して、新規アプリケーションを作成できます。
   - ［**ファイル**］タブの［**アプリケーション**］グループで［**新規**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- を右下で選択します。
- Ctrl + N を押します。
- 空白のスペースを右クリックして、［新規...］をクリックします。

［新規アプリケーションの作成 - テンプレートを選択］ページが表示されます。



2. 新しいアプリケーションのテンプレートを選択して［次へ］をクリックします。

   空白のテンプレートには、デフォルトのグラフィック ツールボックス シンボルと SAL シンボルは含まれません。すべてのテンプレートで、InTouch シンボルおよび産業用グラフィックの両方をアプリケーションで使用できます。作成したアプリケーション テンプレートが正しいフォルダで使用可能である場合、そのアプリケーションも表示されます。

   ［新規アプリケーションの作成 - アプリケーションの詳細を入力］ページが表示されます。

3. 以下の詳細を入力します。
   - **タイプ**: 選択したアプリケーションのタイプが表示されます。
   - **アプリケーション名**：アプリケーション名を入力します。
   - **ディレクトリ名**：アプリケーション フォルダ名を入力します。
   - **アプリケーション パス**：デフォルトの場所以外のフォルダを参照するには、省略記号ボタン（[...]）をクリックします。
   - **デフォルト ディレクトリの設定**：トグルを使用して、アプリケーション パスをデフォルト ディレクトリとして設定します。
   - **解像度**：ドロップダウンリストからアプリケーションのターゲット解像度を選択します。

     使用可能な解像度オプションを以下に示します。
     - 画面解像度（デフォルト）
     - 1024 x 768
     - 1280 x 1024

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 1366 X 768
- 1440 X 900

幅と高さを編集するには、［解像度］ドロップダウン リストから［**カスタム解像度**］を選択します。

- **幅**：アプリケーションの幅を指定します。
- **高さ**：アプリケーションの高さを指定します。
- **説明**：255 文字までのオプションの説明を入力します。

を参照してください

4. **[完了]** を選択します。

作成されたアプリケーションは、アプリケーション マネージャのアプリケーションのリストに表示されます。

詳細については、「Creating an Application (アプリケーションの作成)」というタイトルのビデオをご覧ください。

新規 InTouchView アプリケーションの作成

アプリケーションを作成するとき、アプリケーション マネージャのオプションを設定することにより、InTouchView アプリケーションを識別します。InTouch アプリケーションを InTouchView アプリケーションに設定することや、その反対の設定を行うことができます。InTouchView アプリケーションから変換された InTouch アプリケーションを実行するには、InTouch のフル ライセンスが必要です。アプリケーションを InTouchView アプリケーションに変更するした場合、必要な取得ライセンスは、選択したデータソースに応じて異なります。詳細については、「InTouchView アプリケーション」を参照してください。

InTouch アプリケーションから InTouchView アプリケーションへの変更

アプリケーションに必要な機能が Application Server または InTouch タグ　サーバーのデータへの接続だけである場合、InTouch アプリケーションを InTouchView アプリケーションに変更できます。WindowMaker の一部の機能は InTouchView アプリケーションで使用できません。

InTouch アプリケーションから InTouchView アプリケーションへの変更

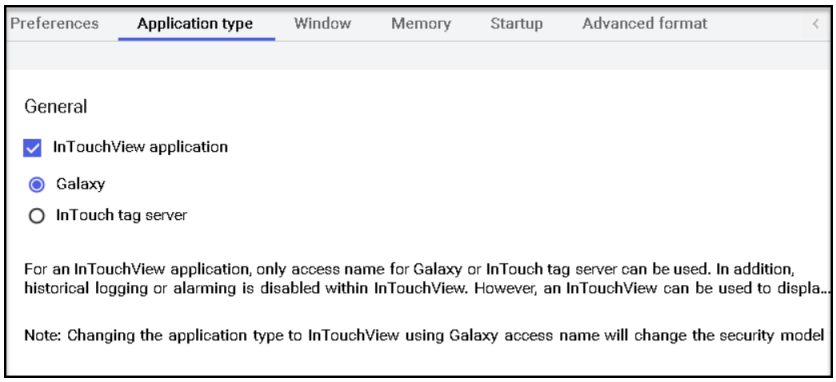
1. WindowMaker で InTouch アプリケーションを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［ファイル］メニューの［設定］をクリックし、［**WindowViewer**］をクリックします。

   WindowViewer の設定画面が表示されます。

3. ［アプリケーション タイプ］タブで［**InTouchView アプリケーション**］チェック ボックスをオンにします。



   - Galaxy のデータに接続するには、［**Galaxy**］を選択します。
   - InTouch タグ サーバーのデータに接続するには、［**InTouch タグ サーバー**］を選択します。

4. ［ホーム］メニューの［タグ］グループで［アクセス名］をクリックします。

   a. ［**Galaxy**］ を選択した場合、 InTouch アプリケーションを InTouchView に変換する前に、Galaxy 以外のすべてのアクセス名を削除する必要があります。削除しない場合、変換の試行中にメッセージが表示されます。

   b. **InTouch Tag Server** を選択した場合、新しいアクセス名を設定できます。
      [追加] を選択し、 [アクセス名] と [ノード名] を指定します。

      アプリケーション名とトピック名はグレーアウトされます。このアプリケーションを InTouchView に変更する前に、Galaxy 以外のアクセス名や InTouch タグ サーバーを参照するアクセス名をすべて削除する必要があります。アプリケーションが*表示*として設定されているアクセス名からの I/O 参照だけがバインドされます。アクセス名 *Galaxy* からの I/O 参照はバインドされません。

   c. **[OK]** を選択します。

      アクセス名の設定の詳細については、「アクセス名の設定」を参照してください。

5. **[閉じる]**を選択します。

6. **[OK]** を選択します。メッセージが表示されたら、［**OK**］を選択します。

WindowMaker と WindowViewer でアプリケーションを開く

新規アプリケーションを WindowViewer で開く前に、WindowMaker で開く必要があります。

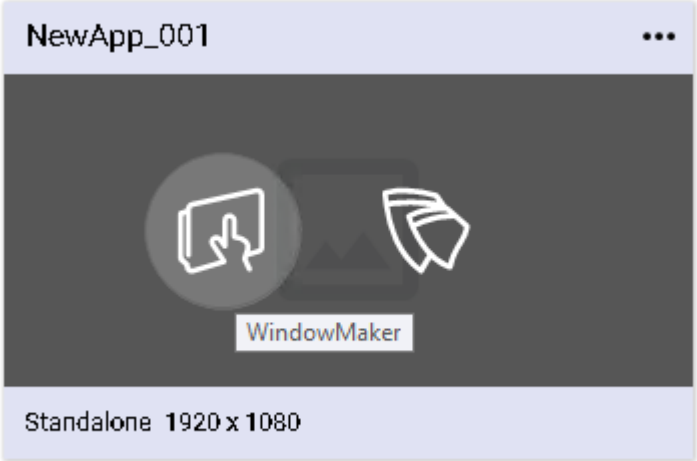
注意：WindowMaker を起動するには、InTouchDevelopers グループのメンバーである必要があります。

WindowMaker でアプリケーションを開く

1. アプリケーション マネージャ ウィンドウでアプリケーションを選択します。
2. ［ファイル］タブの［メイン］グループで［**WindowMaker**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーション タイルの上にカーソルを置いて、［**WindowMaker**］をクリックすることもできます。



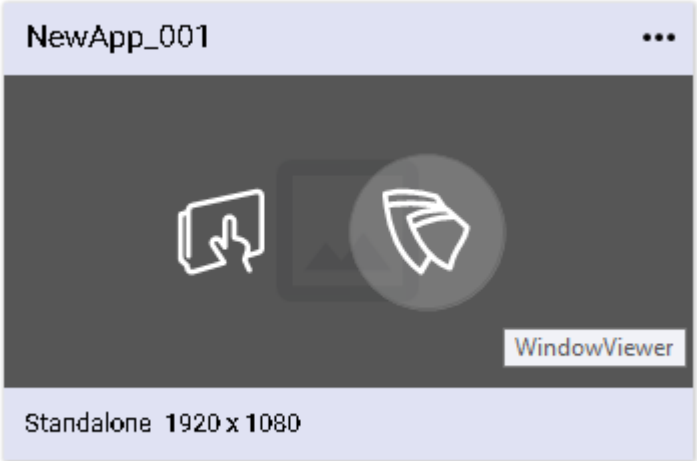
アプリケーションが WindowMaker で開きます。

ヒント： WindowMaker でアプリケーションをダブルクリックして開くこともできます。

WindowViewer でアプリケーションを開く

1. アプリケーション マネージャ ウィンドウでアプリケーションを選択します。
2. ［ファイル］メニューの［メイン］グループで［**WindowViewer**］を選択します。

   あるいは、アプリケーション タイルの上にカーソルを置いて、［**WindowViewer**］をクリックすることもできます。



アプリケーションが WindowMaker で開きます。

アプリケーション マネージャ ウィンドウのカスタマイズ

ツールバーおよびステータス バーを非表示にできます。アプリケーション マネージャ ウィンドウにアプリケーションが一覧表示される方法を変更することもできます。アプリケーションは、詳細、リスト、およびアイコンとして表示できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **詳細** – このビューでは、アプリケーションが表形式で表示され、アプリケーションに関連するすべてのプロパティ（パスや解像度など）が表示されます。
- **リスト** – このビューでは、アプリケーションの名前、タイプ、サムネイル、およびリボンが表示されます。
- **アイコン** – このビューでは、アプリケーションの名前、タイプ、解像度、説明、サムネイル、およびリボンが表示されます。

ツールバーの表示/非表示を切り替える

- ［**表示**］メニューで［**ツールバー**］をクリックして、選択したアプリケーションのステータスの表示と非表示を切り替えます。

ステータス バーの表示/非表示を切り替える

- ［**表示**］メニューで［**ステータス バー**］をクリックして、選択したアプリケーションのステータスの表示と非表示を切り替えます。

アプリケーション リストの表示を変更する

- ［**表示**］メニューで適切なコマンドをクリックするか、ツールバーの対応するボタンをクリックします。

ツールバー オプションの配置を変更する

1. ⋮ アイコンをクリックして、［**カスタマイズ...**］を選択します。
2. ［**アイテム**］タブでオプション（［WindowMaker］など）選択し、［**上へ移動**］または［**下へ移動**］ボタンをクリックして、ツールバーに表示されるオプションの順序を変更します。

ツールバーに表示されるツールを削除する

- ［**表示**］メニューで、目的のオプションのチェック ボックスをクリックします。

1. たとえば、［WindowMaker］オプションのチェック ボックスをオフにして、［**閉じる**］をクリックします。［Window Maker］オプションがツールバーに表示されなくなります。

アプリケーション マネージャにテーマを適用する

- ［**表示**］メニューの［**テーマ**］グループで［**ライトテーマ**］/［**ダークテーマ**］を選択して 2 つのテーマを切り替えます。

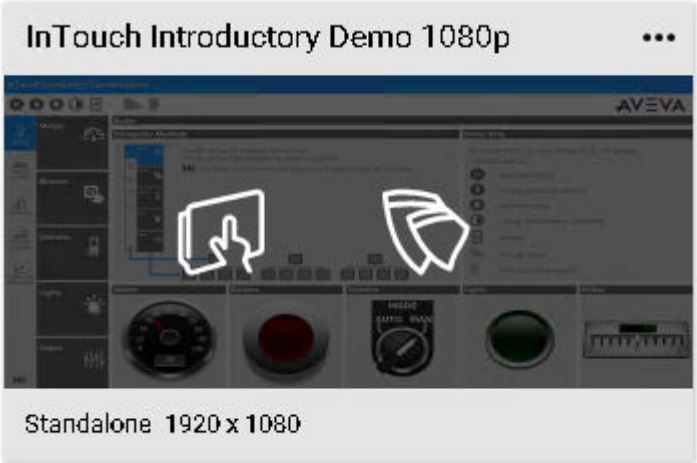
アプリケーション タイルを使用する

アイコンまたは一覧ビューでは、各アプリケーションは、該当する特定のオプションを含むタイルとして表示されます。

WindowMaker または WindowViewer の起動

アプリケーション タイルの上にカーソルを置くと、アプリケーションを WindowMaker または WindowViewer で起動するオプションが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



追加設定

省略記号アイコンをクリックすると、追加設定が表示されます。

- **[WindowMaker]**：WindowMaker を起動する場合に選択します。

  注意：WindowMaker を初めて起動するときに AIM で認証する必要があることがあります。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

- **[WindowViewer]**: WindowViewer を選択して起動します。

  注意：WindowViewer を初めて起動するときに AIM で認証する必要があることがあります。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

- **[アプリケーションフォルダ]**: アプリケーションフォルダを選択して移動します。
- **[プロパティ]**：クリックするとアプリケーションのプロパティ ウィンドウを開きます。
- **[サブネイルの変更]**：アイコンをハイライトして、［参照］ダイアログ ボックスから画像を選択します。画像がアイコンおよび一覧ビューに表示されます。
- **[名前の変更]**：クリックして、アプリケーションの新しい名前を入力します。
- **[DBLoad]**：クリックして DBLoad ユーティリティを起動します。
- **[DBDump]**：クリックして DBDump ユーティリティを起動します。
- **[テンプレートとしてエクスポート]**：InTouch アプリケーションをエクスポートする場合に選択します。「テンプレートとして使用するための InTouch アプリケーションのエクスポート」を参照してください。
- [削除]: アプリケーションを削除する場合に選択します。

InTouch アプリケーションのロックおよびロックを解除する

アプリケーションが別のセッションで編集中の場合、または正しく終了しなかった場合、アプリケーションはロックされることがあります。その場合、アプリケーション マネージャを使用してアプリケーションのロックを解除する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アプリケーションのロック解除

- アプリケーション マネージャを起動します。
- 現在ロックされているアプリケーションを選択します。
- アプリケーション タイルをクリックしてロックを解除し、アプリケーションを再ロードします。

アプリケーションのロックが解除され、アプリケーションを使用できるようになります。詳細については、「アプリケーション編集のロック」を参照してください。

InTouch アプリケーションの変更

アプリケーションの名前を変更したり、アプリケーションのプロパティを変更することができます。

アプリケーション名の変更

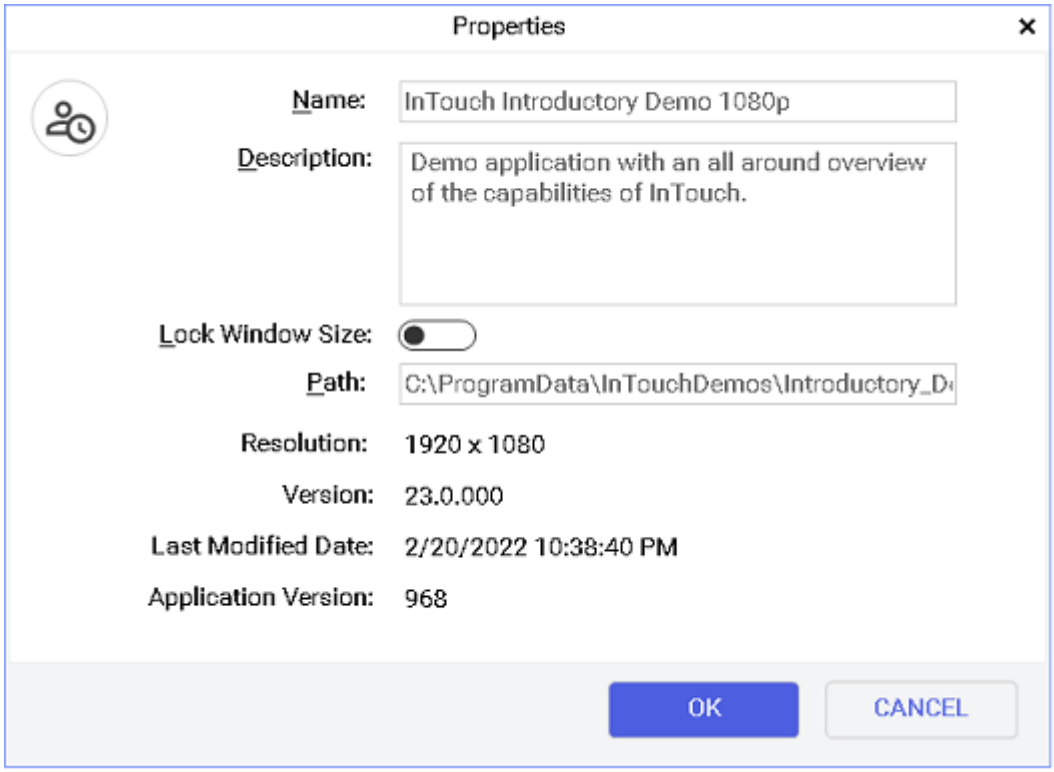
1. 一覧からアプリケーションを選択します。
2. ［ファイル］タブの［アプリケーション］グループで［名前の変更］を選択します。

アプリケーションのプロパティの変更

1. 一覧からアプリケーションを選択します。
2. ［ファイル］タブの［アプリケーション］グループで［プロパティ］を選択します。

   ［プロパティ］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. アプリケーションのプロパティを設定します。
   - ［名前］ボックスで、アプリケーションの新規名を入力します。
   - ［説明］ボックスに、アプリケーションの別の説明を入力します。
4. **[OK]** を選択します。

ターゲット解像度を変更する

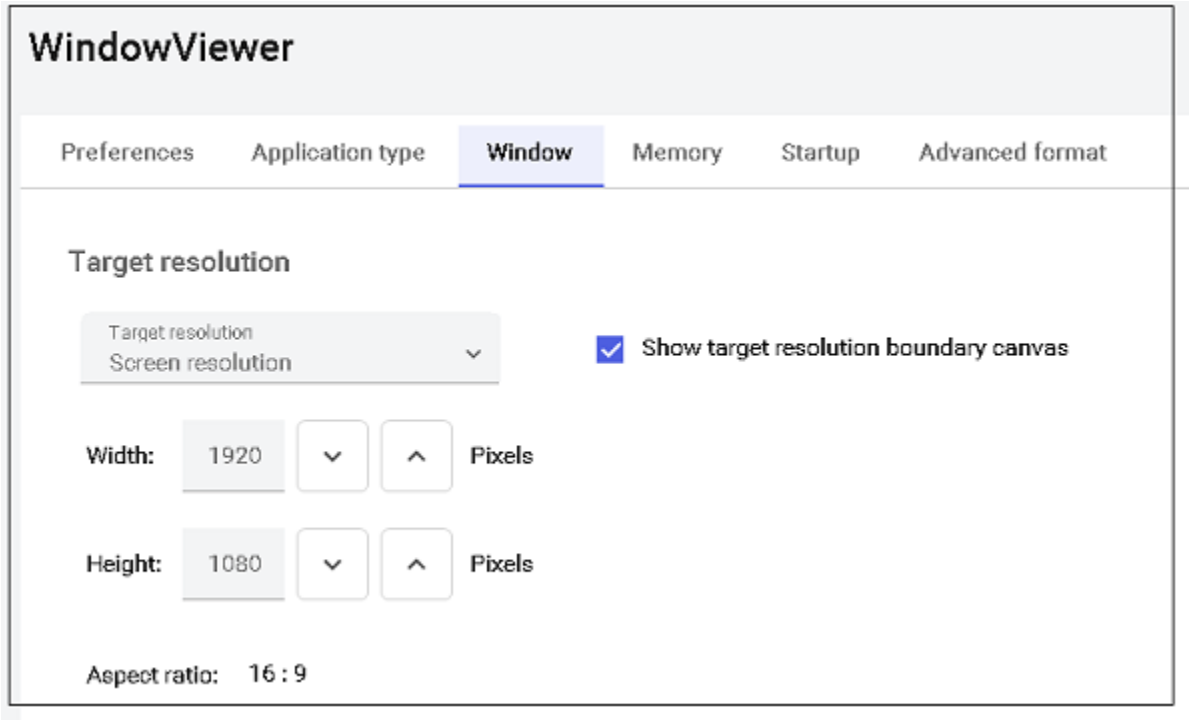
WindowMaker でアプリケーションを編集するとき、指定したターゲット解像度を変更できます。この機能はコマンド ラインでは使用できません。

以下の手順を実行します。

1. WindowMaker を開きます。
2. ［ファイル］メニューの［構成］をハイライトし、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［ウィンドウ］タブを選択します。
4. ［ターゲット解像度］セクションで、必要に応じてターゲット解像度を編集します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. **[OK]** を選択します。

   境界キャンバスが変更され、変更が反映されます。グラフィック、ウィンドウ サイズ、およびウィンドウ コントロールは変更されずに維持されます。

   注意：デフォルトでは［ターゲット解像度の境界キャンバスを表示］が選択されています。

アプリケーション マネージャからの InTouch アプリケーションの削除

アプリケーション マネージャからアプリケーションを削除しても、アプリケーション ファイルはコンピュータ上に残ります。

アプリケーションの削除

1. 一覧からアプリケーションを選択します。
2. ［ファイル］タブの［アプリケーション］グループで［削除］を選択します。
3. メッセージが表示されたら、［はい］をクリックします。
4. アプリケーションは以下の方法でも削除できます。
   - アプリケーションを右クリックして、コンテキスト メニューから［削除］をクリックします。
   - アプリケーション リボンから［削除］アイコンを選択します。
   - アプリケーションを選択して、キーボードの **delete** キーを押します。

   複数のアプリケーションを選択して削除することができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以前のモダン InTouch アプリケーションの削除

古いモダン アプリケーションを含むノードを現在の製品バージョンにアップグレードした場合、古いモダン アプリケーションを削除できます。この操作を行うと、モダン アプリケーションのフォルダとレポジトリが恒久的に削除されます。

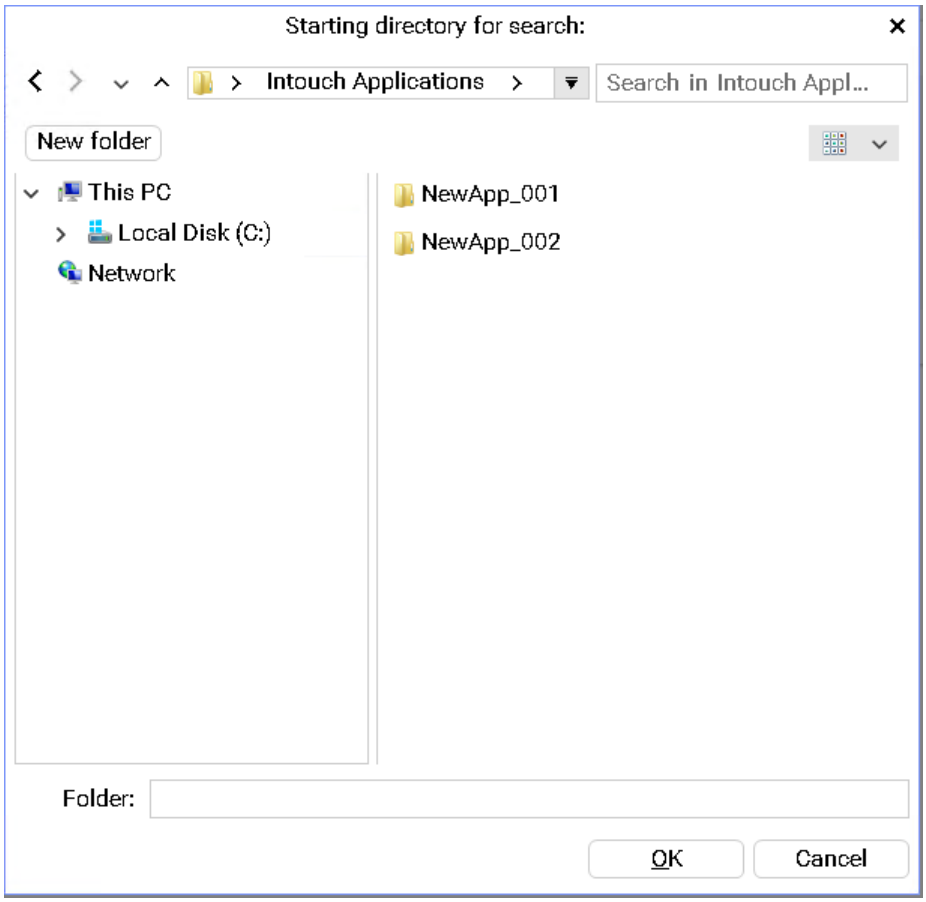
InTouch アプリケーションの検索

既存の InTouch アプリケーションを検索できます。アプリケーション マネージャには、見つかったすべてのアプリケーションが表示されます。

（ネットワーク ドライブ文字、コロン、およびすべての円記号を含めて）フル パスが 114 文字を超える場合、WindowMaker でアプリケーションを開くことができません。文字数の制限をアプリケーションが超えている場合、パスのフォルダの名前を変更するか、またはアプリケーションを別の場所に移動してください。

アプリケーションの検索

1. ［ツール］タブで［検索］をクリックします。［検索を開始するディレクトリ］ダイアログ ボックスが表示されます。



2. アプリケーションを検索するフォルダを選択します。
3. **[OK]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

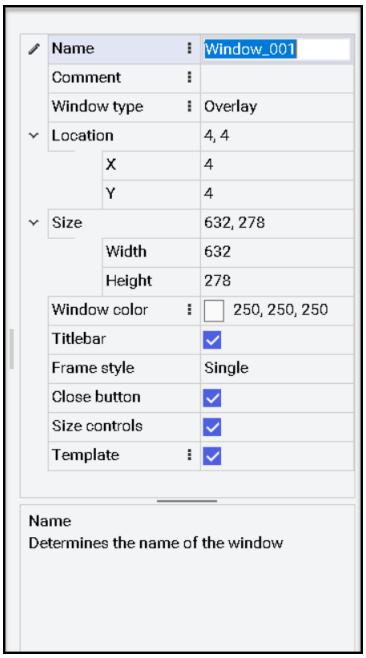
アプリケーション テンプレートでの作業

スタンドアロン アプリケーションから独自のアプリケーション テンプレートを開発できます。アプリケーション テンプレートの開発は、3 ステップのプロセスで構成されます。

1. アプリケーションを作成して、テンプレート ウィンドウに対する適切なアプリケーション ウィンドウを設定します。
2. アプリケーション テンプレート ブラウザのプレビューに使用するアプリケーション サムネイルを作成します。
3. アプリケーションを .aaPKG ファイルとしてエクスポートして、ブラウザでテンプレートとして使用できるようにします。

アプリケーション ウィンドウをテンプレート ウィンドウとして設定する

1. アプリケーションを作成します。
   a. グラフィック、スクリプト、およびウィンドウを使用してアプリケーションを開発します。
2. アプリケーション内の全てのウィンドウに対して、以下の手順を実行します。
   a. 各ウィンドウを右クリックして、［**プロパティ**］を選択します。
   b. ［**ウィンドウのプロパティ**］パネルで、［**テンプレート**］チェックボックスをオンにします。



［**OK**］をクリックすると、各ウィンドウは［ウィンドウ］ペインの［テンプレート ウィンドウ］フォルダに自動的に配置されます。

サムネイルを作成して、テンプレートを作成するアプリケーションに割り当てる必要があります。

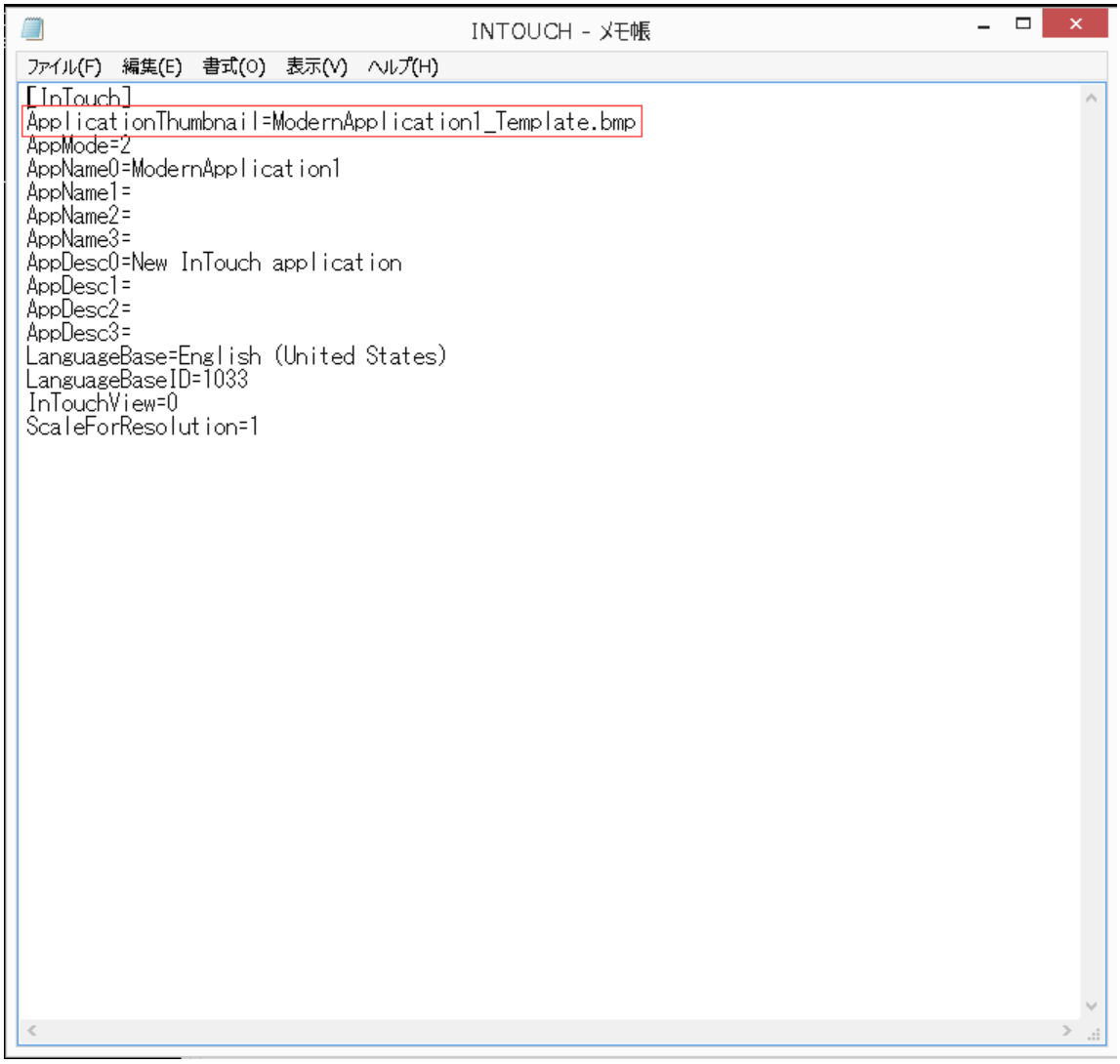
アプリケーション サムネイルを作成して割り当てる

1. 任意のスクリーン キャプチャ プログラムを使用して、設定時または実行時のいずれかのアプリケーションのスクリーン キャプチャを取得します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. 任意のピクチャ ファイル形式 (.bmp や .png ファイルなど) で画像を保存し、アプリケーション フォルダにコピーします。
3. アプリケーション フォルダから、メモ帳などの任意のテキスト エディタを使用して INTOUCH.INI ファイルを開きます。
4. INTOUCH.INI ファイルを編集して、［**ApplicationThumbnail**］フィールドの画像のファイル名を編集します。

注意：［**ApplicationThumbnail**］フィールドでは大文字と小文字が区別され、サムネイル画像の名前と拡張子に正確に一致する必要があります。



5. 変更を保存してファイルを閉じます。

［アプリケーション テンプレート ブラウザ］に表示される .aaPKG ファイルとしてアプリケーションをエクスポートする必要があります。

アプリケーションをエクスポートしてテンプレートを作成する

1. アプリケーション マネージャでアプリケーションをエクスポートして .aaPKG ファイルを作成します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーションのエクスポートの詳細については、「テンプレートとして使用するための InTouch アプリケーションのエクスポート」を参照してください。

2. エクスポートした .aaPKG ファイルを次のディレクトリにコピーします。

```
C:\Program Files (x86)\Wonderware\InTouch\ApplicationTemplates
```

アプリケーションがアプリケーション テンプレート ブラウザ内でテンプレートとして使用できるようになります。

注意： 前の手順で作成したアプリケーション テンプレートのサムネイルは、エクスポートした .aaPKG ファイルから抽出されます。有効な画像が抽出できない場合、[**アプリケーション テンプレート ブラウザ**] に表示されるアプリケーション テンプレートのサムネイルが空白になります。有効な画像ファイル形式を使用してサムネイルを保存したこと、および正しいファイル名が ITOUCH.INI ファイルに入力されていることを確認してください。

InTouch アプリケーション フォルダの表示

WindowMaker を使用すると、InTouch アプリケーション フォルダに配置されていてランタイム時に使用可能なファイルを表示できます。

アプリケーション フォルダを開く

1. WindowMaker で InTouch アプリケーションを開きます。
2. ［**表示**］メニューの［**アプリケーション**］グループで［**フォルダを開く**］を選択します。

   標準の Windows エクスプローラ ウィンドウが開き、InTouch アプリケーション フォルダが表示されます。マネージド アプリケーションの場合、Galaxy リポジトリ ノードのアプリケーション フォルダが表示されます。

テンプレートとして使用するための InTouch アプリケーションのエクスポート

［テンプレートとしてエクスポート］オプションでは、グラフィック関連のすべてのコンテンツ（シンボル、クライアント コントロール、スクリプト ライブラリ、言語、スタイル）を含む .aapkg ファイルをエクスポートして、テンプレートとして保存できます。このテンプレート ファイルを使用して、他のスタンドアロンまたはマネージド アプリケーションを作成できます。

アプリケーションをテンプレートとしてエクスポート

1. アプリケーション マネージャの［ファイル］タブで、［メイン］グループの［**テンプレートとしてエクスポート**］を選択します。

   ［**InTouch アプリケーションのエクスポート**］ダイアログ ボックスが表示されます。
2. .Aapkg ファイルを保存する場所を指定します。

   ［InTouch アプリケーションのエクスポート］進行状況ウィンドウが表示されます。
3. エクスポートが完了したら［**閉じる**］をクリックします。
4. ［**詳細の表示**］をクリックしてエラーをチェックします。

   指定した場所に保存された .aapkg ファイルは新しいアプリケーションを作成するアプリケーション テンプレートとして使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーション マネージャを使用した Web Client 設定の更新

AVEVA アプリケーション マネージャの［Web Client］タブには、Web Client 関連の設定を構成するオプションがあります。

InTouch Web Client の設定を更新

1. アプリケーション マネージャを起動し、［**Web Client**］タブをクリックします。

   ［**Web Client の設定**］画面が表示されます。

2. 以下の設定を構成します。
   - *[現在のアプリケーション]：*Web Client の設定を変更するアプリケーションをリストから選択します。
   - *[グラフィック更新レート（ms）]*：Web ブラウザが Web サーバー にグラフィック データをクエリーする頻度を設定します。デフォルトは 1 秒です。詳細については、『System Platform インストール ガイド』を参照してください。
   - *アラーム更新レート（ms）*：Web ブラウザが Web サーバー にアラーム データをクエリーする頻度を設定します。デフォルトは 1 秒です。詳細については、『System Platform インストール ガイド』を参照してください。
   - *[ヘッダーを表示]*：タイトル バーを表示するには、チェック ボックスをオンにします。
   - *[ナビゲーションバーを有効化]*：このチェック ボックスをオンにするとナビゲーション バーが表示されます。

     ［**ヘッダーを表示**］設定が選択されていない場合、この設定は無効化されます。
   - *[匿名アクセスを許可]*:ユーザーが認証なしで Web Client にアクセスすることを許可する場合は、このチェック ボックスをオンにします。
   - *[作業領域を有効化]*：作業領域を有効にするには、このチェック ボックスをオンにします。
3. ［詳細設定］セクションで［**カスタマイズ**］をクリックして以下の設定を構成します：
   - *[ウェブサイト名]*：標準の URL を置き換える文字列を入力します。
   - *[アプリケーションのタイトル]*:アプリ バーのアプリケーションのタイトルとして表示する文字列を入力します。
   - *[Web サイトのタイトル]*: ブラウザのタイトルバーにタイトルとして表示する文字列を入力します。
   - *[Web サイトのアイコン]*:ブラウザのタイトル バーのアイコンを置換する画像ファイルを入力します。
     i. ［**ファイルを選択...**］をクリックしてアイコン画像を選択します。
     ii. ファイルを選択します。
     iii. **[開く]**を選択します。
4. ［**保存**］をクリックして詳細設定を保存します。
5. 設定を変更したら、**[適用]**を選択 します。
6. Web Client を表示するには、［**起動**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：Web サイトの名前またはアドレスが変更された場合、AVEVA Identity Manager を設定して新しい Web サイトを登録してください。詳細については、「AVEVA Identity Manager での登録, AVEVA Identity Manager での登録」を参照してください。

これらの設定の詳細については、「*Web ブラウザでのアプリケーション グラフィックの表示*」を参照してください。

InTouch OMI ViewApp の起動

アプリケーション マネージャの［OMI］タブを使用して Viewer for InTouch OMI ViewApps を起動します。

1. InTouch OMI アプリを開発および配置します。

   配置した ViewApp は、アプリケーション マネージャの［OMI］タブに表示されます。

2. ViewApp を選択します。
3. OMI の［ファイル］メニューで［**OMI Viewer**］をクリックします。

   Viewer が起動して ViewApp が表示されます。

AVEVA Identity Manager での登録

AVEVA Identity Manager（AIM）を使用すると、デフォルトの Windows OS ベースの認証ではなく、シングル サインオンを使用するように Web Client を設定できます。

注意：Web Client は、AIM 設定を含む ArchestrA ベースのセキュリティのみをサポートします。

Identity Server を設定する

1. System Platform コンフィグレータで［**共通プラットフォーム**］>［**System Management Server**］を設定します。
2. AVEVA アプリケーション マネージャでユーザー資格情報を使用して AIM サーバーを登録します。

［AIM 登録］ダイアログ ボックスを使用して、リバース プロキシ サーバーを設定することもできます。

1. リバース プロキシ サーバーをセットアップします。
2. System Platform コンフィグレータで［**共通プラットフォーム**］>［**System Management Server**］を設定します。
3. ［AIM 登録］ダイアログ ボックスにセキュア ゲートウェイのアドレスを入力します。

リモートまたはローカルの System Management Server を選択できます。System Platform Server の設定の詳細については、『*System Platform インストール ガイド*』を参照してください。

Identity Server に登録する

1. アプリケーション マネージャで［**Web Client**］タブをクリックします。
2. ［**ツール**］、［**ID マネージャ**］の順にクリックします。

   ［**Identity Server 設定**］画面が表示されます。

3. ［**AIM サーバーを認証サーバーとして使用する**］チェック ボックスをクリックして、AVEVA Identity Manager の使用を有効にします。

   ［**Identity Server**］フィールドにコンフィグレータで設定した Identity Server が表示されます。

4. 以下の設定を更新します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. **[ユーザー名]**：Identity Server に接続するユーザー名を入力します。ユーザーは Administrators ユーザー グループに属する必要があります。

b. **[パスワード]**：入力したユーザー名のパスワードを入力します。

5. リバース プロキシのセットアップを完了するには、リバース プロキシまたは DMZ サーバーの URL を［**セキュア ゲートウェイ**］フィールドに入力します。

   これはオプションの設定で、Web Client がリバース プロキシ サーバーの背後でホストされる場合にのみ設定する必要があります。詳細は、「AVEVA DMZ Secure Link」を参照してください。

6. ［**Web サイトへの産業用グラフィックの埋め込みを許可する**］チェック ボックスをクリックして、ランタイム時に HTML iframe 内に Web Client を表示することもできます（オプション）。

7. **[OK]** を選択します。

8. **[適用]** を選択します。

注意：Web サイトの名前またはアドレスが変更された場合、AVEVA Identity Manager を設定して新しい Web サイトを登録する必要があります。

資格情報マネージャの操作

資格情報マネージャでは、資格情報をセキュアな場所に保存できます。保存された資格情報は、WindowMaker、WindowViewer、およびアラーム ユーティリティのその他のコンポーネントへのアクセスの認証に使用できます。

資格情報マネージャを起動するには

1. 管理者としてアプリケーション マネージャを起動します。

2. ［**ツール**］メニューの［**ツール**］グループで［**資格情報マネージャ**］をクリックします。

   資格情報マネージャ画面が表示されます。

   注意：資格情報マネージャにアクセスするには、InTouchDevelopers Windows グループのメンバーである必要があります。

資格情報マネージャは 2 つのセクションで構成されています。

- 名前付き資格情報
- 資格情報の詳細

［**名前付き資格情報**］セクションには、資格情報が以下の列を含むグリッド形式で一覧表示されます。

- 名前：資格情報の名前を示します。
- タイプ：資格情報のタイプ（ユーザー名とパスワード、またはドメインのユーザー名とパスワード）を示します。
- グループ：資格情報が属する Windows ユーザー グループを示します。［グループ］列の省略記号アイコン（...）をクリックすると、1 つまたは複数のグループを選択できます。詳細については、「名前付き資格情報の管理」を参照してください。

［**資格情報の詳細**］のセクションには、選択した資格情報の以下のフィールドが表示されます。

- ドメイン
- ユーザー名

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- Password
- シークレット

名前付き資格情報の管理

資格情報マネージャ画面を使用して資格情報を追加または削除できます。グループを各資格情報に割り当てる必要があります。グループのユーザーは、WindowMaker、WindowViewer、アラーム ユーティリティなどのさまざまな InTouch コンポーネントから資格情報にアクセスできるようになります。

資格情報を追加するには

1. **資格情報マネージャ**の［**名前付き資格情報**］セクションで以下の手順を実行します。
   a. ［**追加（+)**］をクリックします。

      ［名前付き資格情報］グリッドに新しい行が追加されます。

   b. ［**名前**］列に資格情報の名前を入力します。
   c. ［**タイプ**］ 列で、ドロップダウン リストから資格情報のタイプを選択します。
      - **ユーザー名とパスワード：**SQL Server アカウントを資格情報に割り当てることができます。
      - **ドメインのユーザー名とパスワード：**SQL Server アカウントを資格情報に割り当てることができます。
      - **シークレット**:ユーザーが最大 1024 文字のシークレット文字列を指定できるようにします。シークレットに関連付けられたユーザー名はなく、文字列だけです。任意の特殊文字を使用でき、マップ アプリケーション API キーでの使用を意図しています。
   d. ［**グループ**］列の末尾にある省略記号をクリックします。

      ［**グループを選択**］画面が表示されます。

   e. ［**リストから選択**］ドロップダウン リストからドメインを選択します。
   f. ［**使用可能な　OS　グループ**］リストをスクロールして目的のグループを見つけて［+］アイコンをクリックします。

      選択した OS グループが［**選択されたグループ**］セクションに表示されます。

      デフォルトでは、すべての名前付き資格情報は Administrators グループに追加されます。さらに、アプリケーションを実行しているユーザーが属するグループを追加することが推奨されます。そうすることで、WindowViewer などのアプリケーションを管理者以外のモードで実行しているときでも、ユーザーは資格情報にアクセスできるようになります。

      注：グループを削除するには、［**選択されたグループ**］セクションでグループを選択して**[削除(-)**］を選択します。

   g. ［**OK**］をクリックします。
2. ［**詳細**］セクションで以下の手順を実行します。
   a. ［**ドメイン**］フィールドにユーザー ドメインを入力します。

      注：［ドメイン］フィールドは、［ドメインのユーザー名とパスワード］タイプの場合にのみ有効になります。資格情報の［タイプ］で［ユーザー名とパスワード］を選択した場合、［ドメイン］フィールドは無効になります。

   b. ［**ユーザー名**］フィールドに選択した資格情報のユーザー名を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

c. ［**パスワード**］フィールドに選択したユーザー名のパスワードを入力します。

d. ［**パスワードの確認**］フィールドにパスワードを再度入力して確認します。

e. **[シークレット]** フィールドに、シークレットキーとして使用できる値を入力します。

注 : タイプ フィールドを「**シークレット**」として選択すると、「**シークレット**」フィールドが有効になり、残りのフィールドは無効になります。

3. ［**保存**］をクリックします。

資格情報を削除するには

1. グリッドから資格情報を選択します。
2. ［**削除（-)**］をクリックします。

   選択した資格情報が削除されます。

名前付き資格情報のエクスポートとインポート

選択したアプリケーションの資格情報をエクスポートして、アプリケーションをインポートするノードにインポートできます。

資格情報のエクスポート

次のいずれかの方法で資格情報をエクスポートできます。

- 資格情報のみをエクスポートする
- アプリケーションをエクスポートするときに資格情報をエクスポートする

資格情報をエクスポートするとき、セキュリティ キーを設定する必要があります。セキュリティ キーは資格情報を暗号化し、資格情報をインポートするときの認証の際に必要になります。

資格情報マネージャを使用して資格情報をエクスポートするには

1. **資格情報マネージャ**画面で［**エクスポート**］をクリックします。
2. ［**資格情報をエクスポート**］ウィンドウが表示されます。
3. エクスポートする資格情報をグリッドから選択します。
4. 次のセクションで**セキュリティ キー**を設定します。

   注 : セキュリティ キーは 12 文字以上の長さで、少なくとも 1 つの大文字、小文字、数字、および特殊文字を含む必要があります。

5. ［**エクスポート**］をクリックします。

   ［**資格情報の詳細をエクスポート**］ファイル ブラウザが表示されます。

6. 資格情報を保存するフォルダを選択して［**保存**］をクリックします。

   資格情報は .bin ファイル形式でエクスポートされます。

アプリケーションをエクスポートするときに資格情報をエクスポートするには

1. アプリケーション マネージャで、エクスポートするアプリケーションを右クリックして［**テンプレートとしてエクスポートする**］をクリックします。
2. ［**資格情報をエクスポート**］チェック ボックスをオンにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーションが資格情報リストと共にエクスポートされます。

資格情報のインポート

アプリケーションをインポートした特定のノードで使用する資格情報をインポートできます。

資格情報をインポートするには

1. **資格情報マネージャ**画面で［**インポート**］をクリックします。
2. ファイル ブラウザを使用して .bin ファイルを選択し、［**開く**］をクリックします。
3. セキュリティ キーを入力します。

   ヒント：アプリケーションをエクスポートしたときに使用したものと同じセキュリティ キーを使用する必要があります。

   資格情報が資格情報マネージャにインポートされます。

NAD の資格情報をインポートするには

このリリースの時点では、NAD リポジトリの名前付き資格情報は NAD クライアントに自動的に反映されません。したがって、NAD で実行するアプリケーションを（［**アプリケーションの検索**］アクションを使用して）選択しても名前付き資格情報はインポートされません。

NAD リポジトリの名前付き資格情報をインポートするには、各 NAD クライアントのアプリケーション マネージャから名前付き資格情報をインポートします。

資格情報のインポート シナリオ

シナリオ 1:インポートした資格情報とマシンに存在する資格情報が異なる。

資格情報が資格情報マネージャにインポートされ、資格情報グリッドに新しいエントリが追加されます。

シナリオ 2:インポートした資格情報と既存の資格情報が同じ。

資格情報が資格情報マネージャにインポートされ、資格情報グリッドに新しいエントリが追加されます。新しくインポートされた資格情報の名前が変更されます。

シナリオ 3:インポートした資格情報の名前と既存の資格情報の名前が同じで、詳細が異なる。

資格情報が資格情報マネージャにインポートされ、資格情報グリッドに新しいエントリが追加されます。既存の資格情報の名前に増分値が付けられた名前で新しくインポートした資格情報の名前が変更されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

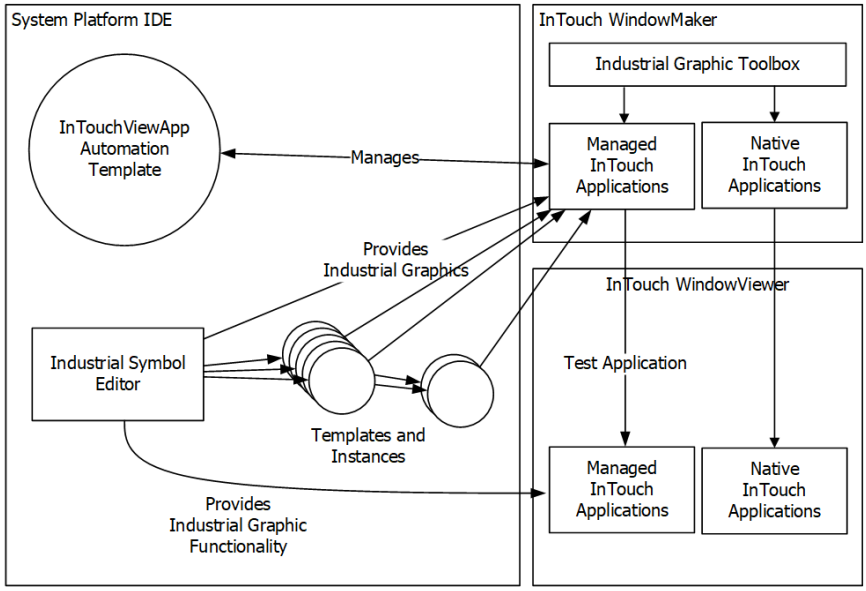
章 6 IDE を使用した InTouch アプリケーションの管理

System Platform IDE を使用して、以下を実行できます。

- 新しいマネージド InTouch アプリケーションを作成する。
- マネージド InTouch アプリケーションとして使用するために、既存の InTouch アプリケーションをインポートする。
- WindowMaker を起動する。
- WindowMaker で行った変更をマネージド InTouch アプリケーションに送信する。
- マネージド InTouch アプリケーションをその InTouchViewApp オブジェクトと共にエクスポートおよびインポートする。
- マネージド InTouch アプリケーションをパブリッシュする。
- マネージド InTouch アプリケーションを削除する。
- マネージド InTouch アプリケーションで使用されているタグ データをインポートおよびエクスポートする。
- .csv ファイルへのタグ データのエクスポートおよび .csv ファイルからのタグ データのインポートを行う。
- マネージド InTouch アプリケーションの言語を切り替える。
- マネージド InTouch アプリケーションにファイルを追加する。
- すべての Galaxy グラフィックを InTouchViewApp に関連付ける。
- アプリケーション テンプレートを使用してマネージド InTouch アプリケーションを作成する。

System Platform IDE は InTouch HMI アプリケーション マネージャから起動できます。

以下の図は、アプリケーションがどのようにインポート、エクスポート、管理、およびパブリッシュされるのかを示しています。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マネージド InTouch アプリケーションの作成

InTouchViewApp オブジェクトを作成および設定することにより、マネージド InTouch アプリケーションを作成します。

また、アプリケーション バージョン、解像度、および説明の情報を InTouchViewApp オブジェクトから直接表示することもできます。

次の InTouch アプリケーション ディレクトリが共有ディレクトリとして作成されます。

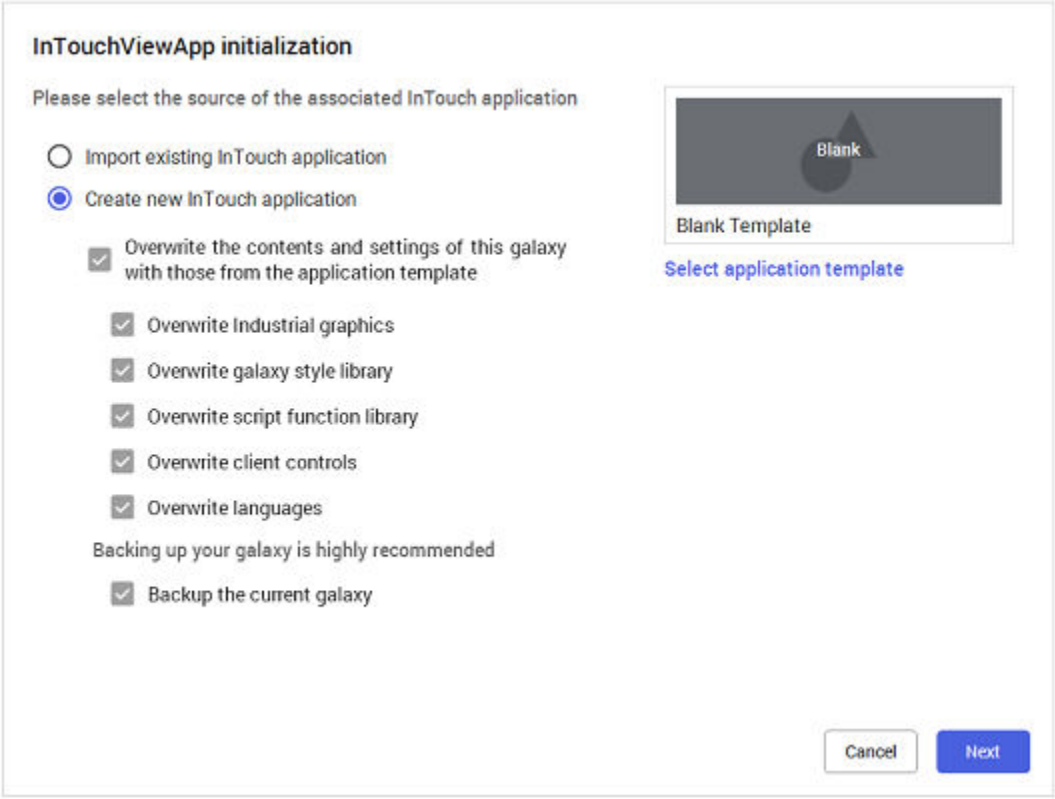
\\GRNodeName\GalaxyName-$InTouchViewAppObjectName

このディレクトリは、InTouch HMI アプリケーション マネージャではなく IDE によって管理されます。

マネージド InTouch アプリケーションの作成

1. System Platform IDE を開きます。
2. ［**テンプレート ツールボックス**］で、［**システム**］ツールセットを展開します。
3. **$InTouchViewApp** ベース テンプレートからテンプレートを派生させます。以下の手順を実行します。
   a. **$InTouchViewApp** ベース テンプレートを右クリックし、［**新規**］をポイントし、次に［**派生テンプレート**］をクリックします。デフォルトの名前で新しい派生テンプレートが表示されます。
   b. 必要に応じて、新しいテンプレートの名前を変更します。
4. 派生テンプレートをダブルクリックします。

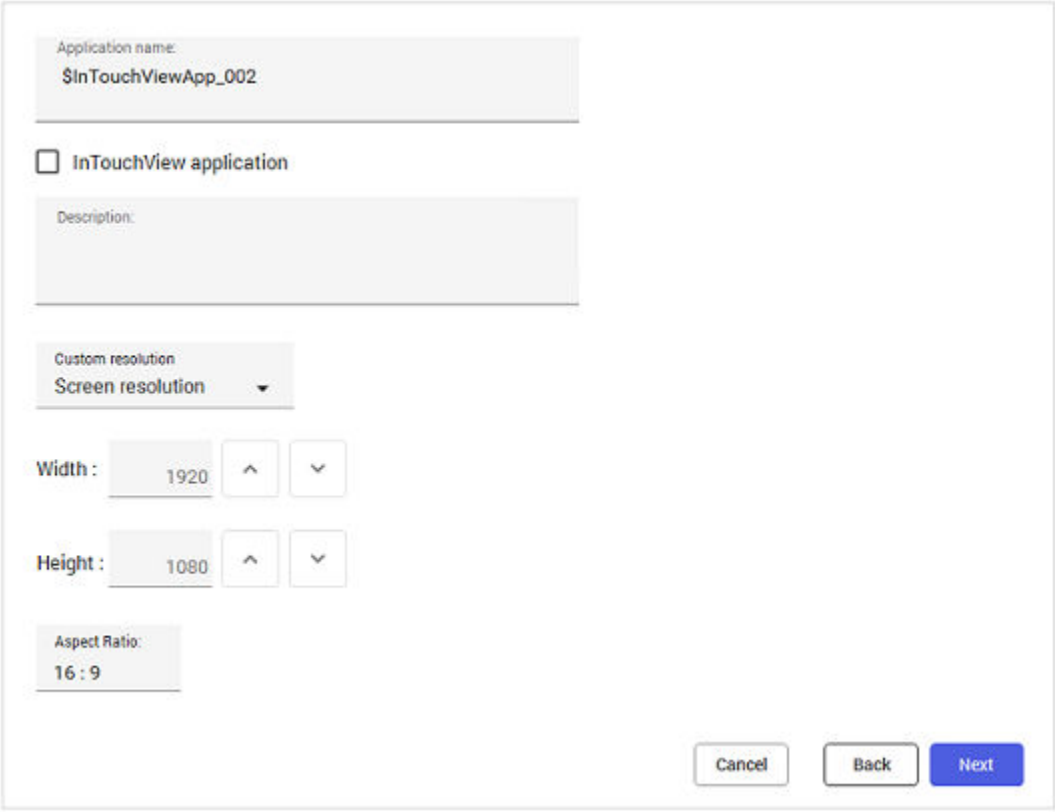
   ［**InTouchViewApp Initialization**］ダイアログ ボックスが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［**新規 InTouch アプリケーションの作成**］を選択し、［**次へ**］を選択します。

   次のパネルが表示されます。



6. InTouch アプリケーションの名前および説明を入力します。
7. 外部データ ソースとして Application Server 参照のみを使用する InTouch アプリケーションを作成するには、［**InTouchView アプリケーション**］を選択します。
8. アプリケーションのターゲット解像度がデフォルトの画面解像度オプションと異なる場合、ターゲット解像度を選択します。このフィールドのオプションを以下に示します。
   a. ［**ターゲット解像度の選択**］ドロップダウン メニューを選択して、事前定義されたターゲット解像度のリストを表示します。
   b. ［**ターゲット解像度の選択**］ドロップダウン メニューをハイライトして、［**カスタム**］を選択します。ピクセルの幅と高さのフィールドが編集可能になります。境界の制限は 150x150 および 10000x10000 です。
9. **[次へ]** を選択します。

   WindowMaker が起動します。

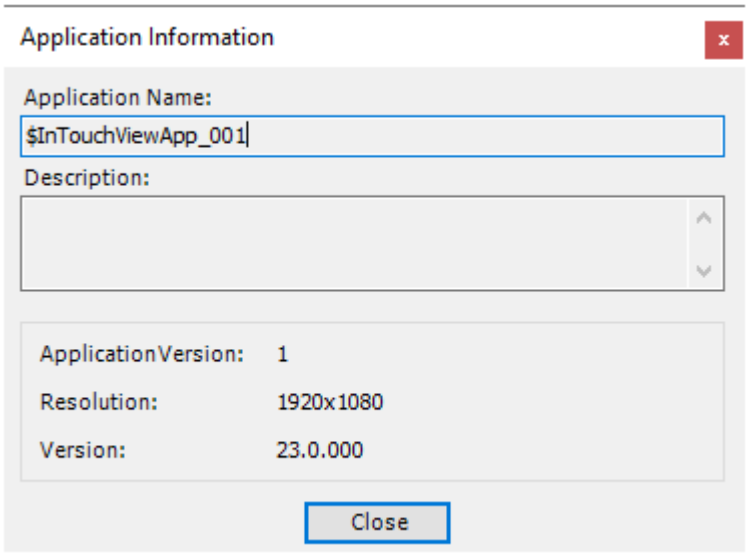
注意 : 画面解像度と異なるターゲット解像度を選択した場合、WindowMaker でアプリケーションを編集する際に画面解像度を変換するよう求めるプロントは表示されません。

アプリケーション バージョン、解像度、および説明を表示する

1. テンプレート ツールボックスを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. InTouchViewApp テンプレートを右クリックし、次に［**アプリケーション情報**］をクリックします。［**アプリケーション情報**］ダイアログ ボックスが表示されます。



ターゲット解像度を変更する

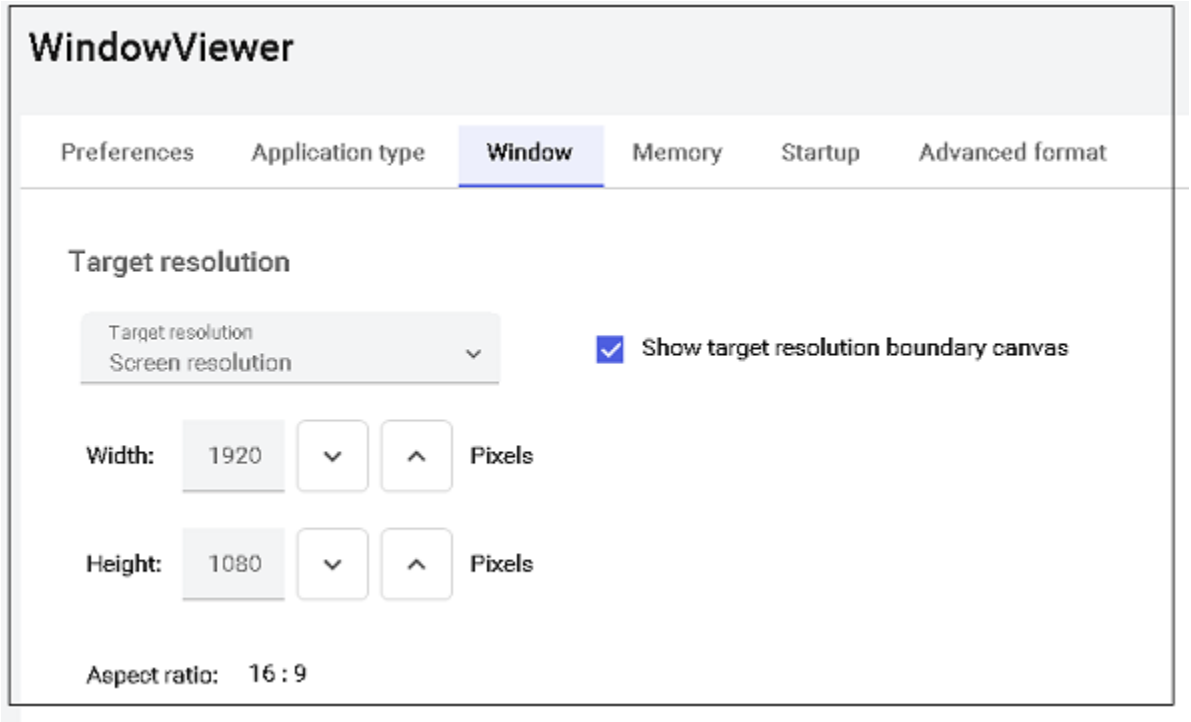
WindowMaker でアプリケーションを編集するとき、指定したターゲット解像度を変更できます。この機能はコマンド ラインでは使用できません。

以下の手順を実行します。

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］をクリックし、［**WindowViewer**］をクリックします。

   WindowViewer の設定画面が表示されます。
3. ［**ウィンドウ**］タブを選択します。
4. ［**ターゲット解像度**］セクションで、必要に応じてターゲット解像度を編集します

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. **[OK]** を選択します。

   境界キャンバスが更新され、変更が反映されます。ウィンドウ、ウィンドウ サイズ、およびウィンドウ コントロールは変更されずに維持されます。

   注意：デフォルトでは［**ターゲット解像度の境界キャンバスを表示**］が選択されています。

アプリケーション テンプレートからのマネージド アプリケーションの作成

アプリケーション テンプレートに基づいてマネージド InTouch アプリケーションを作成すると、デザイン時間を大幅に削減し、新規アプリケーションを既存の標準に基づいて作成することができます。

作成するマネージド InTouch アプリケーションのベースとなるテンプレートを選択するには、アプリケーション テンプレート ブラウザを使用します。アプリケーション テンプレート ブラウザで使用できるテンプレートは、エクスポートされた .aaPKG ファイルです。このファイルは、InTouch インストール ディレクトリの Application Templates フォルダから読み込まれます。以下に例を示します。

```
C:\Program Files(x86)\Wonderware\InTouch\ApplicationTemplates
```

［**アプリケーション テンプレート ブラウザ**］のフォルダ階層には、このディレクトリの階層が反映されます。

アプリケーション テンプレートに基づいてマネージド アプリケーションを作成する

1. IDE を開き、$InTouchViewApp オブジェクトから派生テンプレートを作成します。

   派生テンプレートの名前は変更できます（オプション）。

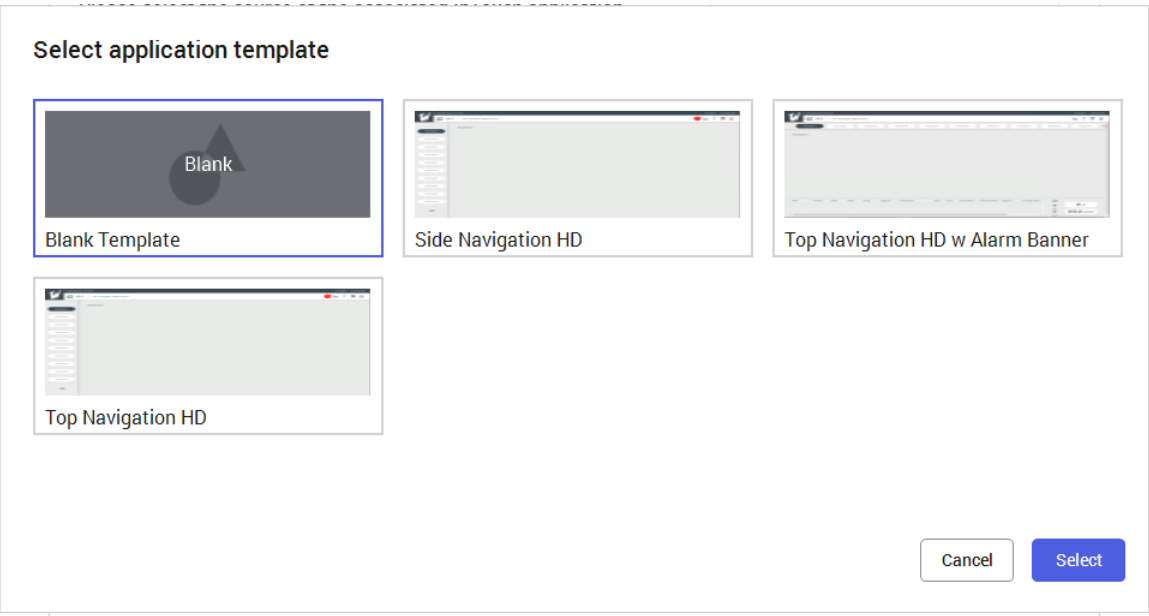
2. 派生テンプレートをダブルクリックします。

   ［**InTouchViewApp 初期化**］ダイアログ ボックスが表示されます。デフォルトでは、［**新規 InTouch アプリケーションの作成**］オプションが選択されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. **[アプリケーション テンプレート]**を選択します。

   ［**アプリケーション テンプレートを選択**］ダイアログ が表示されます。



   ブラウザに表示される各サムネイルでは、縦横比が維持されます。

4. 目的のアプリケーション テンプレートを選択して、［**OK**］をクリックします。

   選択したテンプレートが［**InTouchViewApp 初期化**］ダイアログ ボックスに表示されます。

5. 目的の Galaxy 設定の上書きオプションを注意深くレビューして選択します。

   注意：デフォルトでは、［**この Galaxy の内容と設定をアプリケーション テンプレートの内容と設定で上書き**］オプションが選択されています。上書きされる特定の Galaxy 設定は、子オプションに分割されます。これらのオプションは、この時点で選択を解除できます。

   上書きオプションを以下に示します。

   - **産業用グラフィックの上書き**

     このオプションを選択すると、競合が発生した場合、Galaxy シンボルがアプリケーション テンプレート シンボルで置換されます。

   - **Galaxy スタイル ライブラリの上書き**

     このオプションを選択すると、アプリケーション テンプレート スタイル ライブラリによって Galaxy スタイル ライブラリ全体が置換されます。選択を解除すると、この手順がスキップされます。スタイル ライブラリはマージされません。

   - **スクリプト関数ライブラリの上書き**

     このオプションを選択すると、競合が発生した場合、Galaxy スクリプト関数がアプリケーション テンプレート スクリプト ライブラリによって置換されます。

   - **クライアント コントロールの上書き**

     このオプションを選択すると、競合が発生した場合、Galaxy クライアント コントロールがアプリケーション テンプレート クライアント コントロールによって置換されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **言語の上書き**

  このオプションを選択すると、Galaxy で定義されている言語設定がアプリケーション テンプレートで定義されている言語設定によって置換されます。

  選択を解除すると、競合が発生した場合、Galaxy の言語設定が維持されます。選択を解除したときに競合が発生しなかった場合、アプリケーション テンプレートの言語設定が Galaxy の設定に追加されます。

6. **[次へ]** を選択します。

   ［**現在の Galaxy のバックアップ**］オプションを選択した場合、バックアップ .cab ファイルの場所およびファイル名を尋ねるプロンプトが表示されます。

   重要：Galaxy をバックアップすることを強く推奨します。

7. アプリケーション名を確認して、［**InTouchView アプリケーション**］を選択します。

   ターゲット解像度は、デフォルトでテンプレートの解像度に設定されます。［**ピクセル**］フィールドは無効化されています。この設定はアプリケーション開発中に調整できます。

8. **[次へ]** を選択します。

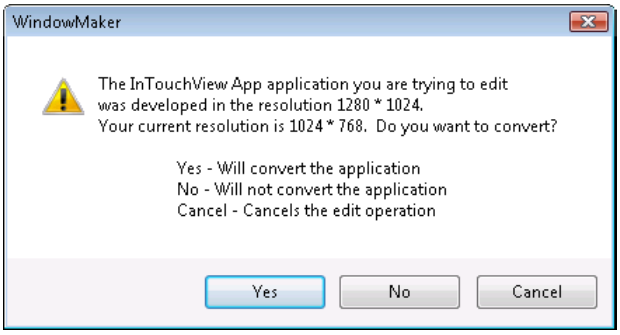
   初期化が完了すると、アプリケーションを配置できます。

IDE からの WindowMaker の起動

IDE 内から WindowMaker を起動することにより、マネージド InTouch アプリケーションを編集できます。

InTouchViewApp のテンプレートまたはインスタンスから WindowMaker を起動できます。

現在のシステムの解像度とは異なる画面解像度で作成されたマネージド InTouch アプリケーションを開くとき、メッセージが表示されます。



- InTouch アプリケーションを現在のシステムの解像度に変換して開くには、［**はい**］をクリックします。
- 元の解像度で InTouch アプリケーションを開いて編集するには、［**いいえ**］をクリックします。

InTouchViewApp テンプレートから WindowMaker を起動する

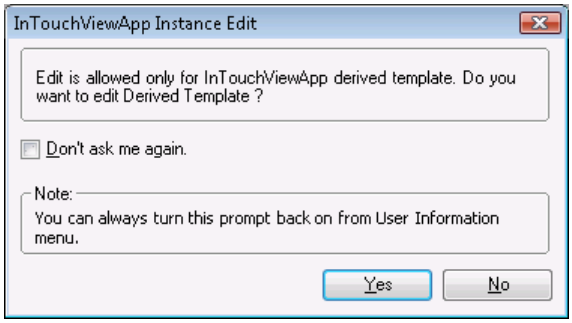
1. IDE を開きます。
2. 変更するマネージド InTouch アプリケーションが含まれている InTouchViewApp テンプレートを検索します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. InTouchViewApp テンプレートをダブルクリックします。オブジェクトのデフォルトのエディタとして WindowMaker が起動し、InTouch アプリケーションが開きます。マネージド アプリケーションを編集する準備ができました。

InTouchViewApp インスタンスから WindowMaker を起動する

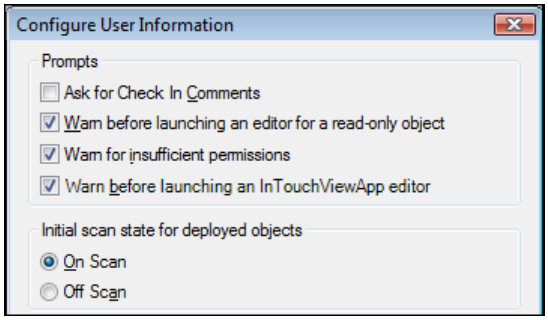
1. IDE を開きます。
2. 変更するマネージド InTouch アプリケーションをホストする親を持つ、InTouchViewApp オブジェクトを検索します。
3. InTouchViewApp オブジェクトをダブルクリックします。［**InTouchViewApp インスタンスの編集**］ダイアログ ボックスが表示されます。



4. 以下のいずれかを実行します。
   - テンプレートを開かない場合は、［**いいえ**］をクリックします。
   - 関連付けられている InTouchViewApp テンプレートを WindowMaker で開くには、［**はい**］をクリックします。

     ［**再度表示しない**］チェック ボックスをオンにし、［**はい**］をクリックした場合、次回 InTouchViewApp インスタンスを開いたときに、関連付けられている InTouchViewApp テンプレートからマネージド InTouch アプリケーションが自動的に開きます。

     この設定は、［**編集**］メニューから開くことができる［**ユーザー情報の編集**］ダイアログ ボックスを使用して変更できます。



マネージド InTouch アプリケーションの言語の切り替え

マネージド InTouch アプリケーションでは、特定の言語の切り替え機能が InTouch HMI ではなく IDE によって処理されます。

IDE を使用して、以下を行うことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 翻訳用にグラフィック テキストをエクスポートおよびインポートする。
- マネージド アプリケーションによって使用される言語を構成する。

詳細については、『アプリケーション サーバー のユーザーマニュアル』を参照してください。

InTouch アプリケーションの変更の送信

マネージド InTouch アプリケーションの変更後、対象ノードに配置する変更を送信できます。

マネージド InTouch アプリケーションに変更を加えた後、派生したすべての InTouchViewApp オブジェクトが［変更の保留］アイコンと共に表示されます。

これは、対象ノード上にある WindowViewer が変更を反映するためには、対象ノードに変更を再配置する必要があることを示します。

変更をオペレータ ノードに配置する方法の詳細については、「マネージド InTouch アプリケーションを配置する」および「オペレータ ノードでの新しいアプリケーション バージョンの受け入れ」を参照してください。

InTouch アプリケーション用に変更を送信する

1. スタンドアロンの InTouch アプリケーションと同じ方法で、WindowMaker でマネージド InTouch アプリケーションを変更します。
2. InTouch ウィンドウを保存します。
3. ［**ファイル**］メニューで、［**終了**］を選択します。WindowMaker が閉じ、フォーカスが IDE に戻ります。［**チェック イン**］ダイアログ ボックスが表示されます。
4. 必要に応じてチェック インのコメントを入力し、［**OK**］をクリックします。［**チェック イン**］進捗ダイアログ ボックスが表示されます。
5. **[閉じる]**を選択します。

InTouch アプリケーションのインポート

マネージド InTouch アプリケーションとして使用するために、既存の InTouch アプリケーションをインポートできます。これを行うには、以下の 2 つの手順を実行します。

- InTouchViewApp オブジェクトを作成し、インポートされた InTouch アプリケーションに関連付けます。
- 既存の InTouch アプリケーションをインポートします。

インポートされた InTouch アプリケーションは、マネージド InTouch アプリケーションになります。

IDE が管理しているフォルダに既存の InTouch アプリケーションのコピーが作成されます。IDE によって既存の InTouch アプリケーションがそのまま残され、場所も変わりません。

インポートする InTouch アプリケーションのバージョンが 6.0 以降である場合、変換に関するメッセージが表示されます。アプリケーションは WindowMaker で開く前に変換されます。6.0 以前のバージョンの InTouch アプリケーションは変換できません。

Galaxy に InTouch アプリケーションをインポートする

1. IDE を開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**テンプレート ツールボックス**］で、$InTouchViewApp ベース テンプレートからテンプレートを派生させます。
3. 派生テンプレートを開きます。［**InTouchViewApp 初期化**］ダイアログ ボックスが表示されます。
4. ［**既存の InTouch application をインポート**］をクリックし、［**次へ**］をクリックします。
5. 既存の InTouch アプリケーションを見つけます。以下のいずれかを実行します。
   - 省略記号ボタンをクリックし、マネージド InTouch アプリケーションが含まれているフォルダを参照します。
   - アプリケーションを検索するには、［**アプリケーションの検索**］チェック ボックスをオンにします。検索を開始する検索ルートを指定し、［**検索**］をクリックします。リストから InTouch アプリケーションを選択します。
6. **[次へ]** を選択します。
7. 必要に応じて、［**アプリケーション名**］ボックスに新しい名前を入力し、［**説明**］ボックスに説明を入力します。マネージド InTouch アプリケーションが配置されるときに、アプリケーション マネージャに名前と説明が表示されます。
8. アプリケーションのターゲット解像度がデフォルトの画面解像度オプションと異なる場合、アプリケーションのターゲット解像度を選択します。このフィールドのオプションは、次のように表示されます。
   a. ［**ターゲット解像度の選択**］ドロップダウン メニューをクリックして、事前定義されたターゲット解像度のリストを表示します。
   b. ［**ターゲット解像度の選択**］ドロップダウン メニューをクリックして、［**カスタム**］を選択します。ピクセルの幅と高さのフィールドが編集可能になります。

   注意：既存のアプリケーションをインポートするときに別のターゲット解像度を選択した場合、ソース アプリケーション解像度からターゲット解像度へのアプリケーション解像度の変換は行われません。
9. **[次へ]** を選択します。次のパネルが表示され、インポートの進捗が表示されます。
10. **[閉じる]**を選択します。［終了］をクリックします。WindowMaker が起動し、InTouch アプリケーションをマネージド InTouch アプリケーションとして編集できます。

グラフィックを含むスタンドアロン InTouch アプリケーションのインポート

マネージド アプリケーションに変換されたスタンドアロン InTouch アプリケーションにグラフィックが含まれる場合、アプリケーションはインポートされますが、グラフィックはインポートされません。

スタンドアロン アプリケーション内に含まれるグラフィックをインポートする

1. スタンドアロン アプリケーションをテンプレートとしてエクスポートします。
2. テンプレートをアプリケーション テンプレートの場所に保存します。
3. IDE で、この作成したアプリケーション テンプレートを選択して新しい $InTouchViewApp 派生テンプレートを作成します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouchViewApp オブジェクトのインポートとエクスポート

IDE で InTouchViewApp オブジェクトをインポートおよびエクスポートできます。InTouchViewApp オブジェクトには、マネージド InTouch アプリケーションをホストするためのすべての情報が含まれており、Galaxy 間でマネージド InTouch アプリケーションを交換するために使用できます。

自動オブジェクトのインポートとエクスポートの詳細については、Application Server のマニュアルを参照してください。

InTouchViewApp オブジェクトをインポートする

1. ［**Galaxy**］メニューで、［**インポート**］をポイントし、［**オブジェクト**］をクリックします。［**Import Automation Object(s)**］ダイアログ ボックスが表示されます。
2. インポートする InTouchViewApp オブジェクトが含まれている .aaPKG ファイルを選択し、［**開く**］をクリックします。オブジェクトおよびそのマネージド InTouch アプリケーションがインポートされます。
3. **[閉じる]**を選択します。

InTouchViewApp オブジェクトをエクスポートする

1. ［**Galaxy**］メニューで、［**エクスポート**］をポイントし、次に［**オブジェクト**］をクリックします。［**自動オブジェクトのエクスポート**］ダイアログ ボックスが表示されます。
2. エクスポート先のフォルダを選択し、.aaPKG ファイル名を指定して［**保存**］をクリックします。オブジェクトおよびそのマネージド InTouch アプリケーションがエクスポートされます。
3. **[閉じる]**を選択します。

タグ データのエクスポートとインポート

マネージド InTouch アプリケーションのタグ データを、.csv ファイルにエクスポートできます。このデータは、他のマネージド InTouch アプリケーションまたはスタンドアロンの InTouch アプリケーションにインポートできます。

マネージド InTouch アプリケーションからタグ データをエクスポートする

1. IDE を開きます。
2. タグ データのエクスポート元のマネージド InTouch アプリケーションが含まれている、派生した InTouchViewApp テンプレートを選択します。
3. ［**Galaxy**］メニューで、［**エクスポート**］をポイントし、次に［**DB ダンプ**］をクリックします。

   ［**ダンプ先 CSV ファイル名**］ダイアログ ボックスが表示されます。
4. .csv ファイルの場所およびファイル名を指定し、［**保存**］をクリックします。

   確認ダイアログ ボックスが表示されます。
5. .csv ファイル内でデータ タイプごとにタグ データをグループ化する場合、［**タグ タイプ毎にグループ出力**］を選択します。
6. **[OK]** を選択します。正常に完了したことを示すメッセージが表示されたら、［**OK**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.csv ファイルからマネージド InTouch アプリケーションにタグ データをインポートする

1. IDE を開きます。
2. タグ データのインポート先のマネージド InTouch アプリケーションが含まれている、派生した InTouchViewApp テンプレートを選択します。
3. ［**Galaxy**］メニューで、［**インポート**］をポイントし、次に［**DB ロード**］をクリックします。

   ［**ロード元 CSV ファイル名**］ダイアログ ボックスが表示されます。
4. .csv ファイルを選択し、［**開く**］をクリックします。

   インポート中、重複した名前があることを警告する、1 つまたは複数のメッセージが表示される場合があります。重複した各タグについて、適切なオプションを選択します。
5. 正常に完了したことを示すメッセージが表示されたら、［**OK**］をクリックします。

タグの値およびパラメータの保持

1 つのノードで複数のマネージド InTouch アプリケーションを使用している場合、すべてのアプリケーションで同じディレクトリが使用されます。タグ データまたはパラメータを保持するように設定し、アプリケーションを切り替えた場合、ランタイム データが失われます。

マネージド InTouch アプリケーションで保持を設定する

1. IDE から WindowMaker を使用してマネージド InTouch アプリケーションを開きます。
2. ［**ファイル**］メニューで、［**設定**］をポイントし、［**WindowViewer**］をクリックします。

   ［**WindowViewer のプロパティ**］画面が表示されます。
3. ［**マネージド アプリケーション**］タブをクリックします。
4. ［**ローカル ディレクトリ**］ボックスで、独自の既存のディレクトリの名前を入力します。これは、対象となるランタイム ノード上にあるディレクトリの名前です。特殊文字を使用して、ローカルの作業ディレクトリ パスを指定することもできます。以下の表には、使用できる特殊文字が示されています。

特殊文字	ローカル作業ディレクトリ パス
%SystemDrive%	システム ドライブ。"C:" など。
%USER%	ログオン ユーザーの名前。WindowViewer がサービスとして実行している場合、アプリケーションは "All Users" にコピーされます。
%APPDATA%	ログオンしているユーザーのアプリケーション データ。WindowViewer がサービスとして実行している場合、アプリケーションは "All Users" アプリケーション データにコピーされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

特殊文字	ローカル作業ディレクトリ パス
%ITUSER%	ログオン ユーザー。WindowViewer がサービスとして実行している場合またはコンソール セッションで実行している場合、アプリケーションは "All Users" にコピーされます。
%ITAPPDATA%	ログオンしているユーザーのアプリケーション データ。WindowViewer がサービスとして実行している場合またはコンソール セッションで実行している場合、アプリケーションは "All Users" アプリケーション データににコピーされます。

マネージド InTouch アプリケーションを削除する

関連付けられている InTouchViewApp テンプレートを削除することにより、IDE で InTouch アプリケーションを削除できます。

これを行うと、テンプレートに関連付けられているテンプレートおよび InTouch アプリケーション ディレクトリは完全に削除されます。

関連付けられている InTouchViewApp オブジェクトが派生したインスタンスを持たない場合、InTouch アプリケーションのみ削除できます。

InTouchViewApp インスタンスを削除しても、関連付けられている InTouch アプリケーションは削除されません。

InTouchViewApp テンプレートの削除

1. IDE を開きます。
2. 削除するマネージド InTouch アプリケーションが含まれている InTouchViewApp テンプレートを選択します。
3. ［**ホーム**］メニューの［**編集**］メニューから［**削除**］をクリックします。

   ［**削除**］ダイアログ ボックスが表示されます。
4. **[はい]**を選択します。

   InTouchViewApp テンプレートおよび関連付けられている InTouch アプリケーション フォルダが削除されます。

InTouchViewApp 内のすべての Galaxy グラフィックの関連付け

アプリケーションに埋め込むことなく、すべての Galaxy グラフィックを InTouchViewApp テンプレートに関連付けると、配置およびパブリッシュ済み InTouch アプリケーションが、Galaxy 内の任意のグラフィックで作成された「グラフィックの表示」要求を実行できるようになります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ShowGraphic() 関数は、グラフィックをパラメータとして使用します。すべての Galaxy グラフィックを関連付ければ、それが InTouchViewApp から参照されているかどうかには関わりなく、ランタイム時に配置され、利用できるようにすることができます。
- すべての Galaxy グラフィックを関連付ければ、ShowGraphic() 関数から参照されているテンプレートシンボルがランタイム時に配置され、利用できるようにすることができます。

注意：「グラフィック」という用語には、グラフィック ツールボックス内に存在するすべてのシンボルまたはクライアント コントロール、およびテンプレートとインスタンスが所有している、または継承されているすべてのシンボルが含まれます。

すべての Galaxy グラフィックと InTouchViewApp の関連付けの詳細については、『アプリケーション サーバー ユーザー ガイド』を参照してください。

InTouchViewApp の右クリック コンテキスト メニューで［すべての Galaxy グラフィックを含める］オプションにアクセスします。

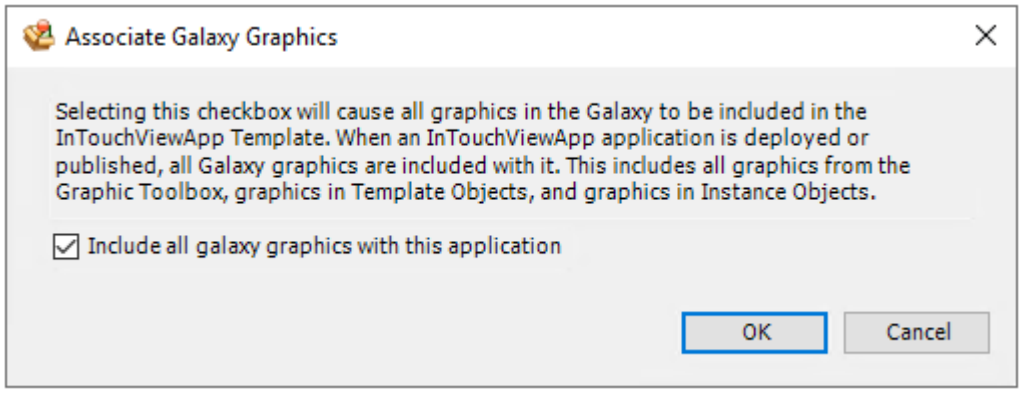
すべての Galaxy グラフィックを InTouchViewApp テンプレートに含める

1. 設定する InTouchViewApp テンプレートを右クリックします。

   InTouchViewApp のコンテキスト メニューが表示されます。

2. ［**Galaxy グラフィックの関連付け**］メニュー項目を選択します。

   ［**Galaxy グラフィックの関連付け**］ダイアログ ボックスが表示されます。



   ［**Galaxy グラフィックの関連付け**］ダイアログ ボックスが表示されると、$InTouchViewApp テンプレートがチェックアウトされます。

   $InTouchViewApp テンプレートがまだ初期化されていない、またはチェックアウトされていない場合には、［**すべての Galaxy グラフィックをこのアプリケーションに含める**］チェック ボックスは利用できません。

3. チェックボックスをオンにして**[OK]**を選択します。

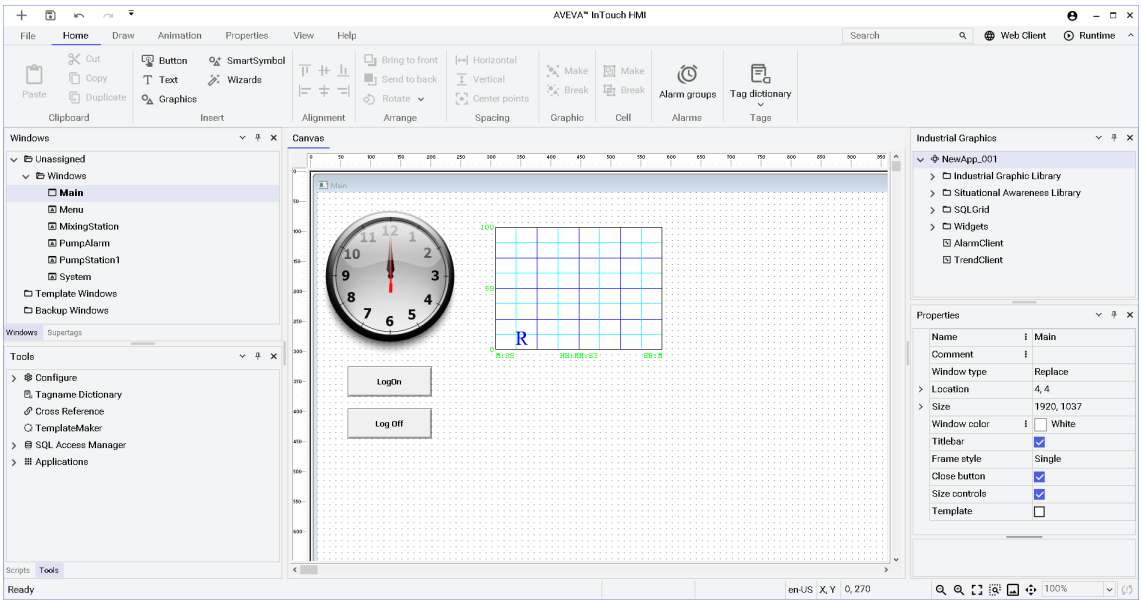
   ステータス ボックスにはチェックインの進行状況が表示されます。

   ［**すべての Galaxy グラフィックを含める**］設定を変更しようとしたときに InTouchViewApp インスタンスが配置されていた場合には、配置されているすべてのインスタンスの配置が解除されることを示す情報メッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 7 ウィンドウ

アプリケーション ウィンドウは、製造プロセスをモデル化する 1 つまたは複数のグラフィックのコンテナです。たとえば、ユニット内の装置を表示するウィンドウを設定できます。別のウィンドウには、そのユニットに関連するアラーム情報のグリッドを表示することができます。



ウィンドウはいくつでも作成でき、背景色、画面位置、ウィンドウ タイトルなどのウィンドウ プロパティを定義できます。

アプリケーション ウィンドウの作成

新しいアプリケーション ウィンドウを作成する場合、デフォルト設定では以前に作成されたウィンドウまたは現在アクティブな InTouch ウィンドウの設定が反映されます。以下の設定には、以前に作成したウィンドウの設定が反映されます。

- ウィンドウ名: 最大 32 文字。使用できる特殊文字はドル記号（$）、シャープ記号（#）、およびアンダースコア（_）だけです。ウィンドウを作成する場合、ウィンドウ名を指定することだけが必要とされます。その他のすべてのアイテムはオプションです。

  注意：ウィンドウ名にスペースなどのサポートされていない文字が含まれる場合、これらの各文字はアンダースコア（_）で置き換えられます。アプリケーションを移行した場合、ウィンドウの編集、またはウィンドウ プロパティの変更を行うと、ウィンドウ名に含まれる特殊文字はアンダースコア（_）で置き換えられます。この処理は、アプリケーション、枠、およびテンプレートを始めとするすべてのタイプのウィンドウに適用されます。

  これを"_"に置き換えたくない場合は、 .ini ファイルを開き、 WindowNameWithUpdateialAjaxacters = 1 を更新します。

- コメント: ウィンドウに関連付けるコメントは入力できますが、情報目的でのみ使用できます。コメントはファイル リストに表示されますが、アプリケーションでは使用されません。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

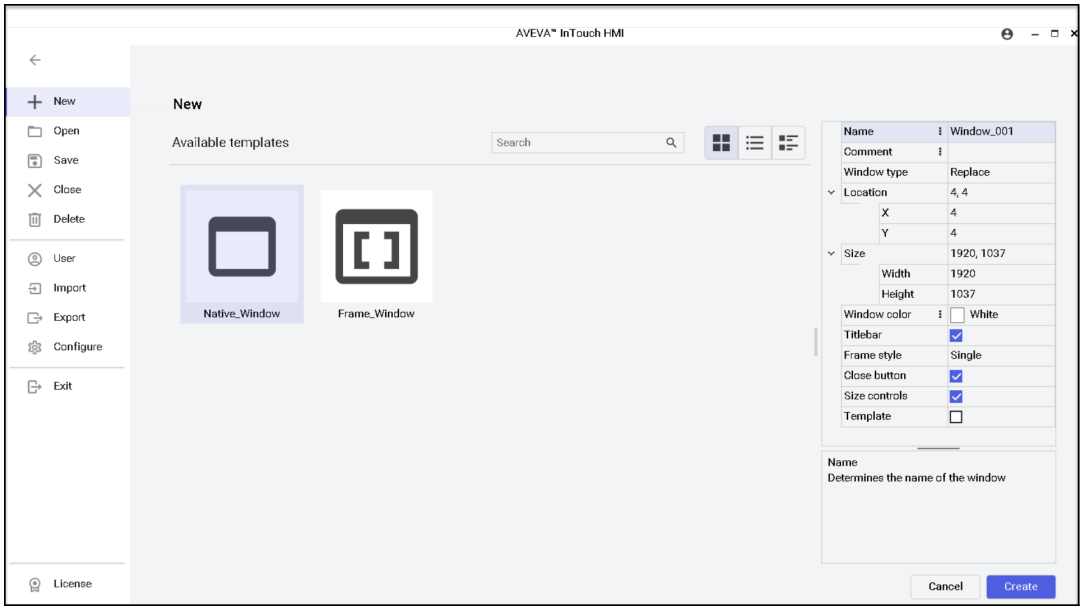
- ウィンドウの寸法値: デフォルトでは、ウィンドウの寸法値は以前作成したウィンドウの寸法に設定されます。これらの値は、ウィンドウの境界をドラッグすることによってウィンドウ サイズを手動で変更した場合にも自動的に修正されます。

注意：画面解像度と異なるアプリケーション ターゲット解像度を指定した場合、アプリケーション ウィンドウはキャンバス境界を超えることがあります。しかし、ランタイム時に表示されるのは、ターゲット解像度のキャンバス境界内のウィンドウだけです。

新しいウィンドウを作成する

1. ［**ファイル**］メニューで、［**新規**］を選択します。

   ［**新規**］ウィンドウが開き、新しいネイティブ ウィンドウおよび枠ウィンドウを作成するために使用できるテンプレートが表示されます。



2. 使用可能なテンプレートから作成するウィンドウのタイプを選択します。

   ウィンドウの作成に使用するテンプレートの名前を［検索］ボックスに入力して Enter キーを押すこともできます。

3. テンプレートの表示レイアウトを設定するには、［**アイコン**］、［**リスト**］、または［**詳細**］を選択します。

4. ［ウィンドウのプロパティ］パネルを使用して、ウィンドウの基本的なプロパティを設定します。

   ウィンドウのプロパティは、［外観］、［レイアウト］、［ウィンドウスタイル］の 3 つのカテゴリに分類されます。

5. 新規ウィンドウの外観を設定するには:
   - ［**名前**］テキスト ボックスに、ウィンドウを識別する一意の名前を入力します。
   - ［**コメント**］テキスト ボックスに、ウィンドウに関連付けるコメントを入力します。コメントは 60 文字以内である必要があります。
   - ［**ウィンドウの色**］ボタンをクリックして、ウィンドウの背景色を選択します。
   - ウィンドウの［**枠スタイル**］を選択すると、ウィンドウの周囲に境界線が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 一本の実線の枠にするには、［**シングル**］をクリックします。
- 3D 枠線にするには、［**ダブル**］をクリックします。
- ウィンドウに枠線を付けない場合には、［**なし**］をクリックします。

［**名前**］、［**コメント**］、または［**ウィンドウの色**］フィールドの横にある省略記号をクリックすると、次の操作を行うことができます。

- リセット - フィールドの値をデフォルト値に戻します。
- 編集 - フィールドの詳細を編集します。
- ソート - ウィンドウのプロパティ フィールドをアルファベット順、カテゴリ、カテゴリとアルファベット順でソートします。
- 説明の表示 - 説明の表示と非表示を切り替えます。
- ツールバーの表示 - ツールバーの表示と非表示を切り替えます。

6. 新規ウィンドウのレイアウトを設定するには:
   - ［**ウィンドウ タイプ**］ドロップダウンからオプションを選択すると、ランタイム時にウィンドウがどのように開くかを確認できます。
     - ［ウィンドウのタイプ］で［**リプレース**］を選択すると、ウィンドウが画面に表示されたときに、そのウィンドウが交差するウィンドウが自動的に閉じます。
     - ［ウィンドウのタイプ］で［**オーバーレイ**］を選択すると、ウィンドウは現在開いているウィンドウの上に表示されます。重なっている他のウィンドウよりも大きくなる可能性があります。オーバーレイ ウィンドウが閉じると、その背後にあったウィンドウが再び表示されます。オーバーレイ ウィンドウの背後にあるウィンドウの表示部分をクリックすると、そのウィンドウがアクティブ ウィンドウとして前面に表示されます。
     - ［ウィンドウのタイプ］で［**ポップアップ**］を選択すると、ウィンドウは常に他のすべてのウィンドウの一番上に表示されます。ポップアップ ウィンドウの削除は通常、ユーザーからの応答を必要とします。
   - ［**位置**］フィールドを展開すると、ウィンドウの位置と寸法を指定できます。
     - ［**X**］ボックスに設計領域の左端と定義中のウィンドウの左端の間隔をピクセル数で入力します。
     - ［**Y**］ボックスに設計領域の上端と定義中のウィンドウの上端の間隔をピクセル数で入力します。
   - ［**サイズ**］フィールドを展開すると、ウィンドウの高さと幅をピクセル単位で指定できます。
7. 新規ウィンドウのウィンドウ スタイルを設定するには:
   - ［**閉じるボタン**］チェック ボックスをオンにすると、デザイン時またはランタイム時にウィンドウを閉じるためのボタンがタイトル バーに表示されます。これらのウィンドウは別のアプリケーションにエクスポートした後でも、ウィンドウを閉じるボタンを保持します。
   - ［**最大化ボタン**］チェック ボックスをオンにすると、デザイン時またはランタイム時にウィンドウを最大化するためのボタンがタイトル バーに表示されます。
   - ［**最小化ボタン**］チェック ボックスをオンにすると、デザイン時またはランタイム時にウィンドウを最小化するためのボタンがタイトル バーに表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［サイズ コントロール］チェック ボックスをオンにすると、ユーザーが WindowMaker でウィンドウのサイズを変更できるようになります。
- ［タイトル バー］チェック ボックスをオンにして、タイトル バーを表示します。
- ［テンプレート］チェック ボックスをオンにすると、設定をテンプレートとして保存し、後で使用することができます。
- ［ウィンドウの状態］ドロップダウンから、ウィンドウのデフォルト状態（通常、最小化、または最大化）を選択します。

8. **[OK]** を選択します。

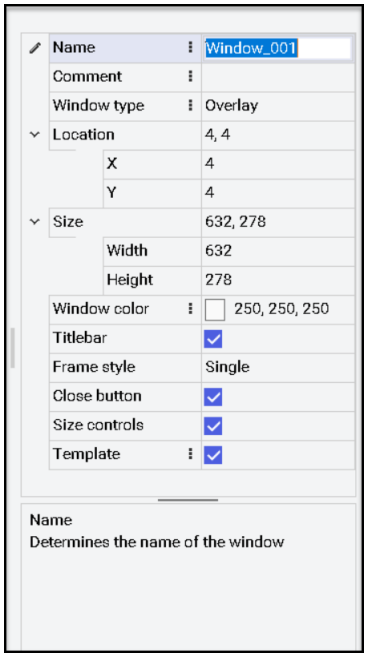
アプリケーション ウィンドウをテンプレート ウィンドウとして設定する

ウィンドウのプロパティを使用して、アプリケーション ウィンドウをテンプレート ウィンドウとして設定できます。このように設定すると、ウィンドウを再使用して設定時間を短縮できます。

アプリケーション ウィンドウをテンプレート ウィンドウとして設定する

1. 次のいずれかを実行して、［ウィンドウのプロパティ］ペインを開きます。
   a. 新しいウィンドウをテンプレート ウィンドウとして設定するには、［ファイル］メニューで［新規］をクリックします。
   b. 既存のウィンドウをテンプレート ウィンドウとして設定するには、［ウィンドウ］ペインで既存のウィンドウをクリックします。

   画面の右側にウィンドウのプロパティ ペインが表示されます。
2. ［ウィンドウのプロパティ］ペインで［テンプレート］チェックボックスをオンにします。



3. **[OK]** を選択します。

   ［ウィンドウ］ペインの下にある［テンプレート ウィンドウ］フォルダにウィンドウが表示されます。新しく作成したウィンドウは、テンプレート ウィンドウ ブラウザでも使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：［ウィンドウのプロパティ］ペインの［テンプレート］チェックボックスの選択を解除すれば、テンプレート ウィンドウをアプリケーション ウィンドウに戻すことができます。このプロパティは、複数の［テンプレート ウィンドウ］フォルダの間でウィンドウをドラッグ アンド ドロップすることによっても変更できます。

テンプレート ウィンドウからのアプリケーション ウィンドウの作成

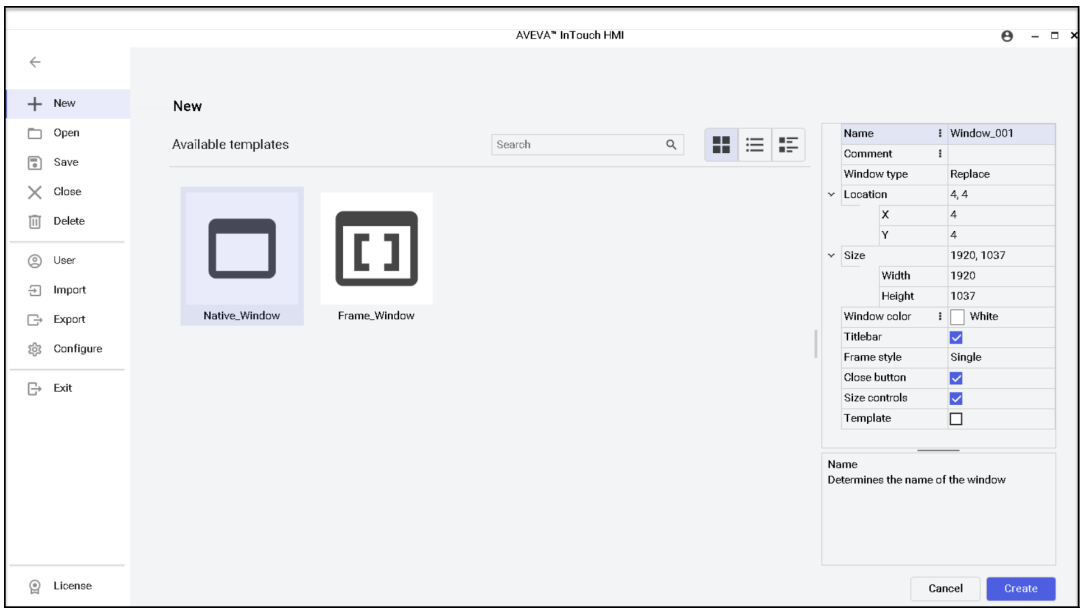
テンプレート ウィンドウ ブラウザで使用可能なテンプレート ウィンドウから InTouch アプリケーション ウィンドウを作成できます。

注意：テンプレート ウィンドウ ブラウザで使用できるのは、［テンプレート］プロパティが選択されているウィンドウだけです。詳細な手順については、「アプリケーション ウィンドウをテンプレート ウィンドウとして設定する」を参照してください。

テンプレート ウィンドウからアプリケーション ウィンドウを作成するには：

1. ［**ファイル**］メニューで、［**新規**］を選択します。

   ［**使用可能なテンプレート**］が表示されます、



2. テンプレート ウィンドウのサムネイルを参照して、目的のウィンドウを選択します。

   注意：デフォルトでは、各テンプレート ウィンドウがサムネイル ビューに表示されます。ウィンドウ サムネイルは、アプリケーション全体に対してデザイン時の実際のウィンドウの位置に相対的に表示されます。

3. **[OK]** を選択します。

   注意：移行したアプリケーションのテンプレート ウィンドウのウィンドウ名にサポートされない文字が含まれる場合、新しいアプリケーション ウィンドウ内のスペース文字などのサポートされない各文字はアンダースコア（_）で置き換えられます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

枠ウィンドウの操作

スタンドアロンまたはマネージド InTouch アプリケーションで作業している場合、アプリケーション ウィンドウに加えて、またはアプリケーション ウィンドウの代わりに枠ウィンドウを作成および開発できます。枠ウィンドウを使用すると、パン、ズーム、およびタッチ機能をサポートできる産業用グラフィックをホストできます。

枠ウィンドウは、アプリケーション ウィンドウと同じプロパティの大半をサポートします。しかし、いくつかの制限が適用されます。

枠ウィンドウには、以下の制限があります。

- 複数の産業用グラフィックはサポートされません。
- ネイティブの InTouch コントロールはホストされません。
- ActiveX コントロールはホストされません。
- SmartSymbol はサポートされません。
- ウィンドウ コンポーネントの「元に戻す」/「やり直す」操作はサポートされません。
- 埋め込まれたシンボルはフリップおよび回転しません。
- 切り取り、コピー、貼り付け、および重複操作はサポートされません。
- 枠ウィンドウに埋め込まれた産業用グラフィックのクロス 参照のサポートは提供されません。
- 産業用グラフィックへのウィンドウ変換はサポートされません。
- 新規インスタンスの作成、インスタンスの編集、および埋め込まれたオートメーション オブジェクト グラフィックの代替インスタンスの選択はサポートされません。

枠を作成する

1. ［ファイル］メニューで、［新規］を選択します。

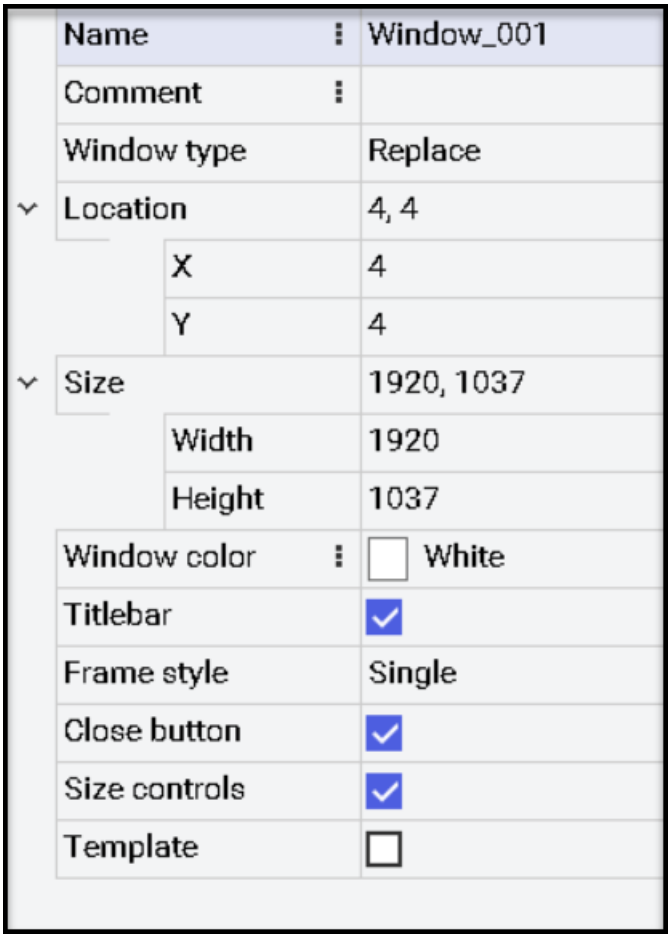
   ［使用可能なテンプレート］のリストを含む［新規］ウィンドウが表示されます。

2. ［枠ウィンドウ］を選択します。
3. ウィンドウのプロパティを設定して、［**OK**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ パネルの使用

枠のプロパティは、WindowMaker キャンバスの［プロパティ］パネルから直接編集できます。枠のプロパティは、ウィンドウの作成時にプロパティ グリッドに入力されます。



デフォルトでは、産業用グラフィック ツールボックスおよびプロパティ グリッドがキャンバスの右側に表示されます。グラフィック ツールボックスまたはプロパティ グリッドは、キャンバスの左側に固定できます。WindowMaker の設定をデフォルトに戻すには、［**表示**］、［**レイアウトのリストア**］をクリックします。

注意：アプリケーション ウィンドウのプロパティは、**プロパティ グリッド**に表示されません。

枠ウィンドウは、アプリケーション ウィンドウと同じプロパティに加えて、3 つの追加プロパティをサポートします。詳細については、以下を参照してください。

各ウィンドウ プロパティの詳細な説明については、「アプリケーション ウィンドウの作成」を参照してください。

枠のプロパティ	機能
[最大化] ボタン	枠の右上の最大化ボタンを有効にします。 デフォルトは［**False**］です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

[最小化]ボタン	枠の右上の最小化ボタンを有効にします。 デフォルトは［**False**］です。
WindowState	ウィンドウの初期状態(**[Normal]**、 **[Minimized]** 、 **[Maximized])**が表示されます。 デフォルトは［**Normal**］です。

FrameStyle 以外の枠のプロパティに対して行った変更は、直ちに枠に反映されます。このプロパティを設定するには、最初に **SizeControl** および **TitleBar** プロパティを **False** に設定する必要があります。

［**FrameStyle**］ドロップダウン メニューが有効になります。

枠ウィンドウに埋め込まれたグラフィックの操作

産業用グラフィックは､アプリケーション ウィンドウと同じ方法で枠ウィンドウに埋め込むことができます。産業用グラフィック ツールボックスは、System Platform IDE と InTouch WindowMaker の両方に統合されています。グラフィック ツールボックスには、定義済みシンボルの産業用グラフィック ライブラリと Situational Awareness Library を含む個別のフォルダが含まれます。産業用グラフィック ライブラリには標準的な産業用オブジェクトの写実的なシンボルが含まれています。

Situational Awareness Library のシンボルは、簡略化された外観で、最低限の視覚的詳細を提供することにより、オペレータに無関係の情報を表示することなく、機能の目的とステータスを効率的に伝達します。ほとんどの Situational Awareness Library のシンボルは、各シンボルに複数の視覚的および機能的な設定が組み込まれたシンボル ウィザードとして設計されています。シンボルの機能設定は､シンボルの［ウィザード オプション］からオプションを選択するだけで簡単に選択できます。

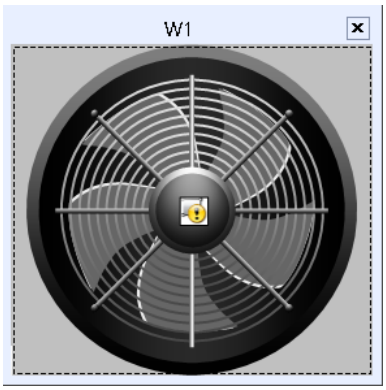
シンボルを枠に埋め込むには、次のいずれかの手順を実行します。

- [枠ウィンドウ]を作成したら、[グラフィックを埋め込む]ボタン キャンバス上に表示され、グラフィックを参照します。
- フレームウィンドウを作成した後、キャンバス上にある［グラフィック作成］ボタンを選択して、産業用グラフィック エディタでグラフィックを作成し、グラフィックを保存します。作成したグラフィックはウィンドウに埋め込まれます。
- 産業用グラフィックをグラフィック ツールバーから枠にドラッグ アンド ドロップします。
- ［産業用グラフィックの埋め込み］ツールバー ボタン を選択して、グラフィックを参照します。
- 枠ウィンドウのコンテキスト メニューを使用してグラフィックを埋め込みます。

枠のサイズを埋め込まれた産業用グラフィックに自動的に合わせることができます。埋め込まれたシンボルに枠のサイズを合わせるには、ウィンドウを右クリックして、［**シンボルに合わせる**］を選択します。

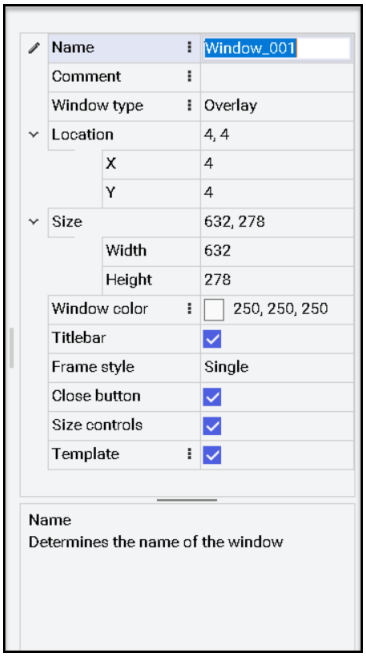
枠のサイズがシンボルに合わせて変更されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



注意：［**シンボルに合わせる**］の操作は元に戻すことができません。枠ウィンドウの寸法または Size プロパティを手動で変更した場合、［**シンボルに合わせる**］操作を再度実行する必要があります。

産業用グラフィックを枠に埋め込むと、以下に示すように、シンボルが ［**プロパティ グリッド**］ドロップダウン メニューに含まれるようになります。



枠ウィンドウとシンボル プロパティを切り替えることができます。以下のいずれかを実行します。

- ドロップダウン メニューからシンボルまたは枠ウィンドウ名を選択します。
- ウィンドウ枠を右クリックして、［**ウィンドウのプロパティ**］を選択します。
- タイトル バーまたは枠の境界をクリックして、枠ウィンドウのプロパティをグリッドに表示します。

  埋め込まれたシンボルをクリックして、シンボルのプロパティを表示します。

シンボルのプロパティを設定して、キーボード、マウス、およびタッチ ジェスチャを使用したパンとズーム機能を有効にする必要があります。設定可能なシンボル プロパティの一覧を以下に示します。

シンボルのプロパティ	機能

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ｱｽﾍﾟｸﾄ比の維持	モダン枠のサイズを変更する際にシンボルのアスペクト比を維持します。デフォルトは **True** です。
ｼﾝﾎﾞﾙ名	モダン枠によってホストされるシンボル名を設定します。
ｲﾝﾀﾗｸｼｮﾝ ﾓｰﾄﾞ	操作モードを設定します。以下のオプションがあります。 • **None**: パンとズームを無効にします。 • **PanZoom**（デフォルト）: パンとズームを有効にします。
ｽﾞｰﾑ ｺﾝﾄﾛｰﾙの表示	シンボルとズーム コントロールを表示します。以下のオプションがあります。 • **Auto**: コントロールは必要に応じて表示されます。 • **Visible**: コントロールは常に表示されます。

ランタイム時のパンとズーム機能の使用方法の詳細については、*AVEVA™ InTouch HMI の『InTouch HMI コンポーネントの標準作成』ガイド*にある「パンとズーム」を参照してください。

枠ウィンドウのリボンの使用

枠フレームワークを作成する際、ウィンドウのリボンで以下のオプションが使用可能になります。

- シンボルの編集 - シンボルを編集できるグラフィック エディタを起動します。詳細については、『産業用グラフィック エディタ ユーザー ガイド』を参照してください。
- ウィンドウを保存 - ウィンドウを保存します。「ウィンドウを開く、保存する、閉じる」を参照してください。
- ウィンドウに名前を付けて保存 - ウィンドウの複製を作成します。「ウィンドウの複製」を参照してください。
- ウィンドウの削除 - ウィンドウを削除します。「ウィンドウの削除」を参照してください。

枠ウィンドウのキャンバスを右クリックすると、使用可能ないくつかのオプションがリボンでハイライトされます。

アプリケーション ウィンドウの変更

アプリケーションを開発するときに、いつでもウィンドウのプロパティを変更できます。

アプリケーション ウィンドウのプロパティを変更する

1. ウィンドウ名を右クリックして、［プロパティ］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［ウィンドウのプロパティ］パネルで必要な変更を行います。
3. **[OK]** を選択します。

ウィンドウ オプションの詳細については、「アプリケーション ウィンドウの作成」を参照してください。

ウィンドウを開く、保存する、閉じる

ウィンドウを開く

アプリケーションの開発中、コンピュータのメモリがサポートする範囲で複数のウィンドウを開くことができます。

ウィンドウを開く

1. ［ファイル］メニューで、［開く］を選択します。

   ［開くウィンドウを選択］ダイアログ ボックスが表示され、アプリケーション内のすべてのウィンドウ名が一覧表示されます。

2. 以下のいずれかを実行します。
   - ウィンドウを 1 つ開くには、ウィンドウ名をダブルクリックします。
   - 複数のウィンドウを開くには、開くウィンドウのチェック ボックスをオンにして、［OK］をクリックします。

ウィンドウの保存

ウィンドウを保存するときには、ウィンドウに関連付けられているすべてのグラフィック、QuickScript、プロパティなども保存されます。

ウィンドウを保存します

1. ［ファイル］メニューで、［保存］を選択します。

   ［保存するウィンドウを選択］ダイアログ ボックスが表示され、すべてのウィンドウの名前が一覧表示されます。

2. 保存する必要があるウィンドウを選択します。
3. **[OK]** を選択します。

ウィンドウを閉じる

変更されたウィンドウを閉じると、変更内容を保存するよう求めるメッセージが表示されます。

ウィンドウを閉じる

1. ［ファイル］メニューで、［保存］を選択します。

   ［閉じるウィンドウを選択］ダイアログ ボックスが表示され、現在開いているすべてのウィンドウ名が一覧表示されます。

2. ウィンドウ名の横にあるチェック ボックスをオンにします。
3. **[OK]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウのサムネイル プレビューを表示する

WindowMaker でウィンドウのサムネイルをプレビュー表示することができます。これは、特に多くのウィンドウを持つアプリケーションの場合に便利な機能です。ウィンドウを開く前に、サムネイルのプレビュー表示で正しいウィンドウかどうかを確認することができます。

ウィンドウのサムネイルをプレビュー表示することができます。サムネイルのプレビューの表示と更新は、ウィンドウを保存した後にのみ行うことができます。アプリケーションを移行またはウィンドウを XML ファイルからインポートする場合、ウィンドウを明示的に保存しなくてもサムネイルのプレビューを表示することができます。別のアプリケーションからウィンドウをインポートする場合、ウィンドウを明確に保存しなくてもサムネイルのプレビューを表示することができます。しかし、インポート後にウィンドウを変更した場合、ウィンドウを保存しなければサムネイルに変更は反映されません。

InTouch アプリケーションのすべてのウィンドウ サムネイルを更新することもできます。サムネイルを更新する前に、すべてのウィンドウを閉じる必要があります。

すべてのウィンドウ サムネイルを更新する

- ［表示］メニューの［更新］グループで［ウィンドウ サムネイル］をクリックします。

特定ののウィンドウのサムネイルを更新する

1. ウィンドウを右クリックして、［サムネイルの更新］をクリックします。

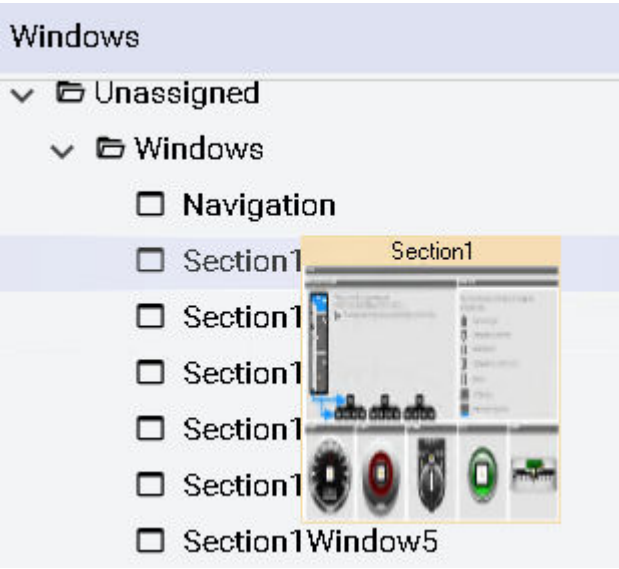
   確認メッセージが表示されます。

2. **[続行する]**を選択します。InTouch アプリケーションのすべてのウィンドウ サムネイルが更新されます。

   サムネイルを更新するために、システムによってすべてのウィンドウが開かれてから閉じられます。

ウィンドウのサムネイル プレビューを表示する

1. WindowMaker を開きます。
2. アプリケーションを開いてウィンドウを保存します。
3. ウィンドウの名前と関連付けられたアイコンを表示している長方形領域の上にポインタを移動します。サムネイルのプレビューが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウの複製

シミュレーションおよびコントロールが類似した複数のプロセスがある場合、ウィンドウを複製して、2 つ目のプロセスまたはユニット用にカスタマイズして使用することができます。

ウィンドウに関連付けられているすべてのグラフィック、QuickScript、プロパティなども含まれるウィンドウを複製できます。

開始する前に、複製するウィンドウを開いた状態にし、少なくとも 1 回保存する必要があります。一度に複製できるのは 1 つのウィンドウのみです。

ウィンドウを複製する

1. ［ウィンドウ］ペインで目的のウィンドウを右クリックし ［**名前を付けて保存**］をクリックします。

   ［**ウィンドウを保存**］ダイアログ ボックスが開きます。
2. ［**新しい名前**］ボックスに、新しいウィンドウの名前を入力します。
3. **[OK]** を選択します。

注意：元のウィンドウ名にサポートされない文字が含まれる場合、複製したウィンドウ名のサポートされない各文字はアンダースコア（_）文字で置き換えられます。

ウィンドウの削除

コンピュータの保存領域を節約するため、またはアプリケーション エクスプローラのウィンドウ リストが長すぎて管理できない場合には、使用していないウィンドウを削除できます。

注意：ウィンドウを削除するには、正しいウィンドウであることを確認してください。削除されたウィンドウは、［**元に戻す**］では復元できません。

ウィンドウを削除します

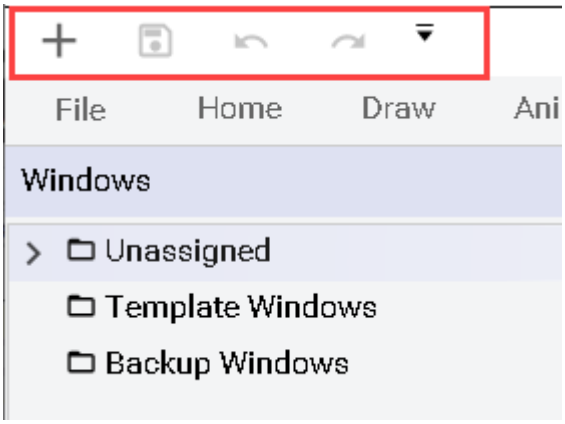
1. ［**ファイル**］メニューで、［**削除**］を選択します。

   ［**削除するウィンドウを選択**］画面が表示されます。
2. 削除するウィンドウ名を選択して、［**OK**］をクリックします。メッセージが表示されたら、［**はい**］をクリックします。
3. **[OK]** を選択します。

クイック アクセス ツールバーの使用

クイック アクセス ツールバーはカスタマイズ可能なツールバーで、事前定義済みのコマンドまたは頻繁に使用するコマンドを含めて、すべての画面からすばやくアクセスすることができます。デフォルトでは、クイック アクセス ツールバーは、WindowMaker 画面の左上角に表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



クイック アクセス ツールバーからのコマンドの実行

クイック アクセス ツールバーからは以下のコマンドを実行できます。

注意：クイック アクセス ツールバーのほとんどのコマンドは他のリボン グループからもアクセスできますが、いくつかのコマンドはクリック アクセス ツールバーからしかアクセスできません。

クイック アクセス ツールバーのコマンド	説明
新規	新しいウィンドウを作成します。
開く	既存のウィンドウを開きます
保存	ウィンドウを保存します
名前を付けて保存	ウィンドウのコピーを作成し、別の名前と場所を指定して保存します
すべて保存	開いているすべてのウィンドウを保存します
閉じる	開いているウィンドウを閉じます
削除	ウィンドウを削除します
インポート	ウィンドウをインポートします
エクスポート	ウィンドウをエクスポートします
印刷	ウィンドウを印刷します
WindowViewerWindowViewer	ウィンドウを WindowViewer で表示します
産業用グラフィックに変換	既存のグラフィックを産業用グラフィックに変換します
元に戻す	最後の操作をキャンセルします
やり直す	最後の操作を繰り返します
終了	WindowMaker を終了します

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クイック アクセス ツールバーのカスタマイズ

デフォルトでは、クイック アクセス ツールバーには、新規、保存、元に戻す、やり直し、のコマンドが表示されます。

リストからクイック アクセス ツールバーに表示するコマンドを追加または削除できます。

クイック アクセス ツールバーにコマンドを追加するには：

1. クイック アクセス ツールバーの下向きの矢印をクリックします。
2. 追加するコマンドを選択します。

クイック アクセス ツールバーからコマンドを削除するには：

1. クイック アクセス ツールバーの下向きの矢印をクリックします。
2. 選択したコマンドを強調表示します。

クイック アクセス ツールバーの配置

デフォルトでは、クイック アクセス ツールバーはリボンの上に表示されます。

- クイック アクセス ツールバーをリボンの下に配置するには、クイック アクセス ツールバーの下向きの矢印をクリックして［**リボンの下に表示**］を選択します。
- クイック アクセス ツールバーをリボンの上に配置するには、クイック アクセス ツールバーの下向きの矢印をクリックして［**リボンの上に表示**］を選択します。

クイック アクセス ツールバーを使用したリボンのサイズ変更

クイック アクセス ツールバーからリボンを最小化または最大化することができます。

- リボンの最小化：このビューではメニュー項目だけが表示され、グループは表示されません。リボンを最小化するには、クイック アクセス ツールバーの下向きの矢印をクリックして［**リボンを最小化**］を選択します。
- リボンの最大化：このビューではメニュー項目とグループが表示されます。リボンを最大化するには、クイック アクセス ツールバーの下向きの矢印をクリックして［**リボンを最大化**］を選択します。

InTouch ウィンドウに関する情報の印刷

InTouch ウィンドウに関する以下の情報を印刷できます。

- ウィンドウ内に配置されたグラフィカル オブジェクトの詳細。たとえば、すべてのタイプのオブジェクトのウィンドウの位置、テキスト オブジェクトに使用されているフォント、産業用グラフィック用に定義されたカスタム プロパティなどです。
- ウィンドウで使用されているアニメーション リンクに関する詳細
- ウィンドウに関連するスクリプト。
- ウィンドウで使用されているタグ。

グラフィカル オブジェクトの詳細は、プリンタまたは .html ファイルに出力できます。各ウィンドウに対して、.html ファイルが作成されます。.html ファイルには、以下のものが含まれています。

- 参照されている .png ファイルとしてのウィンドウのピクチャ。
- ウィンドウ内の各グラフィカル オブジェクトに関する詳細の一覧。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

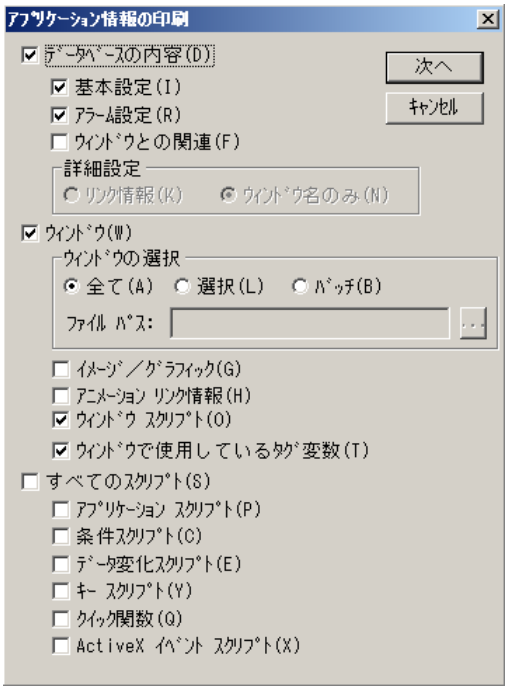
複数のウィンドウについてのグラフィカル オブジェクトの詳細を印刷した場合、ウィンドウ固有の .html ファイルにリンクしているサマリ .html ファイルが作成されます。

プリンタまたは .txt ファイルにアニメーション リンク、スクリプト、またはタグの詳細を出力できます。

InTouch ウィンドウに関する情報の印刷

1. WindowMaker で InTouch アプリケーションを開きます。
2. クイック アクセス ツールバーの［印刷］をクリックします。

   ［アプリケーション情報の印刷］ダイアログ ボックスが表示されます。



3. ［ウィンドウ］を選択します。
4. 印刷するウィンドウを選択します。
   - ［すべて］を選択すると、アプリケーションのすべてのウィンドウの情報（埋め込まれたグラフィックの名前を含む）が出力されます。
   - ［選択］を選択すると、特定のウィンドウの情報（選択したウィンドウの埋め込まれたグラフィックの名前を含む）だけが出力されます。［ウィンドウの印刷］ダイアログ ボックスが開きます。印刷したいウィンドウを選択し、［OK］をクリックします。
   - ［バッチ］を選択すると、.csv ファイルで指定したウィンドウの情報（埋め込まれたグラフィックの名前を含む）が出力されます。

   .csv 形式の詳細については、「ウィンドウを印刷するための .CSV 形式」を参照してください。
5. 選択したウィンドウ内の印刷対象を選択します。
   - ［イメージ/グラフィック］を選択すると、ウィンドウに配置されているすべてのグラフィック オブジェクトに関する情報が出力されます。
   - ［アニメーション リンク情報］を選択すると、ウィンドウのリンクの詳細が出力されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**ウィンドウ スクリプト**］を選択すると、ウィンドウに関連付けられているスクリプトが出力されます。
- ［**ウィンドウで使用しているタグ**］を選択すると、ウィンドウで使用しているタグが出力されます。

6. **[次へ]** を選択します。［**出力の送信先の選択**］ダイアログ ボックスが表示されます。
7. 以下のいずれかを実行します。
   - 情報を印刷するには、［**プリンタに出力を送信する**］をクリックします。
   - 単一の .txt ファイルを作成するには、［**テキスト ファイルに出力を送信する**］をクリックします。
   - 指定した各ウィンドウに対して、概要の .html ファイル、単一の .html ファイル、および .png ファイルを作成するには、［**HTML ファイルに出力を送信する**］をクリックします。.html ファイルおよび .png ファイルが存在する場合、自動的に上書きされます。
8. **[印刷]** を選択します。

コマンド プロンプトからのウィンドウ情報の印刷

コマンド プロンプトからウィンドウ情報を印刷できます。これを行うには、ウィンドウ名が含まれている .csv ファイルを作成し、次に印刷コマンドで .csv ファイルを参照します。

印刷コマンドが実行されると、デフォルトの InTouch アプリケーションが WindowMaker で自動的に開き、印刷操作が実行されてから閉じます。コマンド プロンプトからの印刷は、スタンドアロン InTouch アプリケーションに対してのみ機能します。

コマンド プロンプトからウィンドウ情報を印刷する

1. 印刷するウィンドウの名前が含まれている .csv ファイルを作成します。

   .csv 形式の詳細については、「ウィンドウを印刷するための .CSV 形式」を参照してください。
2. WindowMaker を終了し、.csv ファイルを閉じます。
3. コマンド プロンプトを開きます。
4. コマンドを入力して、情報を印刷します。

   コマンドの構文の詳細については、「コマンド プロンプトから印刷するための構文」を参照してください。
5. **Enter** キーを押します。WindowMaker が起動し、情報が印刷されます。

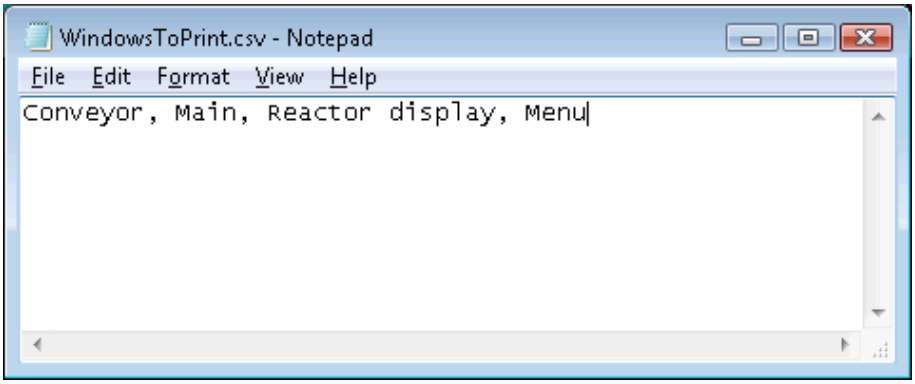
ウィンドウを印刷するための .CSV 形式

.csv ファイルを使用すれば、ウィンドウ情報を Microsoft Excel のような様々な形式で使用することができます。.csv ファイルで単一行を作成し、各ウィンドウ名をカンマで区切って入力します。

Window1,Window2,Window3,WindowN

以下に例を示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



Microsoft Excel を使用してファイルを作成する場合、1 行の各カラム セルに各ウィンドウの名前を入力します。

ウィンドウ名に円記号（\）は使用できません。

カンマを使用する必要があります。その他の区切り文字は使用できません。

コマンド プロンプトから印刷するための構文

コマンドの構文は以下のとおりです。

```
"<WindowMaker のパス名>" COMMANDFILE="<CSV ファイル>" ALL OUTPUTTARGET=<target name>
```

ここで、

<WindowMaker のパス名> は、WindowMaker アプリケーション（WM.exe）のパスです。

- <CSV ファイル> は、印刷するウィンドウを指定する .csv ファイルの名前です。
- <ターゲット名> は、プリンタまたは .html ファイルへの出力先です。
- ALL は、すべてのリンク、データベース エントリ、およびスクリプト情報を印刷するコマンドです。ALL コマンドを含めない場合、グラフィカル オブジェクト情報のみが印刷されます。

以下の例では、すべてのリンク、データベース エントリ、およびスクリプト情報がデフォルトのプリンタに送信されます。

```
"C:\Program Files\Wonderware\InTouch\wm.exe" COMMANDFILE="D:\print.csv" ALL OUTPUTTARGET =
PRINTER
```

以下の例では、グラフィカル オブジェクト情報が .html ファイルに送信されます。

```
"C:\Program Files\Wonderware\InTouch\wm.exe" COMMANDFILE="D:\print.csv" OUTPUTTARGET = HTML
<DEMOAPP.html>
```

コンテキスト メニューの深さの設定

コンテキスト メニューの深さ（ウィンドウの影またはコンテキスト メニューの影）は、画面上で各フローティング ウィンドウの境界を作成します。これは、ウィンドウと重なる他のウィンドウから区別するために役立ちます。

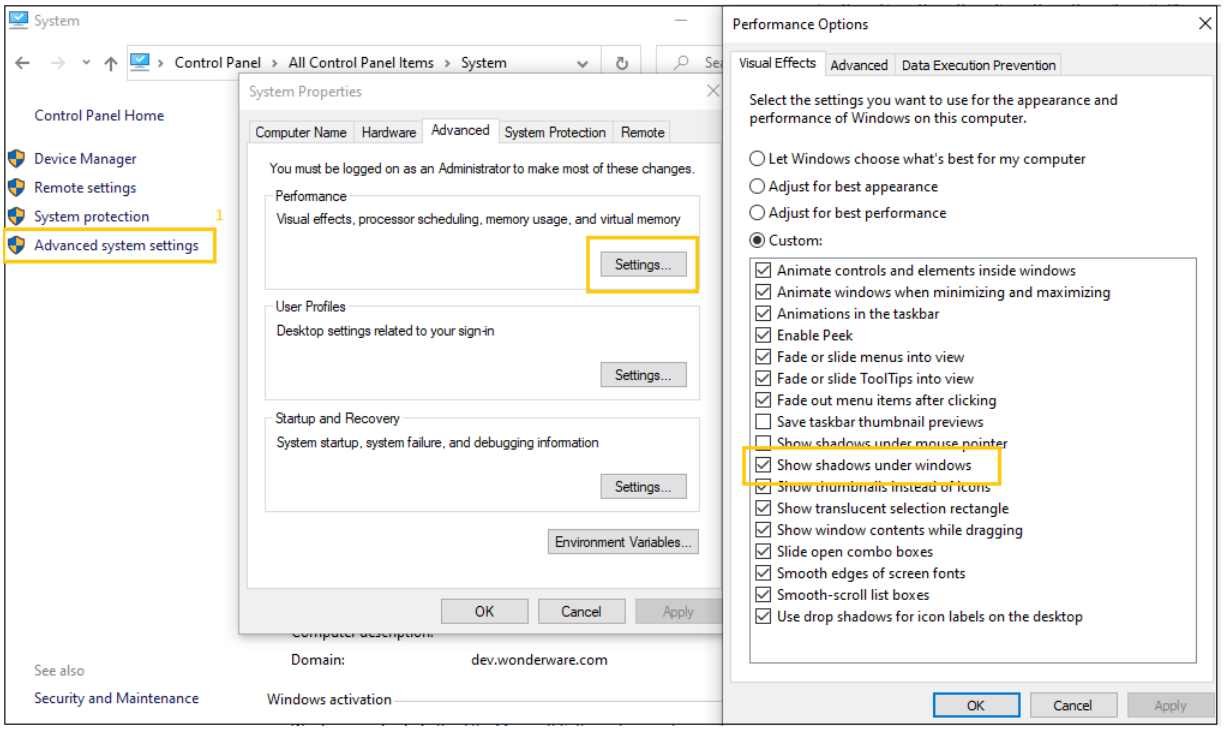
InTouch アプリケーションのコンテキスト メニューの深さは 2 つの方法で設定できます。

- コントロール パネルでのシステム プロパティの変更
- レジストリ エディタでのレジストリ エントリの変更

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

コントロール パネルを使用してコンテキスト メニューの深さを設定する

1. コントロール パネルを開きます。
2. **[システム]**を強調表示し、**[詳細設定]**を選択します。
   ［システムのプロパティ］ダイアログ ボックスが表示されます。
3. ［詳細設定］タブの［パフォーマンス］セクションで［設定］をクリックします。
   ［パフォーマンス オプション］ダイアログ ボックスが表示されます。
4. ［ウィンドウの下に影を表示する］チェックボックスをオンにします。



レジストリ エディタを使用してコンテキスト メニューの深さを設定する

1. レジストリ エディタを開きます。
2. HKEY_CURRENT_USER\Software\Microsoft\Windows\DWM に移動します。
3. ColorPrevalence" -Type DWORD の値を 1 に設定します。
4. WindowMaker を再起動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 8 グラフィック要素

このセクションには、以下の内容が含まれます。

WindowMaker オブジェクト

産業用グラフィック エディタの操作（IT）

WindowMaker での産業用グラフィックの使用

InTouch ウィンドウから産業用グラフィックへの変換

シンボル ウィザード エディタでのシンボル ウィザードの作成

トレンド ペンの履歴データ検索について

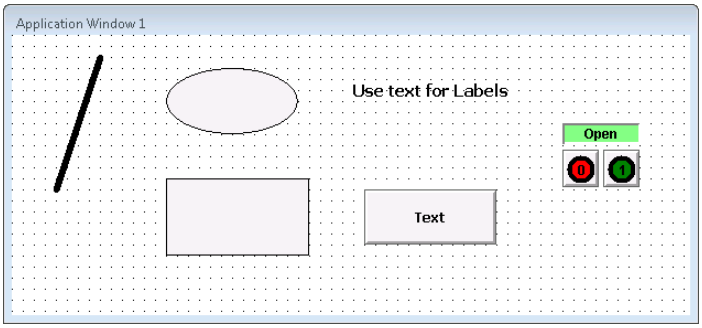
ランタイム時のトレンド ペン プロパティの変更

マルチ ペン トレンドを設定するには

WindowMaker オブジェクト

グラフィカル オブジェクトは、作成するヒューマン マシン インターフェイス（HMI）アプリケーションの重要な部分です。

アプリケーションを作成する場合、シンプル オブジェクトを作成し、シンプル オブジェクトを組み合わせて複合オブジェクトを作成して、定義済みの複合オブジェクトを使用します。



複合オブジェクトの個々の要素は、通常は次の目的のためにグループ化されています。

- 編集中に複合オブジェクトが分離しないようにするため。
- オブジェクト全体で複製するため。
- 一般的な一連のプロパティを個々の要素に割り当てるため。

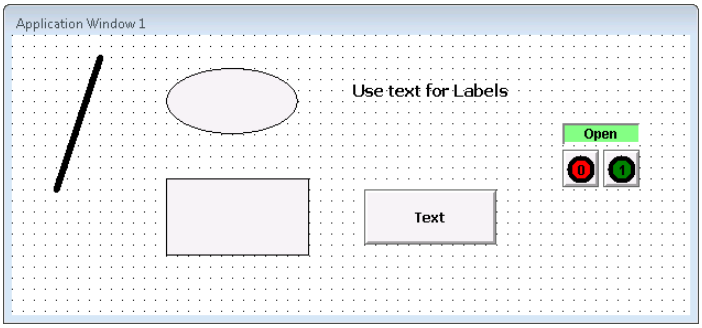
産業用グラフィック エディタで作成した産業用グラフィックは、マネージドまたは Advanced InTouch アプリケーションで使用できます。WindowMaker の産業用グラフィック ツールボックスから産業用グラフィックを直接追加することもできます。産業用シンボルと Situational Awareness Library シンボルの操作に関する詳細については、*『産業用グラフィック エディタ ユーザー ガイド』*または WindowMaker ヘルプを参照してください。

WindowMaker オブジェクト

グラフィカル オブジェクトは、作成するヒューマン マシン インターフェイス（HMI）アプリケーションの重要な部分です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーションを作成する場合、シンプル オブジェクトを作成し、シンプル オブジェクトを組み合わせて複合オブジェクトを作成して、定義済みの複合オブジェクトを使用します。



複合オブジェクトの個々の要素は、通常は次の目的のためにグループ化されています。

- 編集中に複合オブジェクトが分離しないようにするため。
- オブジェクト全体で複製するため。
- 一般的な一連のプロパティを個々の要素に割り当てるため。

産業用グラフィック エディタで作成した産業用グラフィックは、マネージドまたは Advanced InTouch アプリケーションで使用できます。WindowMaker の産業用グラフィック ツールボックスから産業用グラフィックを直接追加することもできます。産業用シンボルと Situational Awareness Library シンボルの操作に関する詳細については、『*産業用グラフィック エディタ ユーザーズ ガイド*』または WindowMaker ヘルプを参照してください。

シンプル オブジェクト

以下のタイプのシンプル オブジェクトを作成できます。

- 線
- 図形
- テキスト
- ボタン

各シンプル オブジェクトは表示方法を制御する属性を持っています。

- 線色および太さ
- 塗りつぶし色
- 高さ
- 幅
- 向き

線と図形の作成

以下の表には、基本描画タスクを実行する方法が示されています。描画ボタンは、［**描画**］メニューにあります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

描画内容	選択内容	ボタン
線	線ボタン	
水平または垂直線	H/V 線ボタン	
長方形	長方形ボタン	
角の丸い長方形	角丸長方形ボタン **注意：**角丸長方形の角の丸みを調整するには、角丸長方形の角丸みの変更を参照してください。	
円または楕円形	楕円形ボタン。円を描画するには Shift キーを押しながら描画します。	

ボタンの作成

ボタンを使用して、アプリケーションと対話するポイントを作成できます。このプロセスはシンプル描画オブジェクトを作成する過程に非常に類似しています。

ポリゴンの作成については、「折れ線と多角形の作成」を参照してください。

ボタンを作成する

1. ［描画］メニューの［挿入］グループで［ボタン］をクリックします。
2. 選択しドラッグして、ボタンを配置しサイズ設定します。
3. デフォルトのボタン テキストを編集します。以下の手順を実行します。
   a. ボタンを右クリックして、［文字列の変更］をクリックします。
   b. ［新しい文字列］ボックスに、ボタンのテキストを入力します。
   c. **[OK]** を選択します。

折れ線と多角形の作成

折れ線の描画は、線の描画とは少し異なります。

折れ線または多角形を作成する

1. ［描画］メニューの［図形］グループで［折れ線］または［多角形］をクリックします。
2. アプリケーション ウィンドウをクリックして、最初のポイントを設定します。
3. 再びアプリケーション ウィンドウをクリックして、折れ線または多角形を定義するポイントをさらに設定します。
4. 最後のポイントをダブルクリックします。

テキストを作成する

テキストを使用して、アプリケーションのビジュアル アイテムにラベルを付けることができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

テキストを作成するとき、テキストの書式設定は WindowMaker の設定画面で設定されている内容と一致します。選択したテキストの外観を変更できます。詳細については、「テキスト外観の変更」を参照してください。

複数行のテキストを入力すると、個別に移動および編集できるオブジェクトとなります。テキスト オブジェクトをシンボルに組み込んで、グループとして編集することもできます。

テキストを作成する

1. ［描画］メニューの［挿入］グループで［テキスト］をクリックします。
2. テキストの開始場所をクリックします。
3. テキストを入力して、Enter キーを押します。新しいテキスト行が表示されます。

複合オブジェクト

複合オブジェクトは、シンプル オブジェクトよりも多くの機能を提供します。以下の表には、複合オブジェクトのタイプが示されています。

複合オブジェクト	説明
セル	シンボルまたはその他のセルを含む 2 つまたはそれ以上のオブジェクト のグループが結合され、単一のユニットを形成しています。セルを使用すると、スライド コントローラなどの仮想デバイスを作成できます。 セルは、異なるタグに関連付けられる複数のデバイスを作成する場合に役立ちます。
シンボル	線、図形、およびテキストなどのシンプル オブジェクトのグループが結合されたもので、単一オブジェクトとして扱われます。シンボルに適用された属性の変更は、シンボルのすべてのコンポーネント オブジェクトに影響を与えます。シンボルには、ビットマップ、ボタン、セル、ウィザード、およびトレンドを含めることはできません。
SmartSymbol	再使用可能なグラフィック テンプレートに変換された InTouch セルです。1 つまたは複数の SmartSymbol テンプレートのインスタンスをアプリケーション ウィンドウに配置できます。テンプレートへの変更はすべてインスタンスに反映されます。詳細については、「SmartSymbol について」を参照してください。
産業用グラフィック	Integrated Development Environment（IDE）で産業用グラフィック エディタを使用して作成される高多様性グラフィックです。
ビットマップ コンテナ	写真、描画、およびスクリーン ショットなどの画像のインポートを可能にするオブジェクトです。 ビットマップを回転させたり、透過背景を付けたりすることができます。詳細については、「ビットマップ コンテナの操作」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

複合オブジェクト	説明
トレンド オブジェクト	時間経過による複数タグのリアルタイムまたは履歴データ値の変化を示すチャートです。詳細については、「トレンド オブジェクト」を参照してください。
ウィザード	選択、配置、そして設定するだけで使用できる事前に作成されたオブジェクトです。 詳細については、「ウィザード」を参照してください。
ActiveX コントロール	アプリケーション内で実行するソフトウェア コンポーネントです。WindowMaker は、AVEVA とサードパーティ製 ActiveX コントロールの両方をサポートしています。詳細については、「ActiveX コントロールを使用する」を参照してください。

セルとシンボル

複数のオブジェクトを組み合わせて、セルとシンボルと呼ばれる 2 種類の単体を作成できます。セルには任意のオブジェクトを含めることができます。シンボルにはシンプル オブジェクトのみを含めることができます。シンボルにセルを含めることはできません。

特定のオブジェクトがセルまたはシンボルであるかどうかを確認するには、そのオブジェクトをダブルクリックします。

- セルが開き、［**タグ変数の変更**］ダイアログ ボックス、またはセルにタグ変数が含まれていない場合は［タグ変数の変更］警告メッセージが表示されます。
- シンボルまたはシンプル グラフィック オブジェクトによって、［**アニメーション リンクの選択**］ダイアログ ボックスを開きます。

セルについて

セルを使用して、複数要素間の固定位置関係を組み合わせて、維持します。また、セルを使用して複数要素を移動したり、他のグラフィカル要素と整列させたりすることができます。

セルの要素を変更するには、セルを分解し、要素を変更して、要素を再びセルに組み合わせる必要があります。

セルの要素をアニメーション化することはできますが、セルをアニメーション化することはできません。セルのサイズを変更することもできません。

シンボルについて

シンボルおよびシンプル オブジェクトはアニメーション化できます。シンボルを使用して、複合グラフィックのパーツをアニメーション化することもできます。

選択したオブジェクトの 1 つ以上にリンクが設定されている場合は、シンボルを作成できません。

2 つのシンボルを組み合わせて新しいシンボルを作成すると、元のシンボル構造は失われます。新しいシンボルを分解すると、元の各シンボルの構成要素に分解されます。元の 2 つのシンボルは失われます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オブジェクトのセルへのグループ化

シンボル、ビットマップ、トレンド、ボタン、ウィザード、およびその他のセルを 1 つのセルに組み合わせることができます。セルにシンボルを含める場合、そのシンボルに関連付けられているすべてのアニメーション リンクはそのままの状態で残ります。

SmartSymbol でセルを使用した後で、SmartSymbol を分解した場合、セルのサイズを変更することはできません。

セルを作成する

1. 含めるオブジェクトを選択します。
2. ［**アニメーション**］メニューの［**セル**］グループで［**セルの作成**］をクリックします。

セルを分解する

1. セルを選択します。
2. ［**アニメーション**］メニューの［**セル**］グループで［**分解**］をクリックします。

オブジェクトのシンボルへのグループ化

シンボルにビットマップ、ボタン、セル、ウィザード、またはトレンドを含めることはできません。選択したオブジェクトの 1 つにアニメーション リンクが接続されている場合、リンクは新しいシンボルに接続されます。

シンボルの作成

1. 含めるオブジェクトを選択します。
2. ［**アニメーション**］メニューの［**グラフィック**］グループで［**グラフィックの作成**］をクリックします。

シンボルを分解する

1. シンボルを選択します。
2. ［**アニメーション**］メニューの［**グラフィック**］グループで［**分解**］をクリックします。

共通の操作

オブジェクトを右クリックすると、オブジェクトに適用できる有効なコマンドまたはアクションを表示するメニューが表示されます。以下の操作を実行できます。

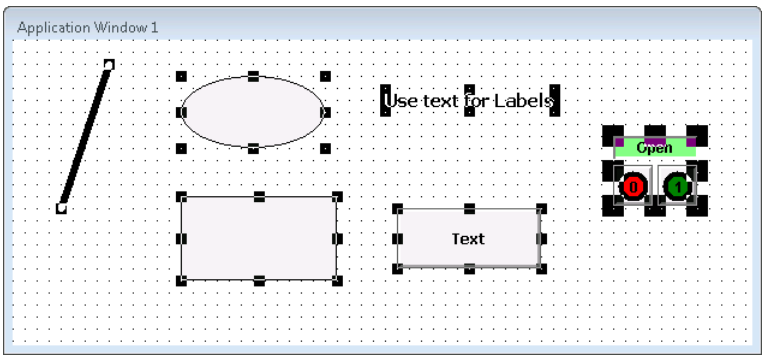
- オブジェクトを選択する
- オブジェクトを移動する
- オブジェクトを整列する
- オブジェクトを配置する
- オブジェクトを重ねる
- 変更を元に戻す
- オブジェクトを反転させる
- シンボルを反転させる
- オブジェクトのサイズを変更する
- オブジェクトを回転させる
- フォントを変更する
- 線または輪郭線を変更する
- 塗りつぶしを変更する
- オブジェクトを削除する

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 左右および上下の間隔を制御する

オブジェクトを選択する

オブジェクトを変更する前に、そのオブジェクトを選択する必要があります。オブジェクトを選択すると、オブジェクトの周囲にハンドルが表示されます。これらのハンドルを使用して、オブジェクトのサイズ変更や変形を行います。



アクティブ ウィンドウのオブジェクトをすべて選択する

- ［アニメーション］メニューの［オブジェクト］グループで［すべて選択］をクリックします。

  または、**F2** キーをクリックします。

オブジェクトを選択する

1. ［アニメーション］メニューの［モード］グループで［選択］をクリックします。

   選択モードが有効になります。
2. 選択するオブジェクトを強調表示します。

オブジェクトの選択を解除するには

- ウィンドウの空白領域をクリックします。

複数のオブジェクトの選択

1. ［アニメーション］メニューの［モード］グループで［選択］をクリックします。

   選択モードが有効になります。
2. 最初のオブジェクトを選択し、Shift キーを押しながら他のオブジェクトをクリックします。

オブジェクト グループを選択する

1. ［アニメーション］メニューの［モード］グループで［選択］をクリックします。

   選択モードが有効になります。
2. オブジェクトの周囲にボックスをドラッグします。長方形内のすべてのオブジェクトが選択されます。

特定のオブジェクトをオブジェクト グループから選択解除する

- Shift キーを押しながらオブジェクトをクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オブジェクトを移動する

オブジェクトは以下の方法で移動できます。

- ドラッグする
- キーボードの矢印キーを使用する
- ステータス バー内のボックスにウィンドウ座標を入力する

オブジェクトを移動するときには、ステータス バーの座標がどのように変化するかを確認してください。

オブジェクトをドラッグして移動する

- オブジェクトを選択して、ドラッグします。

矢印キーを使用してオブジェクトを移動する場合、オブジェクトの移動距離はグリッドが表示されているかどうかによって異なります。

グリッドが表示されている場合、オブジェクトの移動ピクセル数は WindowMaker の設定画面で設定したグリッド間隔に依存します。デフォルト設定のグリッド間隔は 10 ピクセルです。

グリッドが表示されている場合:

- 矢印キーを押すと、オブジェクトが 1 グリッド ポイント移動します。
- Shift キーと矢印キーを押すと、オブジェクトが 2 グリッド ポイント移動します。
- Ctrl キーと矢印キーを押すと、オブジェクトが 4 グリッド ポイント移動します。

グリッドが表示されていない場合:

- 矢印キーを押すと、オブジェクトが 1 ピクセル移動します。
- Shift キーと矢印キーを押すと、オブジェクトが 10 ピクセル移動します。
- Ctrl キーと矢印キーを押すと、オブジェクトが 50 ピクセル移動します。

矢印キーを使用してオブジェクトを移動する

- オブジェクトを選択して
  - 矢印キーを押します。
  - Shift キーを押しながら矢印キーを押します。
  - Ctrl キーを押しながら矢印キーを押します。

オブジェクトを整列する

オブジェクトを左端、右端、中心線、中心点、上端、中央、または下端に揃えることができます。

メニュー コマンドまたはボタンを使用して、複数の方法で整列させることができます。

選択	または選択	目的
左寄せ		オブジェクトの左端をオブジェクトの一番左端に揃えます。
中心線揃え		オブジェクトをグループの垂直中心線に揃えます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

選択	または選択	目的
右寄せ		オブジェクトの右端をオブジェクトの一番右端に揃えます。
上端揃え		上端を一番高いオブジェクトの上端と揃えます。
中央揃え		グループの中央に水平中心に揃えます。
下端揃え		下端を一番低いオブジェクトの下端と揃えます。
中心点揃え		中心点をグループの中心点に揃えます。

オブジェクトを整列する

1. 複数のオブジェクトを選択します。
2. ［ホーム］メニューの［整列］グループで適切な整列コマンドをクリックします。

オブジェクトを重ねる

オブジェクトは他のオブジェクトの前面または背面に配置できます。

オブジェクトを別のオブジェクトの背面に移動する

1. オブジェクトを選択します。
2. ［ホーム］メニューの［配置］グループで［最背面へ移動］をクリックします。

オブジェクトを別のオブジェクトの前面に移動する

1. オブジェクトを選択します。
2. ［ホーム］メニューの［配置］グループで［最前面へ移動］をクリックします。

   または、SHIFT+F9 キーを押します。

左右および上下の間隔を制御する

左右両端の選択オブジェクトを均等間隔に水平に配置することができます。

また、上下両端の選択オブジェクトの間隔も制御できます。

オブジェクトの間隔を指定して、中心に揃えることもできます。

オブジェクトを左右または上下に均等間隔で配置する

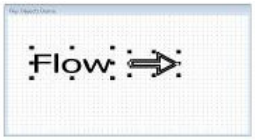
1. オブジェクトを選択します。
2. ［描画］メニューの［間隔］グループで［水平、垂直］または［中心点］をクリックします。

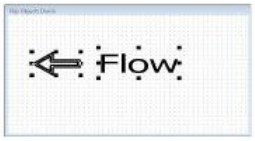
オブジェクトとセルの反転

ほとんどのオブジェクトは左右または上下に反転できます。オブジェクトは単一またはグループで反転できます。

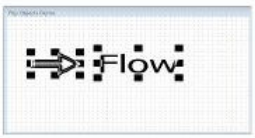
オブジェクトを反転すると、ミラー画像として変換され、表示されます。テキストは反転できません。

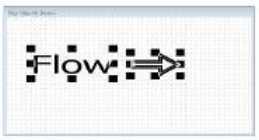
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.





セルを反転しても、ミラー イメージとしては表示されません。オブジェクト グループ内のセルの位置だけがミラー イメージとして表示されます。





反転前後のセルの位置を比較します。位置は反転していますが、内容はそのままです。

オブジェクトまたはセルを反転する

1. オブジェクトを選択します。

2. ［描画］メニューの［配置］グループで、［回転］の横にある下向きの矢印をクリックします。
3. ［左右反転］または［上下反転］をクリックします。

オブジェクトのサイズを変更する

オブジェクトは 2 つの方法でサイズ変更できます。ドラッグするか、正確な幅と高さを指定します。

グリッドに配置がオンになっている場合、オブジェクトは比例サイズ変更操作中にグリッドに配置されます。垂直から水平サイズ間の比率にわずかな偏差が生じます。この偏差を避けるには、グリッドに配置をオフにします。

オブジェクトをドラッグしてサイズ変更する

1. オブジェクトを選択して、矢印形のカーソルの先をハンドルの真ん中に合わせます。
2. ハンドルをドラッグして、オブジェクトのサイズを変更します。

オブジェクトを比例サイズ変更する

- オブジェクトを選択して、Shift キーを押しながらドラッグします。

オブジェクトを寸法でサイズ変更する

1. オブジェクトを選択します。
2. ［プロパティ］パネルに幅と高さの寸法を［**W, H**］ボックスにそれぞれ入力します。

オブジェクトを回転させる

シンボル、テキスト、およびビットマップを含むほとんどのオブジェクトを回転できます。セルは回転できません。

オブジェクトは時計回りまたは反時計回りに 90 度ずつ 360 度回転させることができます。

WindowViewer でのオブジェクトの回転は、ランタイム時にまたは WindowViewer でのオブジェクトの動的な回転とは関係ありません。WindowViewer でオブジェクトを回転させるには、オブジェクトを角度アニメーションにリンクします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オブジェクトを回転させる

1. オブジェクトを選択します。

2. ［描画］メニューの［配置］グループで、［回転］の横にある下向きの矢印をクリックします。
3. ［時計回り］または［反時計回り］をクリックします。

テキスト外観の変更

フォントの外観は、事前にデフォルトを設定することによって、テキストを作成する前に設定できます。また、テキストを作成した後で変更することもできます。デフォルトのフォントを設定する詳細については、「フォントのデフォルトを設定する」を参照してください。

テキストの揃え方の属性設定は、動的な値を示すテキスト オブジェクトに対して重要です。ランタイム時に長さの変化するフィールドがどのように表示されるかは、揃え方によって決まります。

たとえば、中央揃えか右揃えのテキスト文字列の最後に数値を表示している場合、表示される桁数が変更されるたびに、値を含む文字列全体が中央または右に揃え直されます。

メニュー コマンドまたはボタンを使用して、複数の方法でテキストを設定できます。

目的	選択	ボタン
フォント、スタイル、色、またはテキスト サイズを変更する	**フォント**	
テキストを太字にする	**太字**	
テキストを斜体にする	**斜体**	
テキストに下線を付ける	**下線**	
フォント サイズの縮小または拡大	**フォントの縮小**または**フォントの拡大**	
揃え方を変更する	**左揃え**、**中央揃え**、または**右揃え**	

テキストの外観を設定するには

1. テキスト オブジェクトを選択します。
2. ［描画］メニューの［形式］グループで適切なテキスト コマンドをクリックします。

線と輪郭線の変更

輪郭線付きオブジェクトの線または輪郭線の色、およびパターンまたは幅のいずれかを変更できます。輪郭線付きオブジェクトには、楕円形、長方形、および多角形などの塗りつぶし図形とビットマップおよびその他のインポートされた画像が含まれます。

線が太くなるほど、ランタイムでの描写にかかる時間が長くなります。破線および点線の太さは 1 ピクセル幅に固定されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

線の外観のデフォルトを設定する

1. ウィンドウの空白領域をクリックします。
2. ［描画］メニューの［形式］グループで線の幅またはパターンを選択します。
3. ［線の色］ツールをクリックして、色を選択します。

線の色を変更する

1. 線、線グループ、または輪郭線付きのオブジェクトを選択します。
2. ［描画］メニューの［形式］グループで［線の色］ツールをクリックします。
3. 色を選択します。

線または輪郭線のスタイルまたは幅を変更する

1. オブジェクトを選択します。
2. ［描画］メニューの［形式］グループで線のスタイルまたは幅をクリックします。

輪郭線を削除する

1. オブジェクトを選択します。
2. ［描画］メニューの［形式］グループで［線なし］をクリックします。

塗りつぶしを変更する

塗りつぶし図形には線で囲まれた図形が含まれます。塗りつぶし図形の例には、長方形、角丸長方形、円、楕円形、および多角形があります。

オブジェクトの塗りつぶし色を変更する

1. オブジェクトを選択します。
2. ［描画］メニューの［形式］グループで［塗りつぶしの色］ツールをクリックします。
3. 色を選択します。

塗りつぶし図形のデフォルト色を設定する

1. ウィンドウの空白領域をクリックします。
2. ［描画］メニューの［形式］グループで［塗りつぶしの色］ツールをクリックします。
3. 色を選択します。

オブジェクトを削除する

1 つまたは複数のオブジェクトを削除できます。

オブジェクトの削除

- 以下のいずれかを実行します。
  - オブジェクトを右クリックして、**[削除]**を選択します。
  - オブジェクトを選択して Delete キーを押します。

変更を元に戻す

WindowMaker は、各ウィンドウの編集および形式の変更を記録します。デフォルトでは、WindowMaker は元に戻す/やり直しの 10 レベルをサポートしています。1 レベルとは 1 回のアクションを表します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowMaker は最大で 25 レベルのアクションを保持するように設定できます。レベルを 0 に設定して、元に戻す/やり直しをオフにすることもできます。

ウィンドウを閉じると、記録されたすべてのアクションはすべて消去されます。

コマンドを元に戻す

- クイック アクセス ツールバーの［**元に戻す**］をクリックします。

コマンドをやり直しする

- クイック アクセス ツールバーの［**やり直す**］をクリックします。

元に戻す/やり直しするレベルの数を設定する

1. ［**ファイル**］メニューの［**設定**］をクリックし、［**WindowMaker**］をクリックします。

   WindowMaker の 設定画面が表示されます。

2. ［**元に戻すメニューのレベル**］ボックスに、レベル数を入力します。

すべてのオブジェクトの特殊操作

シンプル オブジェクトは変更したり、操作したりできます。以下の操作を実行できます。

- オブジェクトの切り取り、コピー、貼り付け
- オブジェクト リンクの切り取り、コピー、貼り付け
- オブジェクトの複製

オブジェクトの切り取り、コピー、貼り付け

WindowMaker での切り取り、コピー、および貼り付け操作は、他の Windows ベースのアプリケーションと同じですが、いくつか注意すべき相違点があります。

オブジェクトを切り取り、コピー、または貼り付けする際に、属性およびアニメーション リンクも切り取り、コピー、または貼り付けされます。

貼り付けられたオブジェクトはすべて、貼り付け後も選択されたままになるため、動かして位置を調整することができます。

オブジェクトの切り取り

- オブジェクトを右クリックし、［**切り取り**］をクリックします。

オブジェクトのコピー

- オブジェクトを右クリックし、［**コピー**］をクリックします。

オブジェクトの貼り付け

1. ウィンドウの空白領域を右クリックし、［**貼り付け**］をクリックします。カーソルがコーナー シンボルに変わります。
2. 左マウス ボタンを押します。カーソルがコピーしたオブジェクトと同じ大きさの点線の長方形に変わります。
3. 長方形をドラッグして、オブジェクトを配置します。
4. マウス ボタンを離します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オブジェクト リンクの切り取り、コピー、貼り付け

クリップボードは、切り取りまたはコピーしたリンクの一時保存領域です。

- クリップボードには、最後に行われた切り取りやコピー操作のリンクのみが保存されます。
- リンクは、クリップボードのリンクをサポートする任意のオブジェクトまたはシンボルに貼り付けることができます。
- たとえば、テキスト オブジェクトの線／輪郭色リンクなど、貼り付けられたリンクがオブジェクトでサポートされていない場合、リンクは貼り付けられません。
- 貼り付け用に複数オブジェクトを選択すると、すべてのオブジェクトにリンクが貼り付けられます。

リンクを切り取り、コピー、貼り付け、クリアする

- オブジェクトを右クリックし、［**リンク**］をポイントして、適切なコマンドをクリックします。

オブジェクトの複製

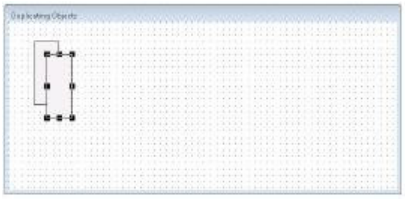
オブジェクトの複製はオブジェクトおよびそのアニメーション リンクのコピーと類似していますが、オブジェクトが 2 回以上複製されるとオフセット距離および方向も複製されるという利点があります。

複製オブジェクトを選択解除しないで移動し、再び複製すると、3 番目の複製は最初の 2 つの複製との間で同じ距離と方向のオフセットとなります。

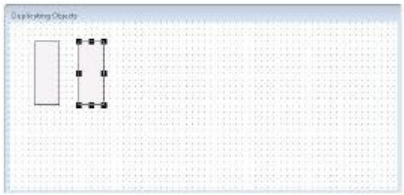
この操作は必要なだけ繰り返すことができます。

オブジェクトを複製する

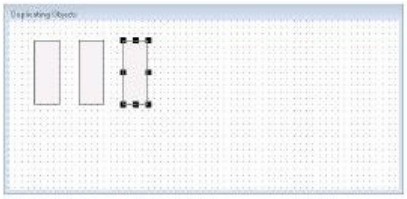
1. オブジェクトを右クリックして、**[複製]**を選択します。オブジェクトは元のオブジェクトからのオフセット位置にコピーされ、貼り付けられます。



2. 複製オブジェクトを選択したままにして、別の位置にドラッグします。



3. 複製オブジェクトを選択解除しないで、オブジェクトを右クリックして、［**複製**］をクリックします。3 回目のオブジェクト複製は最初の 2 つと同じ相対位置に表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

特殊オブジェクトの特殊操作

以下のオブジェクト タイプは、編集できる固有の属性を備えています。

- 折れ線と多角形
- ビットマップ コンテナ
- ビットマップ透明色
- 角丸長方形
- オブジェクト テキスト

折れ線および多角形オブジェクトの形状変更

折れ線および多角形の形状は調整できます。

折れ線または多角形の形状を変更する

1. オブジェクトを選択します。
2. 以下のいずれかを実行します。
   - ［**アニメーション**］メニューの［**オブジェクト**］グループで［**再形成**］をクリックして、すべての再形成オプションを表示します。
   - オブジェクトを右クリックし、［**オブジェクトの変形**］をクリックします。

   それぞれの図形定義点がハンドルとなります。
3. ハンドルをドラッグして形状を変更します。

多角形に頂点を追加する

1. オブジェクトを選択します。
2. 以下のいずれかを実行します。
   - ［**アニメーション**］メニューの［**オブジェクト**］グループで矢印をクリックして非表示のコマンドを表示し、［**頂点を追加**］をクリックします。
   - オブジェクトを右クリックし、［**頂点の追加**］をクリックします。
3. 多角形の端をクリックし、頂点をドラッグして多角形の形状を変更します。

多角形から頂点を削除する

1. オブジェクトを選択します。
2. 以下のいずれかを実行します。
   - ［**アニメーション**］メニューの［**オブジェクト**］グループで矢印をクリックして非表示のコマンドを表示し、［**頂点を削除**］をクリックします。
   - オブジェクトを右クリックし、［**ポイントを削除**］をクリックします。
3. 多角形の頂点をクリックすると、頂点が削除され、多角形の形状が変更されます。

ビットマップ コンテナの操作

ビットマップ コンテナは、ピクチャ、画面キャプチャ、および描画などのグラフィック オブジェクトのアプリケーションへのインポートを可能にするオブジェクトです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

サポートされるビットマップ コンテナ ファイル タイプは、.bmp、.jpeg、.jpg、.pcx および .tga です。

ビットマップをインポートすると、ビットマップ コンテナに自動的に挿入されますが、元のサイズおよび比率にサイズを変更できます。

ビットマップは 90 度ずつ回転させることができます。

ビットマップをセルに含めることはできますが、シンボルに含めることはできません。

WindowMaker を使用して、WindowViewer にロード可能なビットマップよりも多くのビットマップをウィンドウに配置することができます。多数のビットマップをウィンドウに配置する必要がある場合は、アプリケーションをリリースする前に WindowViewer でウィンドウをテストしてください。

ビットマップ画像をインポートする

1. ［**描画**］メニューの［**図形**］グループで［**ビットマップ**］をクリックします。

   カーソルが十字型に変わります。
2. カーソルをドラッグして、ビットマップ コンテナを描画します。
3. ［**アニメーション**］メニューの［**ビットマップ**］グループで［**インポート**］を選択します。

   ［**画像ファイルの選択**］ダイアログ ボックスが表示されます。
4. イメージのファイル名を選択して、［**OK**］をクリックします。

ビットマップを元のサイズにする

1. 画像を選択します。
2. ［**アニメーション**］メニューの［**ビットマップ**］グループで［**元のサイズ**］を選択します。

ビットマップ画像を貼り付ける

1. グラフィックを Windows のクリップボードにコピーします。
2. ［**ビットマップ**］ツールをクリックして、ビットマップ コンテナをウィンドウに描画します。
3. ［**アニメーション**］メニューの［**ビットマップ**］グループで［**貼り付け**］を選択します。

   または、ビットマップ コンテナを右クリックして、［**ビットマップの貼り付け**］をクリックします。

ビットマップを編集する

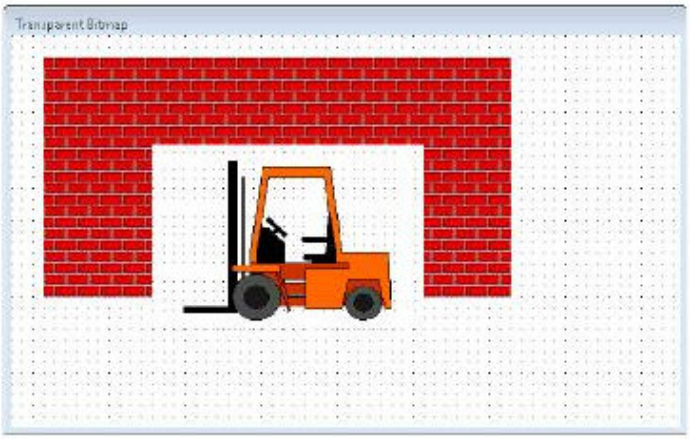
1. ビットマップを選択します。
2. ［**アニメーション**］タブの［**ビットマップ**］グループで［**編集**］をクリックします。

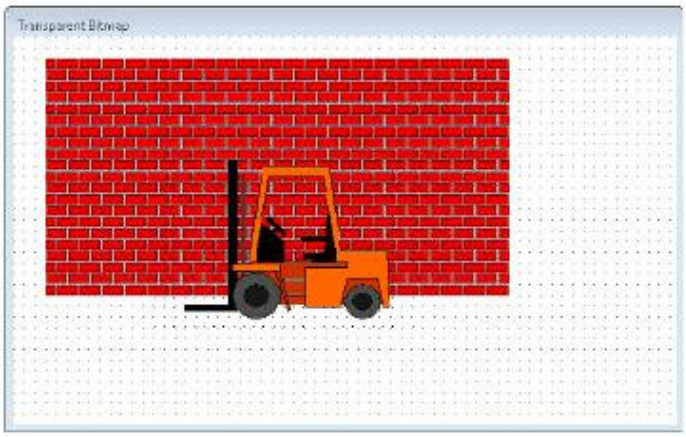
   Microsoft ペイントが開き、ビットマップが表示されます。
3. ペイントでビットマップを編集します。
4. 保存して、ペイントを閉じます。

ビットマップ透明色の定義

ビットマップに透明色を定義すると、ウィンドウの背景色やビットマップの背面にあるオブジェクトは、透明色を使用した部分すべてを通して表示されます。各イメージには、1 つの透明色だけを定義できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.





透明ビットマップを作成する

1. ビットマップが選択された状態で、［**描画**］メニューの［**形式**］グループにある［**透明色**］ボタンをクリックして、透明色パレットを開きます。
2. **カスタム パレット**の色付き四角形を右クリックします。［**カスタム カラーの編集**］ダイアログ ボックスが表示されます。
3. スポイト ツールをクリックします。
4. 透明にするビットマップ内の色をクリックします。選択した色がカラー パレットの選択した色の四角にコピーされます。
5. 色の四角をクリックして、透明色をビットマップに適用します。画像内のその色のすべてのピクセルが透明になります。

角丸長方形の角丸みの変更

角丸長方形の角丸みは拡大または縮小できます。

オブジェクトの角丸みを拡大または縮小する

1. オブジェクトを選択します。
2. ［**アニメーション**］メニューの［**オブジェクト**］グループで非表示のコマンドを表示し、［**半径を拡大**］または［**半径を縮小**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オブジェクト テキストの置換

シンボル、セル、またはボタンなどのテキストを持つオブジェクトのテキストを編集できます。

テキスト文字列を変更しても、元のすべての属性、フォント、スタイル、色などはすべて保持されます。テキスト書式設定も数値に適用できます。

オブジェクトのテキストを変更する

1. テキストを含むオブジェクトまたはボタンを選択します。以下のいずれかを実行します。
   - ［**アニメーション**］メニューの［**置換**］グループで［**文字列**］をクリックします。
   - テキスト オブジェクトを右クリックして、［**置換**］をポイントし、［**文字列の変更**］をクリックします。
2. ［**新しい文字列**］ボックスで、新しい文字列を入力して、［**OK**］をクリックします。

一連のテキスト オブジェクトの一部のテキストを変更する

1. テキスト オブジェクトをすべて選択します。
2. ［**アニメーション**］メニューの［**置換**］グループで［**文字列**］をクリックします。
3. **[置換] を選択します。**

   ［**文字列の置換**］ボックスが表示されます。
4. ［**旧文字列**］ボックスに、置換するテキストの一部を入力します。
5. ［**新文字列**］ボックスに、新しいテキスト文字列を入力します。
6. **[OK]** を選択します。

   選択したすべてのオブジェクトで新しいテキスト文字列が旧テキスト文字列に置き換わります。

WindowMaker オブジェクトに関するウィンドウ情報の印刷

アプリケーション ウィンドウに配置した InTouch グラフィカル オブジェクトに関する情報を印刷できます。InTouch グラフィカル オブジェクトには、線、ボタン、およびテキストなどのシンプルなオブジェクトや、セル、ActiveX コントロール、SmartSymbol、および産業用グラフィックなどの複雑なオブジェクトが含まれています。

グラフィカル オブジェクトの詳細は、プリンタまたは .html ファイルに出力できます。各ウィンドウに対して、.html ファイルが作成されます。.html ファイルには、以下のものが含まれています。

- 参照されている .png ファイルとしてのウィンドウのピクチャ。
- ウィンドウ内の各グラフィカル オブジェクトに関する詳細の一覧。

複数のウィンドウについてのグラフィカル オブジェクトの詳細を印刷した場合、ウィンドウ固有の .html ファイルにリンクしているサマリ .html ファイルが作成されます。

WindowMaker オブジェクトに関する情報の印刷

1. WindowMaker で InTouch アプリケーションを開きます。
2. クイック アクセス ツールバーの［**印刷**］をクリックします。

   ［**アプリケーション情報の印刷**］ダイアログ ボックスが表示されます。
3. ［**ウィンドウ**］のチェックボックスをオンにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. 印刷するウィンドウを指定します。
   - ［全て］を選択すると、アプリケーションにあるすべてのウィンドウのイメージ/グラフィックが印刷されます。

   注意：ウィンドウに多くのグラフィックが含まれる場合、または WindowMaker をホストするコンピュータで使用可能なメモリを超えるサイズの大きいグラフィックが含まれる場合、アプリケーションのすべてのウィンドウを XPS ファイルに出力すると WindowMaker が応答しなくなります。そのような場合は、メモリ不足による問題を回避するために一度に 1 つのウィンドウを出力してください。

   - ［選択］を選ぶと、特定のウィンドウのイメージ/グラフィックのみが印刷されます。［**ウィンドウの印刷**］ダイアログ ボックスが開きます。印刷したいウィンドウを選択し、［**OK**］をクリックします。
   - ［バッチ］を選択すると、.csv ファイルで指定したウィンドウの情報のみが印刷されます。

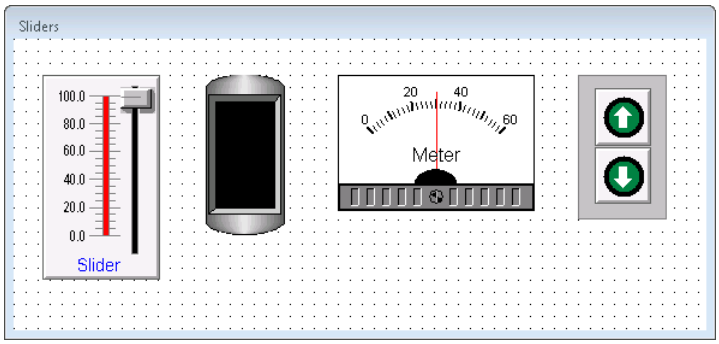
   .csv 形式の詳細については、「ウィンドウを印刷するための .CSV 形式」を参照してください。
5. ［**イメージ/グラフィック**］チェック ボックスをオンにします。
6. **[次へ]** を選択します。［**出力の送信先の選択**］ダイアログ ボックスが表示されます。
7. 以下のいずれかを実行します。
   - 情報を印刷するには、［**プリンタに出力を送信する**］をクリックします。
   - 指定した各ウィンドウに対して、概要の .html ファイル、単一の .html ファイル、および .png ファイルを作成するには、［**HTML ファイルに出力を送信する**］をクリックします。.html ファイルおよび .png ファイルが存在する場合、自動的に上書きされます。
8. **[印刷]** を選択します。

オブジェクトのアニメーション化

オブジェクトまたはシンボルはアニメーション リンクによってアニメーション化できます。アニメーション リンクはタグまたは式の値をオブジェクトまたはシンボルに結合できます。

たとえば、以下の操作を実行できます。

- タンクの液体レベルを示すスライダーまたはタンク シンボルを作成する
- 値の範囲を表示するメーターを作成する
- オペレータ コントロール用のタッチ スクリーン シンボルを作成する



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アニメーション リンクの 2 つのタイプ

アニメーション リンクには、表示リンクおよびタッチ リンクの 2 つの基本タイプがあります。

- 表示リンクはオペレータに情報を表示します。表示リンクの例は、色の変更、塗りつぶしレベルの変更、水平または垂直移動、オブジェクトの点滅などがあります。
- タッチ リンクは、システムへのオペレータ入力を許可します。タッチ リンクの例には、オペレータ入力に応答するスライダまたは押しボタンなどがあります。

オブジェクトまたはシンボルには、複数のリンクを定義できます。さまざまなリンクを組み合わせることによって、ほとんどの画面アニメーション効果を作成できます。

データ表示アニメーション

データ表示アニメーションは、オペレータに情報のみを表示します。これらのアニメーションはオペレータ入力をサポートしません。

値表示の作成

値表示テキスト オブジェクトを使用して、タグ変数の値を表示します。これにより、塗りつぶしレベル、オン／オフ ステータス、またはアラーム メッセージを表示できます。

3 つのタイプの*値表示リンク*のいずれか 1 つを使用して、ランタイムにメッセージを表示できます。

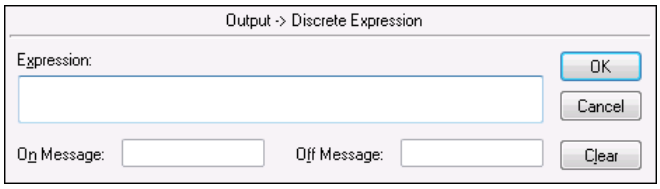
値表示タイプ	内容
論理型	オンまたはオフなどの論理値
アナログ型	塗りつぶしレベルまたは速度などのアナログ式の値
文字列	"Fill Level = 100" などの文字列式の値

式には最大 1023 文字までを使用できます。これより長い式が必要な場合、クイック関数を作成して式を呼び出します。

メッセージは、そのオブジェクトに設定されているフォント、サイズ、色、配置、およびリンクされている属性を使用して、元のテキスト オブジェクトの位置に表示されます。フィールドの元の内容は、ランタイムのメッセージには影響されません。

論理値表示リンクを作成する

1. テキスト オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。［**アニメーション リンク**］ダイアログ ボックスが表示されます。
2. ［**値の表示**］領域で、［**論理値**］をクリックします。 ［**出力 -> 論理値表示**］ダイアログ ボックスが表示されます。



3. ［**式**］ボックスに、論理型タグの名前か論理値に等しい式を入力します。以下に例を示します。

```
Cooling_Pump
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. ［オン メッセージ］ボックスに、式の値が 1、true、on、または yes のときに表示するメッセージを入力します。以下に例を示します。

```
ポンプがオン
```

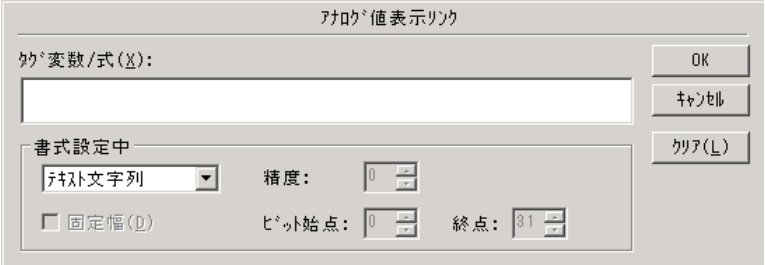
5. ［オフ メッセージ］ボックスに、式の値が 0、false、off、または no のときに表示するメッセージを入力します。例：

```
ポンプがオフ
```

6. **[OK]** を選択します。

アナログ値表示リンクを作成する

1. テキスト オブジェクトを右クリックして、［アニメーション リンク］を選択します。［アニメーション リンク］ダイアログ ボックスが表示されます。
2. ［値の表示］領域で、［アナログ］をクリックします。［アナログ値出力］ダイアログ ボックスが表示されます。



3. ［タグ変数／式］ボックスに、アナログ型（整数型または実数型）タグ変数またはアナログ値に等しい式を入力します。以下に例を示します。

```
Tank_CV*0.06
```

4. ［書式設定］領域のリストで、ランタイム時の詳細な書式設定を行う、それぞれのデータ タイプをクリックします。選択したデータ タイプに基づいて、［固定幅］チェック ボックス、［精度］ボックス、および［ビット数から］および［まで］ボックスが有効になります。これらのオプションの設定の詳細については、「テキストの詳細な書式設定」セクションを参照してください。

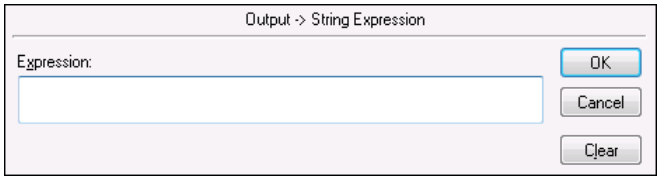
注意：ランタイム時、アナログ値入力リンク フィールドのサイズは、クリックしてポインタとマウスでドラッグすることによって変更できます。

---

5. **[OK]** を選択します。

文字列表示リンクを作成する

1. テキスト オブジェクトを右クリックして、［アニメーション リンク］を選択します。［アニメーション リンク］ダイアログ ボックスが表示されます。
2. ［値の表示］領域で、［文字列］をクリックします。［文字列表示リンク］ダイアログ ボックスが表示されます。



3. ［タグ変数／式］ボックスに、メッセージ型タグ変数またはメッセージ型タグ変数に等しい式を入力します。以下に例を示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
"The Tank Level is:" + Text(TankLevel,"#")
```

4. **[OK]** を選択します。

注意：ランタイム時、文字列値入力リンク フィールドのサイズは、クリックしてポインタとマウスでドラッグすることによって変更できます。

移動の作成

ランタイム時のオブジェクトの移動は、位置リンクを使用することによって実行できます。オブジェクトを水平、垂直、またはその両方向に、アナログ型タグの値または式の変化に合わせて移動できます。たとえば、タンク レベルが増減すると、インジケータが上下に移動します。

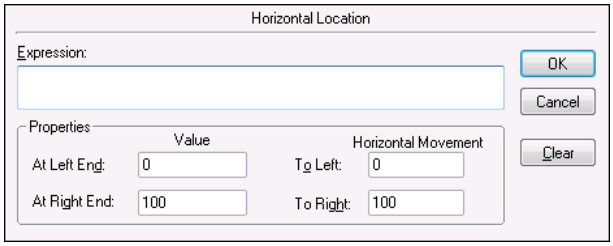
水平移動を作成する

1. オブジェクトを画面上の開始位置に配置します。
2. ［**描画**］メニューの［**モード**］グループで［**アニメーション化**］をクリックします。

   または、オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ダイアログ ボックスが表示されます。
3. ［**位置**］セクションで［**水平**］をクリックします。

   ［**水平位置変化リンク**］ダイアログ ボックスが表示されます。



4. ［**式**］ボックスに、アナログ型タグの名前、またはアナログ値に等しい式を入力します。
5. ［**プロパティ**］セクションでオブジェクトの移動距離を設定します。以下の手順を実行します。
   a. ［**左端の値**］ボックスに、オブジェクトが左端に位置する場合のアナログ型タグの値を入力します。
   b. ［**右端の値**］ボックスに、オブジェクトが右端に位置する場合のアナログ型タグの値を入力します。
   c. ［**左への移動幅**］ボックスに、オブジェクトを開始位置から左に動かすことができる幅をピクセル数で入力します。
   d. ［**右への移動幅**］ボックスに、オブジェクトを開始位置から右に動かすことができる幅をピクセル数で入力します。
6. **[OK]** を選択します。

垂直移動を作成する

1. オブジェクトを画面上の開始位置に配置します。
2. ［**描画**］メニューの［**モード**］グループで［**アニメーション化**］をクリックします。

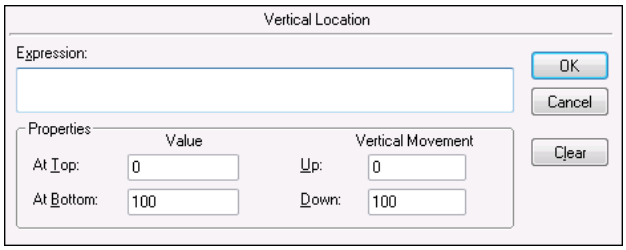
   または、オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［位置］セクションで［垂直］をクリックします。

   ［垂直位置変化リンク］ダイアログ ボックスが表示されます。



4. ［式］ボックスに、アナログ型タグの名前、またはアナログ値に等しい式を入力します。
5. ［プロパティ］セクションで以下の手順を実行します。
   a. ［上端の値］ボックスに、オブジェクトが上端に位置する場合のアナログ型タグの値を入力します。
   b. ［下端の値］ボックスに、オブジェクトが下端に位置する場合のアナログ型タグの値を入力します。
   c. ［上への移動幅］ボックスに、オブジェクトを開始位置から上に動かすことができる幅をピクセル数で入力します。
   d. ［下への移動幅］ボックスに、オブジェクトを開始位置から下に動かすことができる幅をピクセル数で入力します。
6. **[OK]** を選択します。

回転の作成

向きリンクを使用して、オブジェクトを中心点の周囲で、アナログ型タグの値の変化に合わせて移動するようにできます。たとえば、圧力が増加または低下するにつれてポインタがダイヤルの周囲を移動します。

向きリンクは、オブジェクトやシンボルの中心を回転のデフォルト中心として使用します。回転の中心をオフセットすることができます。

産業用グラフィックでは向きリンクはサポートされていません。

ヒント：オブジェクトの中心から回転中心点まで一時的な長方形を描画します。X および Y オフセットの寸法をステータス バーの［W, H］ボックスで読み取ることができます。

向きリンクを作成する

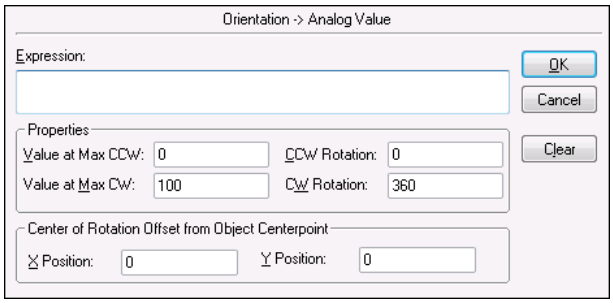
1. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ダイアログ ボックスが表示されます。
2. ［その他］セクションで［向き］をクリックします。

   ［向き -> アナログ値］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［式］ボックスに、アナログ型タグの名前、またはアナログ値に等しい式を入力します。
4. ［プロパティ］セクションで以下の手順を実行します。
   a. ［左回り最大値］ボックスに、オブジェクトを反時計回りの最大角度まで回転させる式の値を入力します。
   b. ［右回り最大値］ボックスに、オブジェクトを時計回りの最大角度まで回転させる式の値を入力します。
   c. ［左回り角度］ボックスに、［左回り最大値］に入力された式の値に対して、オブジェクトが反時計回りに回転する角度を入力します。
   d. ［右回り角度］ボックスに、［右回り最大値］に入力された式の値に対して、オブジェクトが時計回りに回転する角度を入力します。
5. ［回転中心設定（オブジェクトの中心点からのオフセット値）］セクションで以下の手順を実行します。
   a. ［水平方向］ボックスに、回転中心点の水平方向のオフセットを入力します。オブジェクトの中心点からのオフセットをピクセルで入力します。
   b. ［垂直方向］ボックスに、回転中心点の垂直方向のオフセットを入力します。オブジェクトの中心点からのオフセットをピクセルで入力します。
6. **[OK]** を選択します。

サイズのアニメーション化

オブジェクト サイズ リンクを使用して、アナログ型タグや式の値に応じてオブジェクトの高さや幅を変更できます。

たとえば、圧力インジケータは圧力が増加すると大きくなり、コンベヤ上のオブジェクトは大きくなることによってビューアに向かって移動します。

オブジェクト サイズ リンクはオブジェクトのサイズを制御するだけでなく、オブジェクトがアニメーションの基準点を使用してサイズを変更する方向を制御します。

オブジェクト サイズ高さリンクを作成する

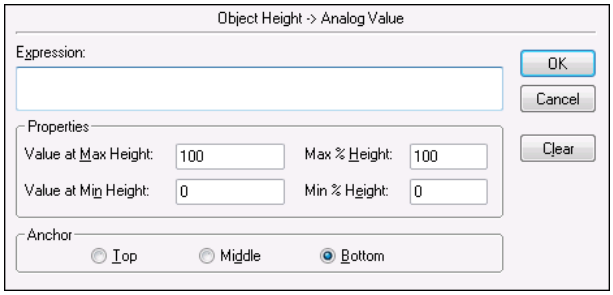
1. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ダイアログ ボックスが表示されます。
2. ［オブジェクト サイズ］セクションで［高さ］をクリックします。

   ［オブジェクトの高さ -> アナログ値］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［式］ボックスに、アナログ型タグの名前、またはアナログ値に等しい式を入力します。
4. ［プロパティ］セクションで以下の手順を実行します。
   a. ［最大高さの値］ボックスに、オブジェクトが最大の高さに達したときのタグまたは式の値を入力します。
   b. ［最小高さの値］ボックスに、オブジェクトが最小の高さに達したときのタグまたは式の値を入力します。
   c. ［最大高さ %］ボックスに、タグ名または式の値が［最大高さの値］ボックスで設定した値に達したときのオブジェクトの高さを元の高さのパーセントで入力します。パーセント値は、オブジェクトの描画サイズの割合で表します。描画サイズは常に 100% サイズです。
   d. ［最小高さ %］ボックスに、タグ名または式の値が［最小高さの値］ボックスで設定した値に達したときのオブジェクトの高さを元の高さのパーセントで入力します。パーセント値は、オブジェクトの描画サイズの割合で表します。描画サイズは常に 100% サイズです。
5. オブジェクトを拡大する基点となる**変化基準点**ポイントを選択します。
   - ［上端］を選択すると、オブジェクトは上端を基点に下方に拡大されます。
   - ［中心点］を選択すると、オブジェクトは中心点を基点に上下両方向に拡大されます。
   - ［下端］を選択すると、オブジェクトは下端を基点に上方に拡大されます。
6. **[OK]** を選択します。

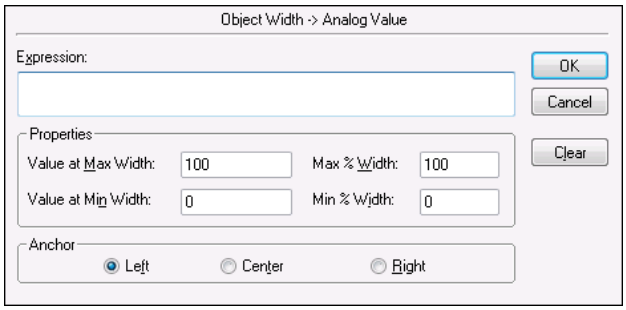
オブジェクト サイズ幅リンクを作成する

1. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ダイアログ ボックスが表示されます。
2. ［オブジェクト サイズ］セクションで［幅］をクリックします。

   ［オブジェクトの幅 -> アナログ値］ダイアログ ボックスが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［**式**］ボックスに、アナログ型タグの名前、またはアナログ値に等しい式を入力します。
4. ［**プロパティ**］セクションで以下の手順を実行します。
   a. ［**最大幅の値**］ボックスに、オブジェクトが最大の幅に達したときのタグまたは式の値を入力します。
   b. ［**最小幅の値**］ボックスに、オブジェクトが最小の幅に達したときのタグまたは式の値を入力します。
   c. ［**最大幅 %**］ボックスに、タグ名または式の値が［**最大幅の値**］ボックスで設定した値に達したときのオブジェクトの幅を元の幅のパーセントで入力します。パーセント値は、オブジェクトの描画サイズの割合で表します。描画サイズは常に 100% サイズです。
   d. ［**最小幅 %**］ボックスに、タグ名または式の値が［**最小幅の値**］ボックスで設定した値に達したときのオブジェクトの幅を元の幅のパーセントで入力します。パーセント値は、オブジェクトの描画サイズの割合で表します。描画サイズは常に 100% サイズです。
5. オブジェクトの幅を拡大する基点となる**変化基準点**ポイントを選択します。
   - ［**左端**］を選択すると、オブジェクトは左端を基点に拡大されます。
   - ［**中心点**］を選択すると、オブジェクトは中心点を基点に上下両方向に拡大されます。
   - ［**右端**］を選択すると、オブジェクトは右端を基点に拡大されます。
6. **[OK]** を選択します。

色のアニメーション化

色リンクを使用して、任意のオブジェクトに対して色の変更をアニメーション化できます。変化はアナログ型または論理型タグの値、アナログ型または論理型式の値、または論理型またはアナログ型アラーム ステータスに基づきます。

3 種類の色リンクを使用して、オブジェクトをアニメーション化できます。

- 線/輪郭色
- 塗りつぶしの色
- テキスト色

これら 3 つの色リンクの種類に対して、4 種類の式で色の変化を制御できます。

式タイプ	色を変更する基準
論理型	論理型タグまたは式の値
アナログ型	アナログ型タグまたは式の値。異なる値を表すために 10 色を定義できます。
論理型アラーム	タグ、アラーム グループ、またはグループ変数のアラーム状態
アナログ型アラーム	アナログ型タグ、アラーム グループ、またはグループ変数のアラーム状態。5 つのアラーム条件を表すために 5 色を定義できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

警告!リンクが InTouch バージョン 7.11 より前に作成された未変換アプリケーションからリモート タグに設定されている場合、アナログ型アラーム リンクを使用しているときにはオブジェクトはアラーム状態にはなりません。

すべての論理値色リンクは同じ方法で作成されます。以下の手順では、塗りつぶし色リンクの作成について説明します。

塗りつぶし色論理値リンクを作成する

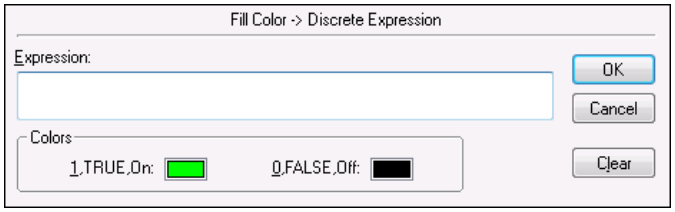
1. ［**描画**］メニューの［**モード**］グループで［**アニメーション化**］をクリックします。

   または、オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ダイアログ ボックスが表示されます。

2. ［**塗りつぶしの色**］セクションで［**論理型**］をクリックします。

   ［**塗りつぶしの色 -> 論理型式**］ダイアログ ボックスが表示されます。



3. ［**式**］ボックスに、True または False に等しい論理型タグまたは論理型式の名前を入力します。

   論理型式には、アナログ型タグを含めることができます。たとえば、TankLevel >= 75 の場合、変数 "TankLevel" の値が 75 以上の場合にオブジェクトの塗りつぶし色が変化します。

4. ［**色**］セクションでカラー ボックスをクリックして、カラー パレットを開きます。各状態に使用する色を選択します。

5. **[OK]** を選択します。

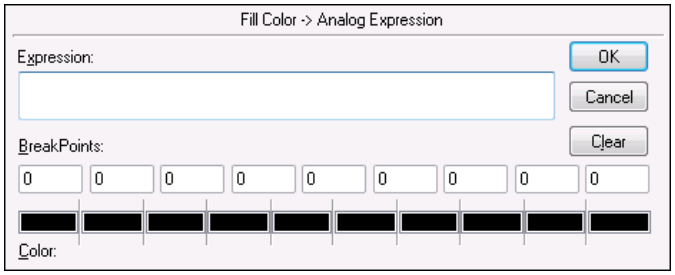
アナログ型式の色リンクを作成する

1. オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ダイアログ ボックスが表示されます。

2. ［**塗りつぶしの色**］セクションで［**アナログ型**］をクリックします。

   ［**塗りつぶしの色 -> アナログ型式**］ダイアログ ボックスが表示されます。



3. ［**式**］ボックスに、アナログ型タグの名前、またはアナログ値に等しい式を入力します。

4. ［**変化境界値**］セクションで以下の手順を実行します。

   - オブジェクトの色が変化するブレークポイント値を指定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ヒント：10 の異なる色を使用する必要はありません。たとえば、オブジェクトの色を 3 回だけ変化させる場合、3 種類の値を入力して残りの値に同じ色を使用します。より多様な範囲が必要な場合は、産業用グラフィックのアナログ塗りつぶし機能を参照してください。

- ［色］セクションで各ブレークポイント値の色を選択します。

5. **[OK]** を選択します。

論理型アラーム ステータスの色リンクを作成する

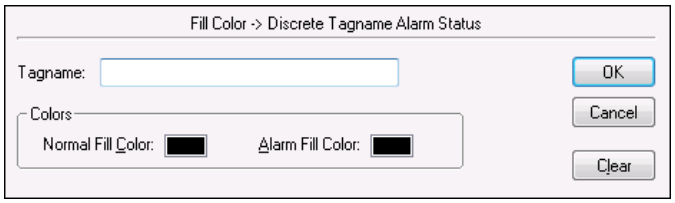
1. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ダイアログ ボックスが表示されます。

2. ［塗りつぶしの色］セクションで［論理型アラーム］をクリックします。

   ［塗りつぶしの色 -> 論理型タグ名アラーム ステータス］ダイアログ ボックスが表示されます。



3. ［タグ］ボックスに、オブジェクトに関連付けられている論理型タグの名前を入力します。
4. ［色］セクションで各アラーム状態の色を選択します。
5. **[OK]** を選択します。

アナログ型アラーム ステータスの色リンクを作成する

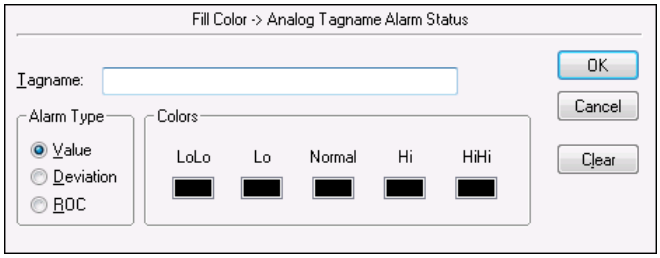
1. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ダイアログ ボックスが表示されます。

2. ［塗りつぶしの色］セクションで［アナログ型アラーム］をクリックします。

   ［塗りつぶしの色 -> アナログ型タグ名アラーム ステータス］ダイアログ ボックスが表示されます。



3. ［タグ］ボックスに、オブジェクトに関連付けられているアナログ型タグの名前を入力します。
4. ［アラーム タイプ］セクションで、オブジェクトに関連付けるアラームの 3 つのタイプから 1 つを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム タイプ	最大使用数
値	値アラームの状態を表す 5 色
偏差	偏差アラームの状態を表す 3 色
変化率	変化率アラームの状態を表す 2 色

5. ［色］セクションで各アラーム状態の色を選択します。
6. **[OK]** を選択します。

塗りつぶしレベルのアニメーション化

塗りつぶしパーセント リンクを使用して、オブジェクトの塗りつぶしレベルを変更できます。塗りつぶしパーセントは、アナログ型タグまたは式の値に基づきます。水平塗りつぶし、垂直塗りつぶし、または両方を作成できます。

たとえば、垂直塗りつぶしリンクを使用してタンク内の液体のレベルを表したり、水平塗りつぶしリンクを使用してプロセスの進捗状況を表すことができます。

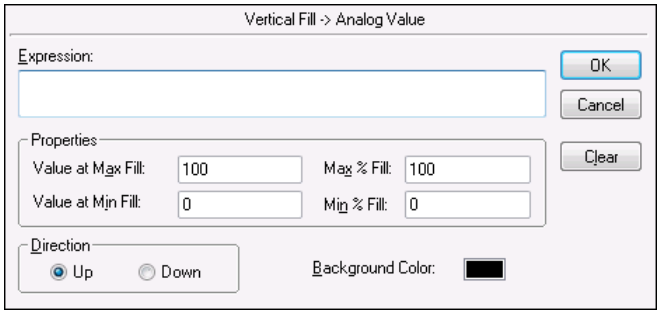
水平塗りつぶしパーセント リンクと垂直塗りつぶしパーセント リンクは同じ方法で作成します。

塗りつぶしパーセント リンクを作成する

1. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

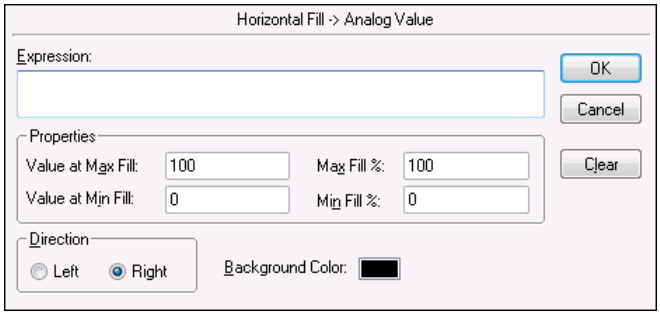
   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ダイアログ ボックスが表示されます。
2. ［塗りつぶしパーセント］セクションで以下のいずれかを実行します。
   - ［垂直］をクリックすると、［垂直塗りつぶし -> アナログ値］ダイアログ ボックスが表示されます。



   - ［水平］をクリックすると、［水平塗りつぶし -> アナログ値］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［式］ボックスに、アナログ型タグの名前、またはアナログ値に等しい式を入力します。
4. ［プロパティ］セクションで以下の手順を実行します。
   a. ［最大塗りつぶしの値］ボックスに、オブジェクトが最大塗りつぶしレベルとなる式の値を入力します。
   b. ［最小塗りつぶしの値］ボックスに、オブジェクトが最小塗りつぶしレベルとなる式の値を入力します。
   c. ［最大塗りつぶし %］ボックスには、式の値が［最大塗りつぶしの値］ボックスで設定したレベルに達したときにオブジェクトが塗りつぶされる値を 0 ～ 100 パーセントで入力します。
   d. ［最小塗りつぶし %］ボックスには、式の値が［最小塗りつぶしの値］ボックスで設定したレベルに達したときにオブジェクトが塗りつぶされる値を 0 ～ 100 パーセントで入力します。
5. ［方向］セクションで塗りつぶしの方向を選択します。
6. ［背景色］ボックスに、オブジェクトが塗りつぶされない部分の色を選択します。
   - 実際の塗りつぶし色は、オブジェクトを描画する場合にオブジェクトに対して選択する色です。
   - 垂直塗りつぶしパーセント リンクと水平塗りつぶしパーセント リンクの両方を同じオブジェクトにリンクした場合、最後に選択した色が背景色となります。
7. **[OK]** を選択します。

オブジェクトを点滅させる

点滅リンクを使用して、タグの値に基づいて点滅するアニメーション オブジェクトを作成できます。たとえば、特定の装置がオンになったとき、またはアラーム設定ポイントに達したときに赤く点滅するオブジェクトを作成できます。

点滅アニメーションの同期は、開いているすべてのウィンドウのアクティブな点滅アニメーションすべてを、共通の点滅速度（低速、中速、または高速）でいっせいに点滅させることができます。

ヒント： 論理型式には、アナログ型タグ名を含めることができます。たとえば、TankLevel > 75 の場合、TankLevel タグの値が 75 より大きい場合にオブジェクトの点滅が開始します。

点滅リンクを作成する

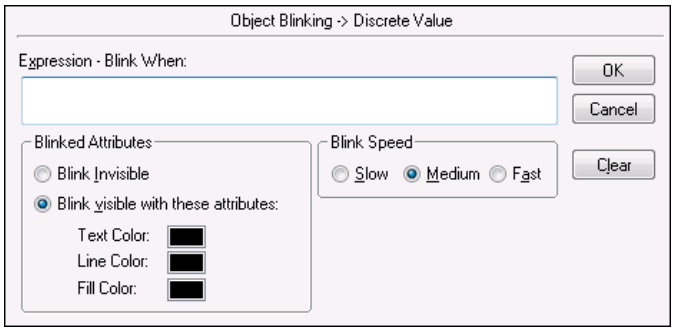
1. アニメーションを適用するオブジェクトを選択します。
2. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。
3. ［その他］セクションで［点滅］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［オブジェクト点滅 -> 論理値］ダイアログ ボックスが表示されます。



4. ［タグ/式 - 点滅条件］ボックスに、論理型タグまたは論理値に等しい式の名前を入力します。
5. ［オブジェクト点滅プロパティ］セクションで以下の手順を実行します。
   - ［目に見えない点滅］をクリックして、ウィンドウ内で表示のオン／オフを切り替えてオブジェクトを点滅させます。
   - ［これらの属性を備えた目に見える点滅］をクリックしてオブジェクトが表示されたままになるようにし、アクティブになったときに色を変更するように設定します。
   - ［テキストの色］、［線の色］、または［塗りつぶしの色］ボックスをクリックして、オブジェクトのパーツの色を選択します。カラー パレットが表示されます。

   注意： オブジェクトの塗りつぶし色と塗りつぶしの点滅色を同じにすると、オブジェクトは点滅しているように見えません。
6. ［点滅速度］セクションで、オブジェクトの点滅速度を設定します。［遅い］、［中間］、または［速い］のいずれかをクリックします。
7. **[OK]** を選択します。

WindowMaker の点滅速度を設定する

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］をクリックし、［**WindowViewer**］をクリックします。

   WindowViewer の設定画面が表示されます。
3. ［環境設定］タブの［点滅速度（ミリ秒）］セクションに、3 つの速度で使用する時間をミリ秒で入力します。

   注意：点滅速度の設定の変更はグローバル設定で、アプリケーションのすべての点滅速度に影響します。
4. **[OK]** を選択します。

表示オン/オフの有効化

表示オン/オフ リンクを使用して、さまざまなタグの値に基づいてオブジェクトを非表示にするリンクを作成できます。表示オン/オフ リンクを使用すると、以下の操作を行えます。

- 間違った方向に移動したときに非表示にすることによって、移動オブジェクトが一方向にのみ移動するという印象を作成します。
- 移動オブジェクトが停止したという印象を作成します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アラームやエラー メッセージなどのオブジェクトは、アクティブになっているときにのみ表示されるようにします。

ヒント：論理型式には、
TankLevel >= 75 などアナログ型タグ変数を含めることができます。この例では、タグ TankLevel の値が 75 以上の場合に、オブジェクトがウィンドウで表示されるようになります。

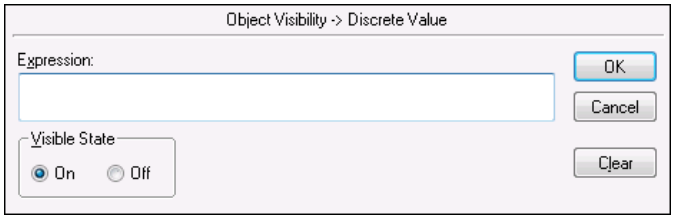
表示オフ/オン リンクを作成する

1. アニメーションを適用するオブジェクトを選択します。
2. ［描画］メニューの［モード］グループで［アニメーション化］をクリックします。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。
3. ［その他］セクションで［表示オン/オフ］をクリックします。

   ［オブジェクト表示オフ/オン -> 論理値］ダイアログ ボックスが表示されます。



4. ［式］ボックスに、論理型タグの名前か論理値に等しい式を入力します。
5. オブジェクトの［表示状態（値が真 1 のとき）］を選択します。［オフ］を選択すると、式の値が True の場合オブジェクトは表示されません。［オン］を選択すると、式の値が True の場合にオブジェクトが表示されます。
6. **[OK]** を選択します。

オブジェクトの無効化

タッチ アクション無効化リンクを持つアプリケーションにセキュリティ レベルを設定できます。たとえば、オペレータのアクセス レベルや名前に基づいて、タッチ オブジェクトを無効にできます。または、誰もログ オンしていない場合にボタンが間違って操作されないように設定できます。

無効化状態がオンの場合、式の値が True である限り、オブジェクトやボタンのタッチ機能がオフとなり、アクティブにはなりません。

ヒント: 論理型式には、アナログ型タグ名を含めることができます。たとえば、TankLevel >= 75 の場合、変数 "TankLevel" の値が 75 以上の場合にオブジェクトが無効になります。

無効化リンクを作成するには

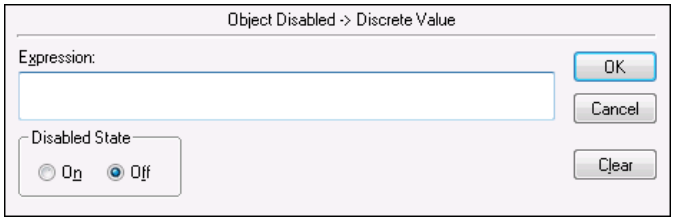
1. アニメーションを適用するオブジェクトを選択します。
2. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。
3. ［その他］セクションで［無効］を選択クリックします。

   ［オブジェクト無効化 -> 論理値］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［式］ボックスに、論理型タグの名前か論理値に等しい式を入力します。
5. ［無効化状態］セクションで、以下のいずれかを実行します。
   - ［オン］を選択すると、無効化状態が設定され、オブジェクトは論理型タグまたは式が True の間はアクティブになりません。
   - ［オフ］を選択すると、無効化状態が削除され、オブジェクトは論理型タグまたは式が True の間は機能します。
6. **[OK]** を選択します。

ツールチップの構成

ツールヒント リンクを使用して、ユーザーに画面上でオブジェクトに関する情報を提供するツールヒントを作成できます。ツールヒントはポインタがオブジェクト上に置かれたときに表示され、ポインタが離れると非表示となります。ツールヒントが表示される持続時間とツールヒントの配置は、オペレーティング システムによって決定されます。

ツールヒントには式または静的テキストを設定できます。

- 静的ツールヒントを作成して、ツールヒントが表示されるたびに同じメッセージを表示します。
- 式ツールヒントを作成して、ツールヒントが表示されるたびに式が評価されてツールヒント テキストとして表示されるようにします。

以下の式の例では、テキストが msgTooltipTag01 メッセージ型タグの現在の値として表示されます。

```
msgTooltipTag01
```

この例では、リテラル文字列の次に iTemp タグの現在の値が表示され、結果がツールヒント テキストとして表示されます。

```
"Current temp. is " + StringFromIntg (iTemp,10)
```

ツールヒント ウィンドウの幅は、Windows 7 およびそれ以降のバージョンの Windows で変更できます。InTouch.INI ファイルを編集して、ツールヒントの幅を指定するエントリを追加できます。

例:

- Tooltipwidth=200
- -1 の値（Tooltipwidth=-1）を指定すると、ツールヒントは改行なしの単一行に表示されます。
- ツールヒントの幅を指定しない場合、ツールヒント ウィンドウのデフォルトの幅は 88 です。

ツールヒント リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。
2. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

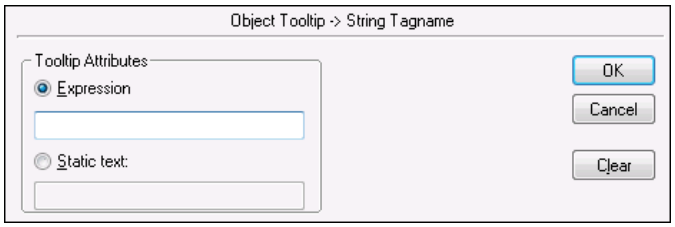
   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［その他］セクションで［ツールヒント］を選択します。

   ［オブジェクト ツールヒント -> 文字列タグ名］ダイアログ ボックスが表示されます。



4. ［ツールヒント属性］セクションで、［式］または［静的テキスト］のいずれかを選択します。
   - ［式］を選択した場合、メッセージ値を評価する式を入力します。シンプルなメッセージ型タグ名またはより複雑な式を使用できます。
   - ［静的テキスト］を選択した場合は、ツールヒント テキストとして静的メッセージ（最大 131 文字）を入力します。
5. **[OK]** を選択します。

タッチ スクリーン ウィンドウの配置

ランタイムでタッチ スクリーン オブジェクトに関連した正確な位置にウィンドウを表示させることができます。たとえば、オペレータがオブジェクトを選択して、ステータス、名前、またはそのオブジェクトにリンクされているその他のデータを参照できます。オペレータがクリックまたはマウスオーバーでオブジェクトを選択すると、指定した位置にウィンドウが表示されます。

スクリプト関数 ShowAt() または ShowTopLeftAt() をシステム読み取り専用タグ $ObjHor および $ObjVer と共に使用して、オブジェクトに関連するウィンドウを検索できます。これらの関数で固定位置を使用することもできます。

Windows の［画面のプロパティ］が［**Windows XP**］のテーマに設定されていると、この機能は場合によっては不安定になります。

構文は以下のとおりです。

```
ShowTopLeftAt(ウィンドウ名、$ObjHor、$ObjVer);
```

構文の要素を以下に示します。

windowname: 開かれるウィンドウの名前です。

$ObjHor: 選択したオブジェクトの中心の水平位置。

$ObjVer: 選択したオブジェクトの中心の垂直位置。

左上隅が選択したオブジェクトの中心に配置された状態で、新しいウィンドウが表示されます。

同様のスクリプト関数は、中心が選択したオブジェクトの中心に配置された状態でウィンドウを開きます。構文は以下のとおりです。

```
ShowAt (ウィンドウ名、$ObjHor、$ObjVer);
```

選択したオブジェクトでウィンドウを開くには

1. 表示するウィンドウを設計、命名、および作成します。
2. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

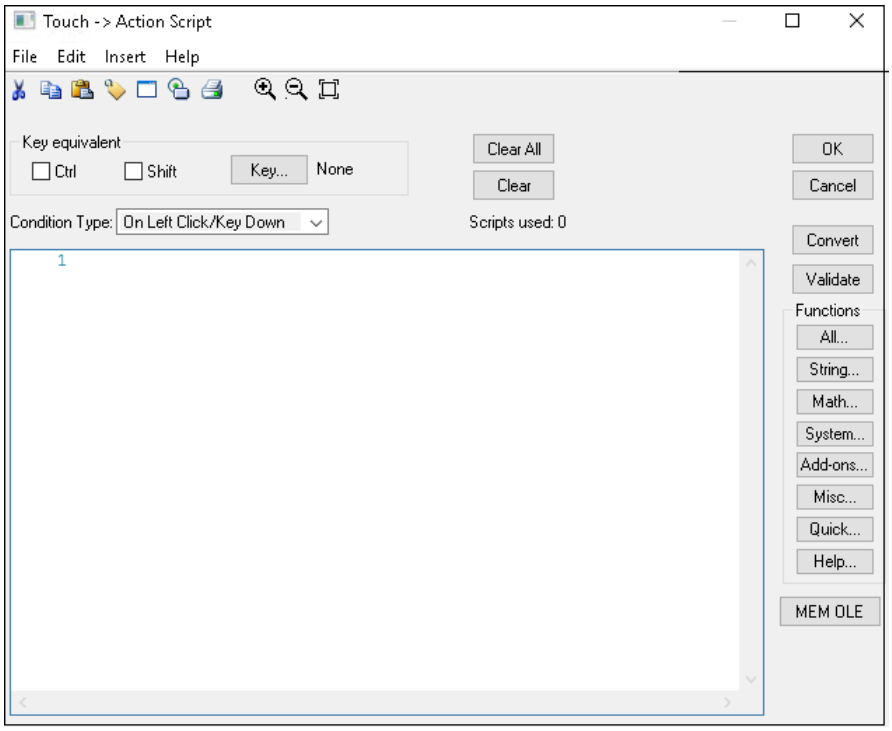
   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［アニメーション リンク］ウィンドウが表示されます。

3. ［タッチ押しボタン］セクションで［アクション］を選択します。

   ［タッチ -> アクション スクリプト］ダイアログ ボックスが表示されます。



4. 前に定義した構文に従って、以下のスクリプトの 1 つを入力します。

```
ShowTopLeftAt(ウィンドウ名、$ObjHor、$ObjVer);
```

   または

```
ShowAt (ウィンドウ名、$ObjHor、$ObjVer);
```

5. ［実行条件］ボックスで、ウィンドウを開くマウス アクションをクリックします。

6. **[OK]** を選択します。

$ObjHor システム タグ変数

対象オブジェクトの中心の水平ピクセル位置が含まれます。

カテゴリ

システム

使用法

$ObjHor：

データ型

整数型（読み取り専用）

参照項目

$ObjVer：
$ObjVer システム タグ変数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

対象オブジェクトの中心の垂直ピクセル位置が含まれます。

カテゴリ

システム

使用法

```
$ObjVer：
```

データ型

整数型（読み取り専用）

参照項目

$ObjHor：

データ入力アニメーション

オペレータ操作を有効にするオブジェクトを作成するには、タッチ リンクを使用します。オペレータは、タッチ リンクを使用してシステムにデータを入力できます。たとえば、オペレータはキーボードを使用したログオン、バルブのオン／オフ切り替え、新しいアラーム限界値の入力、プロセスの開始と停止を実行できます。

タッチ オブジェクトにフォーカスされると周囲に枠が表示されます。オブジェクトはユーザーがカーソルをオブジェクトの上に動かしたとき、または Tab キーまたは矢印キーを押してオブジェクトにフォーカスを移動したときにフォーカスされます。

ユーザーが Tab キーを使用してタッチ オブジェクトを選択するようにする場合は、水平パターンに配置するようにしてください。Tab キーを押すと、フォーカスが 1 つのオブジェクトから別のオブジェクトへ左から右にウィンドウの上部から下へ順番に移動します。

タッチ オブジェクトにそれぞれが重なり合ったテキスト オブジェクトを含む場合は、一番上のテキスト オブジェクトのみが表示されます。

オペレータは、タッチ オブジェクトをクリックしたり、割り当てられているキーを押したり、オブジェクトの枠が表示されたときに **Enter** キーを押したり、またはタッチ スクリーン表示装置を使用している場合は実際にタッチすることによって、オブジェクトをアクティブにすることができます。

9 つのタイプのユーザー入力タッチ リンクを作成することができます。

タッチ リンク	アクション
データ入力	• 論理型 • アナログ型 • 文字列
スライダ入力	• 垂直方向 • 水平方向
押しボタン	• 論理値 • アクション • ウィンドウ表示 • ウィンドウ消去

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

テキスト フィールドが入力に使用されている場合、キーが押されるとテキストが画面に表示されます。

入力しているときにテキストを表示しない場合は、リンクの設定パネルの［**入力のみ**］オプションを選択します。

論理値入力の有効化

論理型タグの値をある状態から別の状態に変更するためのオペレータ入力タッチ リンクを設計できます。たとえば、ポンプをオンまたはオフにするには、論理型リンクを使用します。

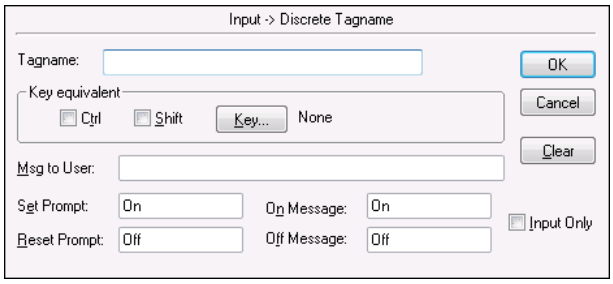
論理値入力リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。
2. ［**描画**］メニューの［**モード**］グループで［**アニメーション化**］を選択します。

   または、オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ウィンドウが表示されます。
3. ［**タッチ リンク**］領域で、［**データ入力**］の下にある［**論理値**］を選択します。

   ［**論理値入力リンク**］ダイアログ ボックスが表示されます。



4. ［**タグ名**］ボックスに、論理型タグの名前を入力します。
5. オプションで、［**キー アクションの割り当て**］領域で、リンクに割り当てるキーを設定します。詳細については、「キーボード ショートカットの作成」を参照してください。
6. 論理値入力詳細を設定します。以下の手順を実行します。
   - ［**プロンプト メッセージ**］ボックスに、［**入力**］ダイアログ ボックスに表示するメッセージを入力します。
   - ［**オンボタン タイトル**］ボックスおよび［**オフボタン タイトル**］ボックスに、オペレータが論理値のオン／オフを切り替えるときにクリックするボタンに表示するメッセージを入力します。
   - ［**オンメッセージ**］ボックスおよび［**オフメッセージ**］ボックスには、オブジェクトに関連付けられているテキスト フィールドに表示するオン／オフ メッセージを入力します。
7. オブジェクトに関連付けられているテキスト フィールドに入力値が表示されないようにするには、［**入力のみ**］チェック ボックスをオンにします。
8. **[OK]** を選択します。

アナログ入力の有効化

入力リンクを使用して、オペレータがアラーム限界点またはコンベヤ速度などの実数値を入力できるオブジェクトを作成できます。WindowViewer のプロパティで［**10 進法入力のみを許可**］オプションを選択すると、1E2 などの数値以外の値を入力フィールドに入力できなくなります。ランタイム設定の詳細

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

については、『*AVEVA™ InTouch HMI コンポーネントの標準の作成*』ガイドの「一般的な WindowViewer プロパティを設定する」を参照してください。

アナログ データ入力リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。
2. ［**描画**］メニューの［**モード**］グループで［**アニメーション化**］を選択します。

   または、オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ウィンドウが表示されます。
3. ［**タッチ リンク**］領域で、［**データ入力**］の下にある［**アナログ**］を選択します。

   ［**アナログ データ入力リンク**］ダイアログ ボックスが表示されます。

4. ［**タグ名**］ボックスに、アナログ型タグの名前を入力します。
5. ［**キー アクションの割り当て**］領域で、リンクに割り当てるキーを設定します。詳細については、「キーボード ショートカットの作成」を参照してください。
6. アナログ値入力詳細を設定します。以下の手順を実行します。
   - 値を入力するために画面上に数値キーパッドを表示するには、［**キーパッドの使用**］領域で、［**する**］をクリックします。［**キーパッドのタイトル**］ボックスに、キーパッドに表示するプロンプト メッセージを入力します。
   - ［**最小値**］ボックスおよび［**最大値**］ボックスに、タグに対する最小および最大の入力値を入力します。これらの設定は、ランタイム時のアニメーションに対するユーザー入力を制限します。定数アナログ値（整数または実数）、および参照もサポートされています。デフォルト値はそれぞれ 1 および 100 です。最大値は最小値より大きい必要があります。

     参照を使用する場合の詳細については、「アナログ入力アニメーションの最小と最大の制限で参照を使用する」セクションを参照してください。
7. オブジェクトに関連付けられているテキスト フィールドに入力値が表示されないようにするには、［**入力のみ**］チェック ボックスをオンにします。
8. ［**書式設定**］領域のリストで、ランタイム時の詳細な書式設定を行う、それぞれのデータ タイプをクリックします。選択したデータ タイプに基づいて、［**固定幅**］チェック ボックス、［**精度**］ボックス、および［**ビット数から**］および［**まで**］ボックスが有効になります。これらのオプションの設定の詳細については、「テキストの詳細な書式設定」セクションを参照してください。
9. **[OK]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アナログ入力アニメーションの最小と最大の制限に対する参照の使用

アナログユーザー入力アニメーションの最小と最大の制限では、定数のアナログ値と参照がサポートされています。これらの制限は、［**アナログ データ入力リンク**］ダイアログ ボックスで設定できます。

参照はメモリ タグ変数または I/O タグ変数などの、リモート アナログ タグ変数またはローカル アナログ タグ変数のいずれかになります。

- InTouch I/O タグ変数を最小または最大値として使用した場合、InTouch I/O タグ変数だけが評価されます。I/O タグ変数が文字列値の割り当てられたリモート タグ リファレンスを参照している場合、I/O タグ変数の値は 0 として評価されます。
- 最小値と最大値でリモート タグ リファレンスが設定されている場合、そのリファレンスは文字列として評価されます。リモート タグ リファレンスが文字列値に割り当てられており、アナログ値への変換に失敗した場合、警告メッセージがロガーに記録されます。［**最小値**］ボックスと［**最大値**］ボックスは、ユーザー入力が設定されたリファレンスのデータ タイプに基づいた値を表示します。
- ［**最小値**］フィールドまたは［**最大値**］フィールドのユーザー入力が、「e」の文字を含んでいる場合、入力は文字列値または指数値のいずれかになります。ユーザー入力に数字、「e」の文字、数字（つまり ±N e ±N の形式、この場合の N は数字）を含んでいる場合、ユーザー入力は指数値としてみなされます。ユーザー入力に「e」とアルファベット（「e」を含む）が含まれている場合、ユーザー入力はリファレンス名としてみなされます。

参照値がランタイム時に不適切である場合、次の状況が発生します。

［**最小値**］ボックスの参照については、

- アニメーションのプライマリ参照が InTouch タグ変数の場合、設定されたタグ変数の最小値がタグ変数データベースから取得され、最小値に使用されます。
- アニメーションのプライマリ参照が外部参照の場合、データ タイプの最小値が使用されます。設定されたデータ ポイントが文字列の場合、最後の既知の値によってデータ タイプが決まります。
  - 最後の既知の値に小数点がある場合、データ タイプは実数になります。
  - 最後の既知の値に小数点がある場合、データ タイプは整数になります。
  - 最後の既知の値がない場合、整数が使用されます。

［**最大値**］ボックスの参照については、

- アニメーションのプライマリ参照が InTouch タグ変数の場合、設定されたタグ変数の最大値がタグ変数データベースから取得され、最大値に使用されます。
- アニメーションのプライマリ参照が外部参照の場合、データ タイプの最大値が使用されます。設定されたデータ ポイントが文字列の場合、最後の既知の値によってデータ タイプが決まります。
  - 最後の既知の値に小数点がある場合、データ タイプは実数になります。
  - 最後の既知の値に小数点がある場合、データ タイプは整数になります。
  - 最後の既知の値がない場合、整数が使用されます。

文字列入力の有効化

文字列入力リンクを使用して、ユーザーがバッチ名、オペレータ ID、またはパスワードなどのテキスト文字列を入力できるようにするオブジェクトを作成できます。

文字列入力リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

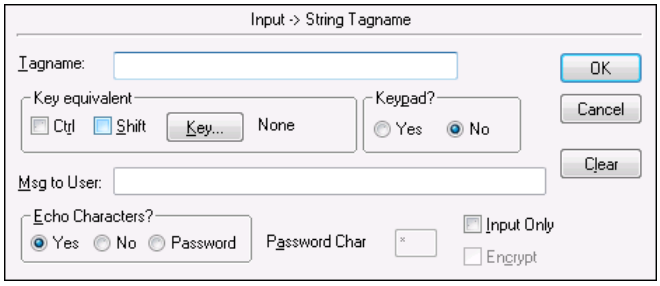
2. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。

3. ［タッチ リンク］領域で、［データ入力］の下にある［文字列］を選択します。

   ［文字列入力リンク］ダイアログ ボックスが表示されます。



4. ［タグ名］ボックスに、メッセージ型タグの名前を入力します。
5. オプションで、割り当てるキーまたはキーボードを設定します。
   - ［キー アクションの割り当て］領域で、リンクに割り当てるキーを設定します。詳細については、「キーボード ショートカットの作成」を参照してください。
   - 新しい値を入力するためのキーボードを表示するには、［キーパッドの使用］領域で、［する］をクリックします。［キーパッドのタイトル］ボックスに、キーボードに表示するメッセージを入力します。
6. ［入力文字列の表示］領域で、ユーザーが文字列を入力したときに入力ボックスに文字を表示するかどうかを選択します。
   - 入力したテキストを入力ボックスに表示するには、［する］をクリックします。
   - 入力したテキストを表示しない場合は、［しない］をクリックします。
   - 入力したテキストの代わりに「マスク」文字を使用するには、［パスワード］をクリックします。［パスワードのエコー文字］ボックスに、マスク文字を入力します。［暗号化］チェック ボックスをオンにして、パスワードを暗号化します。

     重要：パスワード暗号化は、InTouch HMI コンテキスト内でのみ動作します。オペレーティング システムや SQL Server データベースなどの外部のセキュリティ システムに渡す場合は文字列を暗号化しないでください。外部セキュリティ システムは暗号化パスワード文字列を読み取ることができず、アクセスは失敗します。
7. オブジェクトに関連付けられているテキスト フィールドに入力値が表示されないようにするには、［入力のみ］チェック ボックスをオンにします。
8. **[OK]** を選択します。

スライダ入力の有効化

ユーザーがスライダ タッチ リンクを使用して、前後にドラッグできるオブジェクトを作成できます。ユーザーがオブジェクトを動かすと、それにリンクされているタグの値が変わります。オブジェクトを水平または垂直スライダにリンクできます。

オブジェクトをスライダにする際、カーソルがロックするために使用するオブジェクト上のポイントとなる参照位置を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

単一オブジェクトに水平リンクおよび垂直リンクの両方を使用して、2 つのアナログ型タグの値が同時に変わるようにします。

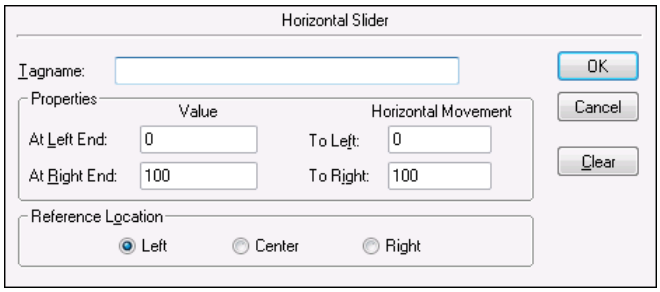
水平スライダ リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。
2. ［**描画**］メニューの［**モード**］グループで［**アニメーション化**］を選択します。

   または、オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ウィンドウが表示されます。
3. ［**スライダ入力**］領域で、［**水平方向**］をクリックします。

   ［**水平スライダー**］ダイアログ ボックスが表示されます。



4. ［**タグ名**］ボックスに、アナログ型タグの名前を入力します。
5. ［**プロパティ**］領域で、以下の手順を実行します。
   a. ［**左端の値**］ボックスに、スライダが左端にあるときのタグの値を入力します。
   b. ［**右端の値**］ボックスに、スライダが右端にあるときのタグの値を入力します。
   c. ［**左への移動幅**］ボックスに、スライダを左に動かすことができる幅をピクセル数で入力します。
   d. ［**右への移動幅**］ボックスに、スライダを右に動かすことができる幅をピクセル数で入力します。
6. ［**カーソルの基準点**］領域で、カーソルがオブジェクトにロックする際に使用するオブジェクト上の位置を選択します。
7. **[OK]** を選択します。

垂直スライダ リンクを作成するには

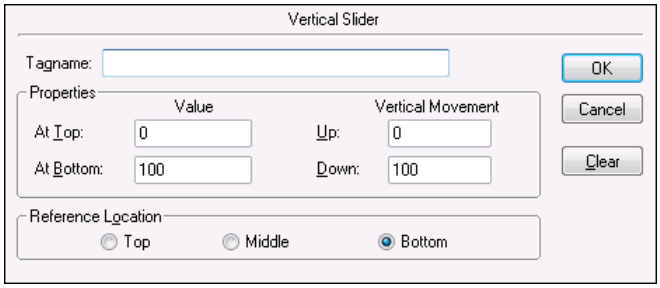
1. アニメーションを適用するオブジェクトを選択します。
2. ［**描画**］メニューの［**モード**］グループで［**アニメーション化**］を選択します。

   または、オブジェクトを右クリックして、［**アニメーション リンク**］を選択します。

   ［**アニメーション リンク**］ウィンドウが表示されます。
3. ［**スライダ入力**］領域で、［**垂直方向**］を選択します。

   ［**垂直スライダー**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［タグ名］ボックスに、アナログ型タグの名前を入力します。
5. ［プロパティ］領域で、以下の手順を実行します。
   a. ［上端の値］ボックスに、スライダが上端にあるときのタグの値を入力します。
   b. ［下端の値］ボックスに、スライダが下端にあるときのタグの値を入力します。
   c. ［上への移動幅］ボックスに、スライダを上に動かすことができる幅をピクセル数で入力します。
   d. ［下への移動幅］ボックスに、スライダを下に動かすことができる幅をピクセル数で入力します。
6. ［カーソルの基準点］領域で、カーソルがオブジェクトにロックする際に使用するオブジェクト上の位置を選択します。
7. ［OK］ を選択します。

押しボタンの有効化

タッチ押しボタン リンクを使用して、アクション スクリプトを開始するタッチ オブジェクトを作成できます。

アクション スクリプトは、タグを特定の値に設定したり、他のアプリケーションの起動や制御、機能の実行などに使用できます。

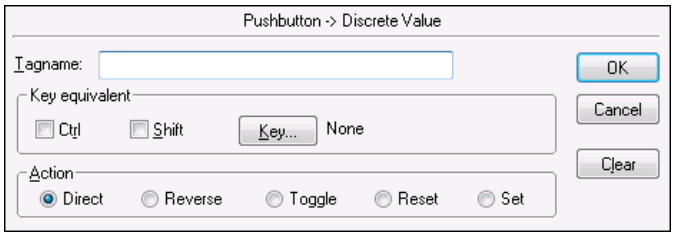
論理値タッチ リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。
2. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。
3. ［押しボタン］領域で、［論理値ボタン］を選択します。

   ［押しボタン - 論理値］ダイアログ ボックスが表示されます。



4. ［タグ名］ボックスに、論理型タグ名を入力します。
5. ［キー］ボタンをクリックして、リンクにキーを割り当てます。
6. ［ボタン アクション プロパティ］領域で、以下のいずれかのタイプを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［ダイレクト］をクリックすると、押しボタンが押されている間、値が 1 に設定されます。ボタンを離すと、値は自動的に 0 にリセットされます。
- ［リバース］をクリックすると、押しボタンが押されている間、値が 0 に設定されます。ボタンを離すと、値は自動的に 1 にリセットされます。
- ［トグル］をクリックすると、論理型タグ変数の状態をリバースします。
- ［リセット］をクリックすると、値が 0 に設定されます。
- ［セット］をクリックすると、値が 1 に設定されます。

7. **[OK]** を選択します。

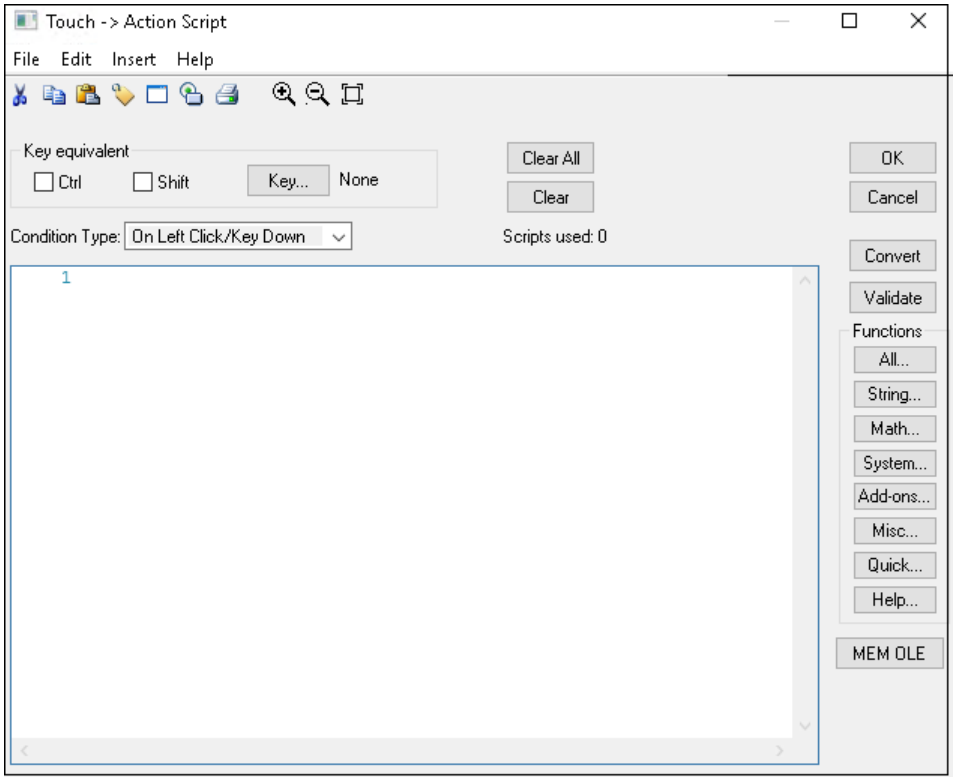
アクション スクリプト リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。
2. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。
3. ［タッチ押しボタン］領域で、［スクリプト ボタン］を選択します。

   ［タッチ アクション スクリプト］ダイアログが表示されます。



4. ［実行条件］リストで、スクリプト タイプを選択します。アクション スクリプトのタイプの詳細については、「スクリプト トリガ」を参照してください。

アクション押しボタンにキーを割り当てて、同じキーがキー スクリプトに使用されている場合、割り当てたキーがキー スクリプトよりも優先されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［スクリプト エディタ］ウィンドウで、オブジェクトが起動されたときに実行するスクリプトを入力します。
6. **[OK]** を選択します。

ウィンドウを開く／閉じる

ウィンドウの表示／消去リンクを使用して、他の InTouch ウィンドウを開いたり閉じたりするタッチ リンクを作成できます。

オブジェクトをプログラムして、一度に複数のウィンドウを開くように設定できます。ただし、複数のウィンドウを開くようにリンクをプログラムした場合、重なりとウィンドウのタイプに注意してください。

開いているウィンドウの 1 つが置換タイプのウィンドウで別に開いているウィンドウと交差している場合は、開く前に他のウィンドウが閉じます。

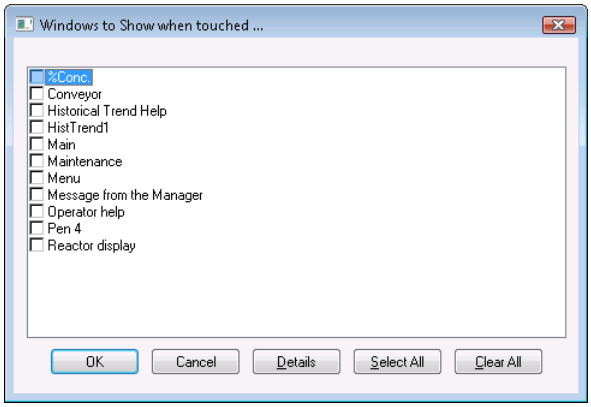
ウィンドウ表示／消去リンクを作成するには

1. アニメーションを適用するオブジェクトを選択します。
2. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。
3. ［押しボタン］領域で、［ウィンドウ表示］または［ウィンドウ消去］を選択します。

   ［タッチ時に表示するウィンドウ］または［タッチ時に非表示するウィンドウ］ダイアログ ボックスが表示されます。



4. 表示または非表示にするウィンドウを選択します。
5. **[OK]** を選択します。

画面上キーボードの設定

画面上キーボードを使用すると、コンピュータにキーボードが接続されていない場合にオペレータ入力が可能になります。

3 つの画面上キーボード タイプのうちの 1 つを指定できます。

- 標準 InTouch キーボードまたはキーパッド。これがデフォルトのキーボードです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- Windows システムのキーボード。Windows キーボードは、完全に機能する QWERTY タイプのキーボードで、ファンクション キー、プリント スクリーン キー、数字ロック キー、方向矢印などを備えています。
- リサイズ可能なキーボードまたはキーパッド。このキーボードは、ランタイムにサイズを変更できます。

キーボードは DialogStringEntry() 関数および DialogValueEntry() 関数をスクリプトで使用して開くこともできます。詳細については、「DialogStringEntry() 関数」および「DialogValueEntry() 関数」を参照してください。

画面上キーボード タイプを設定します。

1. WindowMaker を開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［環境設定］タブの［キーボード］領域で、使用するキーボードのタイプを選択します。
4. ［リサイズ可能なキーボード］を選択した場合は、［オプション］をクリックして、キーボードのフォント、位置、およびサイズ プロパティを選択します。
5. **[OK]** を選択します。

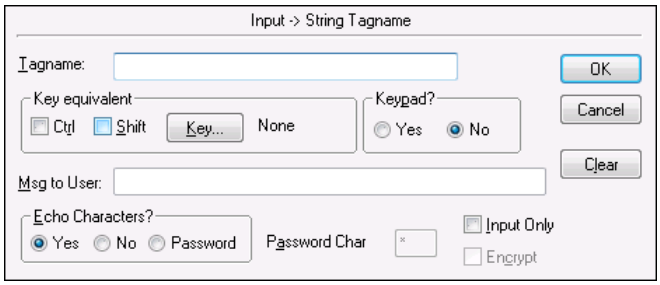
画面上キーボードを表示するには

1. 画面上キーボード タイプを設定します。
2. アニメーションを適用するオブジェクトを選択します。
3. ［描画］メニューの［モード］グループで［アニメーション化］を選択します。

   または、オブジェクトを右クリックして、［アニメーション リンク］を選択します。

   ［アニメーション リンク］ウィンドウが表示されます。
4. ［タッチ リンク］領域で、［データ入力］の下にある［文字列］を選択します。

   ［文字列入力リンク］ダイアログ ボックスが表示されます。



5. ［キーパッドの使用］領域で、［する］を選択します。
6. **[OK]** を選択します。

DialogStringEntry() 関数

この関数は、画面に英数字キーボードを表示し、オペレータがタグ変数ディクショナリにあるメッセージ型タグ変数の現在の文字列の値を画面上で変更できるようにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

その他

構文

```
[Result=]DialogStringEntry(MessageTag_Text、UserPrompt_Text);
```

パラメータ

MessageTag_Text
変更するメッセージ型タグ変数の名前です。この値は文字列値です。タグ変数を引用符で囲んで指定するか、引用符なしの .Name ドットフィールドを使用します。メッセージ型タグ変数をポインタとして使用することもできます。

UserPrompt_Text
キーボード上部に表示するユーザー メッセージです。

戻り値

以下の整数値のいずれかを返します。

0 = ［キャンセル］がクリックされました。

1 = ［OK］がクリックされました。

-1 = 内部エラーです。

-2 = 開始できませんでした。

-3 = タグ変数が定義されていません。

-4 = タグ変数がメッセージ型ではありません。

-5 = 書き込めません。

備考

この関数は、タッチ スクリーンを含むアプリケーションで主に使用されます。

例

```
Errmsg=DialogStringEntry(MyMessageTag.Name, "Enter a new string...");
Errmsg=DialogStringEntry("MyMessageTag","Enter a new string...");
```

たとえば、以下のスクリプトは英数字キーボードを開き、その上部に「新しい文字列を入力してください…」というメッセージを表示する一方、そのキーボードから MyMessageTag を変更できるようにします。

```
MessageTagX="MyMessageTag"; {文字列 MyMessageTag（変更するタグ変数）をメモリ メッセージ型タグ変数 MessageTagX に割り当てます}
MessageDisplay="新しい文字列を入力してください..."; {新しいメッセージ文字列をメモリ メッセージ型タグ変数 MessageDisplay に割り当てます}
Errmsg=DialogStringEntry(MessageTagX, MessageDisplay); {MessageTagX はメッセージ型タグ変数として定義されているので引用符は必要ありません}
```

参照項目

DialogValueEntry()

DialogValueEntry() 関数

画面上に数字キーパッドを表示し、ユーザーが論理型、整数型、または実数型タグ変数の現在の値を変更できるようにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

その他

構文

```
[Result=]DialogValueEntry(ValueTag_Text、LowLimit、HighLimit、UserPrompt_Text);
```

パラメータ

ValueTag_Text
変更する論理型、整数型、または実数型タグ変数の名前です。この値は文字列値です。タグ変数名を引用符で囲んで指定するか、引用符なしの .Name フィールドを使用します。メッセージ型タグ変数をポインタとして使用することもできます。

LowLimit
タグ変数の下限値です（この値は、最小値、最小生データ、または最小工学値に対するタグ変数の定義値と同じかそれ以上です）。

|HighLimit'
タグ変数の上限値です（この値は、最大値、最大生データ、または最大工学値に対するタグ変数の定義値と同じかそれ以下です）。

UserPrompt_Text
キーパッド上部に表示するユーザー メッセージです。

戻り値

以下の整数値のいずれかを返します。

0 = ［キャンセル］がクリックされました。

1 = ［OK］がクリックされました。

-1 = 上限 <= 下限

-2 = 開始できませんでした。

-3 = タグ変数が定義されていません。

-4 = タグ変数が論理型、整数型、または実数型ではありません。

-5 = 書き込みに失敗しました。

備考

この関数は、タッチ スクリーンを含むアプリケーションで主に使用されます。

例

```
Errmsg=DialogValueEntry(MyIntegerTag.Name, MyIntegerTag.MinEU, MyIntegerTag.MaxEU, "新しい値を入力してください...");
Errmsg=DialogValueEntry("MyIntegerTag", -100, 100, "新しい値を入力してください...");
```

たとえば、以下のスクリプトは数字キーパッドを表示し、その上部に「新しい文字列を入力してください」というメッセージを表示する一方、そのキーパッドから －100 ～ 100 の最小および最大制限を使用して、MyIntegerTag を変更できるようにします。

```
TagnameX="MyIntegerTag"; {文字列 MyIntegerTag（変更するタグ変数）をメモリ メッセージ型タグ変数 TagnameX に割り当てます}
Min=-100; {タグ変数に許可される最小値をメモリ実数／整数型タグ変数 Min に割り当てます}
Max=100; {タグ変数に許可される最大値をメモリ実数／整数型タグ変数 Max に割り当てます}
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
MessageDisplay="新しい値を入力してください..."; {新しいメッセージ文字列をメモリ メッセージ型タグ変数 MessageDisplay に割り当てます}
Errmsg=DialogValueEntry(TagnameX, Min, Max, MessageDisplay); {TagnameX はメッセージ型タグ変数として定義されているため、引用符は必要ありません。論理型、整数型、または実数型タグ変数を TagnameX に割り当てることによって割り当てられたタグ変数を変更します}
```

参照項目

DialogStringEntry()

共通のアニメーションタスク

共通アニメーション タスクには、タグ変数の選択、キーボード ショートカットの作成、タグ変数参照の変更、プレースホルダ タグ変数の置換などが含まれます。

タグまたは属性の選択

［タグ変数を選択してください］ダイアログ ボックス（タグ ブラウザ）を使用して、以下の内容を選択します。

- ローカルまたはリモート InTouch アプリケーションに定義されているタグ。
- Galaxy ブラウザに関連付けられた Application Server オブジェクト属性。

［タグの選択］ダイアログ ボックスを開くには、入力用のタグ名を必要とするテキスト ボックス内をダブルクリックします。

InTouch タグを選択するには

ローカルまたはリモート InTouch アプリケーションに定義されているタグを選択できます。タグ名ディクショナリ インターフェイスをサポートするタグ ソース内にタグへの参照を作成できます。

たとえば、リモート タグ 参照を使用すると、ローカルのタグ名ディクショナリでタグを作成することなく、I/O サーバーからデータにアクセスできます。

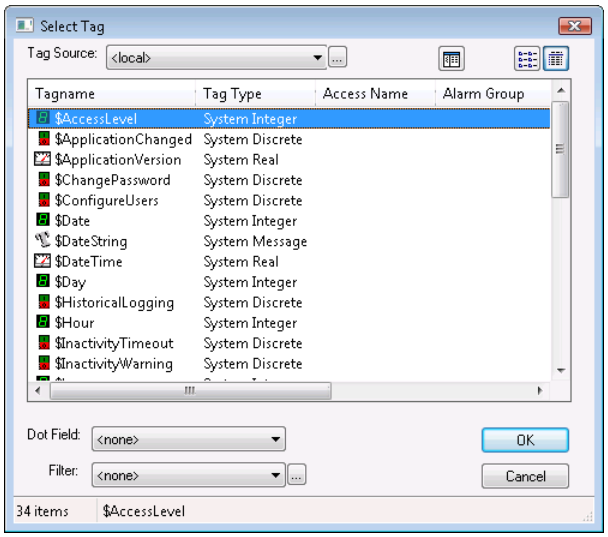
選択した各 InTouch タグ用のドットフィールドを設定できます。ドットフィールドは、タグ プロパティへのアクセス、監視、および変更を行えます。ドットフィールドを選択しない場合、.Value ドットフィールドが使用されます。ドットフィールドの詳細については、「タグ ドットフィールドを使用したタグ プロパティの表示または変更」を参照してください。

InTouch タグを選択するには

1. 入力にタグ名が必要なテキスト ボックスをダブルクリックします。

   ［タグの選択］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. **タグ ソース**リストで、タグ ソースの名前をクリックするか、参照ボタンをクリックして、新しいタグ ソースを定義します。

   タグ ソースの定義の詳細については、「I/O を使用したデータ アクセス」を参照してください。

3. **フィルタ**リストで、ウィンドウに表示されるタグ変数の数を減らすためのフィルタをクリックします。フィルタを定義するには、省略記号ボタンをクリックします。詳細については、「タグ変数フィルタの作成」を参照してください。

4. ウィンドウでタグ名を選択します。

   [タグの選択] ダイアログ ボックスに表示されるタグ名の表示方法は変更できます。詳細については、「[タグの選択] ダイアログ ボックスの表示の変更」を参照してください。

5. **ドット フィールド**リストで、選択したタグ変数に付加するドットフィールドをクリックします。

   ドットフィールドは、タグ プロパティへのアクセス、監視、および変更を行えます。ドットフィールドを選択しない場合、.Value ドットフィールドが使用されます。

6. **[OK]** を選択します。

Application Server オブジェクト属性の選択

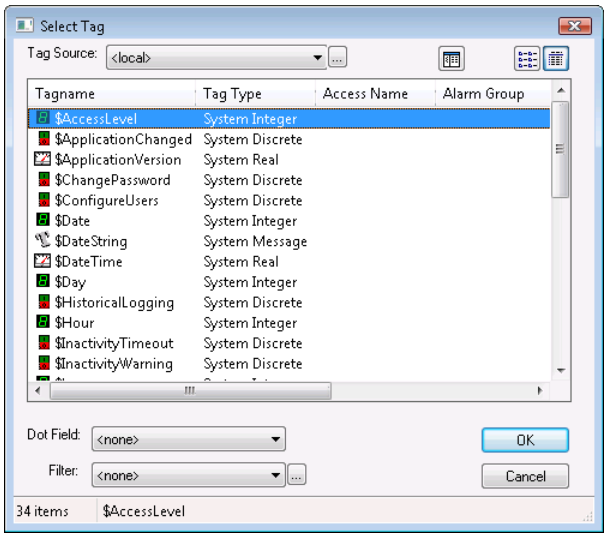
Application Server オブジェクトに関連付けられている属性を選択できます。これを行うには、最初にオブジェクトを InTouch アプリケーション用のデータ ソースとして含む Galaxy を追加します。データ ソースの詳細については、「I/O を使用したデータ アクセス」を参照してください。

オブジェクト属性を選択するには

1. 入力にタグ名が必要なテキスト ボックスをダブルクリックします。

   [タグの選択] ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［**タグ ソース**］リストで、Galaxy を使用するタグ ソースの名前を選択するか、参照ボタンを選択して、新しいタグ ソースを定義します。

   ［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。

3. ［**Galaxy ブラウザ**］を使用し、オブジェクトの属性を検索して選択します。詳細については、Application Server のマニュアルを参照してください。

4. ［**OK**］を選択して、［**Galaxy ブラウザ**］を閉じます。

   タグ名を必要とする属性参照がテキスト ボックス内に表示されます。

   注:［**Galaxy ブラウザ**］から［**タグの選択**］ダイアログ ボックスに戻るには、［**Galaxy ブラウザ**］の右下隅にある［**戻る**］ボタンを選択します。

タグ変数フィルタの作成

タグ変数ディクショナリ内に多数のタグ変数がある場合、アニメーション用のタグ変数を探すのは困難です。たとえば、特定のアクセス名またはアラーム グループに関連付けられているタグ変数のみを参照したい場合、フィルタを設定してタグ変数のサブセットのみが［**タグ変数を選択してください**］ダイアログ ボックスに表示されるようにすることができます。

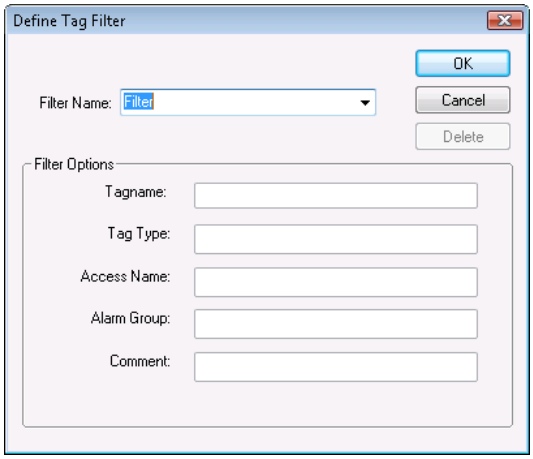
フィルタには次のワイルドカード式を使用できます。

- 複数文字を表すワイルドカードには、アスタリスク（*）を使用します。たとえば、「Asyn*」と指定すると、「Asyn」の文字で始まるタグ変数がすべて検索されます。
- 単一文字を表すワイルドカードには疑問符（?）を使用します。たとえば、「Tag?」というフィルタは、「Tag」で始まる 4 文字のタグ変数をすべて検索します。「Tag*」というフィルタは、「Tag」で始まるタグ変数をすべて検索します。
- フィルタには、タグ変数に有効な文字に加えて 2 つのワイルドカードを使用できます。有効なタグ変数文字は、

検索フィルタを定義するには

1. ［**タグ変数を選択してください**］ダイアログ ボックスで、［**フィルタ**］リストの横にある参照ボタンを選択します。［**タグ フィルタの定義**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［フィルタ名］ボックスに、フィルタ名を入力します。
3. ［オプション］領域で、フィルタ条件を設定します。以下のいずれかを実行します。
   - ［タグ変数］ボックスに、タグ変数を入力します。
   - ［タグ変数タイプ］ボックスに、タグ変数タイプを入力します。
   - ［アクセス名］ボックスに、ローカル アクセス名を入力します。
   - ［アラーム グループ］ボックスに、アラーム グループ名を入力します。
   - ［コメント］ボックスに、コメント式を入力します。
4. **[OK]** を選択します。**フィルタ名**が［**タグ変数を選択してください**］ダイアログ ボックスの［**フィルタ**］リストに表示されます。フィルタ条件に一致するタグ変数を表示するフィルタを選択できます。

検索フィルタを削除するには

1. ［フィルタ名］ボックスで、フィルタを選択します。
2. [削除]を選択します。

［タグの選択］ダイアログ ボックスの表示の変更

［タグ変数を選択してください］ダイアログ ボックスには、

表示するビュー	属性テンプレート	説明
リスト ビュー	一覧ビュー ボタン	各タグ名のタイプに従って小さいアイコンがタグ名の横に表示されます。
詳細ビュー	詳細ビュー ボタン	同じ小さいアイコンおよびタグ名とタグ タイプ、アクセス名、アラーム グループ、およびコメントが表示されます。リストをソートするにはカラム ヘッダーを選択します。
ツリー ビュー	ツリー ビュー アイコン	ツリー ビューは、タグ名を 2 つのビューで表示します。 スーパータグ テンプレート内のメンバー タグ名にアクセスできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

キーボード ショートカットの作成

キー割り当てリンクを使用して、特定のアニメーション リンクを有効にするには、キーボード上の特定のキーを割り当てることができます。割り当てたキーは、そのリンクを持つオブジェクトが表示または選択されている場合のみ機能します。オブジェクトに表示オン／オフや使用オン／オフのリンクを関連付けた場合、そのオブジェクトがオフの状態で割り当てたキーを使用することはできません。

同じキーは複数のウィンドウで定義できます。ただし、最後に開いたウィンドウでの定義がアクティブになります。オーバーレイ ウィンドウの場合には、一番上のウィンドウのキーがアクティブになります。

注記：アクティブ ウィンドウのオブジェクトや操作ボタンにキー アクション スクリプトで使用しているものと同じキーを割り当てる場合、アクティブ ウィンドウのキーに対するキー アクションの割り当てリンクがキー アクション スクリプトよりも優先されます。

割り当てられたキーをサポートしているアニメーション リンクの設定ダイアログには、［**キー アクションの割り当て**］領域が表示されます。

キー リンクは、1 ～ 16 のファンクション キーでのみ利用できます。16 以上のファンクション キーを持つカスタム キーボードを使用している場合は、システムがその拡張ファンクション キーへアクセスするためのデバイス ドライバをキーボードの製造元から入手する必要があります。

リンクにキーを割り当てるには

1. 設定しているリンクのタイプに対する［**アニメーション リンク**］ダイアログ ボックスを開きます。
2. Ctrl キーか Shift キーを押しながらキーを押すように指定する場合は、［**Ctrl**］か［**Shift**］チェック ボックスをオンにします。
3. **[キー] を選択します。**［**キーの選択**］ダイアログ ボックスが表示されます。
4. リンクに割り当てるキーを選択します。

   ［**アニメーション リンク**］ダイアログ ボックスが再び表示され、選択されたキーの名前が［**キー**］ボタンの横に表示されます。
5. **[OK]** を選択します。

タグ名参照の変更

オブジェクトを複製すると、リンク、アニメーション、スクリプトなどを含む元のオブジェクトと全く同じものが作成されます。ただし、複製したオブジェクトに別のタグを使用する必要がある場合には、タグを置換できます。

オブジェクトのタグは選択して置換できます。複数のオブジェクトを選択し、そのすべてのタグを一度に置換することもできます。

システムのライセンスによりタグ名の数が制限されている場合には、ローカル タグをリモート タグ 参照に変換して、ローカル タグ名ディクショナリで定義されているタグの数を減らすことができます。

タグの置換はすべてのタグおよび参照に有効です。

タグを別のタグと置換するには

1. 別のタグに変更するタグに関連付けられているオブジェクトを選択します。
2. ［**アニメーション**］メニューの［**置換**］グループで［**タグ**］を選択します。

   ［**タグ名の置換**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［新しい名前］ボックスに、新しいタグ名を入力します。
   - ［新しい名前］ボックスのタグをダブルクリックすると、タグ名ディクショナリにそのタグの定義が表示されます。
   - タグ名を消去した後に空白のボックス内をダブルクリックすると、［タグの選択］ダイアログ ボックスが表示されます。
4. **[OK]** を選択します。

   オブジェクトに関連付けられているタグが自動的に変更されます。

プレースホルダ タグの変換

ウィンドウまたは QuickScript を現在のアプリケーション間でインポートまたはエクスポートする場合、そのウィンドウまたは QuickScript に関連付けられているすべてのタグがウィンドウと一緒に転送されます。

ローカル タグは、自動的にはアプリケーション データベースに追加されません。その代わりに、これらのタグは自動的にプレースホルダ タグとしてマーク付けされます。

リモート タグ 参照は影響を受けず、プレースホルダ タグとしてマークされません。

プレースホルダ タグを既存のローカル タグに変換して、新しいアプリケーションのタグ名ディクショナリに定義するか、またはリモート タグ 参照にする必要があります。

タグの前には、?d:、?i:、?m:、?r: などのプレースホルダが付いていることに注意してください。これらのプレースホルダは、最初に定義した際のタグのタイプを以下のように表しています。

プレースホルダ シンボル	タグ名 タイプ
d	論理型
i	整数型
m	メッセージ型
r	実数型

注記：リモート タグ 参照は、プレースホルダとして表示されませんが、**PLC2:Temperature**。

［タグ名の置換］ダイアログ ボックスでは、複数の方法を使用してプレースホルダ タグをローカル タグに変換できます。詳細については、『AVEVA™ InTouch HMI アプリケーション ランタイム ガイド』の「InTouch コンポーネントのエクスポートとインポート」を参照してください。

ヒント: タグ名を手動で変換し、元のタグ名ディクショナリで定義されている元のタグを必要がなくなった場合は、タグ名の使用カウントを更新して、使用されていないタグを削除できます。

ウィンドウまたは QuickScript を別のアプリケーションからインポートし、アニメーション リンクまたは QuickScript に関連付けられているタグ名をすべてリモート タグ名 参照に変換すると、ローカルのタグ名ディクショナリでタグを全く定義しない場合でも数多くのリモート タグ名からデータをすぐに受け取ることができるようになります。

テキストの詳細な書式設定

InTouch WindowMaker を使用して、InTouch WindowViewer がランタイム時に使用するインテリジェントな書式設定または詳細な書式設定を設定することができます。これらのオプションは、アナログ値の表示またはアナログ ユーザー入力アニメーションに対して詳細な書式設定を使用している場合にのみ適

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

用されます。詳細な書式設定オプションを設定していない場合、InTouch WindowViewer の現在値が保持されます。

詳細な書式設定オプションは、［**アナログ データ入力リンク**］ダイアログ ボックスと［**アナログ値表示リンク**］ダイアログ ボックスに表示されます。

数値の詳細な書式設定により、データのタイプと品質に基づいて数値データをネイティブな方法で書式設定することができます。詳細な書式設定オプションでは、ランタイム時に InTouch WindowViewer でアナログ値の書式設定を行うことができます。

個々のアプリケーションに対してテキスト文字列を使用するか、または詳細な書式設定オプションを有効にすることができます。旧バージョンの InTouch からアニメーションを移行するか、またはアニメーションを初めて作成する場合、デフォルトとしてテキスト文字列が設定されます。

詳細な書式設定オプションは次のとおりです。

- テキスト文字列。既存の InTouch 書式設定を使用します。
- 実数。ランタイム時の値のサイズに基づいた実数として書式設定されます。グローバルな小数点の精度の設定は、小数点以下の桁数を設定します。
- 固定小数点の書式は、値のタイプによって決まります。
  - 値は［**精度**］コントロールを使用して、ユーザーが指定した小数点の桁数で書式設定されます。
  - 値は小数点のない整数として書式設定され、小数点のあるべき場所に合わせて右揃えされます。
  - 値は小数点のない 0 または 1 で書式設定され、小数点のあるべき場所に合わせて右揃えされます。
- 整数。常に少数点のない整数として書式設定されます。
- 指数。常に固定少数点を配慮した指数として書式設定されます。
- 16 進数値。設定したブール値の範囲に基づいた 16 進数で書式設定されます。
- バイナリ。常に設定したブール値の範囲を配慮したバイナリ文字列として書式設定されます。

［**固定幅**］ボックスでは、デザイン時に指定したテキスト サイズを大きくすることができます。このオプションは、書式設定のスタイルに［**テキスト文字列**］が選択されている場合には利用できません。

デザイン時に、アニメーションの設定で［**固定幅**］チェック ボックスをオンにした場合、ランタイム時に表示される値はテキスト フィールドの長さより長くなりません。値がテキスト フィールドの長さを超えた場合、グローバルの［長すぎる値の固定フィールド幅］の設定がテキスト フィールドの長さを決めます。固定幅の書式設定は、ランタイム時に値のサイズに合わせて、テキスト要素の幅が広がることを防ぎます。

デザイン時にテキスト フィールドにテキストを何も指定しなかった場合、固定幅の設定は無視され、値全体が表示されます。たとえば、デザイン時に ""（テキストなし）と入力した場合、テキスト フィールドの幅は 0 に相当します。この状態で［**固定幅**］オプションが有効になっていると、テキストは何も表示されなくなります。この場合、［**固定フィールドの幅**］のランタイム時のデフォルトは、「なし」に設定されます。

［**クリア**］を選択すると、詳細な書式設定オプションは、次のようにリセットされます。

- ［**書式設定**］オプションには、［**テキスト文字列**］が選択されます。
- ［**固定フィールドの幅**］オプションはクリアされ、無効になります。［**テキスト文字列**］の書式設定が選択されているときには、このオプションは利用できないからです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

デフォルトにより、次の値が設定されます。

- 固定少数点の精度には 0 が設定されます。
- 指数の精度には 0 が設定されます。
- 16 進数のビット範囲には 0 ～ 31 が設定されます。
- バイナリのビット範囲には 0 ～ 31 が設定されます。

書式設定のコントロール:

- 精度コントロールは整数値を受け付けます。［固定少数点］または［指数］のオプションを選択した場合にのみ有効になります。**ポイント** または **指数オプション** 。このコントロールの整数値の範囲は 0 ～ 8 で、デフォルト値は 0 です。ビット範囲は 2 つのスピン コントロールで構成され、それぞれが受け入れる入力値は整数です。これらのコントロールは、［**16 進数**］ または ［**バイナリ**］ オプションを選択した場合にのみ使用可能になります。これら 2 つのコントロールには、0 ～ 31 の範囲のビット値を指定する必要があります。ビット範囲は、正順または逆順で指定することができます。範囲と同じビット仕様を使用して、単一のビットを指定することもできます。ビット仕様に対するデフォルトの数値範囲は 0 ～ 31 です。範囲は 0 に基づいており、InTouch でビットが指定されているほかの場所でも一致します。

設定済みのアナログ ユーザー入力アニメーションまたはアナログ値の表示アニメーションを含むバージョン 3.5 以前の InTouch の場合、移行したアニメーションには［**テキスト文字列**］書式設定が設定され、詳細な書式設定はオフになります。

最新バージョンの場合、詳細な書式設定オプションは移行されます。

数字としてのテキスト オブジェクトの書式設定

テキスト オブジェクトを使用して、静的または動的な数値を表示できます。［値の表示］-［アナログ アニメーション］リンクをテキスト オブジェクトに関連付けることによってアナログ タグの整数値または実数値を表示できます。

テキスト オブジェクトのアニメーションによって表示されるアナログ タグ値の書式を指定するには、以下の文字を使用します。

文字	説明
#	アナログ値の表示を整数に強制します。
0	実数の指定した各場所の桁を表します。数字の整数部分に先行ゼロを適用します。
、	実数の指定した各場所にカンマを挿入します。
をクリックします。	実数の指定した場所に小数点を配置します。

フォント、サイズ、および色などのその他のテキスト書式設定プロパティは、文字列内に表示されている数値に適用されます。

数字としてのテキスト オブジェクトの書式設定の例

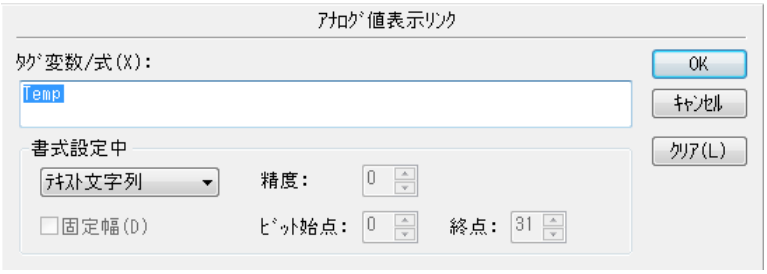
アニメーション化された書式設定文字がテキスト オブジェクト内で使用された場合のアナログ値の表示例を次の表に示します。これらの例では、温度 I/O タグの現在の値は 123.45 Cº です。このタグは、アナログ アニメーション リンクを使用して、テキスト オブジェクトの参照値として割り当てられています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

書式設定されたテキスト オブジェクト	テキスト オブジェクトの表示値
温度 = # C	124 C 温度を整数として表示します。テキスト オブジェクトに入力する必要のある # は 1 つだけです。
温度 = #.# C	123.5 C 「#」文字を使用して、温度の小数部分が 1 桁の数字に丸められます。温度の整数部分は変更されません。
温度 = 00000 C	00124 C 温度の整数部分に先行ゼロを追加します。
温度 = 000.0 C	123.5 C 書式内の 1 つのゼロによって温度の小数部分が 1 桁の数字に丸められます。
温度 = 00.00 C	123.45 C 元の温度値は変更されません。
温度 = 0,000。#	0,123.5 C 温度参照値の整数部分に先行ゼロとカンマが強制されます。「#」文字を使用して、小数部分が 1 桁の数字に丸められます。

数字としてのテキスト オブジェクトの書式設定を指定するには

1. ウィンドウでテキスト オブジェクトを作成して、テキスト オブジェクトに関連付けられた参照値の書式を設定する文字を挿入します。
2. テキスト オブジェクトをダブルクリックして、［**アニメーション リンク**］選択ダイアログ ボックスを表示します。
3. ［**値の表示**］セクションで［**アナログ**］を選択します。［**アナログ値出力**］ダイアログ ボックスが表示されます。



4. ［**タグ変数／式**］フィールドにアナログ タグ名または式を入力します。
5. **[OK]** を選択します。
6. ウィンドウの変更を保存します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

7. アプリケーションを実行して、テキスト オブジェクト内の数字が正しく表示されることを確認します。

産業用グラフィック エディタの操作（IT）

産業用グラフィック エディタは、産業用グラフィックを作成するために使用するツールです。エディタの使用方法については、『産業用グラフィック エディタ ユーザー ガイド』を参照してください。次のセクションでは、産業用グラフィック エディタと共に使用する InTouch HMI 固有の構成または設定を紹介します。

産業用グラフィックの管理

産業用グラフィックは、HMI/SCADA システムのデータを視覚化するために作成できるグラフィックです。

産業用グラフィック エディタを使用して、長方形、線、テキスト要素などの基本要素から産業用グラフィックを作成します。また、産業用グラフィック エディタを使用して、グラフィックのグラフィック ツールボックス ライブラリから産業用グラフィックを埋め込んで設定することもできます。

産業用グラフィック エディタは、産業用グラフィックとしてのスケーラブル ベクター グラフィックス（SVG）のインポートをサポートしています。以下の操作を行うことができます。

- SVG を産業用グラフィック エディタにインポートする。グラフィック要素は、多くのプリミティブを含むことのできる産業用グラフィックに自動的に変換されます。
- 産業用グラフィック エディタの［**ツール**］パネルの［**画像**］アイコンを使用して SVG を挿入する。
- ボタンの **UpImage** と **DownImage** に SVG を使用する。
- ［**スタイル**］設定の**クォリティとステータス**の表示アイコンに SVG を使用する。

SVG および SVG のインポートに関する制限の詳細については、産業用グラフィック エディタ ヘルプの「産業用グラフィックとしての SVG のインポート」セクションを参照してください。

作成した産業用グラフィックは、別のグラフィックまたは HMI システム ウィンドウに埋め込んで、ランタイム時に使用できます。

産業用グラフィックをオブジェクトのテンプレートまたはインスタンスに埋め込み、オブジェクト固有の情報をすばやく簡単に視覚化するいくつかの方法を提供することができます。テンプレートにグラフィックを埋め込むと、1 つのグラフィックを更新するだけで、その変更をアプリケーション全体に適用できます。

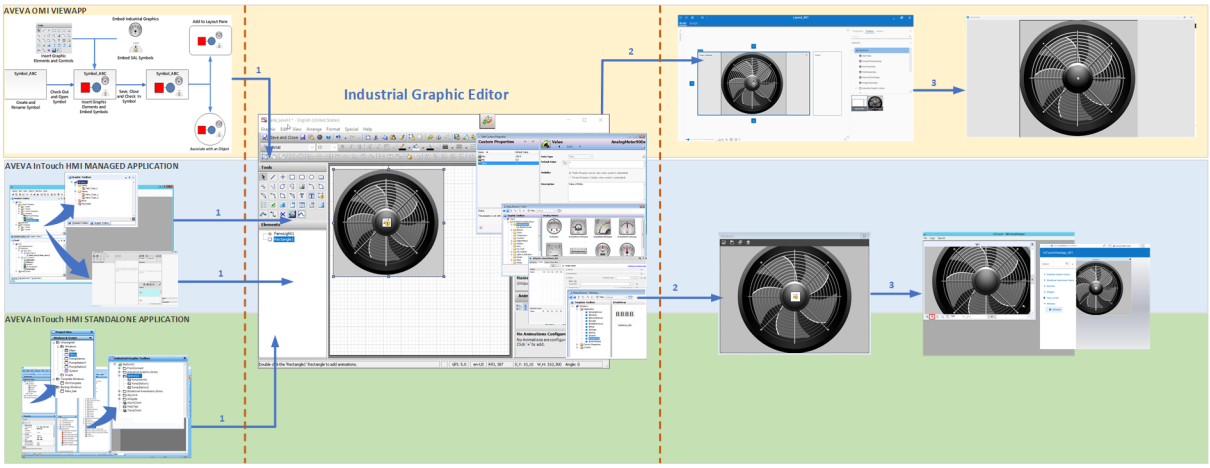
開発要件に応じて、産業用グラフィックを保存する場所と方法を選択できます。

- 作成したグラフィックを再使用可能な標準セットとして保存します（汎用バルブのグラフィックなど）。
- ランタイム時にグラフィックを複数のインスタンスで使用する場合、グラフィックをテンプレートとして保存します。
- 特定のアプリケーションで使用する目的でグラフィックを保存します。

HMI/SCADA アプリケーションと共に使用する産業用グラフィック エディタ ワークフロー

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

産業用グラフィック エディタは、さまざまな HMI/SCADA アプリケーションで使用する標準のグラフィック エディタです。産業用グラフィック エディタを使用すると、工場作業場のアセットを表すグラフィックを作成して HMI/SCADA アプリケーションに埋め込むことができます。以下のセクションでは、全体的なワークフローについて説明します。



マネージド InTouch アプリケーション

InTouch HMI マネージド アプリケーションのワークフローは IDE から始まります。

1. グラフィックを作成または編集するには、
   a. ［視覚化］フォルダから：
      - グラフィックを作成するか、既存のグラフィックを開きます。
      - グラフィックを開いて産業用グラフィック エディタを起動します。

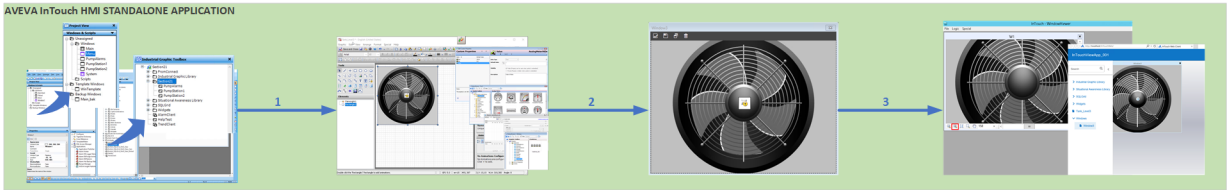
   - または -

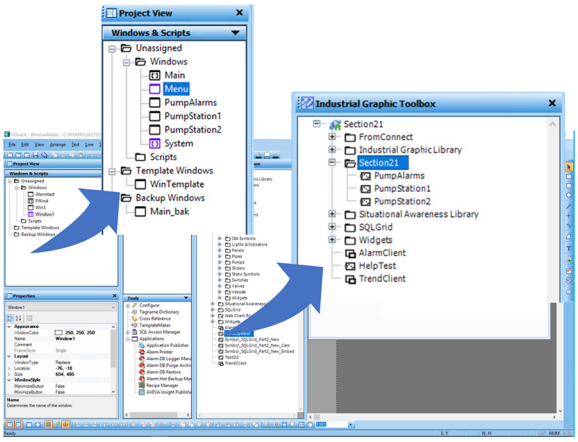
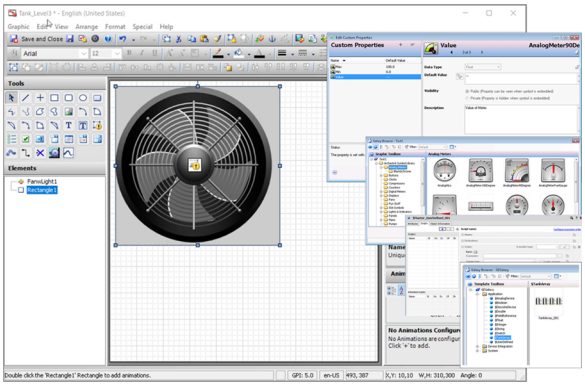
   a. グラフィックを関連付けるオブジェクトの[オブジェクト エディタ]から、次の操作を行います。
      - コンテンツ ペインで［**+**］ボタンを選択してグラフィックをオブジェクトに追加します。
      - ［**コンテンツを編集**］ボタンを選択して産業用グラフィック エディタを起動します。
2. 産業用グラフィック エディタでグラフィックを編集します。必要に応じて、グラフィック要素を変更します。
   - グラフィック要素を挿入し、シンボルを埋め込みます。
   - スクリプト、参照、カスタム プロパティ、およびアニメーションを追加または変更して、グラフィックがアセットを表し、ランタイム時にコマンドおよびデータ変更に応答できるようにします。
3. グラフィックをウィンドウに埋め込みます。
4. ランタイム時にグラフィックを表示するには、WindowViewer または Web Client を起動します。

InTouch スタンドアロン アプリケーション

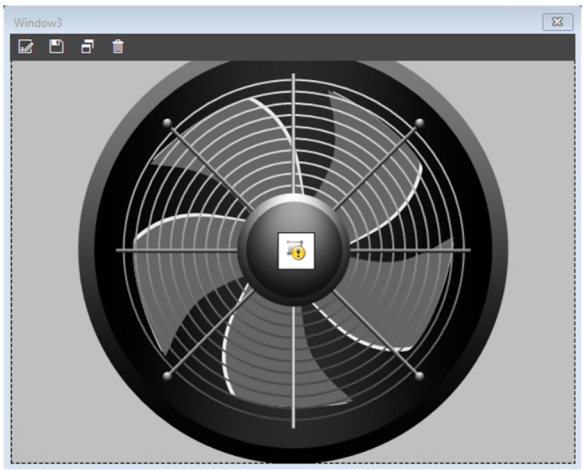
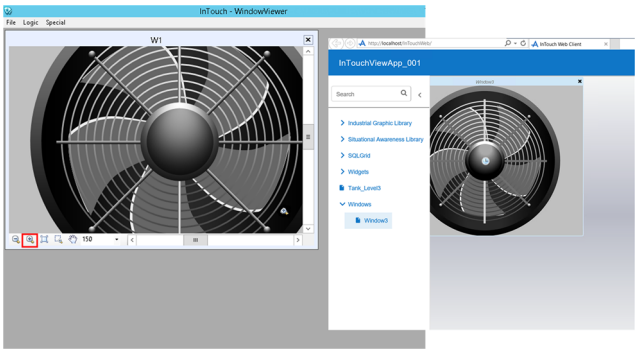
InTouch HMI スタンドアロン アプリケーションのワークフローは WindowMaker から始まります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



手順 1  	InTouch HMI WindowMaker で産業用グラフィック エディタを使用してシンボルを作成します。 グラフィックを編集して産業用グラフィック エディタを起動します。
ステップ 2. 	産業用グラフィック エディタでスクリプト、カスタム プロパティ、タグとオブジェクト参照、スタイル、ツール、アニメーション、および書式設定を使用してグラフィックを構築します。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ステップ 3：  	グラフィックをウィンドウに埋め込みます。
ステップ 4：  	ランタイム時にグラフィックを表示するには、WindowViewer または Web Client を起動します。

グラフィックを開いて編集する

グラフィックから産業用グラフィック エディタを起動できます。

- グラフィック ツールボックス内のシンボル。
- 自動オブジェクト テンプレートまたはインスタンスに含まれるシンボル。
- InTouch ウィンドウに埋め込まれたシンボル。

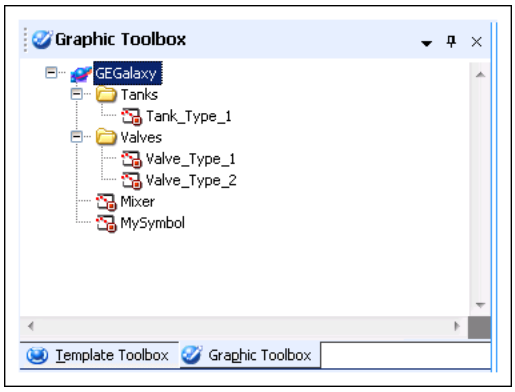
産業用グラフィック エディタで編集目的で開いたグラフィックはチェック アウトされます。他のユーザーはチェック アウトされているグラフィックを編集できません。

産業用グラフィック エディタの複数のインスタンスを同時に開くことができます。しかし、産業用グラフィック エディタの複数のインスタンス中で同じグラフィックを編集することはできません。

グラフィック ツールボックス内のグラフィックを編集するには

1. グラフィック ツールボックスを開きます。
2. 編集するグラフィックを参照します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. グラフィックをダブルクリックします。産業用グラフィック エディタが表示されます。
4. グラフィックを編集します。
5. **保存**を選択して **閉じる**(&S) 産業用グラフィック エディタが閉じ、更新されたシンボルがチェック インされます。

自動オブジェクトに含まれるシンボルを編集する

1. 自動オブジェクトを開きます。
2. [**グラフィック**] タブを選択します。
3. 編集するシンボルを選択し、［**開く**］を選択します。産業用グラフィック エディタが表示されます。
4. シンボルを編集します。
5. **保存**して **閉じる**(&S) 産業用グラフィック エディタが閉じ、更新されたグラフィックがチェック インされます。自動オブジェクトによっては、確認メッセージが表示されることがあります。[**はい**] を選択して保存します。

InTouch ウィンドウに埋め込まれているグラフィックを編集するには

1. 埋め込まれたグラフィックを含む InTouch ウィンドウを WindowMaker で開きます。
2. 編集するグラフィックを右選択し、［**産業用グラフィック "グラフィック名"**］をポイントし、［**シンボルの編集**］を選択します。産業用グラフィック エディタが表示されます。
3. グラフィックを編集します。
4. **保存**して **閉じる**(&S) 産業用グラフィック エディタが閉じ、更新されたグラフィックがチェック インされます。

グラフィックをチェック アウトしたままにするには、産業用グラフィック エディタで［**チェック アウト状態を維持**］を選択します。 このオプションを使用すると、他のユーザーはそのグラフィックを編集用にチェック アウトできなくなります。

シンボル変更の反映

ソースの産業用グラフィックへのすべての変更は、埋め込まれたすべての産業用グラフィック ArchestrA シンボルに反映されます。これは、WindowMaker の産業用グラフィック、およびオートメーション オブジェクトによって継承された産業用グラフィックに影響します。

産業用グラフィックに変更が加えられ、開いている InTouch ウィンドウでそれが使用されている場合、ステータス バーの右下隅に［シンボルの変更］アイコンが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

このアイコンをダブルクリックすると、埋め込まれた産業用グラフィックで変更が更新されます。

以下の例は、シンボルの変更の反映がどのように行われるのかを示しています。

シンボルの変更を反映するには

1. 「新しい Application Server オブジェクト インスタンスの自動作成」の例に従います。
2. WindowMaker で、バルブのシンボルが表示されているウィンドウを開きます。
3. Application Server オブジェクト テンプレート $Valve1 によってホストされている産業用グラフィック ValveSymbol を開きます。
4. いくつかの変更を加え、［**閉じて保存**］を選択します。Application Server オブジェクト インスタンス Valve1_E22 に変更が反映されます。WindowMaker で、［シンボルの変更］アイコンが表示されます。
5. アイコンをダブルクリックして、変更を受け入れます。埋め込まれた産業用グラフィックが更新されます。

シンボルのダイナミック サイズの反映

埋め込まれたシンボルにソース シンボルのサイズの変更が反映される方法を制御できます。たとえば、サイズの変更には以下のようなものがあります。

- シンボルの境界が変更されるように、ソース シンボルの要素の 1 つをサイズ変更する。
- シンボルの境界が変更されるように、ソース シンボルに要素を追加するか、ソース シンボルから要素を削除する。

この機能はダイナミック サイズ変更の反映と呼ばれ、有効化または無効化できます。

ダイナミック サイズの反映の詳細については、『産業用グラフィック エディタ ユーザー ガイド』を参照してください。

WindowMaker と産業用グラフィック エディタの比較

産業用グラフィック エディタを使用すると、InTouch WindowMaker での大部分のタスクを実行できます。また、多くのショートカット キーも使用できます。

産業用グラフィック エディタには、InTouch WindowMaker で使用できない次のような機能が備わっています。

- 追加の要素。
- 追加および改良された要素の外観。
- 追加および改良されたデザイン時の機能。

外観

産業用グラフィック エディタは、InTouch グラフィック設定を拡張します。たとえば、以下を使用できます。

- 線、塗りつぶし、テキストの色のグラデーション。
- 線、塗りつぶし、テキストの色のパターン。
- 線、塗りつぶし、テキストの色のテクスチャ。
- 調節可能な透明度

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- シンボルまたは画面に相対的な塗りつぶし動作

要素

要素は産業用グラフィックを作成するために使用するグラフィック オブジェクトです。産業用グラフィック エディタには、InTouch WindowMaker で使用できない次のような要素が備わっています。

- 曲線および閉曲線。
- 2、3 個の点で定義される円弧、パイ、弦。
- 属性データのクォリティとステータスに応じてアイコンを表示するステータス要素。
- 線ベースの要素を組み合わせて新しい閉じた要素を形作ることで作成するパス グラフィック。
- カレンダー コントロール、日付時間ピッカー コントロールなどの Windows 共通コントロール。

改良された機能

産業用グラフィック エディタは、製造環境を表示することで、生活がより便利になるような、さまざまな改良を提供します。

有用性の向上

産業用グラフィック エディタを使用すれば、要素を容易に選択および設定できます。以下の操作を実行できます。

- リストとキャンバスから要素を選択できます。これにより、他の要素を移動することなく、他の要素の下にある要素を選択できます。
- キャンバスで選択するだけで、要素のプロパティとアニメーションを表示および変更できます。
- グループまたはパス グラフィックを解除することなく、グループとパス グラフィック中の要素を編集できます。これをインライン編集といいます。

スタイルの複製

書式のコピーを利用すれば、1 回の選択で、要素のスタイルを他の要素に適用できます。異なるタイプの要素に対しても可能です。

アニメーションの複製

産業用グラフィック エディタを使用すれば、要素のアニメーションのコピー、切り取り、他の要素への貼り付けを行うことができます。これらの操作は、別のタイプの要素に対して行うこともできます。

要素の配置

産業用グラフィック エディタは、InTouch WindowMaker の配置機能を拡張し、以下の操作が可能になります。

- 水平方向または垂直方向に要素を等間隔で分布する。
- 要素の水平方向および/または垂直方向のサイズを等しくする。
- 水平方向または垂直方向のスペースを増加または減少する。
- 要素間の水平方向または垂直方向のスペースを削除する。
- 要素を誤って移動または編集しないようにロックする。
- デザイン時に回転中心の周りで角度を指定して要素を回転する。
- 同時に複数要素に対してサイズ変更と回転を適用する。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 要素の Z 順序を、前後方向に 1 段階変更する。
- テキスト ボックスまたはボタン内のテキストを整列する。

グループ機能

産業用グラフィック エディタは、InTouch WindowMaker のセルとシンボルの概念の代わりにグループの概念を使用します。以下の操作を実行できます。

- 要素をグループに埋め込む。
- グループを解除することなく、グループ中の各要素 (または埋め込まれたグループ) を編集する。
- 既存のグループにおいて要素を容易に削除または追加する。

カスタム プロパティの拡張性

シンボルまたは埋め込まれたシンボルに対し、カスタム プロパティを追加できます。カスタム プロパティを AutomationObject 属性、要素プロパティ、InTouch タグに関連付けできます。デザイン時とランタイム時に定義済みプロパティを使用すると同じように、カスタム プロパティを使用できます。

注記：ハイフンを含む名前の InTouch タグを参照している Application Server カスタム プロパティは、ランタイム時に動作しません。たとえば、"InTouch:TAG-1" はランタイム時に動作しません。

要素スタイル

要素スタイルはグラフィック要素における 1 つまたは複数の塗りつぶし、線、テキスト、点滅、輪郭、ステータスのプロパティを定義します。要素スタイルをグラフィック要素に適用し、要素スタイルで定義された事前設定プロパティに要素を設定します。要素スタイルで定義された要素のローカル プロパティは無効になります。

要素スタイルは、スクリーン作成に関するドライブ標準を確立する場合、およびシンボルを作成する場合に役立ちます。

その他の改良点

産業用グラフィック エディタを使用すると、以下のことが可能になります。

- スクリプトによって、要素のプロパティとシンボルのカスタム プロパティにアクセスする。
- 要素のタブ順序を設定する。
- 矢印など、線の端点スタイルを使用する。
- 設定情報を失うことなく、要素の固有のアニメーションをダイナミックに無効にする。
- 画像のメタ ファイルと他の画像形式を使用する。
- シンボルの表示方法を向上させるためにアンチエイリアス機能を利用する。

一般的な WindowMaker のタスクと技術の手順

InTouch WindowMaker で実行する大部分の設定は、産業用グラフィック エディタで容易に実行することができます。グラフィック、アニメーション、スクリプトにはいくつかの相違点があります。

グラフィックの使用

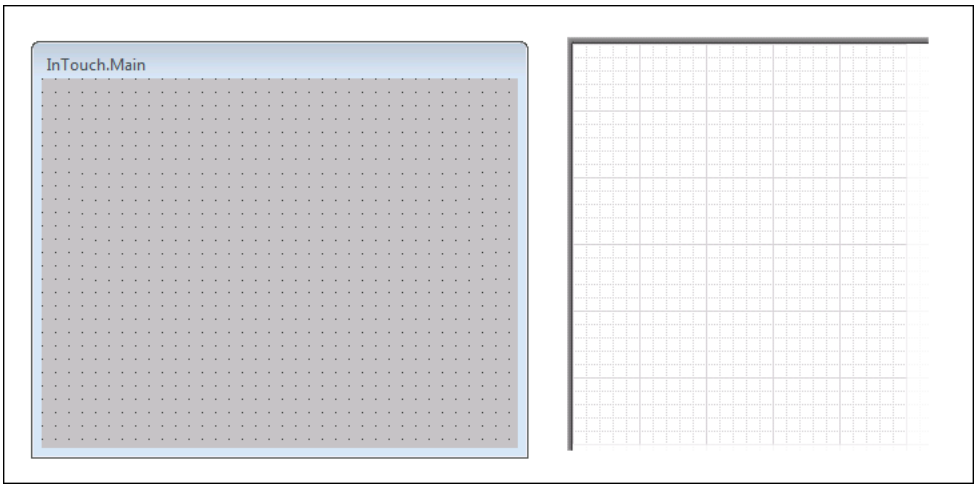
基本的に、産業用グラフィック エディタは InTouch WindowMaker と同じように使用できます。産業用グラフィック エディタに含まれる描画領域では、グラフィック オブジェクトを配置して、生産プロセスの視覚的な表現を作成し、人と機械とのインターフェースを作成できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch で使用するオブジェクト（ActiveX コントロールやいくつかのウィザードなど）には、産業用グラフィック エディタに存在しないものがあります。それらの機能は、より強力で密接に統合できる他のコントロールに置き換えられています。

描画領域の使用

産業用グラフィック エディタの描画領域はキャンバスと呼ばれます。キャンバスは InTouch のウィンドウのように使用できます。そのサイズは 2,000 × 2,000 ピクセルです。



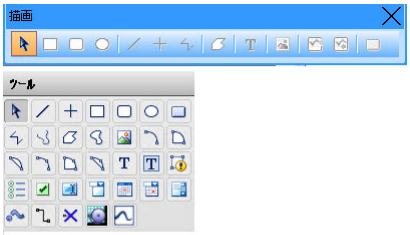
描画領域の色の設定

産業用グラフィック エディタで描画領域の色は設定できません。描画領域の色は透明で、シンボルが埋め込まれた環境の色を継承します。

産業用グラフィックを InTouch のウィンドウに埋め込んだ場合、要素間の領域には、InTouch のウィンドウの色が使用されます。

基本的なオブジェクトの使用

長方形、楕円、折れ線などの InTouch の基本的なオブジェクトは産業用グラフィック エディタと同じように描画できます。産業用グラフィック エディタでは、基本的なオブジェクトは要素と呼ばれます。



複雑なオブジェクトの使用

ActiveX コントロール、ウィザード、セル、シンボルのような InTouch オブジェクトは産業用グラフィック エディタに存在しません。

しかし、スマート シンボルを産業用グラフィックにインポートできます。スマート シンボルをインポートする場合、スマート シンボルの要素とアニメーションが変換されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

産業用グラフィック エディタでは、要素のグループを作成できます。グループ内では、含まれている個別の要素のプロパティは維持されます。グループの TreatAsIcon プロパティを設定すれば、グループの動作を変更できます。

ウィザードの使用

InTouch ウィザードは産業用グラフィックおよびグラフィック ツールボックスにインポートできません。その代り、以下を使用します。

- グラフィック ツールボックスにインポートできる産業用グラフィック ライブラリー
- ツールボックスの一部である Windows コントロール以下を使用できます。
  - ラジオ ボタン グループ
  - チェック ボックス
  - 編集ボックス
  - コンボ ボックス
  - カレンダー コントロール
  - 日付時間ピッカー
  - リスト ボックス

アニメーションの使用

産業用グラフィック エディタでアニメーションを使用して、InTouch WindowMaker と同様に、ランタイム中のシンボルの動作を設定できます。要素またはシンボルに 1 つまたは複数のアニメーションを設定できます。データはさまざまなソースに由来します。

データソースの設定

InTouch WindowMaker では、タグ名ディクショナリを使用して値を保持する変数を定義します。産業用グラフィック エディタのデータ ソースを以下に示します。

- オートメーション オブジェクトの属性
- シンボルのカスタム プロパティおよび継承されたプロパティ
- InTouch タグ(産業用グラフィック エディタでは、InTouch タグに直得接続できる特殊な InTouch 参照が使用されます)

データ型の使用

産業用グラフィックは InTouch のデータ型とは別のデータ型を使用します。

以下の表には、両方のデータ型と相互の対応関係が示されています。

グラフィック エディタでは、「**InTouch:**」と入力しなくても	**Application Server<1/>Application Server**	説明
論理型	ブール型	ブール型。例: 0 または 1
整数型	整数型	整数値。例: -4、7、または 22

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グラフィック エディタでは、「**InTouch:**」と入力しなくても	**Application Server<1/>Application Server**	説明
実数型	浮動小数点または倍精度	精度の異なる浮動小数点値または倍精度値。3.141、-5.332、1.343e+17 など。 **Float**:32 ビット。IEEE 単精度浮動小数点。6 ～ 7 桁の有効桁が必要な場合に使用します。デフォルトは NAN です。 **Double**:64 ビット。IEEE 倍精度。15 ～ 16 桁の有効桁が必要な場合に使用します。デフォルトは NAN です。
メッセージ型	文字列	文字列値。例: "Hello World"
該当なし	日付時刻	日付時刻値。例："04/13/2006 04:03:22.222 AM"。
該当なし	経過時間	経過時間を秒単位で表す浮動小数点値。以下の形式で表示されることがありますが、浮動小数点値で保存されます。 [–][DDDDDD] [HH:MM:]SS[.fffffff] 値を以下に示します。 • DDDDDD: 0 ～ 999999 • HH: 0 ～ 23 • MM: 0 ～ 59 • SS: 0 ～ 59 • fffffff は小数点の右側にある小数部分を示します。 経過時間は正の値にも負の値にもなります。
該当なし	国際化された文字列	特殊文字を保存できる特殊な文字列データ型。

産業用グラフィックは、Galaxy からデータを取得するように設定できます。

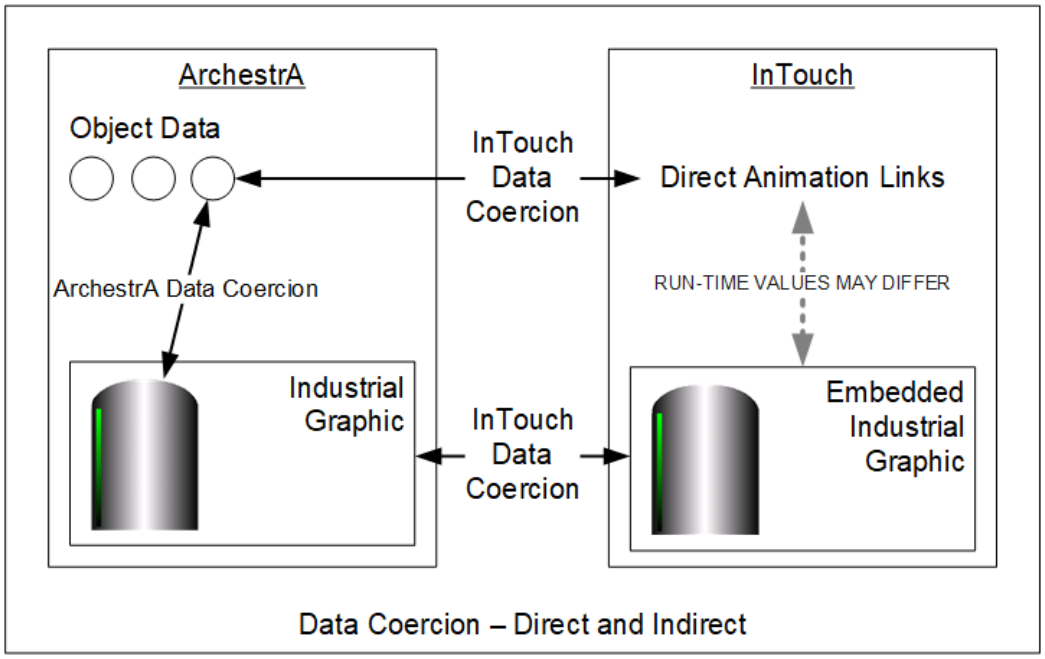
使用するデータ型とソース データ型が異なる場合、データは IDE の強制変換のルールに基づいて強制変換され、アニメーション中では "-10" の文字列値は "True" に強制変換されます。

この産業用グラフィックを InTouch のウィンドウに埋め込んだ場合、アニメーション リンクのデータ型は InTouch の強制変換の規則に基づいて強制変換されます。埋め込まれた産業用グラフィックは InTouch HMI で "True" と表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

しかし、InTouch HMI でオリジナル データ ソースをポイントする論理型アニメーション表示リンクを直接作成する場合、結果の値が異なる場合があります。

この例では、文字列値 "-10" は InTouch HMI で "False" と表示されます。



アニメーションの使用

InTouch のアニメーションを設定するには、［**アニメーション リンク**］ダイアログ ボックスを使用します。このダイアログ ボックスを開くには、InTouch オブジェクトをダブルクリックします。

産業用グラフィック エディタのアニメーションを設定するには、通常は要素をダブルクリックすることで表示される［**アニメーションの編集**］ダイアログ ボックスを使用します。

いくつかのアニメーション タイプは異なっており、他のものは設定を単純化するためにグループ化されています。以下の表を使用して、産業用グラフィック エディタの同等のアニメーション タイプを見つけてください。

InTouch アニメーション	**産業用グラフィック エディタ アニメーション**
ユーザー入力 - 論理型	ユーザー入力 - ブール型
ユーザー入力 - アナログ型	ユーザー入力 - アナログ型
ユーザー入力 - 文字列	ユーザー入力 - 文字列
スライダー - 垂直	垂直スライダー
スライダー - 水平	水平スライダー
タッチ押しボタン - 論理値	押しボタン - ブール値
アクション	アクション スクリプト

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アニメーション	産業用グラフィック エディタ アニメーション
ウィンドウ表示	(サポートされていません)
ウィンドウ消去	(サポートされていません)
線の色 - 論理型	線スタイル - ブール型
線の色 - アナログ型	線スタイル - 真理値表
線の色 - 論理型アラーム	線スタイルに変換
線の色 - アナログ アラーム	線スタイルに変換
塗りつぶしの色 - 論理型	塗りつぶしスタイル - ブール値
塗りつぶしの色 - アナログ型	塗りつぶしスタイル - 真理値表
塗りつぶしの色 - 論理型アラーム	塗りつぶしスタイルに変換
塗りつぶしの色 - アナログ アラーム	塗りつぶしスタイルに変換
テキストの色 - 論理型	テキスト スタイル - ブール値
テキストの色 - アナログ型	テキスト スタイル - 真理値表
テキスト色 - 論理型アラーム	テキスト スタイルに変換
テキスト色 - アナログ アラーム	テキスト スタイルに変換
オブジェクト サイズ - 高さ	高さ
オブジェクト サイズ - 幅	幅
位置 - 垂直	水平位置
位置 - 水平	垂直位置
塗りつぶしパーセント - 垂直	垂直塗りつぶしパーセント
塗りつぶしパーセント - 水平	水平塗りつぶしパーセント
その他 - 表示オン/オフ	表示オン/オフ
その他 - 点滅	点滅
その他 - 向き	向き
その他 - 無効	無効化
その他 - ツールヒント	ツールヒント
値の表示 - 論理型	値の表示 - ブール型
値の表示 - アナログ型	値の表示 - アナログ型

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アニメーション	産業用グラフィック エディタ アニメーション
値の表示 - 文字列型	値の表示 - 文字列型

スクリプトの使用

産業用グラフィック エディタでは InTouch WindowMaker と同じようにスクリプトを設定できます。ただし、小さな違いがあります。

InTouch スクリプト	産業用グラフィック エディタ スクリプト
アプリケーション スクリプト	(使用できません)
ウィンドウ スクリプト	シンボルの定義済みスクリプト
キー スクリプト	キー実行条件付きのアクション スクリプト アニメーション
アクション スクリプト	OnTrue、OnFalse、WhileTrue、WhileFalse 実行条件付きのシンボル名前付きスクリプト
データ変化スクリプト	DataChange 実行条件付きのシンボル名前付きスクリプト
ｸｲｯｸ関数	(使用できません)
ActiveX イベント スクリプト	(使用できません)
アクション スクリプト	アクション スクリプト アニメーション

アプリケーション スクリプトの使用

InTouch HMI で、アプリケーション スクリプトのトリガは以下の通りです。

- WindowViewer でアプリケーションが起動したときに 1 回実行。
- WindowViewer でアプリケーションが実行されている間、定期的に実行。
- WindowViewer でアプリケーションがシャットダウンしたときに 1 回実行。

産業用グラフィックは InTouch のアプリケーションに対応し、シンボルに直接関連付けられた定義済みスクリプトを設定できます。例：

- 表示時
- 表示中
- 非表示時

キー スクリプトの使用

産業用グラフィック エディタでキー スクリプトは使用できませんが、キーの組み合わせによって有効化されるアクション スクリプトに対し、要素を関連付けることができます。

条件スクリプトの使用

シンボル スクリプト機能を使用することで、条件が満たされたときに実行するスクリプトを設定できます。これにより、値または式が以下のときに実行する実行条件を定義できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 条件を満たしている間(True の間)
- 条件を満たしたとき(True の時)
- 条件を満たしていない間(False の間)
- 条件を満たさなくなったとき(False の時)

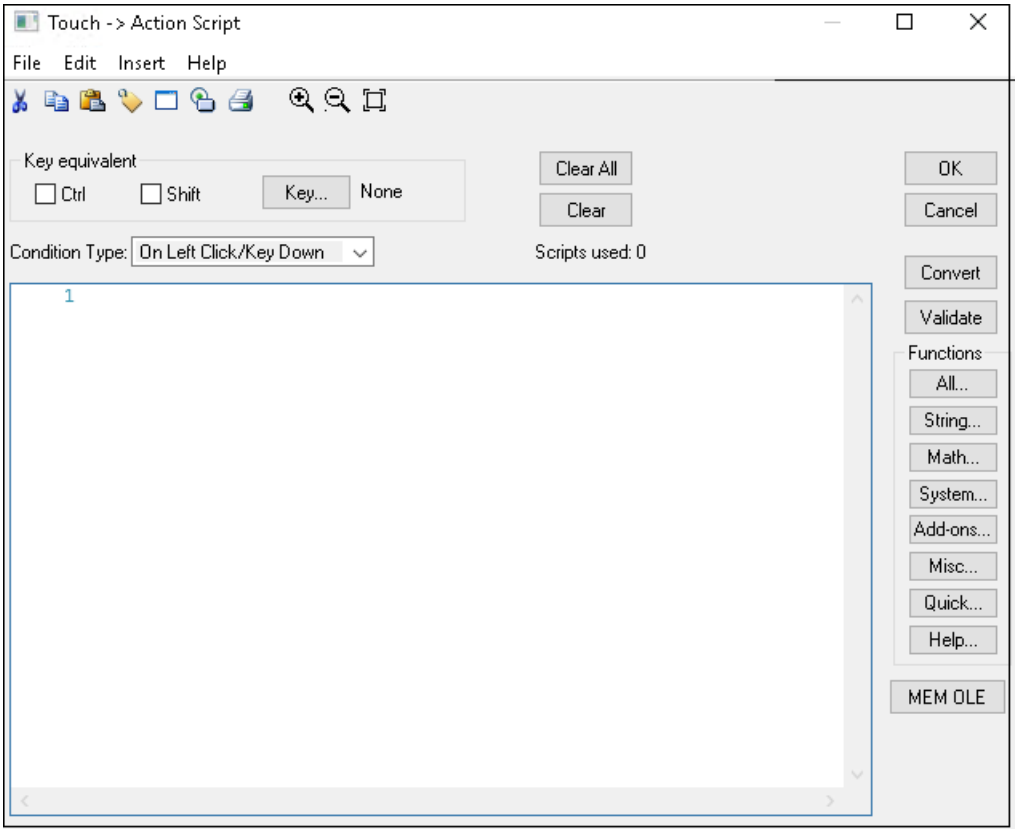
データ変化スクリプトの使用

シンボル スクリプト機能を使用することで、値または式が変更されたときに実行するスクリプトを設定できます。これにより、値または式が変化したときに実行する実行条件を定義できます。

アクション スクリプトの使用

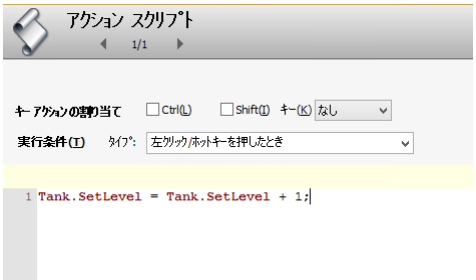
産業用グラフィック エディタでは、InTouch WindowMaker と同じようにアクション スクリプトを設定できます。ランタイムのユーザーがマウスを使用するかキーを押して要素を操作すると、アクション スクリプトを実行できます。

InTouch アクション スクリプト ウィンドウを使用してアクション スクリプトを作成します。



産業用グラフィック エディタのアクション スクリプト ウィンドウを使用してアクション スクリプトを作成します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



個別の要素、またはシンボル全体に対してアクション スクリプトを設定できます。

InTouch WindowMaker の定義済み関数の多くは産業用グラフィック エディタで使用できます。産業用グラフィックに使用できる InTouch の定義済み関数の詳細なリストについては、「アクション スクリプトのインポートあくしょんすくりぷとのいんぽーと」を参照してください。

アプリケーション スクリプト、キー スクリプトなど、他の InTouch スクリプト タイプは Application Server オートメーション オブジェクトで設定できます。

InTouch タグへのアニメーションの接続

要素のアニメーションと外観を InTouch タグに接続できます。InTouch タグは、要素のアニメーションと外観を制御する値をランタイム時に提供します。

以下の方法で要素のアニメーションを InTouch タグに接続できます。

- ［**intouch:tag**］構文で参照を設定します。この構文は、産業用グラフィックを使用する InTouch HMI スタンドアロン アプリケーション内の参照タグでは必要ありません。
- カスタム プロパティを使用して、InTouch 内の埋め込まれた産業用グラフィックのカスタム プロパティを、InTouch タグを参照するように設定します。属性として InTouch タグを含む管理対象の InTouchViewApp オブジェクトへの Application Server の属性参照を設定します。InTouchViewApp オブジェクトは InTouchProxy オブジェクトの機能を使用します。
- アイテムとして InTouch タグを含む InTouchProxy オブジェクトへの Application Server の属性参照を設定します。これは Application Server 属性参照の設定の特別な例です。

重要 : 履歴サマリ データ型は、AVEVA OMI ViewApps または InTouch HMI マネージド アプリケーションを対象とした Application Server オブジェクト属性とのみ機能します。InTouch HMI モダン アプリケーションでは、履歴サマリ データ型が割り当てられているカスタム プロパティを使用しないでください。

InTouch:Tagname 構文の使用

intouch:tagname 構文を使用すると、アニメーションはグラフィックが使用されるノードの InTouch タグに接続します。この構文の使用には制限があります。

- Application Server とは異なり、ブール値として True と False は使用できません。代わりに 1 と 0 を使用します。
- InTouch SuperTag を参照する場合、代わりに以下の構文を使用します。

  属性 `("intouch:SuperTag\Member")`
- " 接頭辞は、産業用グラフィックを含む InTouch スタンドアロン アプリケーションでは不要です。タグの参照およびランタイム タグのバインディングは "InTouch:" 接頭辞なしで機能します。
- ハイフンを含む名前の InTouch タグを参照している Application Server カスタム プロパティは、ランタイム時に動作しません。たとえば、"InTouch:TAG-1" はランタイム時に動作しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

入力モードの設定

いくつかのボックスで、静的、および/または属性と要素のプロパティに対する参照を使用する値または式を入力できます。両方の入力メソッドをサポートするボックスには［**入力モード**］選択アイコンがあります。

以下を選択します。

- ［**静的モード入力**］アイコンを選択すると、3.141 や「Test」など、静的な文字の値または式を指定できます。
- ［**参照モード入力**］アイコンを選択すると、属性または要素プロパティへの参照を指定できます。

注記：静的な文字列値を参照モードで（参照と共に、または単独で）使用するには、二重引用符で囲むことができます（例:「+Tank_001.Desc

設定フィールドまたはスクリプトの固有のプロパティの 1 つを参照する要素を設定するとき、そのプロパティ名のみを使用することができます。産業用グラフィックの場合、オートメーション オブジェクトで使用されている「me.」のような自己を参照するキーワードはありません。

しかし、「me.」キーワードを使用して 現在、設定している産業用グラフィックをホストしているオートメーション オブジェクトの属性を参照することができます。

InTouchViewApp 属性へのアニメーションの接続

InTouch タグを参照するには、最初に以下を実行する必要があります。

- WindowMaker で InTouchViewApp テンプレートを派生させて設定することによって、管理対象の InTouch アプリケーションを作成します。
- InTouchViewApp から派生したテンプレートのインスタンスを派生させます。

InTouch タグは InTouchViewApp オブジェクトのインスタンスの属性で表されます。

Galaxy ブラウザの［InTouch タグ ブラウザ］タブの使用

産業用グラフィック アニメーションまたはクライアント スクリプトに InTouch タグを必要とする参照を設定するときに、Galaxy ブラウザから直接 InTouch タグを選択できます。

アニメーション エディタまたはスクリプト エディタから呼び出したとき、Galaxy ブラウザには［**InTouch タグ ブラウザ**］タブが［**属性ブラウザ**］タブおよび［**要素ブラウザ**］タブと共に表示されます。

［**InTouch タグ ブラウザ**］タブでは、左ペインに現在の Galaxy の InTouchViewApp インスタンスとテンプレートがすべて一覧表示されます。右ペインには選択した InTouchViewApp に対する InTouch タグが表示されます。［**ドットフィールド**］リスト ボックスには、選択したタグに関連付けられたドットフィールドが表示されます。

右ペインの下にある［**ドットフィールド**］リスト ボックスでは、選択したタグのドットフィールドを指定できます。

［**InTouch タグ ブラウザ**］タブは以下のように動作します。

- InTouch タグ ブラウザの機能は、アニメーション エディタまたはスクリプト エディタのみから利用できます。
- Galaxy ブラウザはタグ名ディクショナリから InTouch タグを読み込みます。
- InTouch HMI がインストールされているかどうかに関係なく、タグ名ディクショナリのコンポーネントはインストールされ、Galaxy ブラウザで利用可能です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- タグは Galaxy ブラウザを閉じて再び開いたときにだけ更新されます。
- すべてのタグは Galaxy ブラウザを閉じて再び開くまでメモリに保持されます。
- 現在のユーザーが InTouchViewApp をチェック アウトした場合、Galaxy ブラウザが InTouchViewApp の最新のタグ名ディクショナリの内容を読み込みます。
- ユーザーが InTouchViewApp にチェック インしてアクセスした場合、Galaxy ブラウザは常にタグ名ディクショナリのチェック イン バージョンを読み込みます。
- 現在のユーザー以外のユーザーが InTouchViewApp をチェック アウトした場合、Galaxy ブラウザはもっとも最近チェック インされたタグ名ディクショナリを読み込みます。
- InTouchViewApp テンプレートを選択する場合、出力参照文字列構文は InTouch:selectedTag です。InTouchViewApp インスタンスを選択する場合、出力参照文字列構文は SelectedInTouchViewAppInstance.selectedTag です。

アニメーションを InTouch タグに接続するには

1. 要素をダブルクリックします。

   ［**アニメーションの編集**］ダイアログ ボックスが表示されます。
2. アニメーション リストからアニメーションを選択します。
3. パラメータを選択します。
4. [参照] ボタンを選択します。

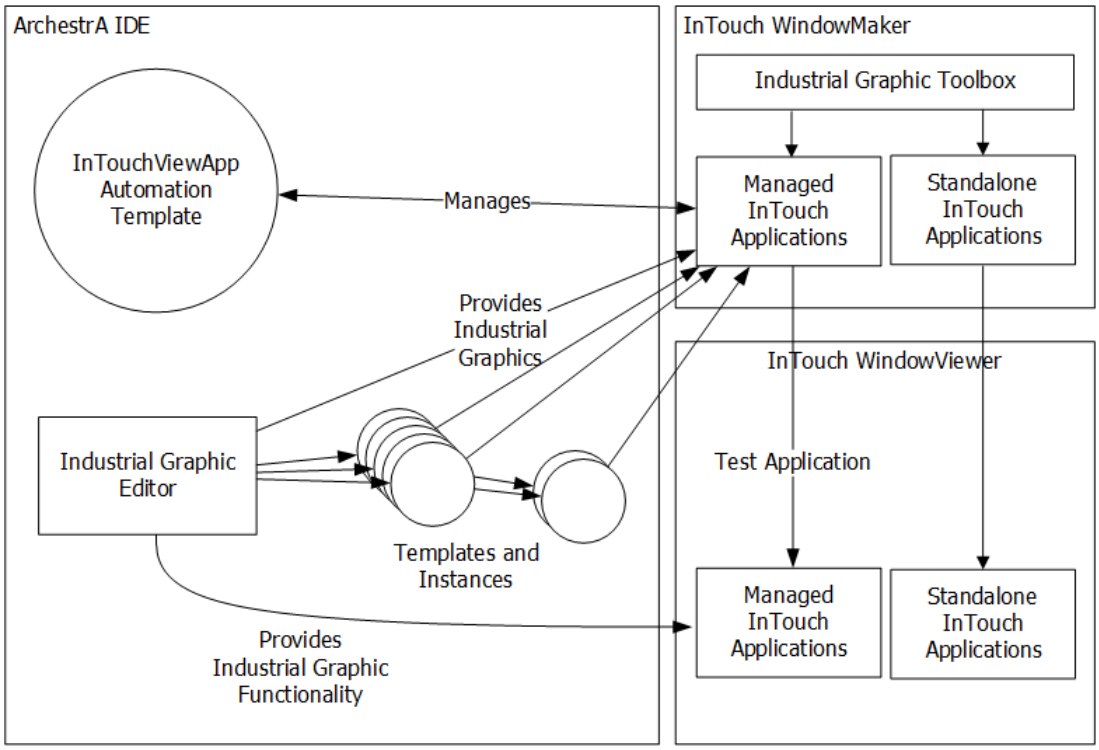
   ［**Galaxy ブラウザ**］が表示されます。
5. ［**InTouch タグ ブラウザ**］タブを選択すると［**InTouch タグ ブラウザ**］ページが表示されます。
6. 管理対象の InTouch アプリケーションに対応する InTouchViewApp オブジェクトを選択します。右ペインに InTouch タグが表示されます。
7. タグを選択して、［**OK**］を選択します。

   InTouch タグに対して選択した参照が設定ボックスに表示されます。

WindowMaker での産業用グラフィックの使用

産業用グラフィック エディタで作成した産業用グラフィックは、マネージドまたはスタンドアロン InTouch アプリケーションで使用できます。WindowMaker の産業用グラフィック ツールボックスから産業用グラフィックを直接追加することもできます。産業用グラフィックと Situational Awareness Library シンボルの操作に関する詳細については、『*産業用グラフィック エディタ ユーザー ガイド*』または WindowMaker ヘルプを参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



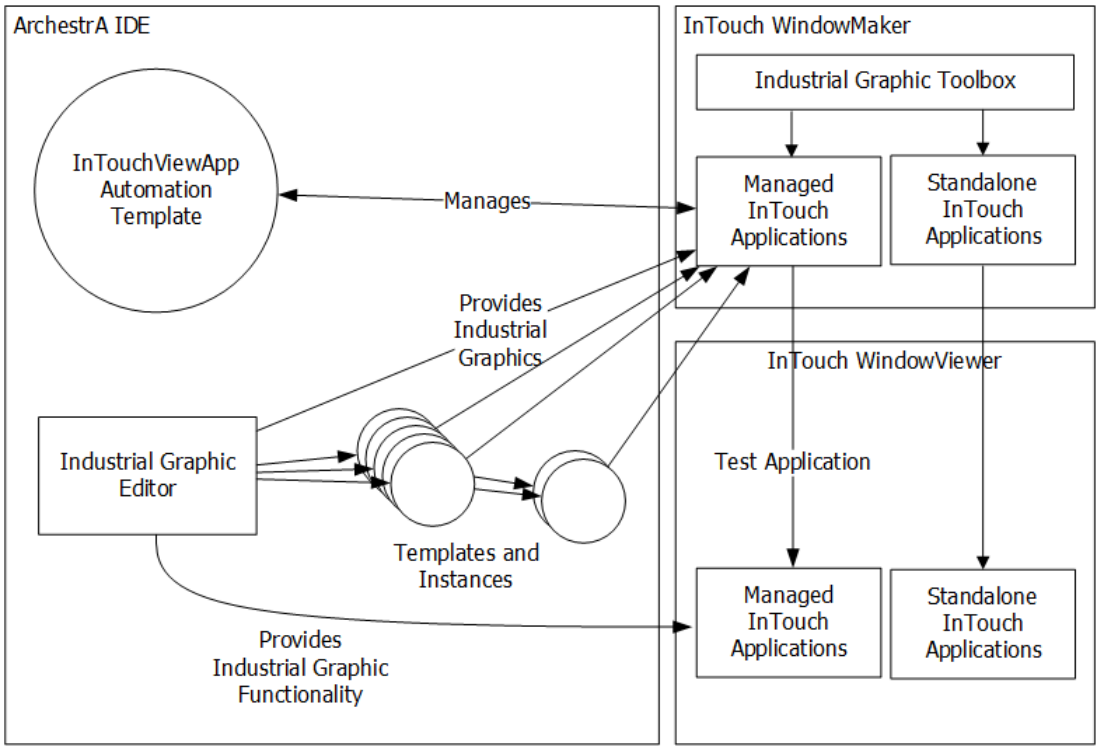
以下の操作を実行できます。

- InTouch ウィンドウへの産業用グラフィックの埋め込み。
- 産業用グラフィック ツールボックスから InTouch ウィンドウへの産業用グラフィックの直接のドラッグ アンド ドロップ。
- 埋め込まれた産業用グラフィックのサイズ変更。
- 産業用グラフィックへの制限された数の InTouch アニメーションの追加。
- 埋め込まれた産業用グラフィックのカスタム プロパティの設定。
- 産業用グラフィック エディタの起動。
- WindowViewer での産業用グラフィックのテスト。
- 埋め込まれた産業用グラフィックをホストするオートメーション オブジェクトの新規インスタンスの作成。

WindowMaker での産業用グラフィックの使用について

産業用グラフィック エディタで作成した産業用グラフィックは、マネージドまたはスタンドアロン InTouch アプリケーションで使用できます。WindowMaker の産業用グラフィック ツールボックスから産業用グラフィックを直接追加することもできます。産業用グラフィックと Situational Awareness Library シンボルの操作に関する詳細については、『*産業用グラフィック エディタ ユーザー ガイド*』または WindowMaker ヘルプを参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



以下の操作を実行できます。

- InTouch ウィンドウへの産業用グラフィックの埋め込み。
- 産業用グラフィック ツールボックスから InTouch ウィンドウへの産業用グラフィックの直接のドラッグ アンド ドロップ。
- 埋め込まれた産業用グラフィックのサイズ変更。
- 産業用グラフィックへの制限された数の InTouch アニメーションの追加。
- 埋め込まれた産業用グラフィックのカスタム プロパティの設定。
- 産業用グラフィック エディタの起動。
- WindowViewer での産業用グラフィックのテスト。
- 埋め込まれた産業用グラフィックをホストする自動オブジェクトの新規インスタンスの作成。

InTouch ウィンドウへの産業用グラフィックの埋め込み。

産業用グラフィックをマネージドおよびスタンドアロン InTouch アプリケーションの InTouch ウィンドウに埋め込むことができます。

産業用グラフィックは、以下の一部にすることができます。

- グラフィック ツールボックス
- オブジェクト テンプレート。
- オブジェクト インスタンス。

埋め込むことのできる産業用グラフィックは、要素に要素スタイルが適用されたシンボルおよびシンボル ウィザード エディタを使用して作成されたシンボル ウィザードです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パブリッシュ済みアプリケーションに産業用グラフィックを含むマネージド InTouch アプリケーションをパブリッシュできます。ただし、新しい産業用グラフィックを追加することや、パブリッシュ済みアプリケーション内の既存のシンボルを編集することはできません。

オートメーション テンプレートからの産業用グラフィックの埋め込み

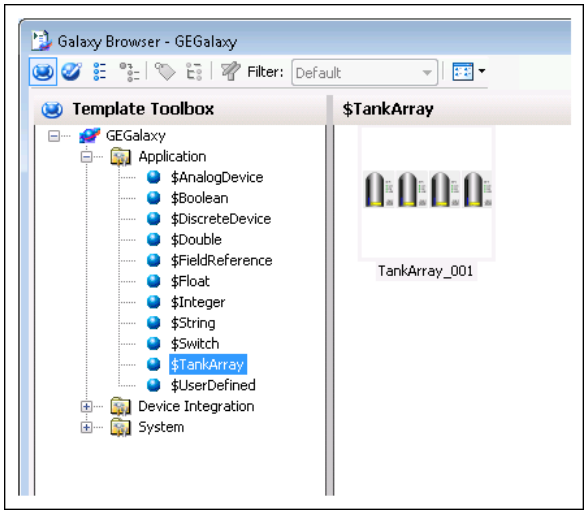
産業用グラフィックをホストするオートメーション テンプレートから産業用グラフィックを埋め込むことができます。同時に、選択したテンプレートから新しく派生したインスタンスが作成されます。

InTouch ウィンドウ上に既に存在する産業用グラフィックに基づく新規インスタンスの作成の詳細については、「WindowViewer での産業用グラフィックのテスト」を参照してください。

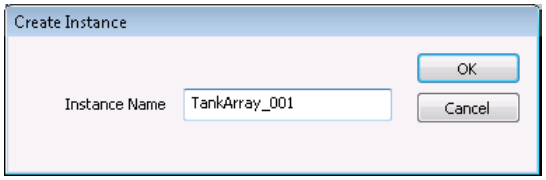
オートメーション テンプレートから産業用グラフィックを埋め込むには

1. WindowMaker を開きます。
2. オブジェクトを右クリックして［**産業用グラフィックの埋め込み**］を選択します。

   ［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。
3. ［テンプレート　ツールボックス］アイコンを選択します。左側に［**テンプレート　ツールボックス**］リストが表示されます。



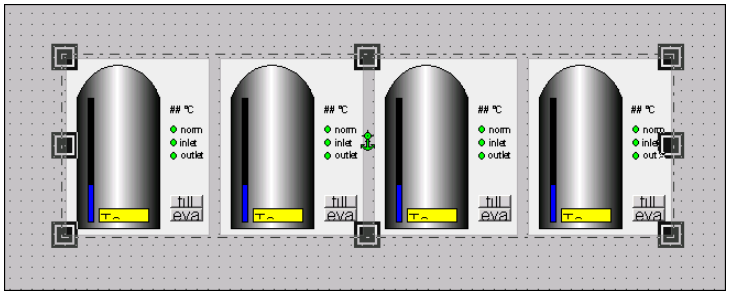
4. 埋め込む産業用グラフィックが含まれているテンプレートを選択します。選択したテンプレートに含まれている産業用グラフィックが右側に表示されます。
5. 埋め込む産業用グラフィックを選択し、［**OK**］を選択します。InTouch ウィンドウ上にポインタを置くと、Galaxy ブラウザが閉じて挿入アイコンが表示されます。
6. InTouch ウィンドウ内で産業用グラフィックを埋め込む場所を選択します。［**インスタンスを作成**］ダイアログ ボックスが表示されます。



7. ［**インスタンス名**］ボックスで、インスタンスの名前を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

8. **[OK]** を選択します。指定した名前を持つテンプレートから、インスタンスが自動的に派生します。InTouch ウィンドウにシンボルが埋め込まれます。



インスタンスからの産業用グラフィックスの埋め込み

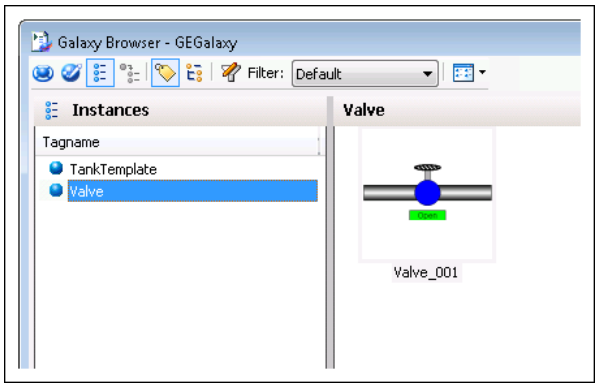
産業用グラフィックが関連付けられているインスタンスから産業用グラフィックを埋め込むことができます。

インスタンスから産業用グラフィックを埋め込むと、シンボルがそのインスタンスに関連付けられます。

インスタンスから産業用グラフィックを埋め込むには

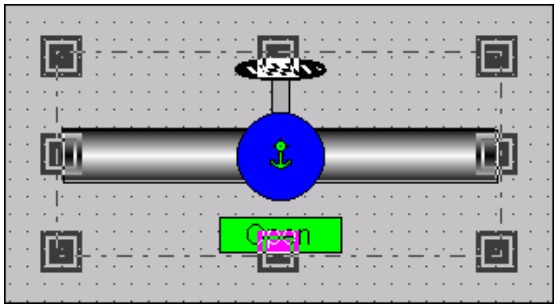
1. WindowMaker を開きます。
2. オブジェクトを右選択して［**産業用グラフィックの埋め込み**］を選択します。

   ［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。
3. ［インスタンス］アイコンを選択します。左側に［**インスタンス**］リストが表示されます。



4. 埋め込む産業用グラフィックが含まれているテンプレートを選択します。選択したインスタンスに関連付けられている産業用グラフィックが右側に表示されます。
5. 埋め込む産業用グラフィックを選択し、［**OK**］を選択します。InTouch ウィンドウ上にポインタを置くと、Galaxy ブラウザが閉じて挿入アイコンが表示されます。
6. [InTouch]ウィンドウ内の場所を選択して、Industrial Graphic を埋め込みます。InTouch ウィンドウにシンボルが埋め込まれます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



グラフィック ツールボックスからの産業用グラフィックの埋め込み

グラフィック ツールボックスから産業用グラフィックを埋め込むことができます。

グラフィック ツールボックスから産業用グラフィックを埋め込むには

オプション 1

1. WindowMaker を開きます。

   手記： Industrial Graphics のロード中や、グラフィックツールボックスの進行状況バーが表示されている場合は、Industrial Graphics に関連する操作は実行しないでください。

2. ［産業用グラフィック］ペインから、目的の産業用グラフィック フォルダを選択します。

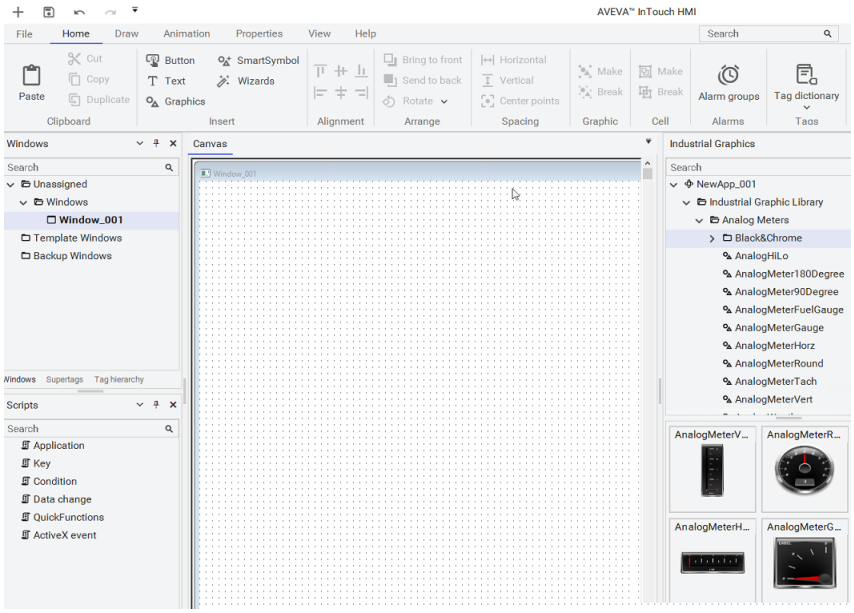
   そのフォルダの下に産業用グラフィックがサムネイルとして表示されます。

   または

   検索ボックスで目的の産業用グラフィックを検索します。

3. 目的のグラフィックを選択して、キャンバスにドラッグ アンド ドロップします。

   枠ウィンドウが作成されます。

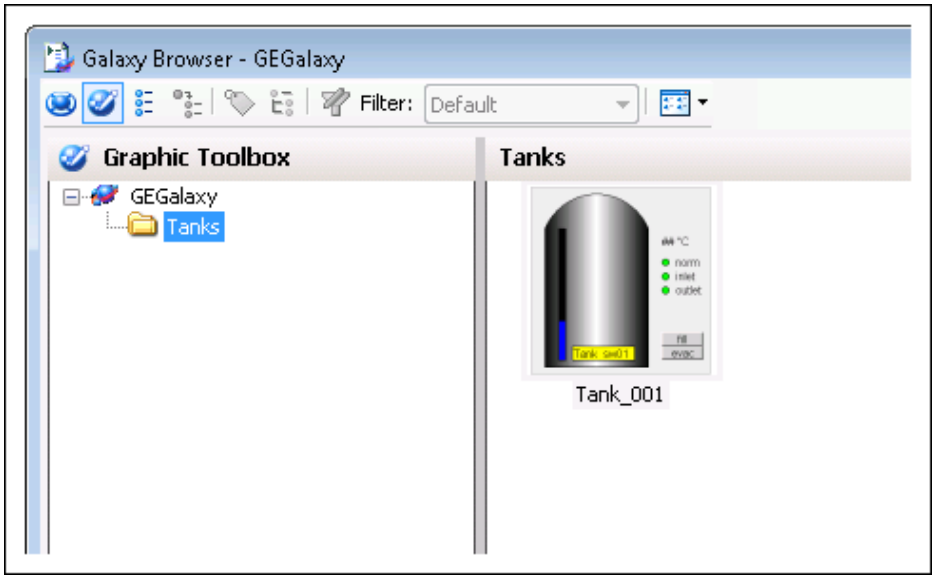


オプション 2

1. WindowMaker を開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. オブジェクトを右クリックして［**産業用グラフィックの埋め込み**］を選択します。
3. ［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。
4. グラフィック ツールボックスを開きます。左側に［**グラフィック ツールボックス**］リストが表示されます。



5. 埋め込む産業用グラフィックを選択し、［**OK**］を選択します。ポインタを InTouch ウィンドウ上に置くと、挿入アイコンが表示されます。
6. InTouch ウィンドウ内の場所を選択して、Industrial Graphic を埋め込みます。InTouch ウィンドウにシンボルが埋め込まれます。

要素スタイルが適用された産業用グラフィックの埋め込み

要素スタイルが適用された産業用グラフィックを埋め込むことができます。要素スタイルは、グラフィックの塗りつぶし、線、テキスト、点滅、および外郭プロパティを定義します。要素スタイルで定義された表示プロパティがグラフィックに適用されます。要素スタイルを使用すると、要素に一貫したスタイルを適用できます。要素スタイルを使用すると、画面ビルダやシンボルを作成するその他のユーザーの表示標準を確立することもできます。

要素スタイルが適用された産業用グラフィックは、その他の産業用グラフィックと同様に InTouch ウィンドウに埋め込まれます。

産業用グラフィックでの要素スタイルの使用の詳細については、『*産業用グラフィック エディタ ユーザー ガイド*』の「要素スタイルの操作」を参照してください。

マネージド InTouch アプリケーションの要素スタイル

WindowViewer で一度に開くことのできるアプリケーションは 1 つだけです。プラットフォームがローカル ノードに配置されている場合、他のスタンドアロン アプリケーションで設定されているスタイルよりも Galaxy で設定したスタイルが優先されます。

シンボルの埋め込みウィザード

System Platform IDE からシンボル ウィザードをマネージド InTouch アプリケーションのウィンドウに埋め込むことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

シンボル ウィザードを埋め込むには

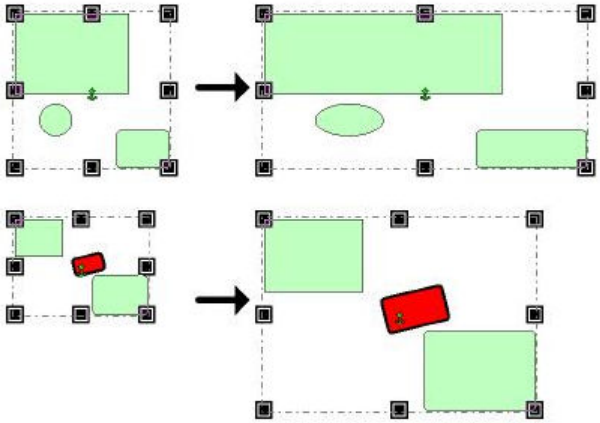
1. WindowMaker を開き、埋め込まれたシンボルを含むウィンドウを表示します。
2. オブジェクトを右クリックして［**産業用グラフィックの埋め込み**］を選択します。

   ［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。
3. グラフィック ツールボックスを開きます。 左側に［**グラフィック ツールボックス**］リストが表示されます。
4. アプリケーションに埋め込むシンボル ウィザードを選択して、［**OK**］を選択します。
5. [InTouch] ウィンドウ内でシンボルウィザードを埋め込む場所を選択します。デフォルトのシンボル ウィザード設定を使用して、シンボルが InTouch ウィンドウに埋め込まれます。

埋め込まれた産業用グラフィックのサイズ変更。

InTouch ウィンドウに産業用グラフィックを埋め込んだ後、他の InTouch オブジェクトで行う場合と同様に、ハンドルまたは幅および高さの値エントリを使用してサイズを変更できます。

ただし、シンボルに少なくとも 1 つの回転要素が含まれている場合、産業用グラフィックのサイズ変更は縦横の比率を維持した状態でのみ行うことができます。



埋め込まれた産業用グラフィックのサイズは、その最小サイズよりも小さいサイズに変更することはできません。最小サイズは、含まれている要素のペン幅によって決まります。

埋め込まれた産業用グラフィックは、産業用グラフィック エディタで作成されたときの元のサイズにリセットできます。

埋め込まれた産業用グラフィックのサイズを変更するには

1. 産業用グラフィックを選択すると、ハンドルが表示されます。
2. 以下のいずれかを実行します。
   - ハンドルの 1 つをドラッグして、産業用グラフィックのサイズを変更します。
   - ステータス バーの［**W**］ボックスおよび［**H**］ボックスに、幅および高さを入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

埋め込まれた産業用グラフィックのサイズを元のサイズに戻すには

- 元のサイズに戻す産業用グラフィックを右クリックし、[**産業用グラフィック**] をポイントし、次に [**シンボル - オリジナル サイズ**] を選択します。埋め込まれた産業用グラフィックが元のサイズに戻ります。

WindowMaker での産業用グラフィックの設定

WindowMaker に埋め込まれた産業用グラフィックは、以下の方法で設定できます。

- コピー、切り取り、貼り付け、複製、サイズ変更、移動、および削除などの標準の編集を行う。
- WindowMaker アニメーション リンクを設定する。
- InTouch タグ変数と産業用グラフィックを接続する。
- 同じ親の代替インスタンスを選択する。
- 同じインスタンスの代替シンボルを選択する。
- ダイナミック サイズの反映を有効化または無効化する。

産業用グラフィックの WindowMaker アニメーション リンクの設定

埋め込まれた産業用グラフィックの WindowMaker アニメーション リンクは、他の InTouch オブジェクトと同じ方法で設定します。設定できるのは、埋め込まれた産業用グラフィックの外部にある次のようなアニメーション リンクだけです。

- オブジェクト サイズ
- オブジェクトの位置
- 表示オン/オフ
- 使用可能性

WindowMaker で設定されたアニメーション リンクは、産業用グラフィック エディタで設定されたアニメーション リンクから独立しています。これらには産業用グラフィックの設定は継承されません。また、WindowMaker で実行されるときに優先されます。

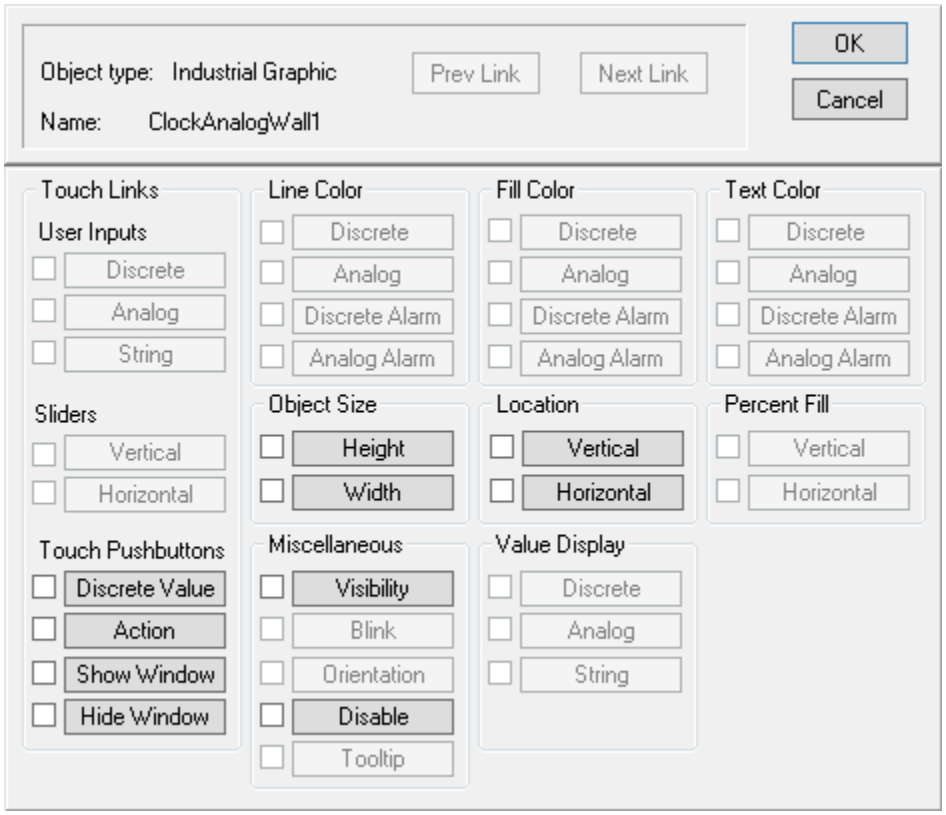
埋め込まれた産業用グラフィックの WindowMaker アニメーション リンクを設定するには

1. 埋め込まれた産業用グラフィックを選択します。
2. [**描画**] メニューの [**モード**] グループで [**アニメーション化**] を選択します。

   または、オブジェクトを右クリックして、[**アニメーション リンク**] を選択します。

   [**アニメーション リンク**] ウィンドウが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

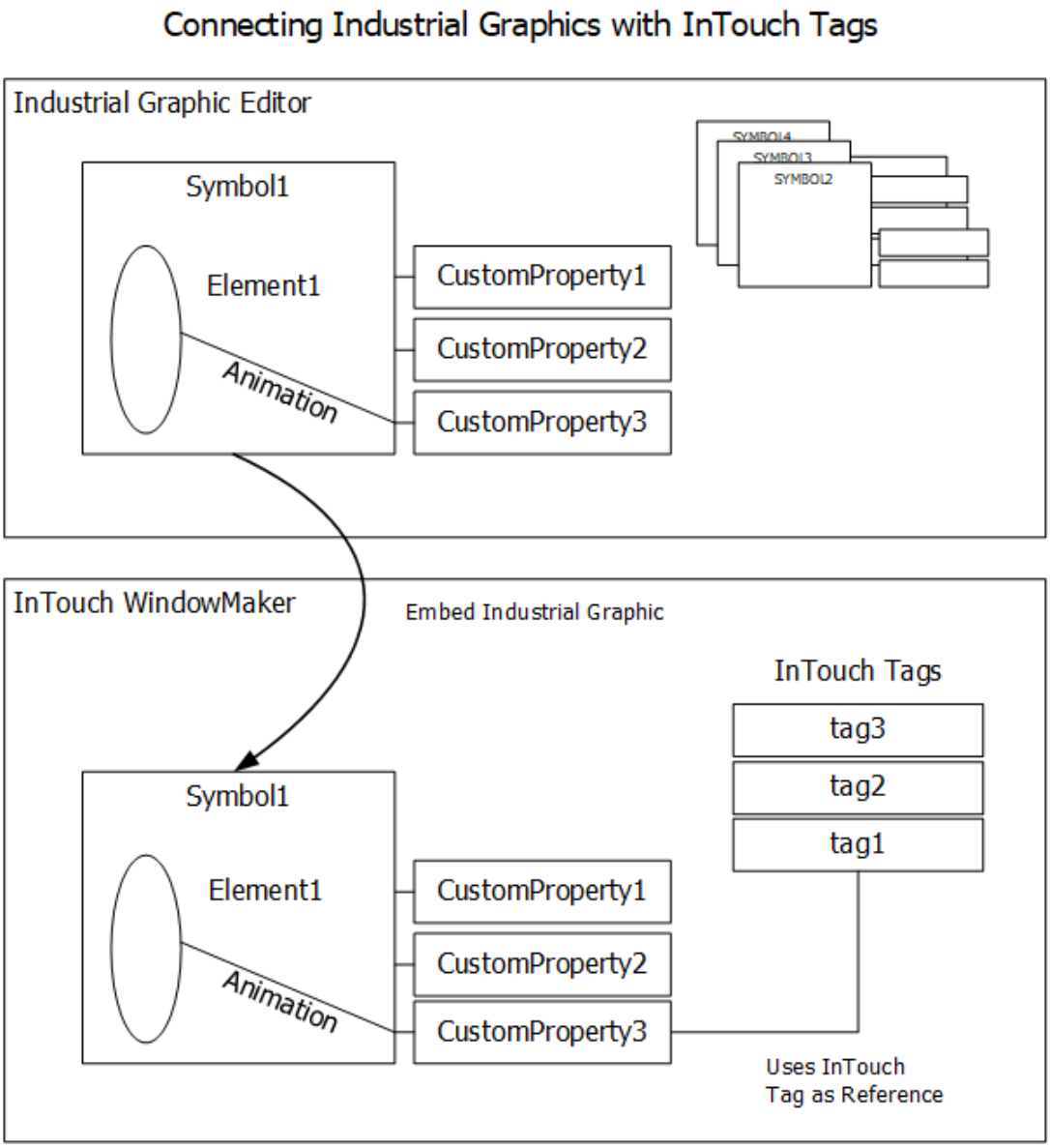


3. 他の InTouch オブジェクトと同じ方法で変更します。
4. **[OK]** を選択します。

InTouch タグと産業用グラフィックの接続

埋め込まれた産業用グラフィックのカスタム プロパティを上書きすることにより、産業用グラフィックと InTouch タグを接続できます。産業用グラフィックのプロパティがカスタム プロパティによって InTouch に公開されます。カスタム プロパティは、産業用グラフィックのアニメーションによって内部的に使用できる場合と、できない場合があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



産業用グラフィックを InTouch ウィンドウに埋め込むと、以下の表に示すようにアニメーション リンク内の参照が変換されます。

産業用グラフィック	埋め込まれた産業用グラフィック
Object.Extension	galaxy:Object.Extension
intouch:タグ名	タグ名

カスタム プロパティの詳細については、『*産業用グラフィック エディタ ユーザー ガイド*』を参照してください。

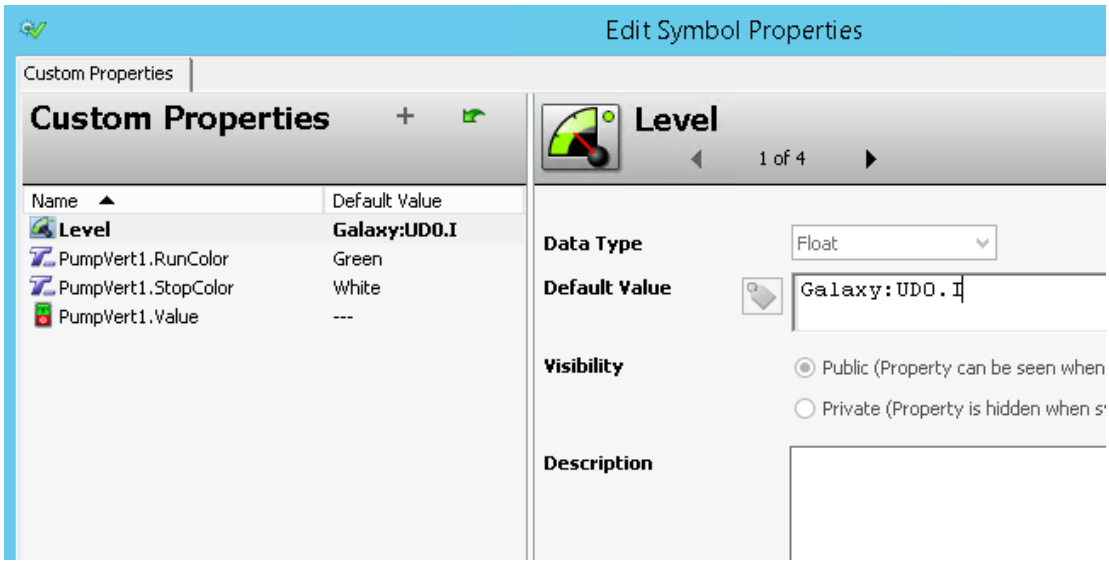
アニメーション エディタまたはスクリプト エディタ内で作業している場合、Galaxy ブラウザから InTouch タグを直接選択できます。これらのいずれかのエディタから呼び出された場合、Galaxy ブラウザには現在の Galaxy のすべての InTouchViewApp インスタンスが一覧表示され、選択した InTouchViewApp インスタンスのすべての InTouch タグとドットフィールドが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Galaxy ブラウザでの InTouch タグの選択の詳細については、『InTouch からの Application Server のオブジェクト属性の参照』を参照してください。

InTouch タグと産業用グラフィックを接続するには

1. 埋め込まれた産業用グラフィックを InTouch ウィンドウ内で右クリックし、［**産業用グラフィック**］をポイントし、次に［**シンボル プロパティの編集**］を選択します。［**シンボル プロパティの編集**］ダイアログ ボックスが表示されます。



2. InTouch タグに接続するカスタム プロパティを選択します。選択したカスタム プロパティの設定が右ペインに表示されます。
3. ［**デフォルト値**］ボックスで、以下のいずれかを実行します。
   - InTouch タグの名前を入力します。
   - 参照ボタンを選択し、［**タグ変数を選択してください**］ダイアログ ボックスからタグ変数を選択します。

4. カスタム プロパティを元の値に戻すには、［**Restore**］を選択します。
5. **[OK]** を選択します。選択したカスタム プロパティを使用して設定した産業用グラフィック内のアニメーションで、処理中に InTouch タグの値が使用されるようになります。

InTouch タグと産業用グラフィックの接続の例

この例では、産業用グラフィック エディタで作成された垂直塗りつぶしパーセント アニメーションを持つタンク シンボルを InTouch タグに接続する方法を示します。

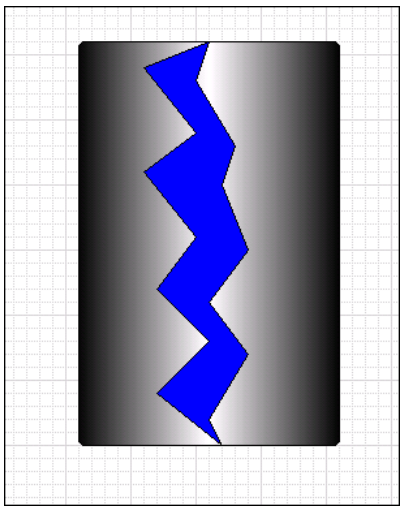
これを行うには、以下の 3 つの主な手順を実行します。

- 産業用グラフィックとしてサンプル タンクを作成します。
- InTouch アプリケーションを作成します。
- WindowViewer でサンプル タンクを派生させて表示します。

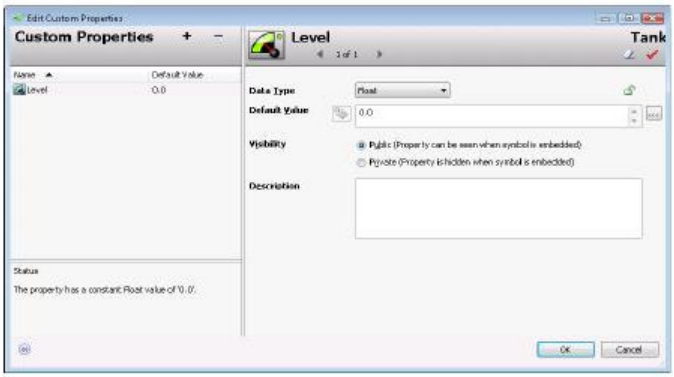
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

産業用グラフィックとしてサンプル タンクを作成します。

1. IDE で、"Tank" という名前の新しいシンボルを作成し、産業用グラフィック エディタで開きます。
2. キャンバスに長方形を貼り付けます。必要に応じて、外観を変更します。
3. タンク レベルを示すためのタンクの切り抜きを表す色付きの多角形要素を作成します。



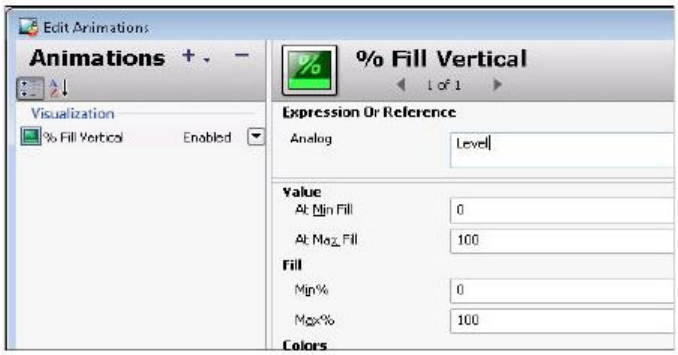
4. キャンバスを選択します。
5. ［**プロパティ**］メニューの［**グラフィック**］グループで［**編集**］を選択します。
   または、グラフィック オブジェクトをダブルクリックします。
   ［**カスタム プロパティの編集**］ダイアログ ボックスが表示されます。
6. **Level** という名前のカスタム プロパティを追加します。
7. プロパティの詳細を設定します。以下の手順を実行します。
   - ［**Data Type**］リストで、［**Float**］を選択します。
   - ［**デフォルト値**］ボックスに 0 と入力します。



8. **[OK]** を選択します。
9. タンク レベルを表す多角形要素をダブルクリックします。
   ［**アニメーションの編集**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

10. ［**垂直塗りつぶしパーセント**］アニメーションを追加します。
11. 右ペインにある［**アナログ**］ボックスに、カスタム プロパティの名前を入力します。この例では、これは Level です。



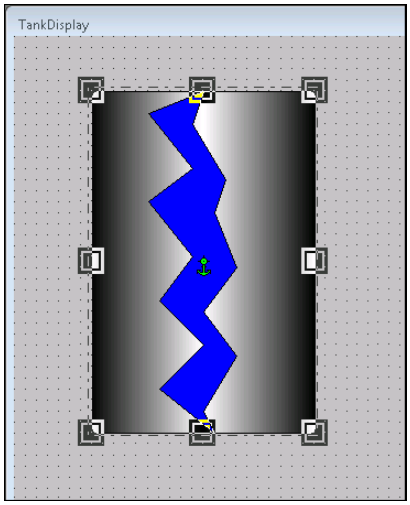
12. ［**OK**］を選択し、［**Edit Animations**］ダイアログ ボックスを閉じます。
13. ［**保存して閉じる**］を選択し、産業用グラフィック エディタを閉じます。

InTouch アプリケーションを作成します。

1. System Platform IDE で、新しいマネージド InTouch アプリケーションを作成します。詳細については、「マネージド InTouch アプリケーションの作成」を参照してください。
2. マネージド InTouch アプリケーションを WindowMaker で開きます。
3. TankDisplay という名前の新しいウィンドウを作成します。
4. タグ名ディクショナリを開き、TankLevel という名前の新しい実数型の InTouch タグを作成します。
5. ［産業用グラフィックの埋め込み］アイコンを選択します。

   ［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。
6. タンク シンボルを選択し、［**OK**］を選択します。
7. ウィンドウ内のシンボルを埋め込む新しい位置で選択します。

   ウィンドウにタンク シンボルが埋め込まれます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

8. 埋め込まれた産業用グラフィックを右クリックし、［**産業用グラフィック "Tank"**］をポイントし、次に［**シンボル プロパティの編集**］を選択します。

   ［**シンボル プロパティの編集**］ダイアログ ボックスが表示されます。

9. カスタム プロパティ レベルを選択します。
10. ［**デフォルト値**］ボックスに `TankLevel` と入力します。また、参照ボタンを選択し、［**タグ変数を選択してください**］ダイアログ ボックスを使用して TankLevel を参照することもできます。
11. **[OK]** を選択します。
12. ウィンドウにスライダを貼り付け、ローカル InTouch タグである TankLevel を使用して設定します。
13. 変更を保存し、WindowMaker を終了します。

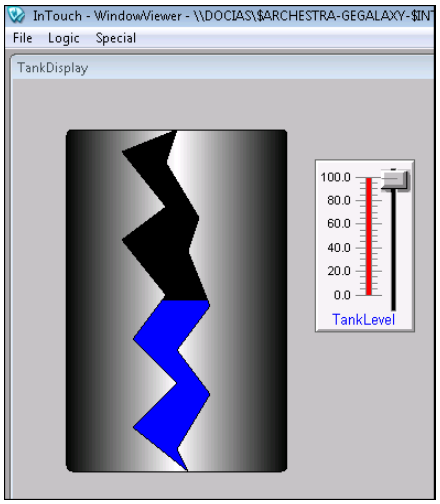
    マネージド InTouch アプリケーションは自動的にチェック インされます。

サンプル タンクを派生させ、テストするには

1. System Platform IDE で、マネージド InTouch アプリケーションのインスタンスを派生させ、WinPlatform インスタンスおよび ViewEngine インスタンスと共に配置します。
2. InTouch アプリケーション マネージャを開き、WindowViewer のリストに表示されているアプリケーションを起動します。

   タンクおよびスライダが WindowViewer のウィンドウに表示されます。

3. スライダを移動するとタンク レベルを変更できます。



同じ親からの代替インスタンスの選択

産業用グラフィック内のすべての参照は、代替インスタンスにリダイレクトできます。継承された産業用グラフィックは編集できないので、サイズを除き、産業用グラフィックの外観は変更されません。

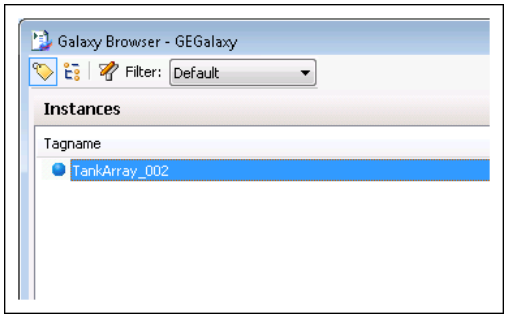
グラフィック ツールボックスから生成される産業用グラフィックはオブジェクトには関連付けられていないため、この機能は使用できません。

同じ親から代替インスタンスを選択するには

1. 埋め込まれた産業用グラフィックを右クリックし、［**産業用グラフィック**］をポイントし、次に［**代替インスタンスの選択**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。同じ親を持つその他のすべてのインスタンスが表示されます。



2. リストから代替インスタンスを選択し、［**OK**］を選択します。

   新しい代替インスタンスをポイントするように、産業用グラフィックの参照が更新されます。

同じインスタンスの代替シンボルの選択

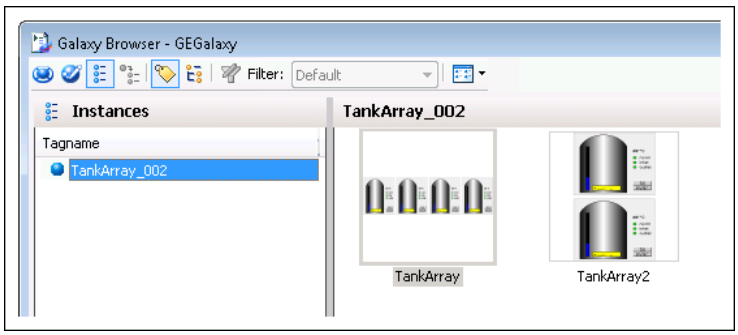
埋め込まれた産業用グラフィックは、同じインスタンスに属する他の産業用グラフィックで置き換えることができます。

グラフィック ツールボックスから生成される産業用グラフィックはオブジェクトには関連付けられていないため、この機能は使用できません。

代替の産業用グラフィックを同じインスタンスから選択するには

1. 埋め込まれた産業用グラフィックを右クリックし、［**産業用グラフィック**］をポイントし、次に［**代替シンボルの選択**］を選択します。

   ［**Galaxy ブラウザ**］ダイアログ ボックスが表示されます。



2. 右ペインで代替シンボルを選択し、［**OK**］を選択します。
3. 代替シンボルのサイズが元のシンボルのサイズと異なる場合、現在埋め込まれている産業用グラフィックのサイズを保持するかどうかを確認するメッセージが表示されます。以下のいずれかを実行します。
   - 選択されている産業用グラフィックの現在のサイズを保持するには、［**はい**］を選択します。
   - 選択されている産業用グラフィックのサイズを新しい産業用グラフィックのサイズに更新するには、［**いいえ**］を選択します。

   どちらの場合でも、埋め込まれた産業用グラフィックは代替の新しい産業用グラフィックで更新されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

産業用グラフィック内の文字列の置換

埋め込まれた産業用グラフィック内のすべての文字列を代替文字列で置き換えることができます。

埋め込まれた産業用グラフィック内のすべての文字列を置き換えるには

1. 埋め込まれた産業用グラフィックを選択します。
2. ［**アニメーション**］メニューの［**置換**］グループで［**文字列**］を選択します。

   ［**文字列の変更**］ダイアログ ボックスが表示されます。
3. 対応するボックスに新しい文字列を入力し、［**OK**］を選択します。

   埋め込まれた産業用グラフィック内の文字列が新しい代替文字列で置き換えられます。

産業用グラフィック内の参照の置換

埋め込まれた産業用グラフィック内のすべての参照を代替の参照で置き換えることができます。

埋め込まれた産業用グラフィック内のすべての参照を置き換えるには

1. 埋め込まれた産業用グラフィックを選択します。
2. ［**アニメーション**］メニューの［**置換**］グループで［**タグ**］を選択します。

   ［**タグの置換**］ダイアログ ボックスが表示されます。
3. 対応するボックスに新しい参照を入力し、［**OK**］を選択します。

   埋め込まれた産業用グラフィック内の文字列が新しい代替参照で置き換えられます。

埋め込まれた産業用グラフィックのダイナミック サイズ変更の反映を有効化または無効化できます。

埋め込まれた産業用グラフィックのダイナミック サイズ変更の反映を有効化または無効化できます。

ダイナミック サイズ変更の反映が有効になっている、ソース シンボルの絶対アンカー ポイントの位置は以下のように変更されます。

- 埋め込まれたシンボルのアンカー ポイントは変更されません。
- 埋め込まれたシンボルの位置は適宜移動します。

ダイナミック サイズ変更の反映が無効になっていると、ソース シンボルの絶対アンカー ポイントの位置は以下のように変更されます。

- 埋め込まれたシンボルのアンカー ポイントが適宜移動します。
- 埋め込まれたシンボルの位置は変更されません。

ダイナミック サイズの反映の詳細については、*『産業用グラフィック エディタ ユーザー ガイド』*を参照してください。

埋め込まれたシンボルのダイナミック サイズ変更の反映を有効化または無効化するには

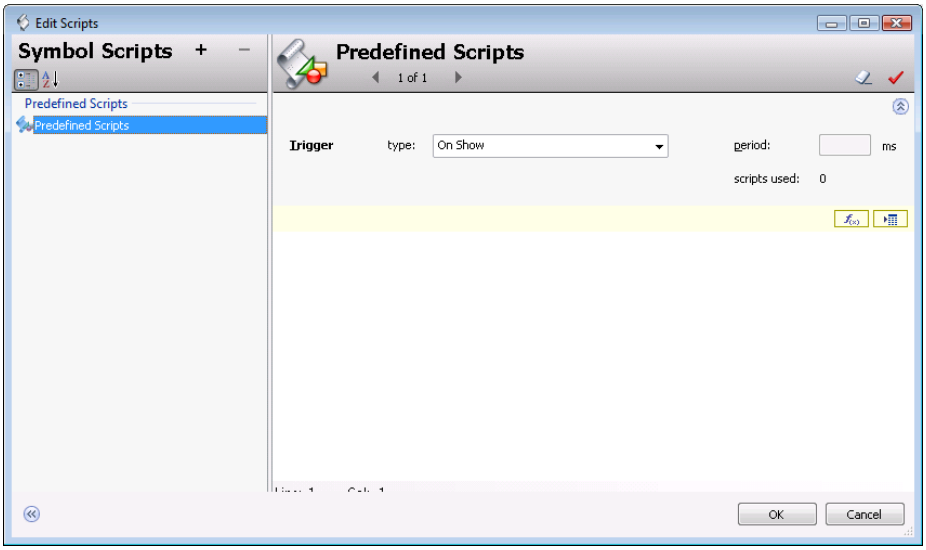
- 埋め込まれた産業用グラフィックを右クリックし、［**産業用グラフィック**］をポイントして［**ダイナミック サイズ変更**］をオンまたはオフにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スクリプトと産業用グラフィックの関連付け

スクリプトと InTouch アプリケーションに配置された産業用グラフィックを関連付けることができます。スクリプトを使用して、InTouch アプリケーションの実行中にグラフィックをアニメーション化することや要素を変更することができます。

InTouch ウィンドウに埋め込まれた産業用グラフィックを選択した後、スクリプトの実行をトリガする値を持つ式またはリファレンスを選択します。産業用グラフィック エディタを使用して、スクリプトを実行するトリガを選択します。



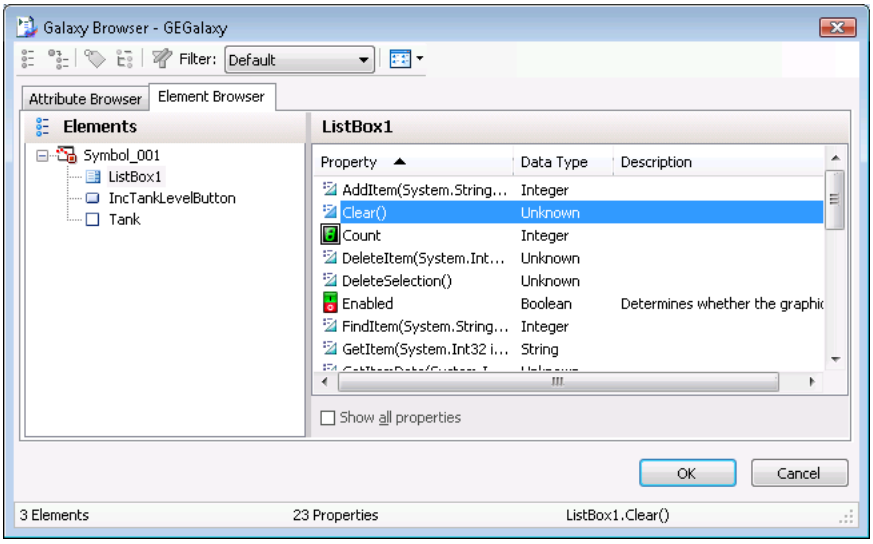
- グラフィック スクリプトは、事前定義することや名前を付けることができます。事前定義されたスクリプトは、実行中のアプリケーションのグラフィックのステータスに基づいて実行されます。名前付きスクリプトは、スクリプトに関連付けられた式またはリファレンスで状態が変更された場合に実行されます。
- 事前定義されたスクリプトは、HMI InTouch ウィンドウ スクリプトに似ています。スクリプト トリガの設定方法に応じて、事前定義されたシンボル スクリプトが以下のように実行されます。
  - グラフィックを開いた後、または表示した後に 1 回。
  - 実行中のアプリケーションにグラフィックが表示されている間、定期的に。
  - グラフィックを閉じた後、または非表示にした後に 1 回。
- トリガ値または式が True または False であるか、True と False のステータスの間を移行している場合に、スクリプトの設定方法に応じて名前付きグラフィック スクリプトを実行できます。また、トリガ式に関連付けられたデータの値が変更されたとき、または品質状態の値が変更されたときに、名前付きグラフィック スクリプトを実行できます。

産業用グラフィック スクリプトでのメソッドの使用

いくつかの要素では、スクリプト メソッドがサポートされています。これらのメソッドは、ランタイム中に要素上でさまざまな関数を実行できます。通常、メソッドにアクセスするアクション スクリプトを設定します。

Galaxy ブラウザを開いて要素を選択すると、要素のサポートされているプロパティおよびメソッドが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



- Edit Box コントロールのメソッドが含まれているスクリプトを実行して、ランタイム中にファイルからコントロールにテキストをロードすることができます。また、スクリプトを実行して、ランタイム中に Edit Box コントロールの現在のコンテンツをファイルに保存することもできます。
- Edit Box コントロールのメソッドは、スクリプト内で以下の形式で宣言されています。

  ```
  ControlName。SaveText(FileName);
  ```

  ここで、*ControlName* は Edit Box コントロールの名前であり、*FileName* はロードまたは保存されるコントロールのコンテンツが含まれているファイルの名前です。上記の例では、SaveText が、Edit Box コントロールのコンテンツをファイルに保存するメソッドの名前です。
- Combo Box コントロールおよび List Box コントロールのメソッドが含まれているスクリプトを使用して、ランタイム中にリストのコンテンツを変更できます。リスト アイテムは、追加、削除または変更できます。
- Combo Box コントロールおよび List Box コントロールのメソッドは、Edit Box コントロールに似たスクリプトで宣言されています。

産業用グラフィック エディタでの産業用グラフィックの編集

埋め込まれた産業用グラフィックは、System Platform IDE に統合されている産業用グラフィック エディタを使用して編集できます。これを行うには、以下の手順を実行します。

1. 埋め込まれた産業用グラフィックを産業用グラフィック エディタで開き、シンボルを変更して保存します。テンプレート、インスタンス、またはグラフィック ツールボックスで産業用グラフィックが更新されます。

   詳細については、「埋め込まれた産業用グラフィックの編集」を参照してください。
2. ステータス バーの右下隅にある［シンボルの変更］アイコンを選択し、WindowMaker の変更を受け入れます。WindowMaker に変更が反映されます。

   詳細については、「WindowMaker でのシンボルの変更の受け入れ」を参照してください。

埋め込まれた産業用グラフィックの編集

埋め込まれた産業用グラフィックは、InTouch WindowMaker 内で簡単に編集できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

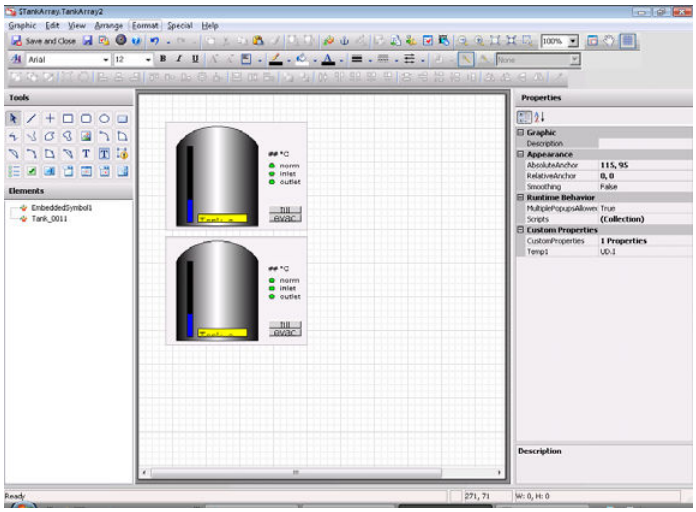
ソース シンボルまたはその埋め込まれたシンボルが他のマネージド InTouch アプリケーションで使用されている場合、埋め込まれたシンボルおよび InTouch アプリケーションに変更が反映されます。

産業用グラフィックに加えたすべての変更は、埋め込まれたシンボルに自動的に反映されるわけではありません。詳細については、「WindowMaker でのシンボルの変更の受け入れ」を参照してください。

埋め込まれた産業用グラフィックを産業用グラフィック エディタで編集するには

1. 埋め込まれた産業用グラフィックを右クリックし、[**産業用グラフィック**] をポイントして [**シンボルの編集**] を選択します。

   産業用グラフィック エディタに産業用グラフィックが表示されます。



2. 産業用グラフィックを編集します。詳細については、『産業用グラフィックの作成および管理ユーザー ガイド』参照してください。

   産業用グラフィックの編集では、シンボルに要素スタイルを適用することもできます。要素スタイルの使用の詳細については、『*Application Server ユーザー ガイド*』を参照してください。

3. **[閉じて保存] を選択します**。変更が保存され、 産業用グラフィック エディタが終了します。

4. Application Server オブジェクトがインスタンスまたはテンプレートによってホストされている場合、保存して、IDE のオブジェクト エディタを終了します。

WindowMaker でのシンボルの変更の受け入れ

産業用グラフィックが変更され、それを WindowMaker の InTouch ウィンドウで現在使用している場合、WindowMaker で変更をすぐに受け入れることができます。

変更をすぐに受け入れない場合、ウィンドウを閉じてから再度開いたときに WindowMaker でシンボルが更新されます。

また、アプリケーションをテストするために WindowViewer に切り替えた場合、または対象のノードで WindowViewer のアプリケーションを開いた場合、シンボルも更新されます。

WindowMaker でシンボルの変更をすぐに受け入れるには

- 埋め込まれた産業用グラフィックが含まれた開いている InTouch ウィンドウでは、以下のいずれかを実行します。
  - ステータス バーの右下隅にある [シンボルの変更] アイコンをダブルクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 埋め込まれた産業用グラフィックが含まれている InTouch ウィンドウを閉じ、再度開きます。

どちらの場合でも、産業用グラフィックに加えられた変更は、InTouch ウィンドウの埋め込まれた産業用グラフィックに反映されます。

WindowViewer でのシンボルの変更の受け入れ

産業用グラフィックが変更され、それを WindowViewer で現在テストしている場合、WindowViewer で変更を受け入れることができます。

埋め込まれた産業用グラフィックのテストの詳細については、「WindowViewer での産業用グラフィックのテスト」を参照してください。

テスト時に WindowViewer のシンボルの変更を受け入れるには

以下のいずれかを実行します。

- WindowMaker に高速切り替えで戻り、WindowViewer に戻します。
- InTouch ウィンドウを閉じ、再度開きます。これは、［**WindowViewer のプロパティ**］ダイアログ ボックスで［**アプリケーションを常にディスクからロード**］オプションがオンになっている場合のみ機能します。

どちらの場合でも、産業用グラフィックに加えられた変更は、InTouch ウィンドウの埋め込まれた産業用グラフィックに反映されます。

InTouch タグを使用したグラフィック要素と産業用グラフィックの作成

産業用グラフィック エディタの［プロパティ］設定ペインにある［タグ］タブに InTouch アプリケーションで使用できるすべてのタグが表示されます。タグをキャンバスにドラッグ アンド ドロップするだけでグラフィック要素または産業用グラフィックを作成できます。複数のシンボルを一度に作成する場合、ドットフィールド プロパティが自動的に制限されます。この方法を使用すると、グラフィック開発ワークフローを簡素化して、アプリケーション開発時間を大幅に削減できます。

注 : この［**タグ**］タブは CONNECT とマネージド InTouch の産業用グラフィック エディタでは使用できません。

InTouch タグを使用してグラフィック要素を作成する

1. 産業用グラフィック エディタの［**タグ**］ペインからタグをキャンバスにドラッグ アンド ドロップします。

   ［**グラフィック要素**］オプションと［**産業用グラフィック**］オプションが表示されます。

   注 : 複数のタグを一緒にドラッグ アンド ドロップすると、同じ要素タイプとアニメーションの複数のグラフィック要素を作成できます。

2. デフォルト値を設定するには
   - 画面上の任意の場所を選択します。

     または

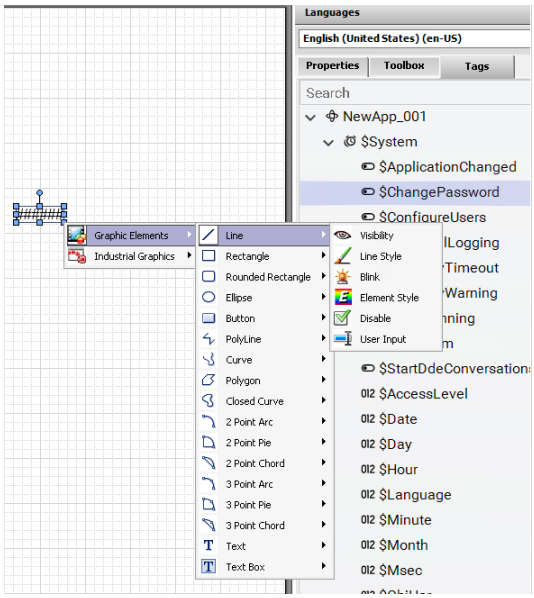
   - ［**グラフィック要素**］オプションの上にカーソルを置き、目的のグラフィック要素とアニメーションを選択します。

     グラフィック要素と使用可能なグラフィック アニメーションのデータ型のリストが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. 目的のグラフィック要素の上にカーソルを置きます。

   使用可能なアニメーションが表示されます。



4. 目的のアニメーションを選択します。

   タグに割り当てられた新しいグラフィック要素が作成されます。

InTouch タグを使用して産業用グラフィックを作成する

1. 産業用グラフィック エディタの［**タグ**］ペインからタグをキャンバスにドラッグ アンド ドロップします。

   ［**グラフィック要素**］オプションと［**産業用グラフィック**］オプションが表示されます。

   注：複数のタグを一緒にドラッグ アンド ドロップすると、同じタイプの複数の産業用グラフィックを作成できます。

2. デフォルト値を設定するには
   - 画面上の任意の場所を選択します。

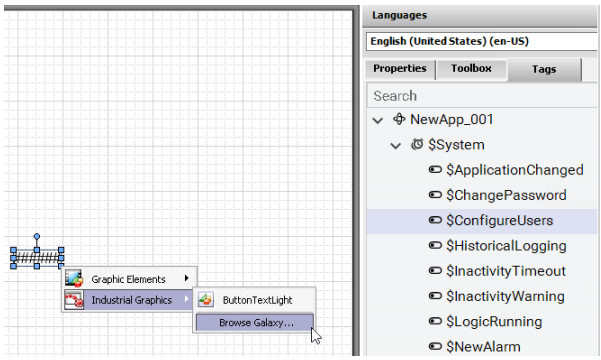
     または

   - ［**産業用グラフィック**］オプションの上にカーソルを置き、目的の産業用グラフィックを選択します。

     ［**Galaxy を参照**］オプションが表示されます。

     注：同じタイプのタグを複数回ドラッグして［**産業用グラフィック**］オプションの上にカーソルを置くと、以前に選択された 5 つのグラフィック名が［**Galaxy を参照**］オプションと共に表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 以前に選択したグラフィックのいずれかを選択するか、[**Galaxy を参照**] を選択して新しいグラフィックを選択します。
4. [**Galaxy を参照**] オプションを選択した場合、目的の産業用グラフィックを参照して選択し、[**OK**] を選択します。

   タグに割り当てられた新しい産業用グラフィックが作成されます。

   注 : 選択した産業用グラフィックの [**カスタム プロパティ**] で、[**表示オン/オフ**] が [**パブリック**] に設定されている場合:

   - [**値**] プロパティの [**デフォルト値**] が空または --- の場合、ドラッグ アンド ドロップしたタグの名前で置き換えられます。

   - ドット プロパティの [**デフォルト値**] が空または --- の場合、**タグ名.<該当するドット プロパティ名>** で置き換えられます。

   上記の両方のケースにおいて、選択した産業用グラフィックのカスタム プロパティの [**表示オン/オフ**] が [**プライベート**] の場合、そのプロパティは、作成した産業用グラフィックに表示されません。[表示オン/オフ] が [プライベート] で、他のグラフィックに対して表示しない場合、[**表示オン/オフ**] が [**プライベート**] のカスタム プロパティはその他の産業用グラフィックで表示されません。

産業用グラフィック エディタの [ツールボックス] タブからのグラフィックの埋め込み

産業用グラフィック エディタの [**プロパティ**] 設定ペインにある [**ツールボックス**] タブを使用して、既存の産業用グラフィックを別のグラフィックに埋め込むことができます。[**ツールボックス**] タブには、InTouch ライブラリで使用できるすべての産業用グラフィックが表示されます。別のグラフィックに埋め込んだグラフィックは、グラフィックのその他のコンポーネントと同様に編集できます。産業用グラフィック エディタの [産業用グラフィックの埋め込み] アイコンを使用してグラフィックを埋め込むこともできます。詳細については、産業用グラフィック エディタ ヘルプの「グラフィックの埋め込み」トピックを参照してください。

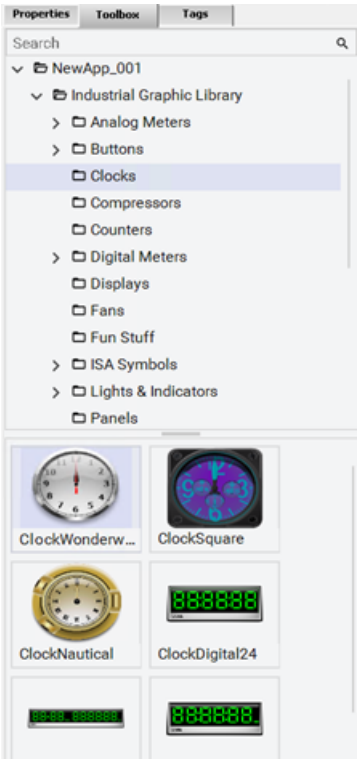
産業用グラフィック エディタの [ツールボックス] タブから既存のグラフィックを別のグラフィックに埋め込むには

1. [**プロパティ**] 設定ペインで [**ツールボックス**] タブを選択します。
2. フォルダをナビゲートして目的の産業用グラフィックを選択するか、[**検索**] ボックスでグラフィックを検索します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

選択したフォルダまたはオブジェクト内のグラフィックがナビゲーション領域の下に表示されます。

3. 追加するグラフィックを選択して、キャンバスの目的の位置にドラッグします。

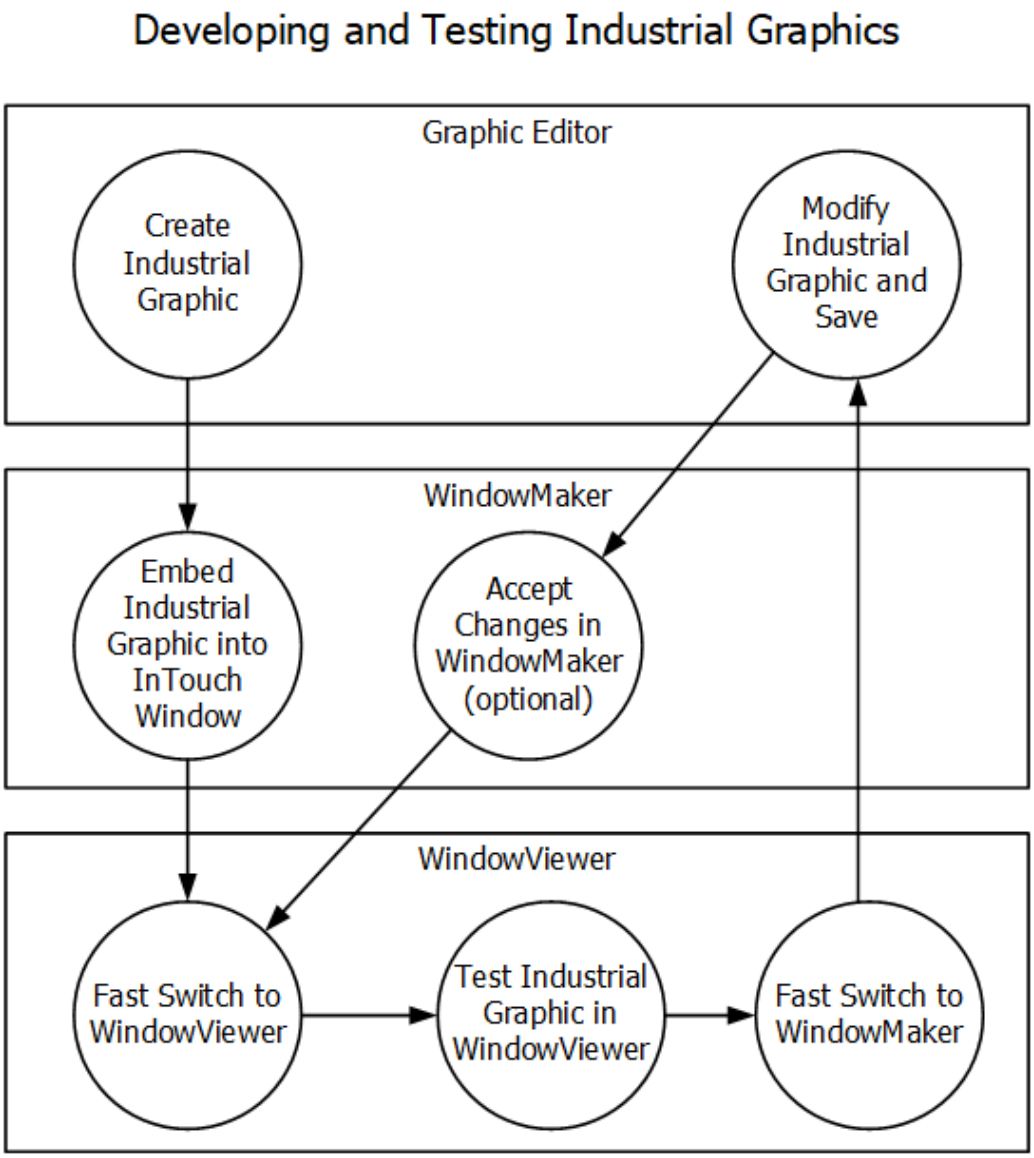


WindowViewer での産業用グラフィックのテスト

埋め込まれた産業用グラフィックを InTouch ウィンドウでテストできます。InTouchViewApp インスタンスを派生させる必要はありません。事前に以下の操作を行っている場合、埋め込まれた産業用グラフィックをテストできます。

- グラフィック ツールボックス、オートメーション テンプレートまたはオートメーション インスタンスでの産業用グラフィックの作成。
- マネージド InTouch アプリケーションの作成。
- マネージド InTouch アプリケーションへの産業用グラフィックの埋め込み。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



埋め込まれた産業用グラフィックを **WindowViewer** でテストするには

1. WindowMaker で［**実行**］を選択し、WindowViewer に切り替えます。
2. 通常のランタイム環境で行う場合と同様に、埋め込まれたシンボルのアニメーション、動作、および表示方式をテストします。
3. WindowMaker への高速切り替えを行って、産業用グラフィックを埋め込む方法を変更できます。

埋め込まれた産業用グラフィックを **WindowViewer** で変更してテストするには

1. 産業用グラフィック エディタで産業用グラフィックを変更します。
2. 変更を保存します。

   WindowViewer が開いてしばらくすると、WindowViewer に変更を受け入れるようメッセージが表示されます。**はい**を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowViewer が終了したら、WindowMaker から WindowViewer に高速切り替えし、変更を確認することができます。開いている WindowViewer セッションに変更が反映されるのを待機するよりも、WindowViewer を終了してから WindowViewer を再度開くほうがより高速です。

グラフィック パフォーマンスの予測

グラフィック パフォーマンス インデックス（GPI）を使用して、ランタイム時に産業用グラフィックのパフォーマンスを測定できます。

このツールは、産業用グラフィック エディタで作成するシンボルがランタイム時に起動するときの推定呼び出し時間を計算します。呼び出し時間は、ユーザーまたはシステムが関連グラフィックの表示を要求してからグラフィックがライブ データで画面に表示されるまでの間隔です。計算は、ランタイム時に InTouch WindowViewer で起動されたシンボルの内容、および外部参照サブスクリプションが完了した場合のベスト ケース時間予測に基づきます。

GPI の使用の詳細については、『*産業用グラフィック エディタ ユーザー ガイド*』の「グラフィック パフォーマンスの予測」を参照してください。

新しいオートメーション インスタンスの作成

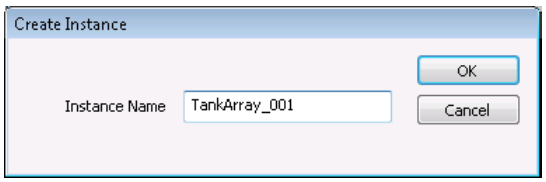
埋め込まれた産業用グラフィックをホストするオートメーション オブジェクトの新しいインスタンスをすばやく作成できます。この方法を使用すると、System Platform IDE に切り替えてインスタンスを派生させる必要がありません。

新しいインスタンスの割り当てが解除されるので、インスタンスを使用する前に System Platform IDE で割り当ておよび配置を行う必要があります。

新しいオートメーション インスタンスを作成できるのは、テンプレートまたはインスタンスによってホストされる産業用グラフィックに対してのみです。グラフィック ツールボックス内の産業用グラフィックには、この機能はありません。

新しいオートメーション インスタンスを作成するには

1. 埋め込まれた産業用グラフィックを右クリックし、［**産業用グラフィック**］をポイントして［**新しいインスタンス**］を選択します。［**インスタンスの作成**］ダイアログ ボックスが表示されます。



2. ［**インスタンス名**］ボックスで、インスタンスの名前を入力します。
3. **[OK]** を選択します。指定した名前を持つテンプレートから、インスタンスが自動的に派生します。

新しい Application Server オブジェクト インスタンスの自動作成

テンプレートから産業用グラフィックを埋め込む場合、InTouch HMI で、そのオブジェクトのインスタンスおよび新しいインスタンスを参照するシンボルのインスタンスを作成できます。

以下の例は、新しい Application Server オブジェクト インスタンスを自動的に作成する方法を示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

新しい Application Server オブジェクト インスタンスを自動的に作成するには

1. オブジェクト テンプレート **$Valve1** を作成し、IDE オブジェクト エディタで開きます。
2. ［**グラフィック**］タブで、**ValveSymbol** という名前の産業用グラフィックを追加します。
3. InTouchViewApp オブジェクトの派生テンプレートを作成し、WindowMaker で開きます。
4. 新しい InTouch ウィンドウを作成し、オブジェクト テンプレート $Valve1 の産業用グラフィック **ValveSymbol** を埋め込みます。インスタンス名を入力するよう求めるメッセージが WindowMaker に表示されます。
5. たとえば、`Valve1_E122` などの名前を入力し、［**OK**］を選択します。InTouch ウィンドウに産業用グラフィックが貼り付けられ、Galaxy でオートメーション オブジェクト インスタンス Valve1_E122 が作成されます。

InTouch ウィンドウから産業用グラフィックへの変換

マネージド InTouch アプリケーションのウィンドウを産業用グラフィックに変換できます。変換された産業用グラフィックは WindowMaker グラフィック ツールボックスおよび IDE グラフィック ツールボックスに表示されます。ウィンドウに表示されるグラフィックに加えて、InTouch スクリプトも Application Server スクリプトに変換されます。

ウィンドウの変換の準備

InTouch ウィンドウを変換する前に以下の点に注意してください。

- 産業用グラフィックに変換できるのは InTouch スタンドアロンおよびマネージド アプリケーションのウィンドウだけです。
- 変換するウィンドウは、WindowMaker で閉じる必要があります。

ウィンドウの変換ルールと例外

ウィンドウのシンボルおよびスクリプトだけが変換されます。ウィンドウの色、タイプ、フレーム、タイトルバー、サイズコントロール、および［**閉じる**］ボタンは変換されたシンボルから除外されます。

InTouch のグラフィック タイプに基づいて、ウィンドウグラフィックは次のように変換されます。

- すべての InTouch グラフィック プリミティブは、対応する産業用グラフィック プリミティブに変換されます。
- 1 つの InTouch スマート シンボルは 1 つの産業用の埋め込まれたグラフィックに変換されます。
- ウィンドウ内の 1 つの産業用グラフィックは 1 つの埋め込まれたシンボルに変換されます。埋め込まれたシンボルには、新しいシンボルは作成されません。
- InTouch シンボルは、プロパティ TreatAsIcon=True のグループに変換されます。
- InTouch セルは、プロパティ TreatAsIcon=False のグループに変換されます。
- InTouch ウィンドウは、ArchestrA ウィンドウ コントロールに変換されます。
- 一部の InTouch グラフィック コンポーネントは産業用グラフィックに変換できません。
  - InTouch のリアルタイム トレンドおよび履歴トレンドは変換できません。
  - ActiveX コントロールおよび分散アラーム表示は変換できません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アニメーション スクリプトの変換

ウィンドウに埋め込まれたすべての InTouch アニメーション リンクは、Application Server の対応するアニメーションに変換されます。

変換中、検証に関する警告やエラー メッセージはログに記録されません。変換されたスクリプトを産業用グラフィック スクリプト検証で検証して、サポートされていないスクリプト構文を見つける必要があります。

InTouch アニメーション スクリプトを変換するとき、以下の例外が発生します。

- 線の色リンクと塗りつぶし色の理論値アラームとアナログ アラームのアニメーション リンクは ライン スタイル アニメーションと塗りつぶしスタイル アニメーションのブール型および真理値表に変換されます。
- ShowWindow アニメーション リンクは、ShowGraphic スクリプト関数を含むアクション スクリプトに変換されます。HideWindow アニメーション リンクは、HideGraphic スクリプト関数を含むアクション スクリプトに変換されます。
- アニメーション リンク式で設定されているすべての InTouch タグのタグ名には、"InTouch" というプリフィックスが追加されます。たとえば、Tag1 は Intouch:Tag1 に変換されます。
- アニメーション リンクで設定されている Application Server 属性参照のプリフィックス "galaxy:" は削除されます。たとえば、galaxy:UD001.Value は UD001.Value に変換されます。
- アクション スクリプト内で設定されているすべての InTouch スクリプト関数は変換されません。InTouch タグおよび Application Server 属性参照処理を除くすべてのスクリプトがコピーされます。

ウィンドウ変換の既知の制限事項

InTouch ウィンドウを産業用グラフィックに変換する場合、必ずしも完全な忠実性が維持されるわけではありません。ウィンドウ コンポーネントをシンボルに変換できないことがあります。このセクションでは、InTouch ウィンドウを産業用グラフィックに変換する際の既知の制限事項およびその対処方法について説明します。

- 垂直マウス リリース スライダー SmartSymbol を含むウィンドウの変換

  垂直マウス リリース スライダー SmartSymbol には、2 種類のアニメーションが含まれます。シンボルは、塗りつぶしアニメーションを使用して、スケールに対する現在の測定値および値を設定する移動可能なスライダー ノブを示します。埋め込まれた垂直マウス リリース スライダー SmartSymbol を含むウィンドウを産業用グラフィックに変換すると、移動可能なスライダー アニメーションは維持されません。

- Symbol Factory シンボルを含むウィンドウの変換

  Symbol Factory シンボルに組み込まれているいくつかのタイプのアニメーションは産業用グラフィックに変換できません。以下のタイプの Symbol Factory シンボル アニメーションは変換できません。

  - 塗りつぶしパーセント
  - 線/輪郭色
  - 水平移動
  - 垂直移動

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- InTouch ActiveX コントロール、InTouch OCX DLL、または InTouch DLL オブジェクトを含むウィンドウの変換

  ウィンドウ変換では、InTouch ActiveX コントロール、InTouch OCX DLL、および InTouch DLL をサポートしていません。これらのコンポーネントは、新しいシンボルには含まれません。

- InTouch 翻訳文字列から産業用グラフィックへの移行

  InTouch 翻訳文字列は、アプリケーションのフォルダ内の XML ファイルに格納されます。各言語の翻訳文字列は、個別の XML ファイルに配置されます。すべてのウィンドウおよび SmartSymbol 文字列の翻訳は、ローカライズされた文字列をインポートした後、アプリケーション ディレクトリ内の XML ファイルから使用できます。

  以下の手順は、言語アシスタント ツールを使用して、InTouch ウィンドウ ローカリゼーション文字列を産業用グラフィックに移行する方法を示します。

  a. InTouch ウィンドウから産業用グラフィックへの変換
  b. InTouch 言語 XML ファイルの内容を言語アシスタントにインポートして、語句のグローバル ディクショナリを作成します。
  c. 産業用グラフィックを XML ファイルにエクスポートします。
  d. シンボル XML ファイルを言語アシスタントにインポートします。グローバル ディクショナリの翻訳が産業用グラフィックの語句に自動的に適用されます。
  e. 言語アシスタントから産業用グラフィック XML ファイルをパブリッシュします。
  f. 翻訳されたテキスト文字列を含む産業用グラフィック ツールボックスに XML ファイルをインポートします。

- InTouch スクリプトから産業用グラフィックへの移行

  InTouch または QuickScript 関数を含むスクリプトは産業用グラフィックに変換できません。

- InTouch 履歴オブジェクトから産業用グラフィックへの移行

  InTouch 履歴トレンドを含む InTouch ウィンドウは、産業用グラフィックには完全に変換されません。変換されたシンボルに履歴トレンドの一部だけが表示されることがあります。スクータ バーなどのトレンド コンポーネントは、変換された産業用グラフィックに表示されません。

ウィンドウを変換した後

変換されたシンボルは、System Platform IDE **ツールボックス**および WindowMaker の**産業用グラフィック ツールボックス**に追加されます。変換された元の InTouch ウィンドウのバックアップが作成されます。新しい InTouch ウィンドウが InTouch アプリケーションに作成され、新しく作成された産業用グラフィックが埋め込まれます。

ウィンドウのサブセットしか変換されない場合、変換されたウィンドウが未変換のウィンドウと共に機能することを手動で確認する必要があります。

変換されたウィンドウの名前が新しい産業用グラフィックの名前としてデフォルトで割り当てられます。InTouch ウィンドウの名前にサポートされない文字が含まれている場合、サポートされない各文字はアンダースコア（_）で置き換えられます。シンボルが既に存在する場合、変換されたシンボルの名前に数字のサフィックスが追加されます（Main_001 など）。

例外は、ドル記号（$）、シャープ記号（#）、およびアンダースコア（_）だけです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーションを移行した場合、ウィンドウの編集、またはウィンドウ プロパティの変更を行うと、ウィンドウ名に含まれる特殊文字はアンダースコア（_）で置き換えられます。この処理は、アプリケーション、枠、およびテンプレートを始めとするすべてのタイプのウィンドウに適用されます。

新しいツールセットが作成され、変換されたすべてのシンボルに InTouch アプリケーション名が割り当てられます。ウィンドウ変換の後、InTouch のフォルダ階層は、ツールセットによって維持されます。

- 割り当てられていない InTouch ウィンドウから変換されたシンボルは、InTouch アプリケーション名が割り当てられたツールセットに追加されます。
- ウィンドウが割り当て済み領域に属する場合、元のフォルダ構造が両方のツールボックス内に作成され、InTouch アプリケーションの名前が最上位のルート フォルダ名として割り当てられます。

ウィンドウの変換手順

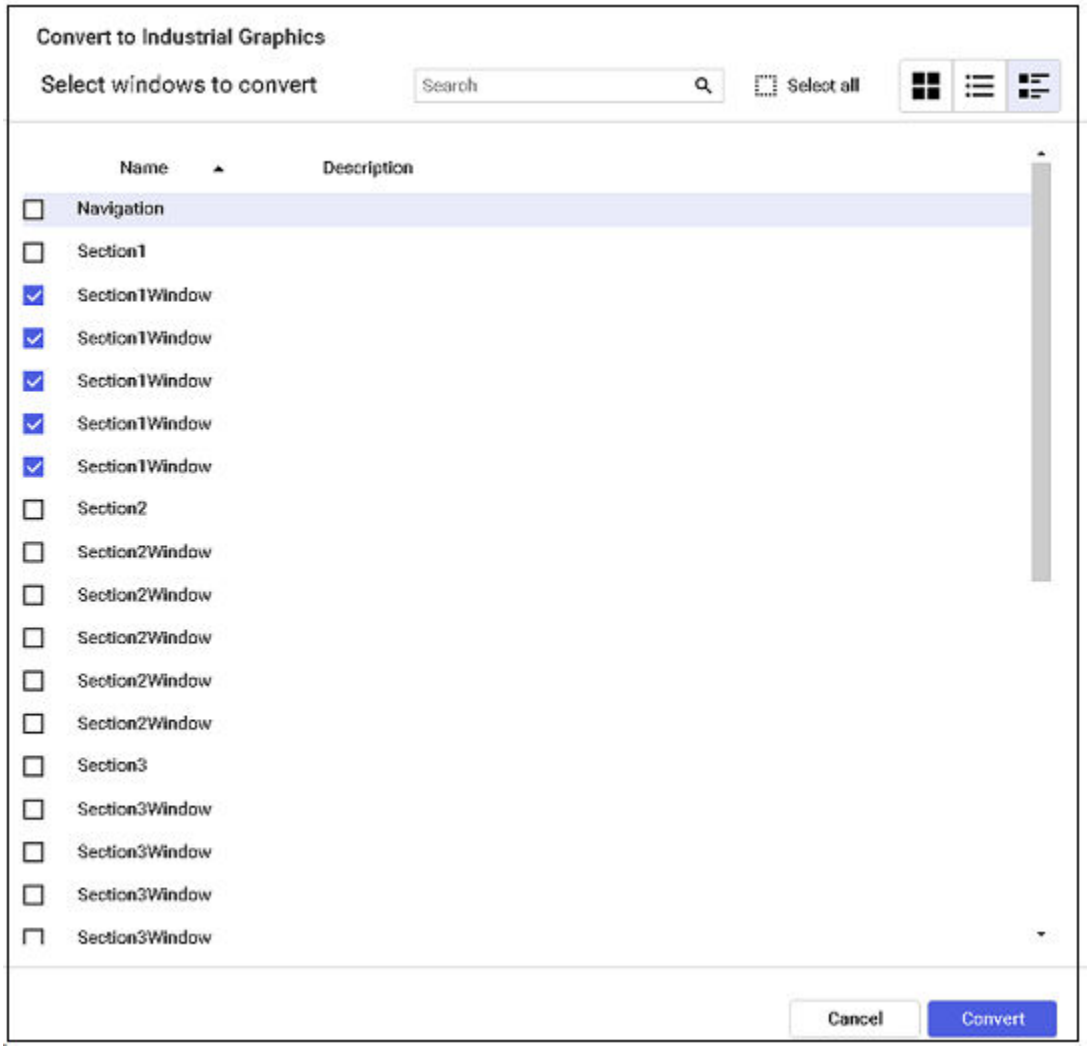
ウィンドウを産業用グラフィックに変換するには

1. 変換するすべての InTouch ウィンドウを閉じます。
2. 変換するウィンドウを［**ウィンドウ**］ペインで選択します。
3. 右クリックしてショートカット メニューを表示し、［**産業用グラフィックに変換...**］を選択します。

   ［**変換するウィンドウを選択**］画面が表示されます。
4. 変換するウィンドウを選択します。

   デフォルトでは、以前の画面で選択したウィンドウが選択されます。追加のウィンドウを選択できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. **[変換] を選択します。**

   選択した複数のウィンドウが連続して変換されていることを示すメッセージが表示されます。ウィンドウが変換された後、複数の［**チェック イン**］ダイアログ ボックスが連続して表示され、変換された各ウィンドウのコメント（オプション）を入力できます。

6. WindowMaker の**産業用グラフィック ツールボックス**および System Platform IDE の**グラフィック ツールボックス**を確認します。

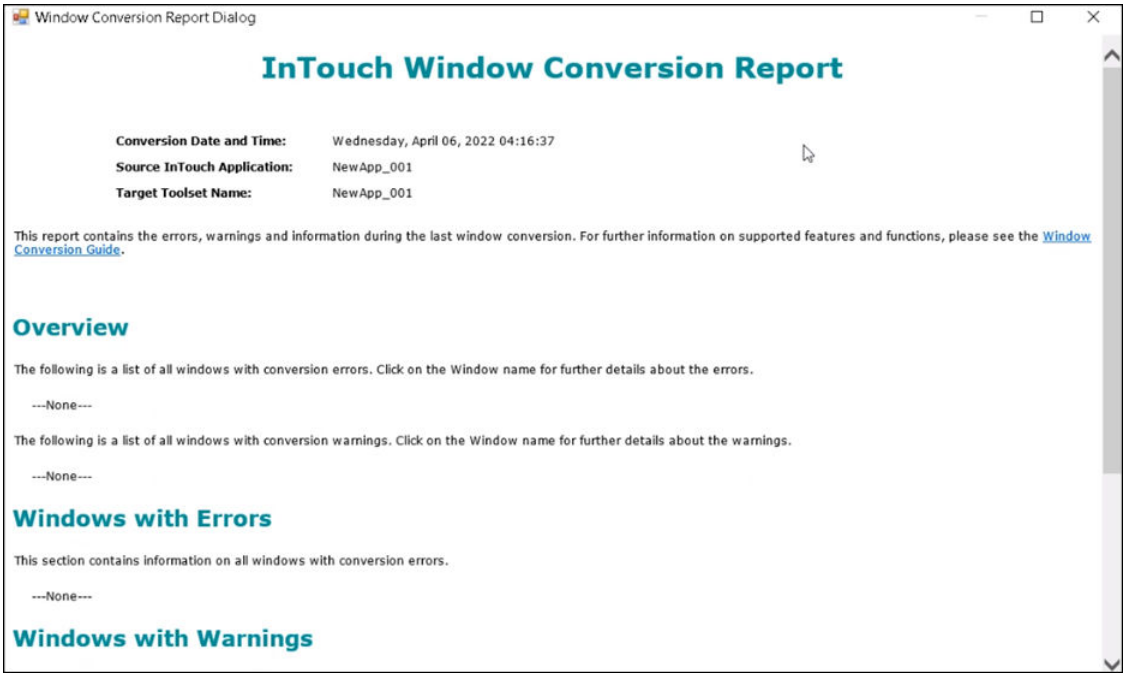
   変換されたウィンドウが両方のツールボックスに産業用グラフィックとして表示されていることを確認します。

ウィンドウ変換エラーの診断

変換中、進行状況バーにウィンドウ変換中に発生した警告メッセージやエラー メッセージが表示されます。ウィンドウの変換中、各エレメント、アニメーション、またはスクリプトに関する有益な診断情報を含むカスタム ログ フラグが作成されます。

変換後、進行状況バーの［**Window conversion Report**］（**ウィンドウ変換レポート**）を選択して、すべてのメッセージを HTML 変換レポートにエクスポートします。このレポートは、メッセージ ウィンドウで表示できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

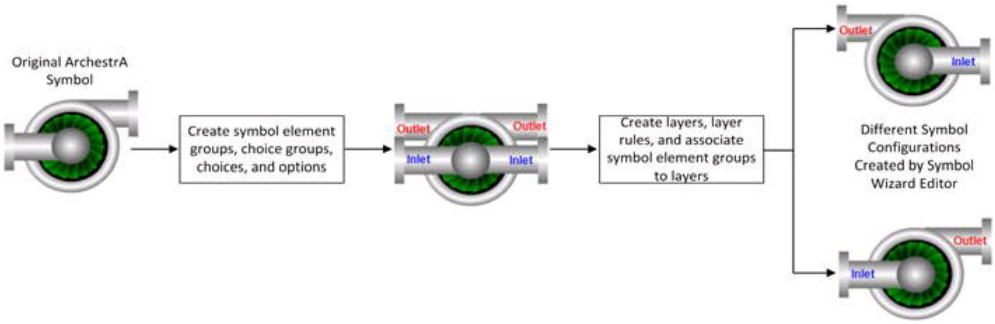


変換の進行状況ダイアログボックスと変換レポートの両方には、ウィンドウ変換に関する以下の情報が含まれます。

- ウィンドウ名。
- ウィンドウが正常に変換されたかどうか、またはエラーが発生したどうかを示す変換ステータス。
- 変換中に発生した警告メッセージ。
- ウィンドウ変換中に発生したエラー メッセージ。

シンボル ウィザード エディタでのシンボル ウィザードの作成

産業用グラフィック エディタには、シンボルのさまざまな表示設定および機能設定を作成できるシンボル ウィザード エディタが含まれています。複数の設定を含むシンボルは、シンボル ウィザードと呼ばれます。シンボル ウィザードの作成の例は、中央のブレード ハウジングの左または右の排水管を示すポンプ シンボルです。シンボル ウィザード エディタを使用して、両方の設定を 1 つのシンボル ウィザードに含めることができます。



シンボル ウィザードは、特定の Application Server オブジェクト テンプレートや Application Server オブジェクト インスタンスに関連付けられていません。特定のシンボル設定を選択できることを除き、シンボル ウィザードは標準の産業用グラフィックと同様に動作します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

シンボル ウィザードには複数の異なる設定を含めることができるので、アプリケーションに対して作成する必要のあるシンボルの数が削減されます。その結果、保守労力が削減され、InTouch HMI アプリケーションを構築する際の設計時の問題も少なくなります。

シンボル ウィザードの作成

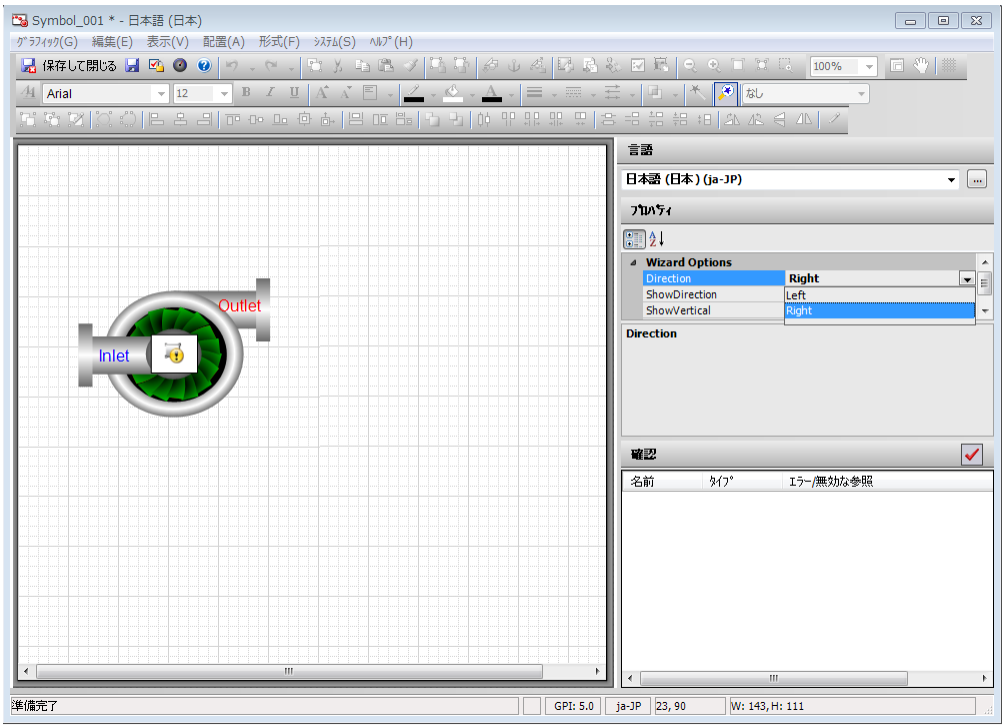
シンボル ウィザードの作成および実装のプロセスには、デザイナ ワークフローとコンシューマ ワークフローという 2 つのワークフローがあります。

- デザイナは、シンボル ウィザード エディタを使用して、1 つのシンボルの複数の設定を作成します。
- コンシューマは、シンボル ウィザードの適切な設定を選択して、シンボルをマネージド InTouch アプリケーションに埋め込みます。

シンボル ウィザードのデザイナ ワークフロー

デザイナはシンボル ウィザード エディタを使用して、さまざまなシンボル設定を定義し、グラフィック要素、カスタム プロパティ、およびスクリプトをレイヤーの形式でシンボル設定に関連付けます。規則は、レイヤーをシンボル設定に含めるときを決定する選択グループ、選択、およびオプションに関連付けられます。デザイナは、シンボルのデフォルトとなる設定を選択します。このデフォルト設定は、シンボルがマネージド InTouch アプリケーションに埋め込まれたときに表示されます。

すべてのシンボル設定を作成した後、デザイナは、シンボル ウィザード プレビューを使用して、シンボルの各設定をコンシューマに示す方法を検証します。デザイナは、「**ウィザード オプション**」を使用して、シンボルに対して設定されているレイヤー規則に基づいて設計されたとおりに各設定が表示されることを確認します。



デザイナは、マネージド InTouch アプリケーションに埋め込むことができるように、終了したシンボル ウィザードを Galaxy ライブラリに保存します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

シンボル ウィザードの作成に関する詳細については、『産業用グラフィック エディタ ユーザー ガイド』を参照してください。

シンボル ウィザードのコンシューマ ワークフロー

コンシューマがシンボル ウィザードのインスタンスをマネージド InTouch アプリケーションに埋め込むと、シンボルのデフォルト設定が選択されます。コンシューマは、シンボル ウィザードの［**ウィザード オプション**］ビューからオプションを選択して設定を変更できます。選択した設定よっては、コンシューマが選択できる追加の設定関連プロパティがあります。

InTouch アプリケーションが実行しているとき、シンボル ウィザードは、コンシューマによって選択された設定として表示されます。シンボルの設定は、ランタイムで変更することはできません。

シンボル ウィザードをマネージド InTouch アプリケーションに埋め込む方法の詳細については、「シンボルの埋め込みウィザード」を参照してください。

トレンド ペンの履歴データ検索について

トレンド ペンを含むアプリケーションが WindowViewer で実行開始したとき、Historian クエリーが全体のトレンド ペン期間のデータを取得します。リアルタイム データは履歴データが取得されている期間中にトレンド ライン上にプロットされます。履歴データは検索後、リアルタイム データに追加され、全体期間についてトレンド ラインをバックフィルします。

以下はトレンド ペンを設定するための手順です。通常、トレンド ペンの設定には、トレンド ペン プロットがグラフィックのプロセス値の経時的変化を表示することを視覚的に示すためのメーター グラフィックの隣にトレンド ペンを配置する複数のステップが含まれます。

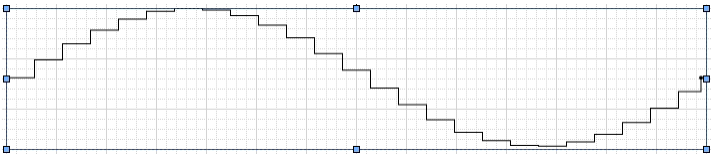
トレンド ペンを設定するには

1. ［視覚化］フォルダから［**トレンド ペン**］を選択します。

   カーソルを産業用グラフィック エディタのキャンバス領域に置くと、十字型に変化します。

2. トレンド ペン プロットの水平の境界として設定するキャンバス上の位置にカーソルを置きます。
3. キャンバスでドラッグして、トレンド ペン コントロールの長方形を選択します。

   トレンド ペンの長方形の水平方向および垂直方向の境界は、ランタイム時におけるトレンド ペン プロットの描画領域を表します。長方形の水平軸はトレンドの期間を表します。垂直軸はトレンドの可能な値の範囲を表します。

   ［**トレンド ペン**］ダイアログ ボックスが表示されます。トレンド ペン グラフィックをダブルクリックするか、または［**システム**］メニューの［**アニメーションの編集**］を選択し、［**トレンド ペン**］ダイアログ ボックスを表示できます。

4. ［**参照**］フィールドに参照を入力します。

   参照はトレンドによって値として表示されるデータ ソースで、オブジェクトの属性、アナログ タグ値、またはカスタム プロパティのような外部参照にできます。定数と式は許可されません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［**Historian**］または［**InTouch Log History/LGH**］を選択して、履歴ソースを表示します。［Historian］を選択した場合、手順 7 に進みます。
6. ［**InTouch Log History/LGH**］を選択した場合、**UNC パス**に対してアイコンを使用して、フィールドへの入力を式または静的テキストとして切り替えることができます。
   - 式モードを選択した場合、省略記号ボタンを選択して HMI の属性/タグ ブラウザを起動し、カスタム プロパティ、属性、またはタグを選択できます。
   - 静的テキスト モードを選択する場合、省略ボタンを選択してファイル ブラウザを起動し、LGH ファイル名を指定できます。
7. Historian の位置を識別するメソッドに、［**自動検出**］または［**式**］ を選択します。
   - **自動検出**: Historian Server は参照属性が実行されている AppEngine から自動検出されます。たとえば、［**参照**］ フィールドが UDO.UDA1 に設定されている場合、［**自動検出**］は、UDO が実行されている AppEngine に設定された Historian Server 名に設定されています。自動検出は Application Server リファレンスにのみ有効です。
   - **式 :** 式または参照を［**サーバー名**］フィールドに入力すると、トレンド ペンが、指定した履歴サーバーに関連付けられます。

   ［**サーバー名**］フィールドの左にあるアイコンは、フィールドへの入力を式モードまたは静的テキスト モードの間で切り替えます。

   トレンド ペンは、［**式**］モードで［**サーバー名**］フィールドが空白の場合にライブ データのみ表示します。
8. トレンド期間のタイプとして、［**移動**］または［**固定**］を選択します。
   - **移動 :** トレンド期間の開始時間は現在の時刻で、終了時間は開始時間からの期間です。次の期間の開始時間は、過去のトレンド期間の終了時間に設定されます。
   - **固定 :** 固定トレンド期間において、StartTime および EndTime プロパティは自動で変化しません。初期値では、トレンド期間の開始時間は現在の時間です。StartTime プロパティはスクリプトによって変更できます。

   トレンド期間の EndTime プロパティ（移動と固定の両方）は読み取り専用です。トレンド期間の終了時間は、指定した開始時 j 館と期間から計算されます。
9. ［**経過時間（分）**］フィールドにトレンド ラインの X 軸期間を設定します。

   トレンド期間は定数、外部参照、式、またはカスタム プロパティとして指定できます。浮動小数点数を入力した場合、期間は最近似の分に丸められます。

   最小トレンド期間は 1 分で、最大期間は 10080 分（1 週間）です。
10. プロセス値をトレンド ラインに配置するために、［**自動範囲**］または ［**範囲外の値を除外**］をスケール メソッドとして選択します。

    ［**自動範囲**］を選択する場合、［**最小範囲**］と［**最大範囲**］フィールドが無効になります。トレンド ラインの Y 軸は、自動で調整され、トレンド ペン グラフィックの上限と下限の間にあるトレンド値の全範囲を表示します。

    ［**範囲外の値を除外**］ を選択する場合、［**最小範囲**］と［**最大範囲**］フィールドが有効になります。［**最小範囲**］と［**最大範囲**］はトレンドの Y 軸値範囲の下限と上限を設定します。どちらのフィールドも、定数、外部参照、カスタム プロパティに設定できます。

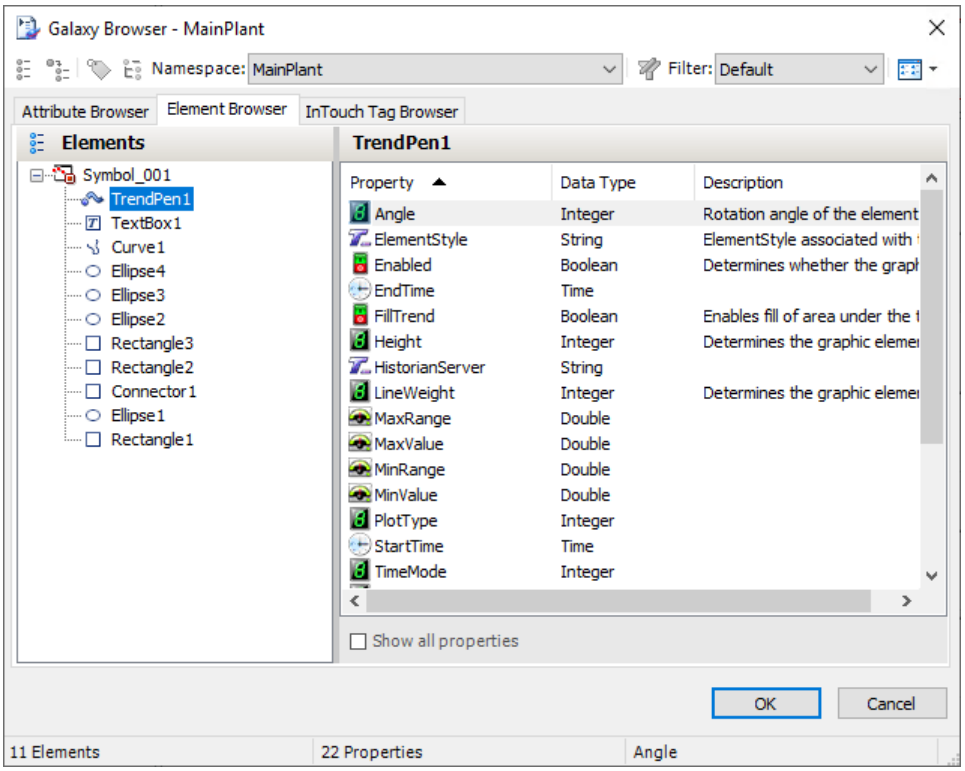
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［範囲外の値を除外］を使用したトレンドの下限と上限を値が超える場合、トレンド ラインは値の範囲の制限で切り捨てられ、プロセス値がトレンド値の範囲の外にあるときに水平線として表示されます。

11. ［プロット タイプ］で、［ステップ ライン］または［線］をトレンド プロットのタイプとして選択します。
    - ［ステップ ライン］ プロットはトレンド データ ポイントから次のデータ ポイントの時間まで、トレンドの X 軸に水平の線を描画し、データ ポイントまで垂直の線を描画します。
    - ［線］プロットはトレンド期間中の各連続ポイントを線で直接結びます。

ランタイム時のトレンド ペン プロパティの変更

トレンド ペンに含まれるプロパティの値は、ランタイム時に修正して、トレンドの視覚的および機能的な特性を変更できます。次の図は、ランタイム時に変更できる属性、タグ、または値を検索するための一般的な HMI ソフトウェア ブラウザまたはエクスプローラの一例です。



以下のサブトピックでは、一般的にランタイム時に修正して、トレンド ペンの視覚的、機能的な特徴を変更するプロパティについて説明します。

MinValue プロパティ

ランタイム時に、MinValue プロパティの値を修正して、トレンドに表示される最小の測定値を変更できます。デザイン時にトレンド ペンの［Y 軸範囲］オプションに割り当てられた値に基づいて、ランタイム時に、MinValue は読み書き可能、または読み取り専用プロパティにできます。

- デザイン時に［Y 軸範囲］を［自動範囲］に設定する場合

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

トレンドによって表示される最小測定値は、履歴またはトレンド期間の現在のデータから受け取ったデータの最小値に設定されます。MinValue は読み取り専用です。

- デザイン時に［**Y 軸範囲**］を［**範囲外の値を除外**］に設定する場合

  トレンドによって表示される最小測定値は、デザイン時における［**最小範囲**］オプションの下限に設定されます。

  MinValue は読み書き可能で、ランタイム時に変更できます。ランタイム時に MinValue を変更する場合、トレンド ラインは MinValue および MaxValue プロパティに割り当てられた値に基づいて再描画されます。

  MinValue および MaxValue プロパティに割り当てられた値が等しい場合、トレンドの Y 軸範囲は自動的に自動範囲に変更されます。

MaxValue プロパティ

ランタイム時に、MaxValue プロパティの値を修正して、トレンドに表示される最大の測定値を変更できます。デザイン時にトレンド ペンの［Y 軸範囲］オプションに割り当てられた値に基づいて、ランタイム時に、MaxValue は読み書き可能、または読み取り専用プロパティにできます。

- デザイン時に［**Y 軸範囲**］を［**自動範囲**］に設定する場合

  トレンドによって表示される最大測定値は、履歴またはトレンド期間の現在のデータから受け取ったデータの最大値に設定されます。MaxValue は読み取り専用です。

- デザイン時に［**Y 軸範囲**］を［**範囲外の値を除外**］に設定する場合

  トレンドによって表示される最大測定値は、デザイン時における［**最大範囲**］オプションの上限に設定されます。

  MaxValue は読み書き可能で、ランタイム時に変更できます。ランタイム時に MaxValue を変更する場合、トレンド ラインは MinValue および MaxValue プロパティに割り当てられた値に基づいて再描画されます。

  MinValue および MaxValue プロパティに割り当てられた値が等しい場合、トレンドの Y 軸範囲は自動的に自動範囲に変更されます。

StartTime プロパティ StartTime ぷろぱてぃ

ランタイム時に、StartTime プロパティの値を修正して、デザイン時にトレンド ペン［**実行周期**］に設定された値に基づいて、トレンド期間の開始時間を変更します。

- デザイン時に［**実行周期**］を［**固定**］に設定する場合

  StartTime プロパティに割り当てられているデフォルト値は、トレンド ペンが WindowViewer に最初に表示されるときで、開始時間は時間の経過と共に変化します。StartTime は読み取り/書き込みで、ランタイム時に変更できます。StartTime の値が変化すると、トレンド ペンは新しい StartTime 値を使用してトレンドを再プロットします。

- デザイン時に［**実行周期**］を［**移動**］に設定する場合

  StartTime プロパティに設定された値は現在のシステム日付時間です。StartTime は読み取り専用です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

EndTime プロパティ

ランタイム時に、EndTime プロパティの値を修正して、デザイン時にトレンド ペン［**実行周期**］に設定された値に基づいて、トレンド期間の終了時間を変更します。

- デザイン時に［**実行周期**］を［**固定**］に設定する場合

  EndTime プロパティに割り当てられたデフォルト値は、デザイン時に設定られた終了時間です。EndTime は読み書き可能で、ランタイム時に変更できます。EndTime の値が変化すると、トレンド ペンは新しい EndTime 値を使用してトレンドを再プロットします。

- デザイン時に［**実行周期**］を［**移動**］に設定する場合

  EndTime プロパティに設定された値は現在のシステム日付時間です。EndTime は読み取り専用です。

PlotType プロパティ

ランタイム時に、**PlotType** プロパティの値を修正して、トレンド プロットのタイプを変更できます。

- **PlotType** が 0 のとき、トレンド ペン プロット タイプはステップ ラインになります。(デフォルト)
- **PlotType** が 1 のとき、トレンド ペン プロット タイプは線になります。

PlotType の値は値が 0 でも 1 でもない場合は無視されます。

ランタイム時に **PlotType** の値が変化すると、トレンドを描画する前に、トレンド データは再度読み出されます。

TimeMode プロパティ

ランタイム時に、**TimeMode** プロパティの値を修正して、トレンド期間のタイプを変更できます。

- **TimeMode** が 0 のとき、トレンド期間のモードは**移動**です。トレンドの終了時間は現在の時間です。(デフォルト)
- **TimeMode** が 1 のとき、トレンド期間のモードは**固定**です。トレンドの開始時間は、トレンド ペンが WindowViewer に最初に表示されるときで、開始時間は時間の経過と共に変化します。

TimeMode の値は、値が 0 でも 1 でもない場合は無視されます。

ランタイム中、期間が移動から固定に変化した場合、トレンドの開始時間と終了時間は以前と同じになり、データも維持されます。ランタイム中、期間が固定から移動に変化した場合、トレンドを描画する前にデータが再度読み出されます。トレンドの開始時間と終了時間は移動モードによって自動で調整されます。

FillTrend プロパティ

ランタイム時に FillTrend プロパティを使用して、トレンド ペン曲線の下の領域の外観を変更できます。

FillTrend プロパティは、スクリプトを使用して変更することもできます。たとえば、`TrendPen6.FillTrend = not TrendPen6.FillTrend;` と指定します。

一般的な要素スタイルの手順ルールに従って、塗りつぶしスタイル アニメーションおよび要素スタイル アニメーションの Fill プロパティ、または要素スタイルが FillTrend プロパティに適用されるようになりました。

塗りつぶしスタイル アニメーションの詳細については、「塗りつぶしスタイルの設定」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素スタイル アニメーションの詳細については、「アニメーション スタイルを使用したアニメーションの設定」を参照してください。

注記：FillTrend プロパティはデザイン時またはランタイム時に有効化することはできます。デザイン時に色が選択されていない場合、ランタイム時、色はデフォルトで白に設定されます。

マルチ ペン トレンドを設定するには

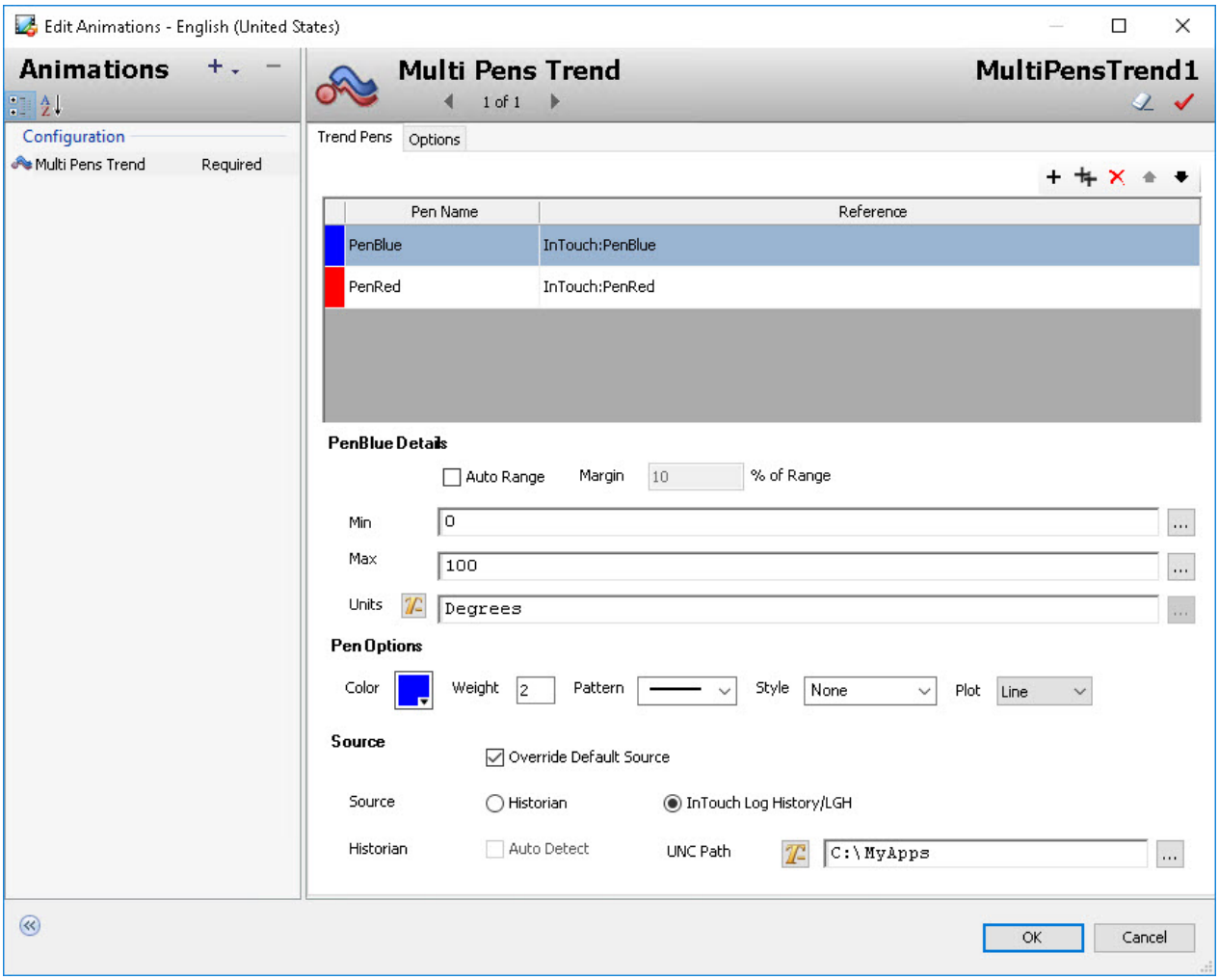
マルチ ペン トレンドはトレンド ペンと同様に一連のプロセス値を表示しますが、複数のトレンド線が表示されます。トレンド ペンのタイプの詳細については、「トレンド ペンの設定」内のセクションを参照してください。Industrial Graphics Programmic API を使用して、マルチペントレンドが埋め込まれたシンボルをエクスポートおよびインポートできます。グラフィック要素をエクスポートおよびインポートする方法の詳細については、AVEVA Industrial Graphic Editor のヘルプの「プログラムによる産業用グラフィックのエクスポートとインポート」を参照してください。

注記：
- マルチ ペン トレンドは英語以外のオペレーティング システムではテストされていません。
- マルチ ペン トレンドのパフォーマンス インデックス計算は GPI に含まれていません。
- AVEVA OMI ViewApps はマルチ ペン トレンドをサポートしません。
- Web Client では、マルチ ペン トレンドがサポートされません。

マルチ ペン トレンドを設定するには

1. ［**マルチ ペン トレンド**］要素を選択します。
2. グラフィック エディタ キャンバス上をクリックします。
3. ［**アニメーションの編集**］ダイアログ ボックスが表示されます。各ペンに対して、［**トレンド ペン**］タブの以下のセクションを設定します。
   - ペン名と参照
   - ペンの詳細
   - ペンのオプション
   - ソース

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



ペン名と参照の設定

［トレンド ペン］タブで各ペンに対して以下の設定を行います。

1. ［**ペン名**］フィールドにペンの名前を入力します。

   ペン名フィールドには以下のルールが適用されます。

   - 入力できるのは静的テキストだけです。
   - 1 つの文字を含む必要があります。
   - 最大文字数は 32 です。
   - 有効な文字は、英数字および特殊文字（$、#、_）です。
   - 2 つのトレンド ペンに同じ名前を付けることはできません。
   - ペン名では大文字と小文字が区別されません。

   設定したペン名はツールヒントに表示されます。

2. ［**参照**］フィールドに参照を入力します。フィールドの横にある省略記号ボタンを使用して参照を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照はトレンドによって値として表示されるデータ ソースで、オブジェクトの属性、InTouch アナログ型タグ値、またはカスタム プロパティのような外部参照にできます。定数と式は許可されません。詳細については、「入力モードの設定 ()」を参照してください。

3. デフォルトでは、2 つの空の行がグリッドに作成され、最大 8 つのペンを作成できます。
4. トレンド ペンを追加するには、+ アイコンを使用します。トレンド ペンを削除するには、× アイコンを使用します。ペンの表示順序を変更するには、↑ および ↓ の矢印を使用します。既存のペン設定を複製するには、アイコンを使用します。

ペンの詳細の設定

ペンの詳細セクションで個々のペンの範囲と単位を設定できます。

1. スケール方法に［**自動範囲**］を選択すると、トレンド線にプロセス値が配置されます。

   ［**自動範囲**］を選択すると［**最小**］と［**最大**］フィールドが無効になります。トレンド線の Y 軸は、自動で調整され、トレンドの上限と下限の間にあるトレンド値の全範囲を表示します。

   a. ［**マージン**］のパーセンテージ値を指定します。ランタイム時、トレンド領域のマージンはトレンド値と指定したマージンのパーセントに基づいて動的に表示されます。これは表示のみを目的としていて、軸の値には影響しません。

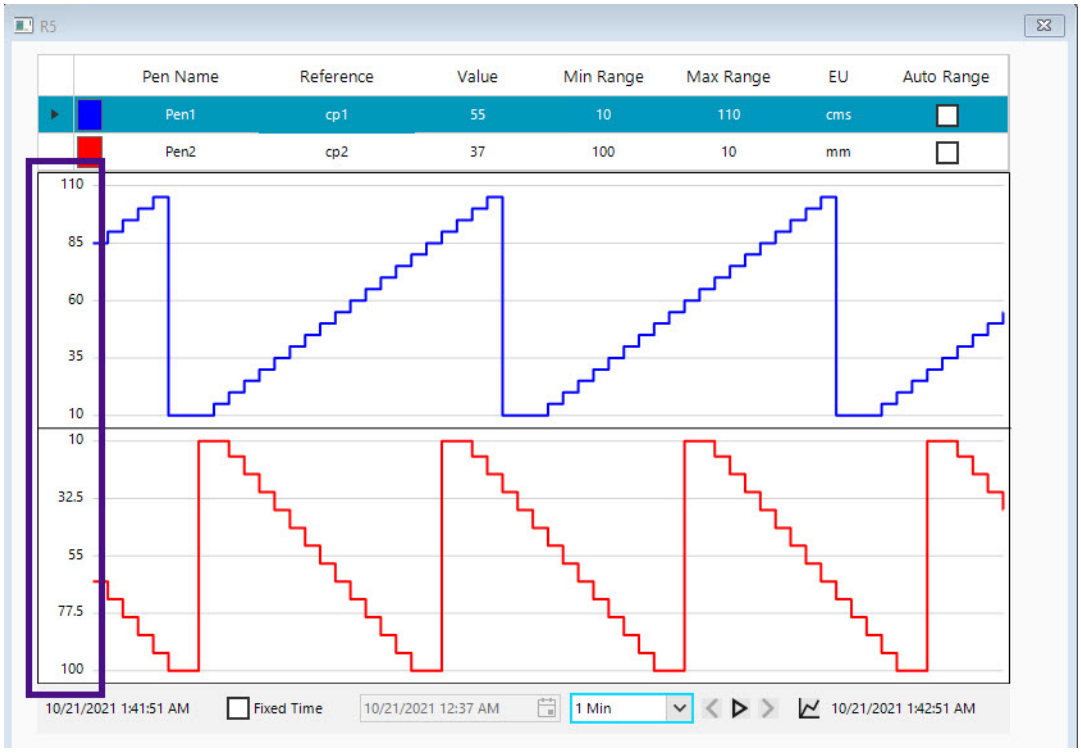
      最小値が 10、および最大値が 20 に設定されている場合を考えてみます。範囲は 10(20~10)です。範囲は 10 (20-10) です。
      20% のマージンを指定すると、マージンは「範囲 x マージン %」として計算されるので、この例では 10 x 20% = 2 になります。
      Y 軸の最大値は「最大値 + マージン」です。この例では 20 + 2 = 22 になります。Y 軸の最小値は「最大値 - マージン」です。この例では 10 - 2 = 8 になります。

2. ［**自動範囲**］を選択しない場合、選択したペンの Y 軸の［**最小**］と［**最大**］の値を指定します。

   値を直接指定するか、［**...**］Galaxy ブラウザが表示されます。

   最小値が最大値よりも大きい場合、トレンド ペンはランタイム時に違う形で描画され、スケールは最大値が最小値よりも大きい場合と逆になります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 属性またはタグを参照して選択します。
4. ［**単位**］を入力します。トグル ボタンを使用して静的テキストを入力するか、Galaxy ブラウザから参照値を選択します。例: 度。
5. ［**OK**］をクリックします。

ペン オプションの設定

［ペン オプション］セクションを使用して個々のペンのスタイルを設定できます。

1. **色**：ペンの表示色を選択します。
   a. カラー ピッカーを使用して画面の部分から色を選択できます。
   b. ［**パレットのロード...**］をクリックしてカスタム パレットを読み込むこともできます。
   c. **+** を押して色をカスタム パレットに追加することや、Delete ボタンを押して色を削除することもできます。
   d. 設定したカスタム パレットは保存して後で使用できます。［**パレットの保存**］をクリックします。
2. ペンの太さを入力します。 トレンド ペン線の太さが指定されます。
3. **パターン**：事前定義済みのオプションのリストからトレンド ペン線の表示パターンを選択します。
4. **スタイル**：事前定義済みの［スタイル］オプションのリストから選択できます。スタイルは IDE の［要素スタイル］で設定されます。詳細については、「要素スタイルの視覚的プロパティの変更」を参照してください。
5. **プロット**：これは表示されるトレンド線のタイプです。［ステップ ライン］または［線］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

6. ［**OK**］をクリックします。

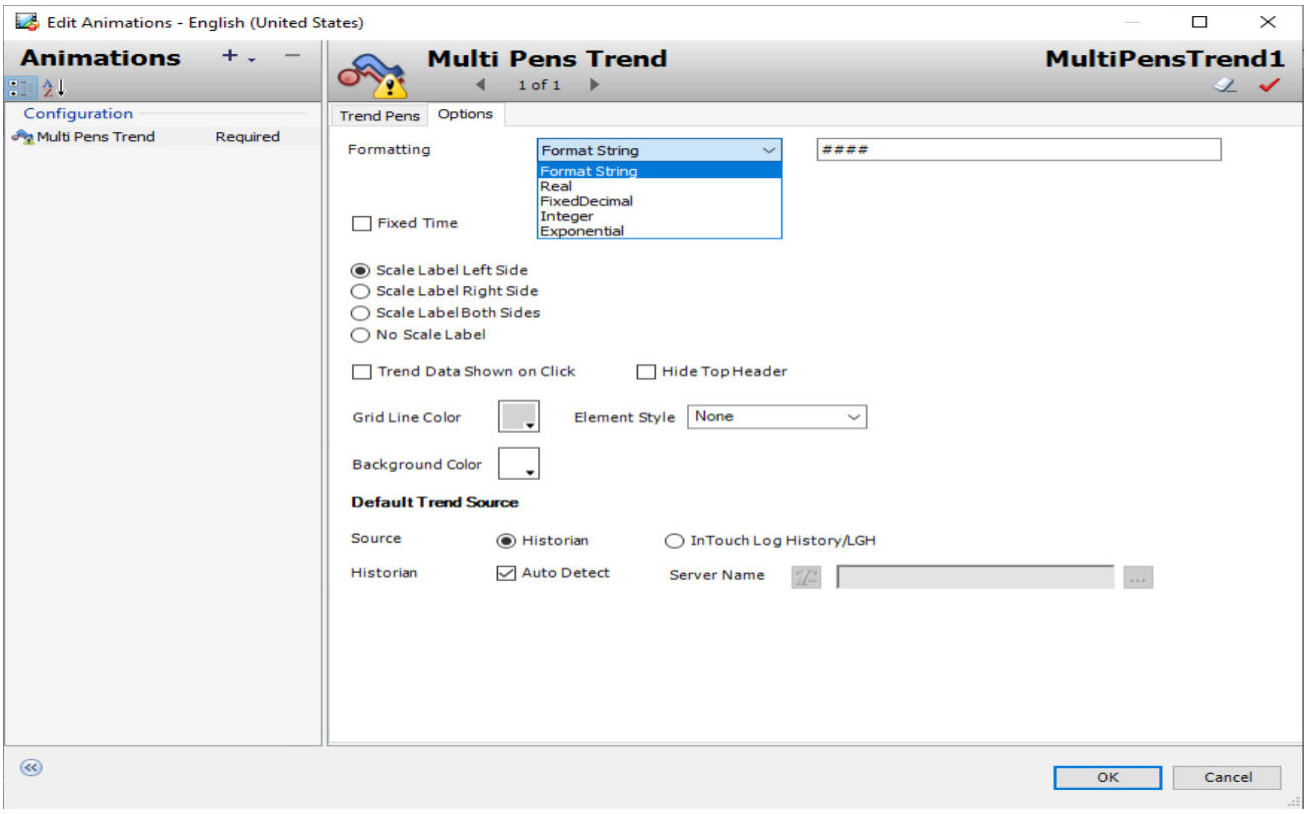
ソースの設定

［**オプション**］タブで設定したデータの履歴ソースを上書きできます。詳細については、「マルチ ペン トレンドのカスタマイズ ()」を参照してください。

- 履歴データのソースを変更するには、［**デフォルト ソースの上書き**］を選択します。

マルチ ペン トレンドのカスタマイズ

［**オプション**］タブを使用してランタイム時のマルチ ペン トレンドの表示をカスタマイズできます。



1. [書式設定] オプション を 使用して、次の値の書式を設定します。
   - 値
   - 最小範囲
   - 最大範囲
   - カーソル読み出し時の値

   既定では、 [文字列の **書式設定]** が選択されます。ユーザーは、0、#、コンマ、および小数点を使用して、上記の値を書式設定できます。

   Real、FixedDecimal、Integer、指数などのその他の選択は、**IDE** > **Galaxy** > Configure > **Styles** から設定できます。上記の値は、選択したスタイルに従って書式設定されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**固定時間**］設定は、固定時間モード（選択時）および移動モード（非選択時）を切り替えることができます。［固定時間］チェック ボックスをオンにすると、指定した期間のトレンド データがトレンド領域に表示され、移動しません。
3. ドロップダウン リストの事前定義済みオプションのリストから［時間間隔］を選択します。ここで選択した時間間隔がランタイム時の初期値として設定されます。
4. スケール ラベルを表示するトレンド領域内の場所を選択します（左、右、両方、または非表示）。
5. ［**クリック時にトレンド データを表示**］をクリックすると、ランタイム時のカーソル読み出しダイアログの動作をカスタマイズできます。

   デフォルトでは、カーソルをトレンド領域の上に置くとカーソル読み出しダイアログが表示されます。このオプションを選択すると、トレンド領域をクリックしたときにカーソル読み出しダイアログを表示することや非表示にすることができます。
6. ランタイムの上部ヘッダーを非表示にするには、［**上部ヘッダーの非表示**］を選択します。トレンド領域とツールバーだけが表示されます。
7. 現在の値に従ってデータ品質のトレンドを描画しプロットするには、[品質のデータを **レンダリング**]チェックボックスをオンにします。良好以外のデータ品質については、トレンドは点線でプロットされます。このチェックボックスが選択されていない場合、トレンドは、最後に有効な値に対して、直線状の点線でプロットされます。
8. ［**グリッド線の色**］オプションを使用してグリッド線の色を選択します。
9. ［**要素スタイル**］ドロップダウンリストから定義済みスタイルを選択します。
10. ［**背景色**］オプションを使用して背景の色を選択することもできます。
11. ［**デフォルトのトレンド ソース**］の［**ソース**］フィールドで［Historian］または［InTouch ログ履歴/LGH］を選択します。

    デフォルトでは、［Historian］ソースおよび［自動検出］オプションが選択されます。［**自動検出**］を選択すると、［サーバー名］が無効になります。

    式または参照を［サーバー名］フィールドに入力すると、指定した Historian Server にマルチ ペン トレンドが接続されます。［**サーバー名**］フィールドの左にあるボタンを使用して、フィールドへの入力モード（式または静的テキスト）を切り替えます。

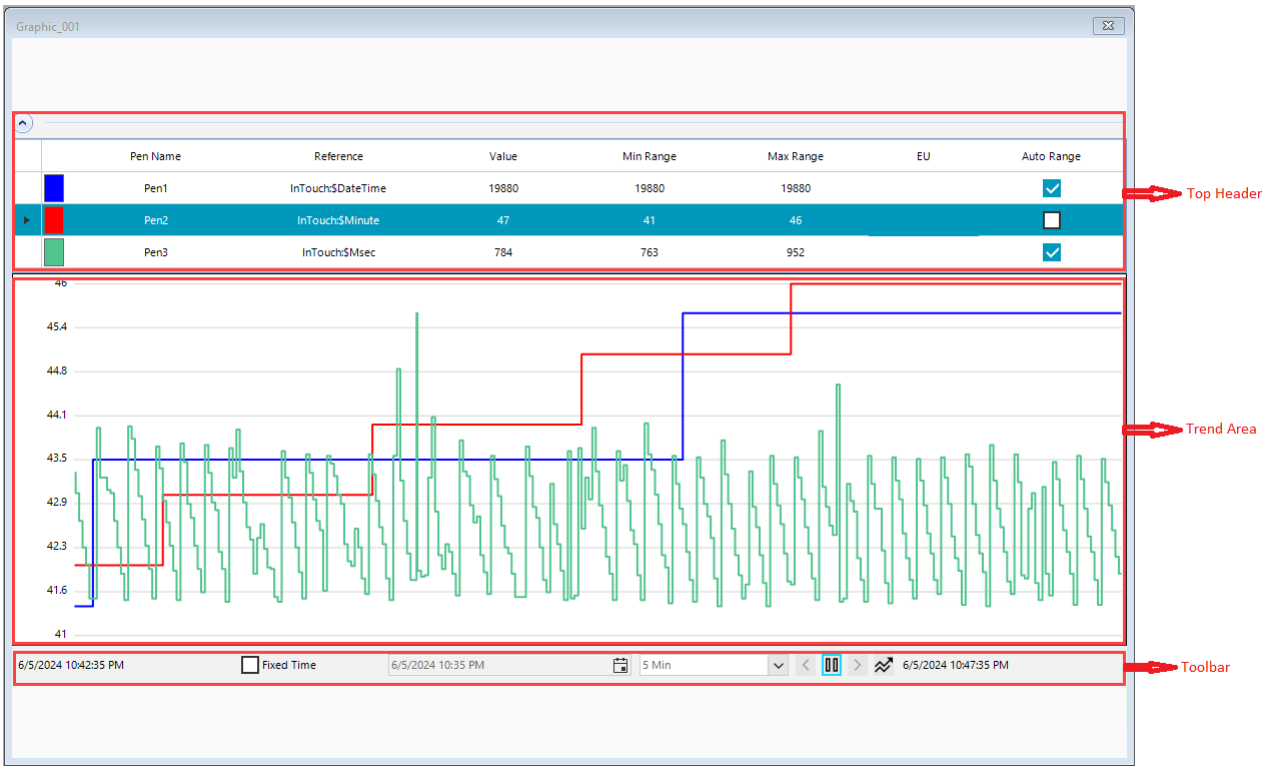
    Historian Server は参照属性が実行している AppEngine から自動的に検出されます。たとえば、［参照］フィールドが UDO.UDA1 に設定されている場合、［自動検出］は、UDO が実行している AppEngine に設定された Historian Server の名前に設定されます。［自動検出］は Application Server 参照に対してのみ有効です。トレンド ペンは、［式］モードで［サーバー名］フィールドが空白の場合にライブ データのみ表示します。
12. ［**InTouch ログ履歴/LGH**］を選択した場合、［**UNC パス**］フィールドの横にあるボタンを使用して、フィールドへの入力モード（式または静的テキスト）を切り替えます。
    a. 式モードを選択した場合、省略記号ボタンをクリックして HMI の属性/タグ ブラウザを起動し、カスタム プロパティ、属性、またはタグを選択できます。
    b. 静的テキスト モードを選択する場合、省略記号ボタンをクリックしてファイル ブラウザを起動し、LGH ファイルの場所を指定できます。

ランタイム時のマルチ ペン トレンドの使用

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowViewer でのランタイム時のマルチ ペン トレンドの動作を変更することができます。たとえば、ペンの追加、ペンの削除、または履歴トレンド データの操作を行うことができます。マルチ ペン トレンドは 3 つの領域で整理できます。

- テーブル形式でペンの詳細が一覧表示されます。
- この領域ではペンが視覚的に表現されます。
- 再生およびスタック オプションが含まれます。



カーソル読み出しダイアログ

カーソル読み出しダイアログは、カーソルがペン上の特定のポイントに置かれると、以下のフィールドを表示します：

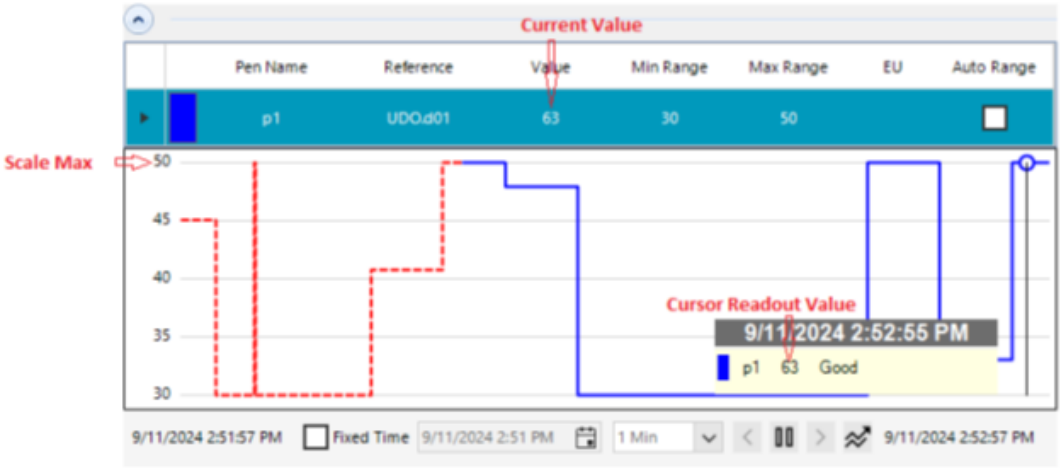
- ﾍﾟﾝ名
- 値
- 品質
- 工学単位

[カーソルの読み取り]ダイアログを表示するには、カーソルをトレンド領域に置くと、カーソルの読み出しが指定した時間の上記の値とともに表示されます。

[自動範囲]が選択されていない場合、ペンの参照値が最小/最大範囲を超えた場合、カーソルの読み出し時には実際の値が表示されますが、プロットされているトレンドには平坦線が表示されます。

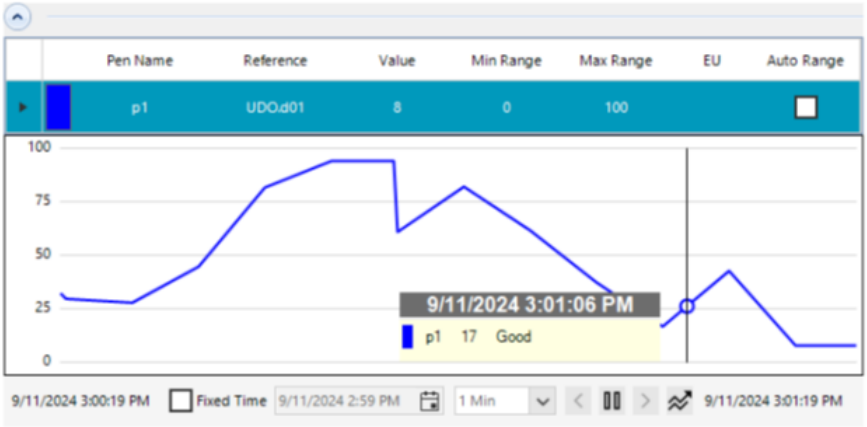
たとえば、次のスクリーンショットでは、現在の p1 値は 63 です。最新のカーソル読み出し値にも 63 が表示されますが、ペンのプロットは [最大範囲] の設定に基づいて 50 と表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



[プロット]モードが[線]モードの場合、カーソルの読み出しには前回のデータポイント値が表示され、カーソルは傾きの中央に表示されます。

たとえば、次のスクリーンショットでは、カーソルは 2 つのデータポイント(17 と 45)の間にあります。カーソルはスケールに基づいて 26 に表示されますが、カーソルの読み出し値は 17 を前回のデータポイントとして表示します。



データ品質の表示

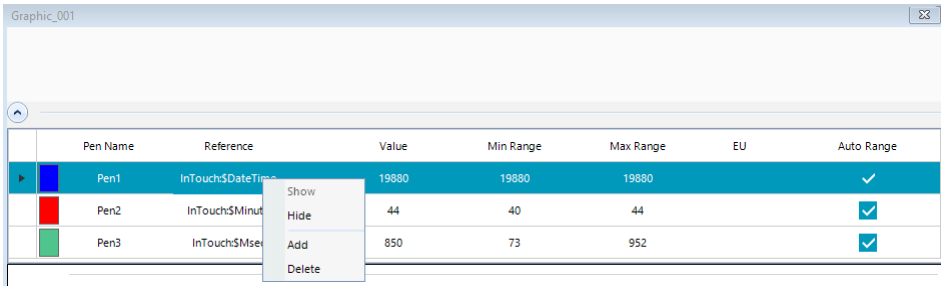
参照に高品質のデータがない場合、トレンド ペンのスタイルは低品質スタイルになります。表示されるトレンド値は、[品質]チェックボックスの **[Render data]** の設定によって異なります。参照に[Good]以外の品質がある場合、悪いスタイルが適用されます。

上部ヘッダー オプション

上部ヘッダー領域を使用してランタイム時にペンの詳細を編集できます。以下の操作を実行することもできます。

- 実行時にトップヘッダーを展開および折りたたみ
- 個々のペンの表示と非表示の切り替え
- ペンの追加
- ペンの削除

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



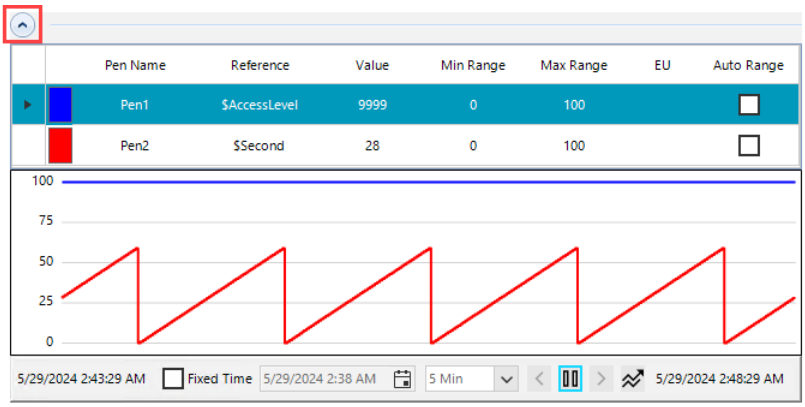
実行時にトップヘッダーを展開および折りたたみ

ペンの数が多いと、トレンドチャートの領域が小さくなり、トレンドの読み取りと分析が困難になります。［上部ヘッダーを展開または折りたたみ］オプションを使用すると、上部ヘッダーを折りたたんだり、チャート領域を大きくしたりして、トレンドを簡単に分析できます。また、必要に応じて、上部ヘッダーを展開してペン設定を表示できます。

上ヘッダーを折りたたみ

1. 実行時に、左上隅にある上向き矢印を選択します。

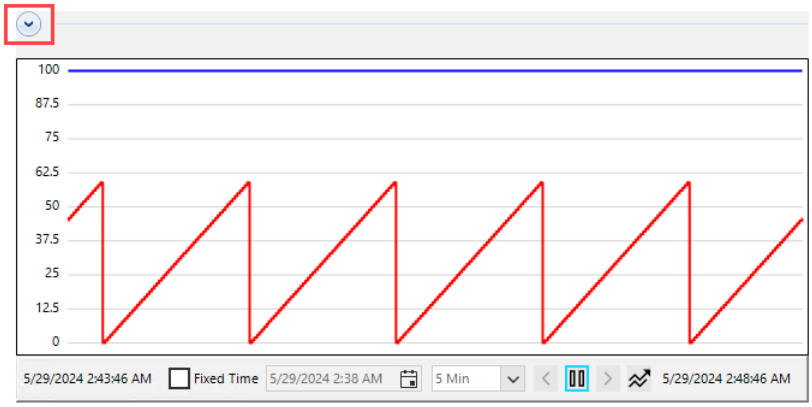
   Top ヘッダーが折りたたまれます。



[上ヘッダー] を展開します。

1. 実行時に、左上隅にある下向き矢印を選択します。

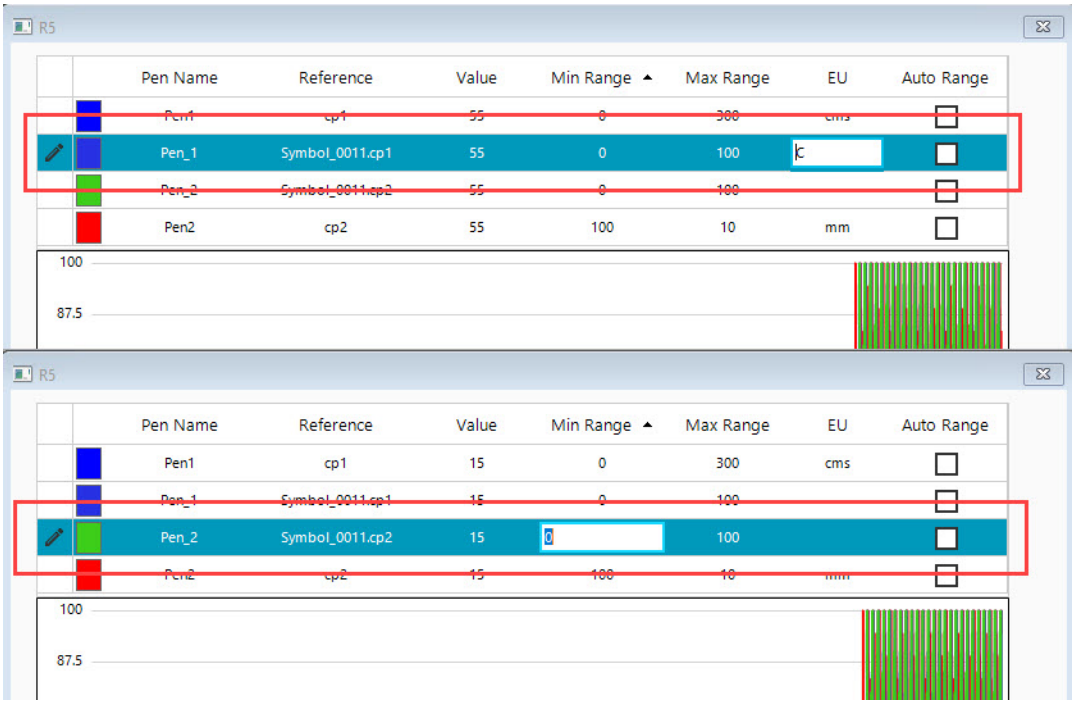
   上部ヘッダーが展開されます。



© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ランタイム時のペンの詳細の編集

1. フィールドをダブルクリックして編集します。
2. 値を更新します。
3. 行の外側をクリックすると新しい値が保存されます。



個々のペンの表示と非表示の切り替え

1. 行を右クリックします。
2. コンテキスト メニューから［**非表示**］を選択します。
   上部ヘッダー領域でペンが非表示になり、トレンド領域に描画されなくなります。
3. ペンを表示するには、上部ヘッダー領域を右クリックします。
4. コンテキスト メニューから［**表示**］を選択して、ペンの名前を選択します。
   上部ヘッダーとトレンド領域にペンが表示されます。

ペンの追加

1. 上部ヘッダー領域を右クリックし、［**追加**］を選択します。
   新しい行が上部ヘッダー領域に追加されます。
2. 名前や参照などのペンの詳細を入力します。
3. Enter キーを押します。
   データが使用可能な場合、トレンド領域でペンの描画が開始されます。

ペンの削除

- 上部ヘッダー領域を右クリックし、［**削除**］を選択します。Delete キーを押すこともできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マルチ ペン トレンド コントロールからペンを削除されます。

上部ヘッダー領域のソート

1. ペンの詳細をソートするヘッダー行の見出し名をクリックします。

   ペン名、参照、EU の見出し名の場合、

   その他のすべての見出し名は、小さいものから大きいものの順にソートされます。

   色以外のすべての見出しを上部ヘッダー領域のソートに使用できます。

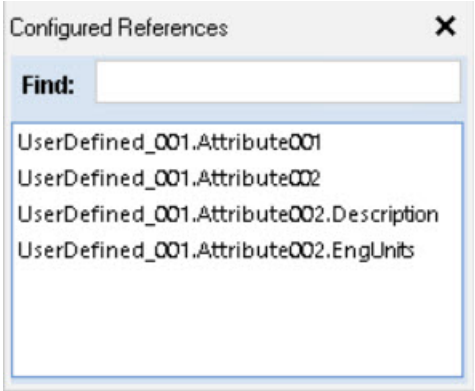
2. 見出し名を再度クリックすると、ソートの方向が変更されます。
3. 見出し名をもう一度クリックすると、ソート前の状態に戻ります。

［設定済み参照］ダイアログ ボックス

［設定済み参照］ダイアログ ボックスでは、グラフィックまたはウィンドウで設定されている属性を表示できます。エレメントを選択せずに開いた領域を右クリックすると、スクリプトレベルで使用可能な属性が表示されます。エレメントを右クリックすると、そのエレメントに対応する属性が表示されます。このダイアログ ボックスは、サイズを変更することや、ウィンドウ上の位置を変更することができます。ダイアログ ボックスの位置と寸法は WindowViewer を再起動しても維持されます。

［設定済み参照］ダイアログ ボックスを表示するには

- ウィンドウを右クリックします。［**設定済み参照**］ダイアログ ボックスが表示されます。



属性を見つけるには

- ［検索］フィールドに属性名の最初の数文字を入力して Enter キーを押します。入力したテキストに一致する属性のリストが表示されます。

属性名をコピーするには

- ［設定済み参照］ダイアログ ボックスで属性を右クリックして［**コピー**］をクリックします。

  属性の名前がオペレーティング システムのクリップボードにコピーされます。

ドラッグ アンド ドロップによるペンの追加

参照を選択してドラッグ アンド ドロップすることでランタイム時にトレンド ペンを追加できます。

1. グラフィックまたは要素を右クリックします。［設定済み参照］ダイアログ ボックスが表示され、そのグラフィックで使用できるすべての属性が一覧表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. 参照を選択して、マルチ ペン トレンドの上部ヘッダーにドラッグ アンド ドロップします。新しいペンが作成され、選択したペンのデータがトレンド領域に描画されます。

トレンド領域のスケール

トレンド領域のスケールは、選択したトレンド ペンの値に応じてランタイム時に計算および表示されます。表示されるグリッド ラインの数は、トレンド領域の高さに応じて異なります。マルチ ペン トレンドの高さを変更すると、その変更にしたがってスケールも変更されます。高さに応じて、2、4、または 8 の間隔を使用できます。

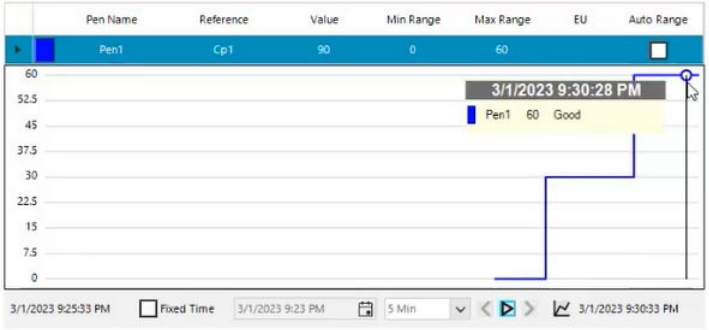
最小および最大範囲の変更

［自動範囲］チェック ボックスがオンにされていない場合、最小と最大の範囲を変更できます。トレンド領域でスケールが更新され、新しいスケールでトレンドがプロットされます。

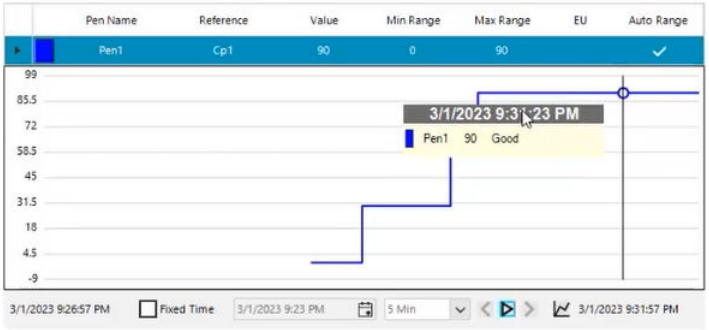
- 選択した行に対して、上部ヘッダーの最小範囲および最大範囲フィールドをクリックして新しい値を入力します。

トレンド領域のペンのスケールが更新されます。

タグ値が最小範囲と最大範囲を超えている場合、トレンド線は最小範囲と最大範囲の値に切り詰められます。カーソルの読み出し時には、常に実際の値が表示されます。下の画像では、最大範囲が 46 なので、タグ値が 58 であってもトレンドは 46 に切り詰められ、カーソル読み出しも 58 を示します。



切り詰めなしでトレンド線を表示し、カーソル読み出しで実際の値を表示するには、［自動範囲］チェック ボックスをオンにします。ペンに対して指定されている最大範囲と最小範囲の値が等しい場合、自動範囲の条件と見なされ、ペンに対して指定されている値は無視されます。下の画像では、［自動範囲］チェック ボックスがオンになっているので、トレンド線は切り詰められず、カーソル読み出しでは実際の値が表示されています。



自動範囲の詳細については、「自動範囲の有効化」トピックを参照してください。

自動範囲の有効化

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

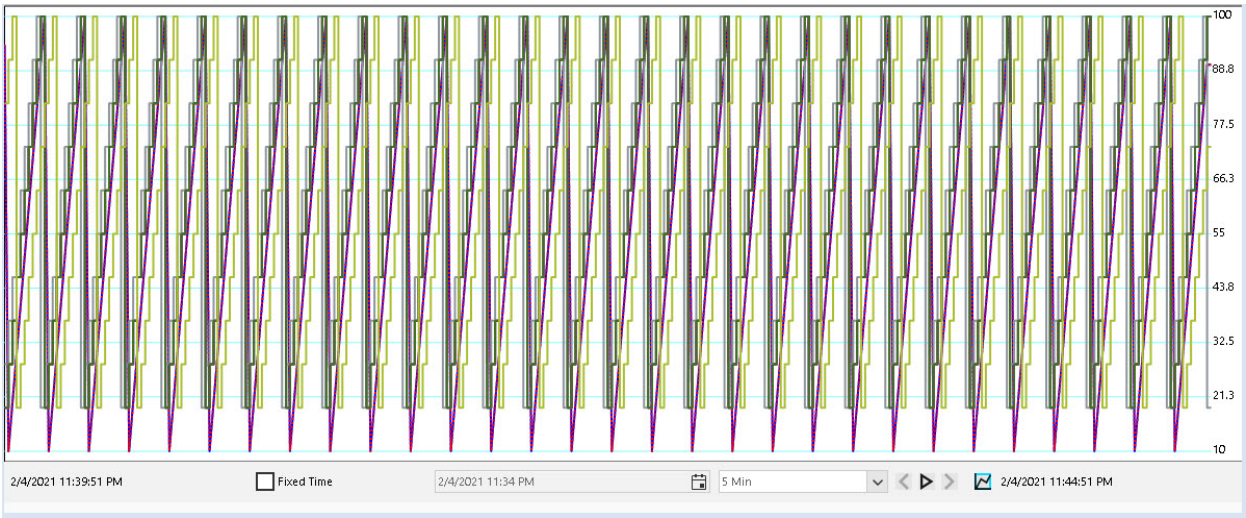
［自動範囲］を選択すると、トレンド ペン値に基づいてトレンド領域のスケールが動的に更新されます。ペンに対して指定された最小および最大の範囲は無視されます。

- 目的のペンの［**自動範囲**］チェック ボックスをオンにします。

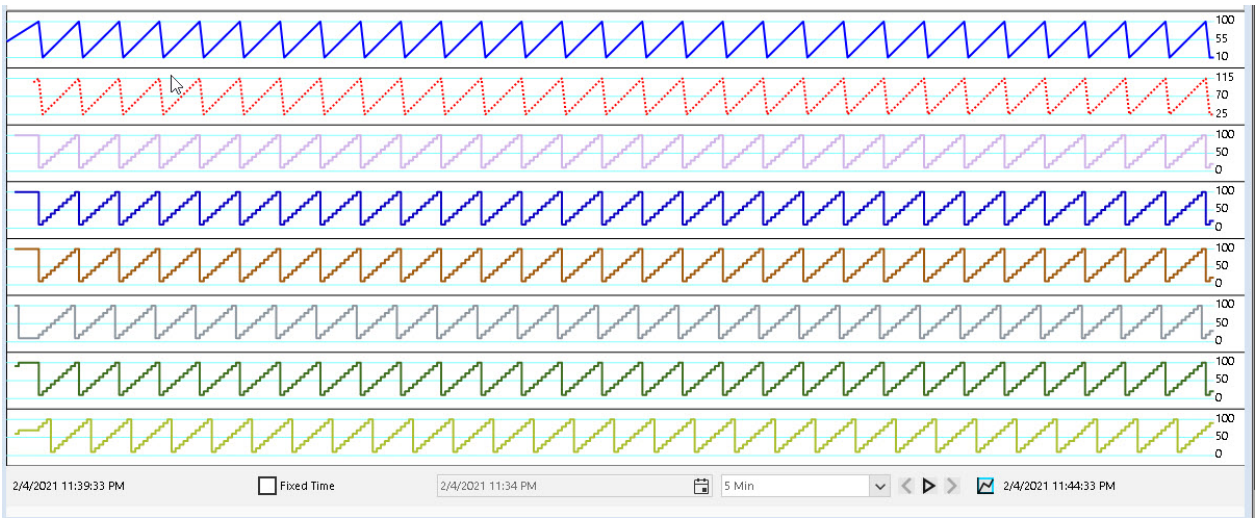
ペンの行を選択すると、トレンド領域のペンのスケールが更新されます。

トレンドのスタックおよびスタック解除

ランタイム時にスタック/スタック解除オプションを使用して個々のペンを表示できます。デフォルトでは、複数のペンが一緒に表示されます。



個々のペンを表示するには、ツールバーの ［**トレンドのスタック**］ボタンをクリックします。ペンが個々に重なった状態で表示されます。 を再度選択すると、スタックが解除されたビューに戻ります。

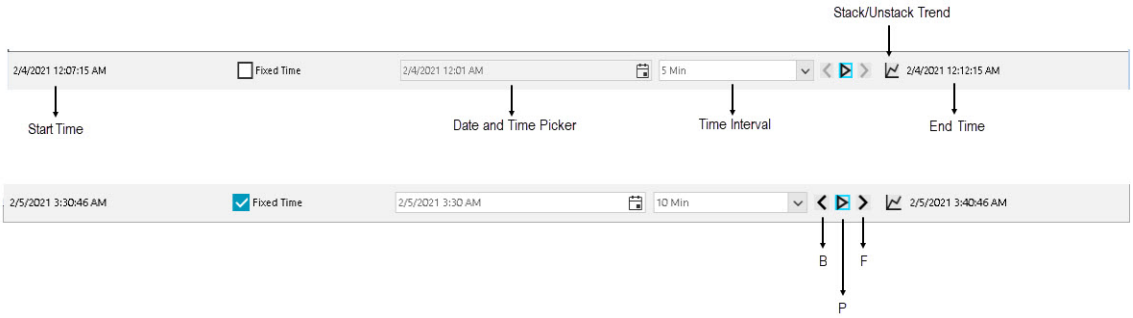


注意：トレンド領域が小さい場合、ランタイム時にペンを追加すると個々のスケールが重なって表示されします。いくつかのペンを非表示にして結果を明瞭に表示することができます。

トレンドデータを再生する

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ツールバーの再生オプションを使用すると、指定した開始時間と終了時間の間のトレンド データを再生できます。



移動モード

デフォルトでは、トレンド領域には、［時間間隔］フィールドで指定した値の現在のトレンド データが表示されます。開始時間と終了時間の関係は、「終了時間 = 開始時間 + 時間間隔」で表されます。移動モードでは、終了時間は常に現在の時間で、入力に応じて変更されます。固定モードでは、開始時間と終了時間は、再生コントロールを使用するまで固定されます。

時間間隔は事前定義済みの値のリストから選択できます。トレンド領域のトレンド ペン データが更新され、その期間のデータだけが表示されます。

固定モード

1. 過去または将来の特定の期間のトレンド ペンを表示するには。**［固定時間］**チェック ボックスをオンにします。
2. **［時間間隔］**を選択します。

   トレンド領域で、開始時間と終了時間の間のペン データに選択した時間間隔が反映されます。
3. **［トレンドを後方向に移動］**（B）、**［トレンドの一時停止/トレンドの再生］**（P）、および **［トレンドを前方向に移動］**（F）を使用します。

   一時停止ボタンをクリックすると、選択した時間間隔のトレンド領域が固定されます。

< および > をクリックすると、トレンド領域が開始時間と終了時間の間の時間ブロック内で時間間隔の半分だけ移動します。

時間間隔が 10 分の場合の例を以下に示します。

Mode	開始時刻	終了時刻
ライブ	2/8/2021 3:15:16 AM	2021 年 2 月 8 日 3:25:16
固定:後方向に移動	2021 年 2 月 8 日 3:10:16	2021 年 2 月 8 日 3:20:16
固定:前に進む	2021 年 2 月 8 日 3:20:16	2021 年 2 月 8 日 3:30:16

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

日付時刻ピッカーの使用

日付時刻ピッカーを使用して特定の開始日時を選択できます。この機能は、［固定時間］チェック ボックスがオンにされている場合にのみ使用できます。

選択した日時が開始時間に表示されます。終了時間には、選択した日時に時間間隔を加えた値が表示されます。

選択した日時に時間間隔を加えた期間のデータがトレンド領域に表示されます。

SmartSymbol について

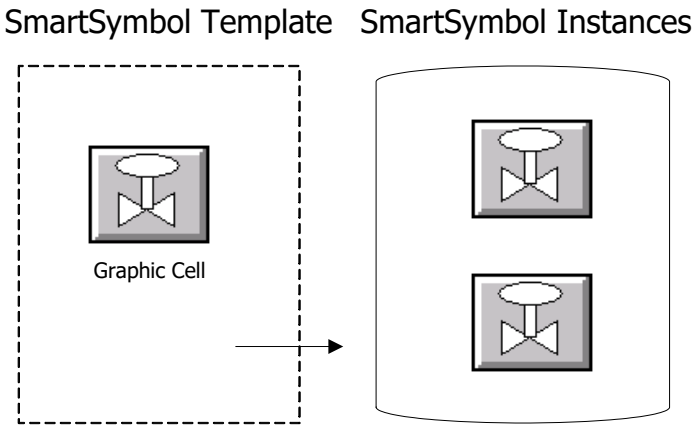
SmartSymbol は、再使用できるテンプレートに変換された InTouch グラフィックです。SmartSymbol テンプレートに対して行った変更内容は、アプリケーション全体の SmartSymbol のすべてのインスタンスに反映されます。このため、アプリケーションで繰り返し使用されるグラフィックの作成、変更、検証、および再検証を何度も行う手間が省けます。

SmartSymbol は、InTouch タグおよび Application Server 属性を参照します。タグを参照しない SmartSymbol、または InTouch タグのみを参照する SmartSymbol は InTouch SmartSymbol と呼ばれます。

1 つまたは複数のオートメーション オブジェクトまたはオートメーション オブジェクト インスタンスの属性を参照する SmartSymbol は産業用グラフィック SmartSymbol と呼ばれます。産業用グラフィック SmartSymbol は InTouch タグを参照することもできます。

InTouch SmartSymbol および産業用グラフィック SmartSymbol は、産業用グラフィックではありません。産業用グラフィックの設計および管理には System Platform 統合開発環境（IDE）を使用します。

InTouch バージョン 10 以降を使用して新しいグラフィックを作成している場合、アプリケーションには SmartSymbol より産業用グラフィック の方が適している場合があります。



SmartSymbol マネージャおよびライブラリ

SmartSymbol ライブラリには、InTouch アプリケーション用の SmartSymbol が含まれています。SmartSymbol マネージャは、SmartSymbol ライブラリの内容をインポート、エクスポート、および整理するために使用します。

SmartSymbol は、ユーザーの InTouch アプリケーションのアプリケーション フォルダにある \Symbols フォルダに保存されています。

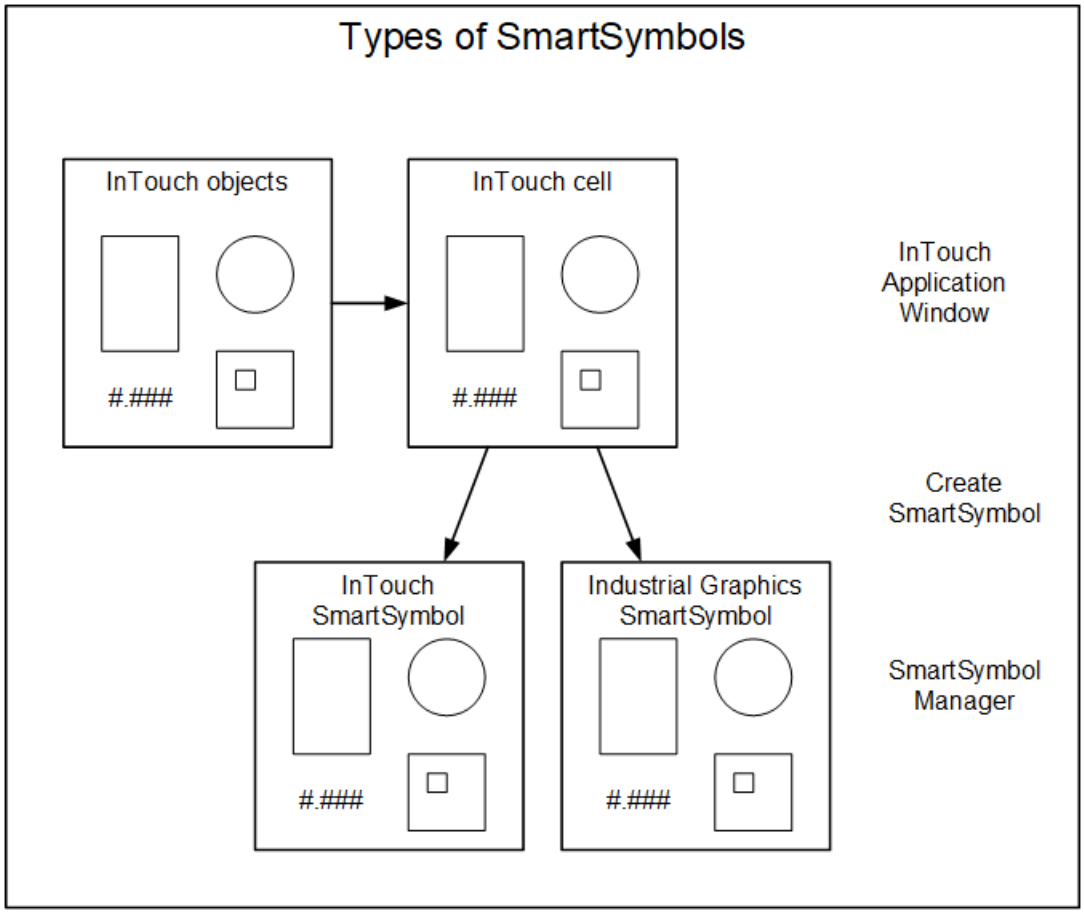
ライブラリの SmartSymbol についての参照情報は、関連する XML ファイルに保存されます。XML ファイルは編集しないでください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch SmartSymbol および産業用グラフィック SmartSymbol

InTouch SmartSymbol は、産業用グラフィック SmartSymbol と同じものではありません。産業用グラフィックは産業用グラフィック SmartSymbol に取って代わるものであり、System Platform IDE を使用して開発されています。InTouch SmartSymbol は、 タグ変数のデータを参照します。産業用グラフィック SmartSymbol は、Galaxy オブジェクトまたはテンプレートの属性を参照します。

次の図は、SmartSymbol の種類を示しています。

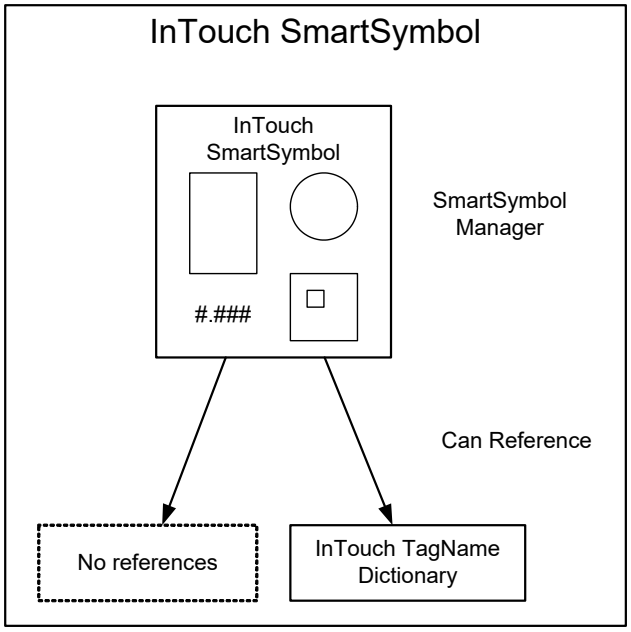


InTouch SmartSymbol

InTouch SmartSymbol は、SmartSymbol マネージャの InTouch シンボル フォルダに保存されています。

ローカルおよびリモートの InTouch タグ変数への参照を使用して、InTouch SmartSymbol でグラフィカルな要素に対するアニメーションを設定できます。詳細については、「SmartSymbol テンプレートとインスタンスの作成」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



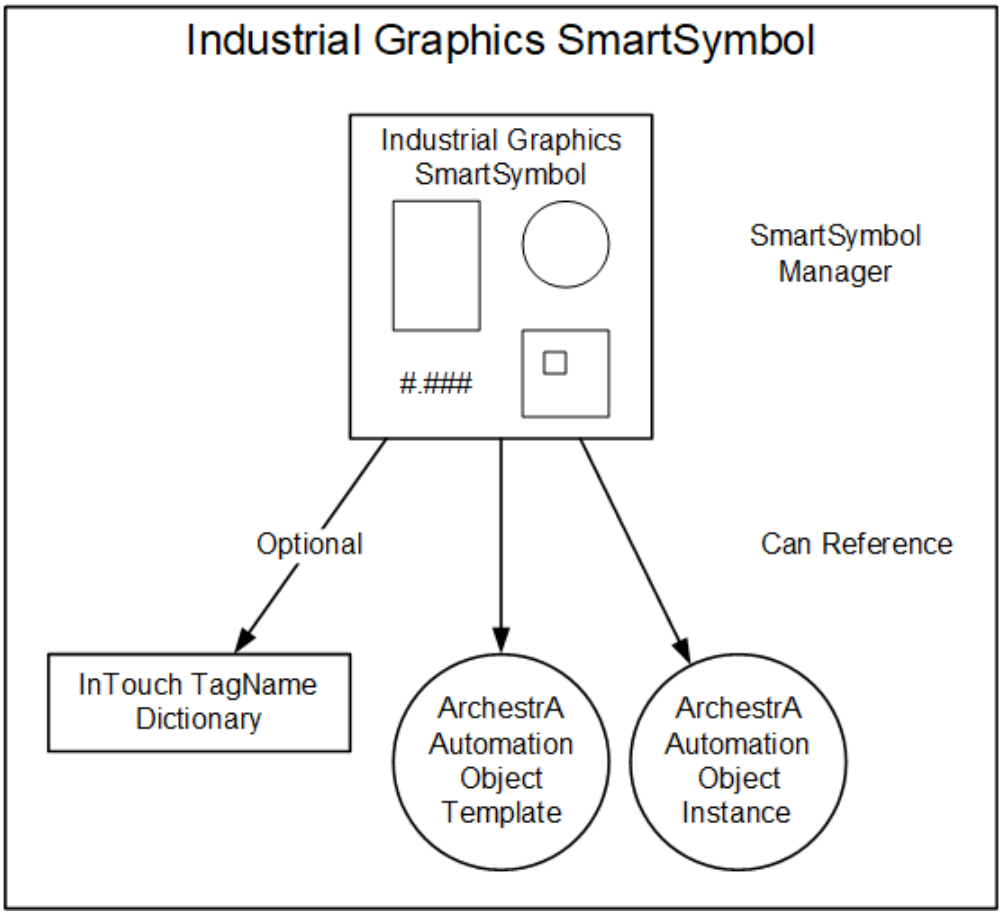
産業用グラフィック SmartSymbol

産業用グラフィック SmartSymbol は、SmartSymbol マネージャの産業用グラフィック フォルダに保存されています。

注: フォルダの名前は産業用グラフィック (Industrial Graphics) ですが、フォルダには産業用グラフィックではなく、産業用グラフィック SmartSymbol が含まれています。

産業用グラフィック SmartSymbol を作成するには、1 つまたは複数の Galaxy オブジェクト テンプレートを選択して、さまざまなグラフィカル要素に対するアニメーション参照を定義します。詳細については、「SmartSymbol テンプレートとインスタンスの作成」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



SmartSymbol の制限

次に、SmartSymbol の既知の制限について説明します。

- SmartSymbol にはトレンド オブジェクトを含めることはできません。トレンド オブジェクト（履歴またはリアルタイム）が含まれている SmartSymbol を作成しようとすると、エラー メッセージが表示されます。
- SmartSymbol には、分散アラーム表示コントロール、Windows コントロール、AlarmViewer などの InTouch ActiveX コントロール、または InTouch アプリケーションで設定されるサードパーティの ActiveX コントロールを含めることはできません。
- Application Server バージョン 1.5 で作成された Galaxy でインスタンスを参照することはできません。インスタンスを参照するには、Application Server バージョン 2.0 以降をインストールしてください。
- SPC チャート ウィザードを使用した SmartSymbol の生成はサポートされていません。
- SmartSymbol は、ローカル スクリプト変数を参照できません。
- **属性ブラウザ**には、生成されたオブジェクト インスタンスは表示されません。この問題に対処するには、生成されたテンプレートを SmartSymbol マネージャで作成するか、またはブラウザでカスタム フィルタを作成します。

SmartSymbol テンプレートとインスタンスの作成

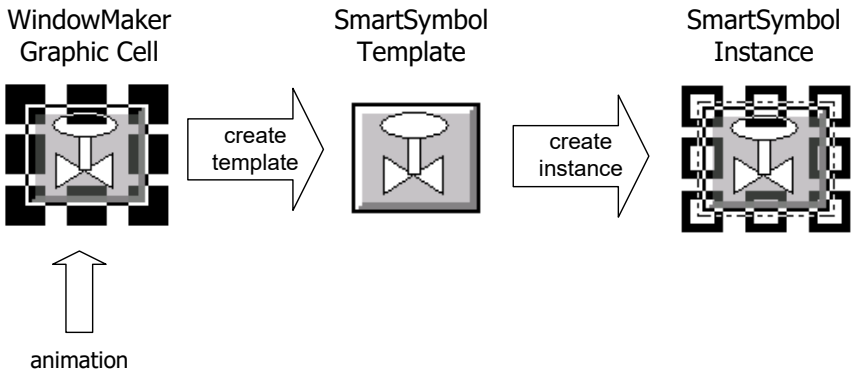
WindowMaker を使用すると、SmartSymbol テンプレートおよびインスタンスを作成できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SmartSymbol テンプレートは、WindowMaker で 1 つまたは複数のグラフィックを描画し、それらを 1 つのセルに結合してから、そのセルを SmartSymbol に変換することによって作成します。テンプレートを InTouch タグまたは Application Server オブジェクトに接続またはリンクする必要はありません。

SmartSymbol テンプレートを作成した後で、SmartSymbol のインスタンスをアプリケーション ウィンドウで作成できます。

また、既存の産業用グラフィック SmartSymbol インスタンスから Application Server オブジェクト インスタンスを作成して、InTouch WindowMaker と System Platform IDE の切り替えを回避することもできます。



InTouch データで使用する SmartSymbol テンプレートの作成

InTouch SmartSymbol は、グラフィカル要素、アニメーション、および InTouch タグへの参照を含めることのできるセルから作成されます。

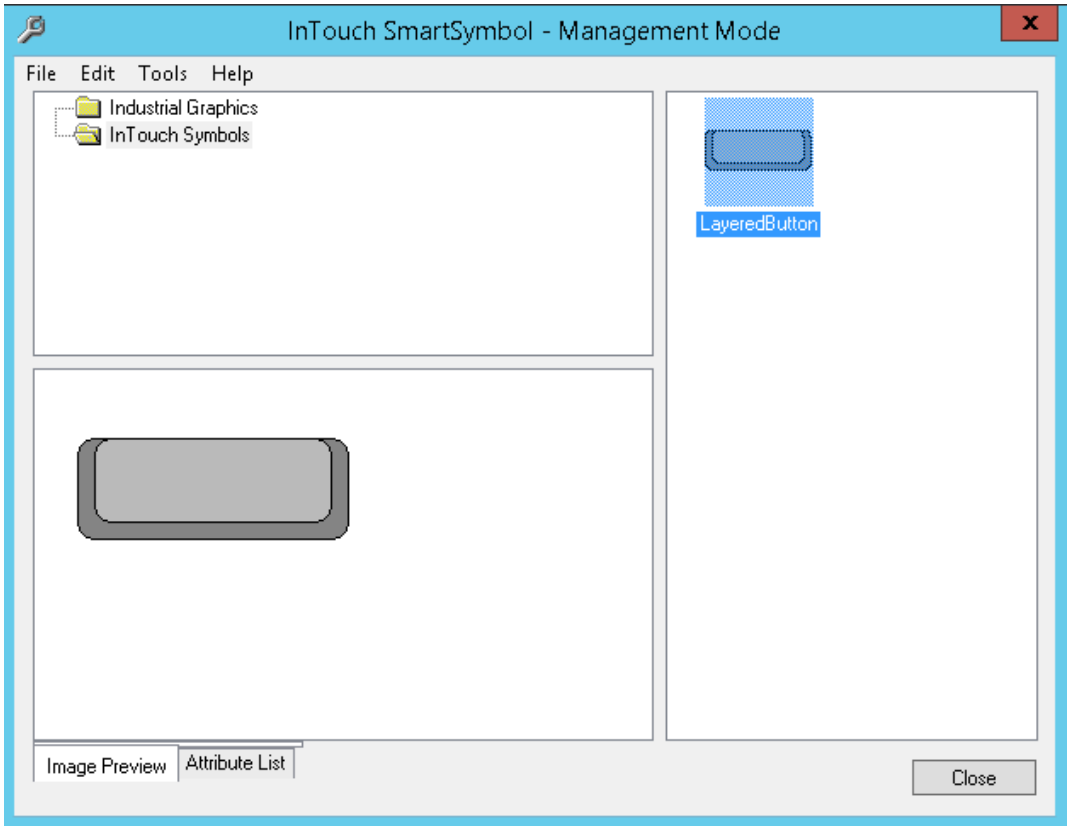
以下の手順では、新しいウィンドウ、グラフィック、およびセルの作成が指定されていますが、既存のウィンドウ、グラフィック、またはセルを使用して SmartSymbol を作成することもできます。

新しい InTouch SmartSymbol テンプレートを作成するには

1. WindowMaker で新しいウィンドウを作成します。
2. グラフィック描画ツールまたはウィザードを使用して、SmartSymbol に含める 1 つまたは一組のグラフィックを作成します。
3. グラフィックのアニメーション リンクを設定します。
   詳細については、「オブジェクトのアニメーション化」を参照してください。
4. SmartSymbol テンプレートに含めるオブジェクトをすべて選択します。
5. ［**アニメーション**］メニューの［**セル**］グループで［**セルの作成**］を選択します。
6. 先ほど作成したセルを選択します。
7. ［**描画**］メニューの［**SmartSymbol**］グループで［**生成**］を選択します。

   ［**InTouch SmartSymbol - 管理モード**］ダイアログ ボックスが表示され、新しい SmartSymbol が強調表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



デフォルトでは、新しい SmartSymbol は InTouch シンボルの最上位レベルのフォルダに配置されます。デフォルトの名前が、シンボルに自動的に割り当てられます（例: 新規シンボル 1）。

8. 新しい名前を入力するか、デフォルトの名前を受け入れます。SmartSymbol の名前はいつでも変更できます。SmartSymbol の名前の変更の詳細については、「SmartSymbol テンプレートの名前の変更」を参照してください。

9. **[閉じる]**を選択します。グラフィック セルを新しい SmartSymbol で置換するよう求めるメッセージが表示されます。**[はい]** または **[いいえ] を選択します。はい**を選択すると、グラフィック セルは SmartSymbol に置換されます。［**いいえ**］を選択すると、グラフィック セルは変更されません。いずれの場合でも、新しい SmartSymbol は SmartSymbol ライブラリに保存され、今後使用も使用できます。

産業用グラフィック SmartSymbol テンプレートの作成

オートメーション オブジェクト テンプレートまたはインスタンスへの参照を少なくとも 1 つ含む InTouch グラフィック セルから SmartSymbol を作成すると、その SmartSymbol は産業用グラフィック SmartSymbol になります。

オートメーション テンプレートへの参照には、"$" 記号が含まれます。

オブジェクト テンプレートまたはインスタンスに関連付けられている SmartSymbol テンプレートを生成できます。

InTouch ウィンドウで SmartSymbol インスタンスを作成すると、それが参照するオブジェクト テンプレートをインスタンス化できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

新しい産業用グラフィック SmartSymbol テンプレートを生成するには:

1. WindowMaker で新しいウィンドウを作成します。
2. 描画ツールまたはウィザードを使用して、SmartSymbol に作成するグラフィックを作成します。
3. グラフィックのアニメーション リンクを設定します。

   Galaxy ソース名の設定の手順については、「InTouch から Application Server データへのアクセス」を参照してください。Application Server 属性へのリンクの作成および設定の手順については、「オブジェクトのアニメーション化」を参照してください。
4. セルに含めるグラフィックを選択します。
5. ［**アニメーション**］メニューの［**セル**］グループで［**セルの作成**］を選択します。
6. ［**描画**］メニューの［**SmartSymbol**］グループで［**生成**］を選択します。

   SmartSymbol マネージャによって Galaxy データの新しい SmartSymbol が作成されます。
7. 新しい名前を入力するか、デフォルトの名前を受け入れて後で変更します。
8. **[閉じる]**を選択します。グラフィック セルを新しい SmartSymbol で置換するよう求めるメッセージが表示されます。**[はい]** または **[いいえ] を選択します**。［**はい**］を選択すると、グラフィック セルは SmartSymbol に置換されます。［**いいえ**］を選択すると、グラフィック セルは変更されません。いずれの場合でも、新しい SmartSymbol は SmartSymbol ライブラリに保存され、今後使用も使用できます。

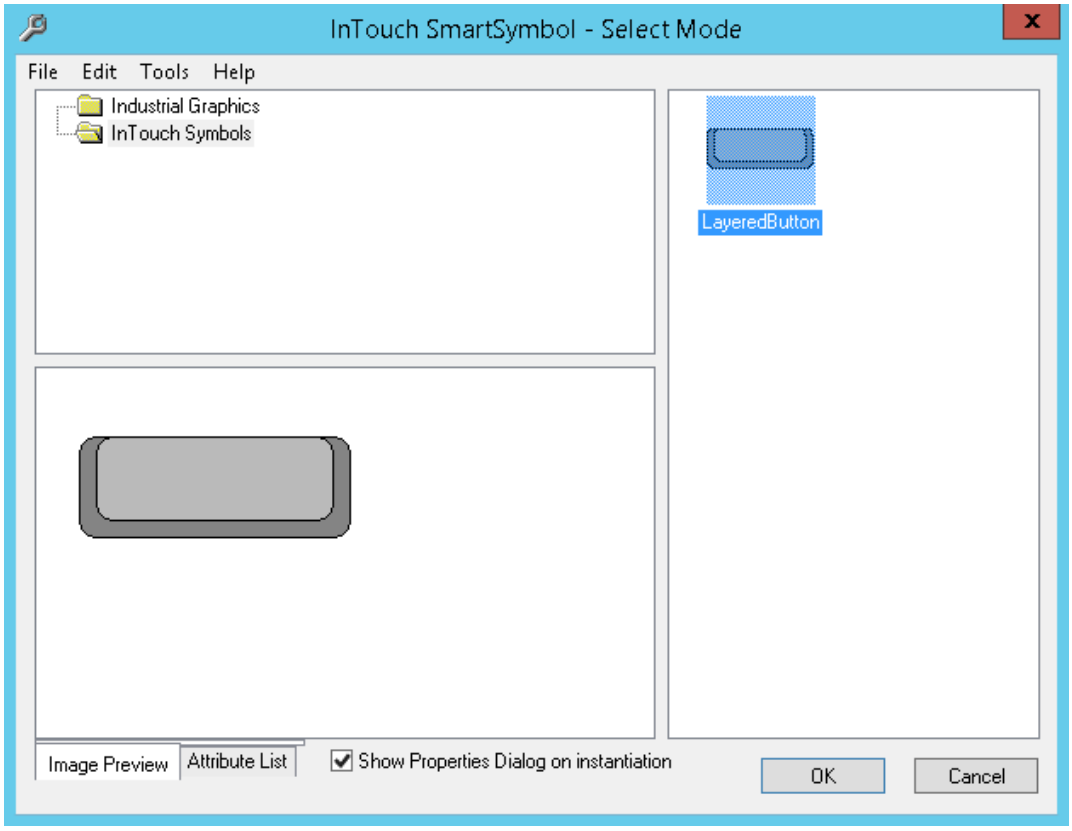
InTouch SmartSymbol テンプレートからの SmartSymbol インスタンスの作成

1 つの SmartSymbol テンプレートから複数の SmartSymbol インスタンスを作成できます。各インスタンスには、すべての参照とテキスト ラベルが継承されます。InTouch ウィンドウにインスタンスを配置する前に、参照とテキスト ラベルを変更できます。

InTouch SmartSymbol テンプレートから SmartSymbol を作成するには

1. WindowMaker を開き、SmartSymbol を使用するウィンドウを開きます。
2. ［**ホーム**］メニューの［**挿入**］グループで［**SmartSymbol**］アイコンを選択します。
3. WindowMaker ウィンドウで、シンボルを配置する場所を選択します。［**InTouch SmartSymbol - 選択モード**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

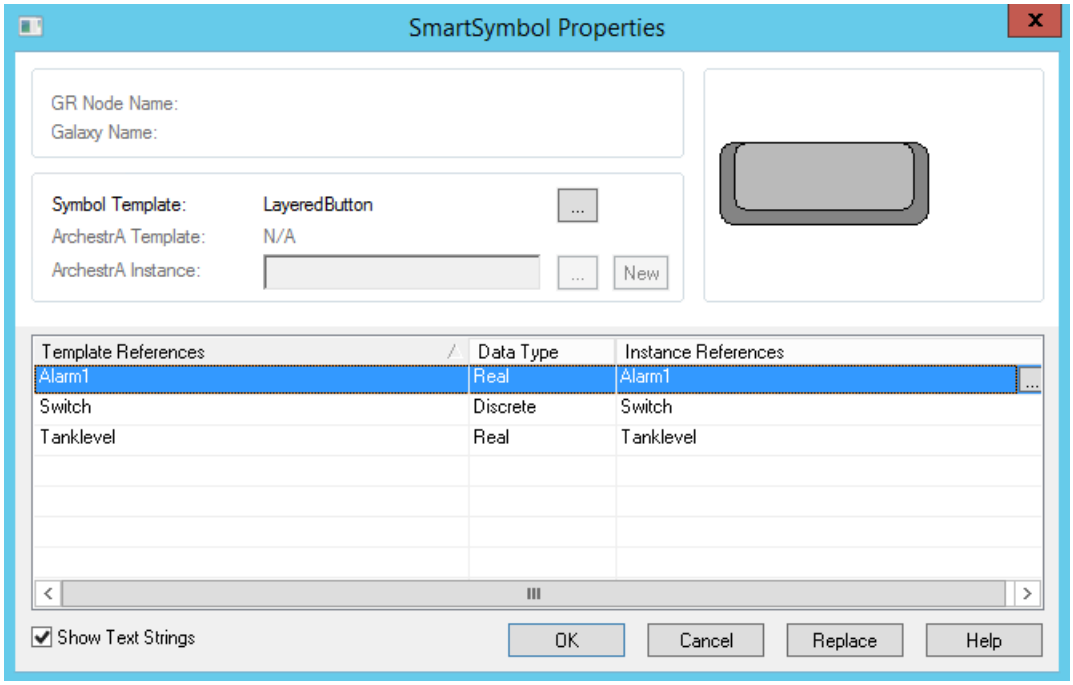


注意：デフォルトでは、［**インスタンス化の際に［プロパティ］ダイアログを表示する**］チェック ボックスはオンになっています。新しい SmartSymbol インスタンスの参照またはテキスト ラベルを変更しない場合は、チェック ボックスをオフにします。

4. ［**InTouch シンボル**］フォルダで、SmartSymbol をダブルクリックします。アプリケーション ウィンドウに新しいシンボルが表示されます。

   前の手順で［**インスタンス化**の際に［プロパティ］ダイアログを表示する］チェック ボックスをオンにした場合、［**SmartSymbol プロパティ**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. ［**インスタンス参照**］カラムで、［**参照**］ボタンを選択します。［**タグの選択**］または［**タグ名ディクショナリ**］ダイアログ ボックスが表示されます。
6. タグを選択して、SmartSymbol にリンクします。ウィンドウを閉じると、［**SmartSymbol プロパティ**］ダイアログ ボックスが表示されます。

   注意：まだ定義されていない新しいタグ変数を入力すると、［**未登録のタグ変数**］ダイアログ ボックスが表示されます。［**OK**］を選択して、タグ変数ディクショナリから新しいタグ変数を定義します。

7. [OK] を選択します。 アプリケーション ウィンドウに新しいシンボルが表示されます。

ArchestrA SmartSymbol テンプレートからの SmartSymbol インスタンスの作成

複数の ArchestrA SmartSymbol インスタンスは、1 つの ArchestrA SmartSymbol テンプレートから作成できます。

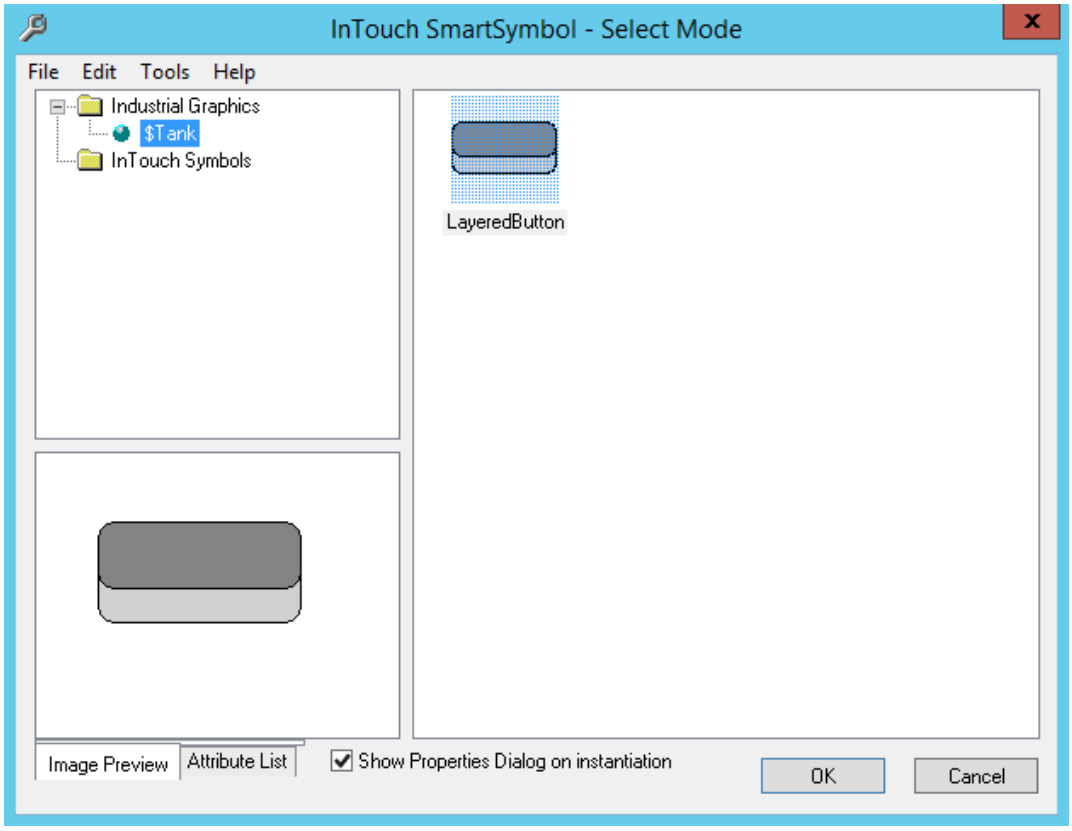
ArchestrA SmartSymbol テンプレートから SmartSymbol を作成するには

1. ［**SmartSymbol**］アイコンを選択し、WindowMaker ウィンドウでシンボルを配置する場所を選択します。［**InTouch SmartSymbol - 選択モード**］ダイアログ ボックスが表示されます。

注意：デフォルトでは、［**インスタンス化の際に［プロパティ］ダイアログを表示する**］チェック ボックスはオンになっています。

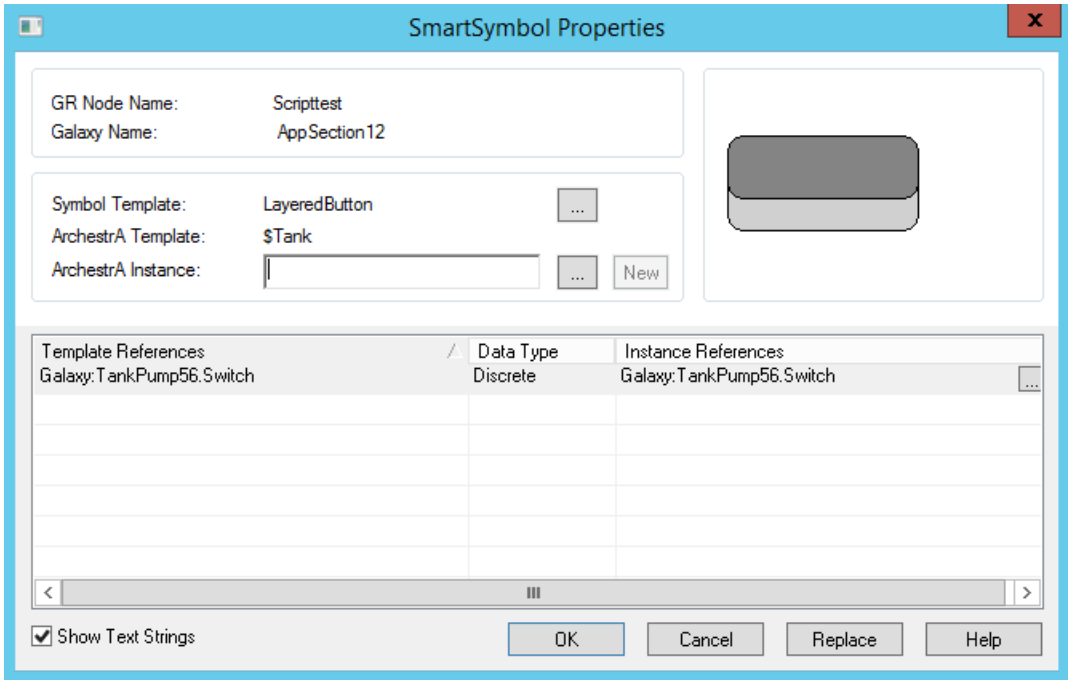
1. **産業用グラフィック**を選択します。右ペインに産業用グラフィックが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. SmartSymbol を選択して、[**OK**] を選択します。アプリケーション ウィンドウに新しいシンボルが表示されます。

   前の手順で [**インスタンス化**の際に [プロパティ] ダイアログを表示する] チェック ボックスをオンにした場合、[**SmartSymbol プロパティ**] ダイアログ ボックスが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［**ArchestrA インスタンス**］テキスト ボックスでは、以下のいずれかを実行できます。
   - ArchestrA オブジェクトを参照し、選択します。
   - 関連付けられたオブジェクト テンプレートから派生した新しい ArchestrA オブジェクト インスタンスを作成します。インスタンスの名前を入力して、［**新規**］を選択します。

   ［**インスタンス参照**］カラムにインスタンス属性参照が表示されます。

   Galaxy にまだ接続されていなかった場合、ノード名と Galaxy 名を入力するよう求めるダイアログ ボックスが表示されます。

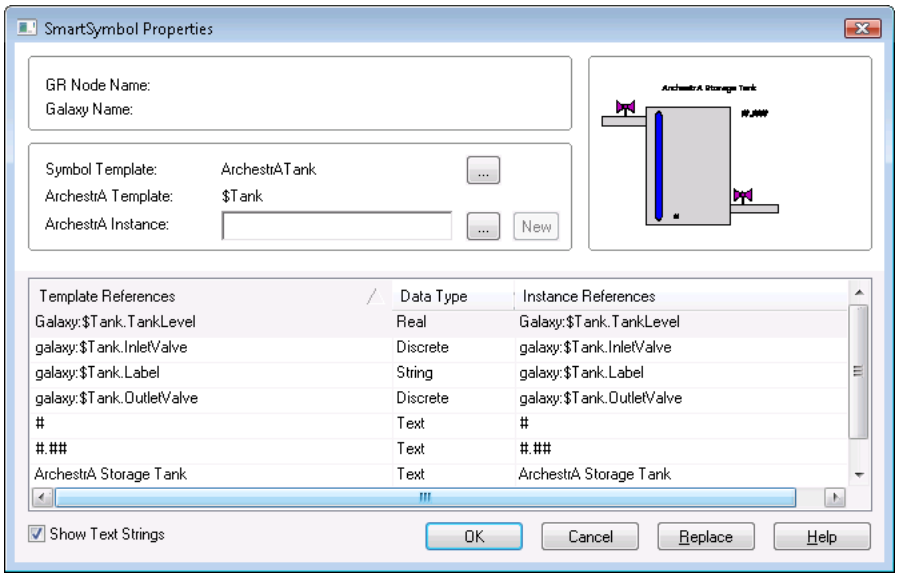
4. 必要に応じて、［**インスタンス参照**］カラムで参照を変更します。正しい構文で手動で参照を入力するか、［**参照**］ボタンを選択して、**属性ブラウザ**を使用します。
5. **[OK]** を選択します。ウィンドウに新しいシンボルが表示されます。

ArchestrA SmartSymbol インスタンスからの ArchestrA オブジェクト インスタンスの作成

既存の ArchestrA SmartSymbol インスタンスから新しい ArchestrA オブジェクト インスタンスを作成できます。この操作を行うことによって、WindowMaker と IDE を切り替える必要はありません。

新しい ArchestrA オブジェクト インスタンスを作成するには

1. WindowMaker で、SmartSymbol インスタンスが配置されているウィンドウを開きます。
2. アプリケーション ウィンドウで、SmartSymbol インスタンスをダブル クリックします。
   ［**SmartSymbol プロパティ**］ダイアログ ボックスが表示されます。

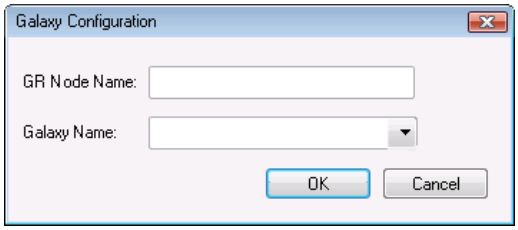


3. ［**ArchestrA インスタンス**］ボックスで、新しいオートメーション オブジェクトの有効な名前を入力します。

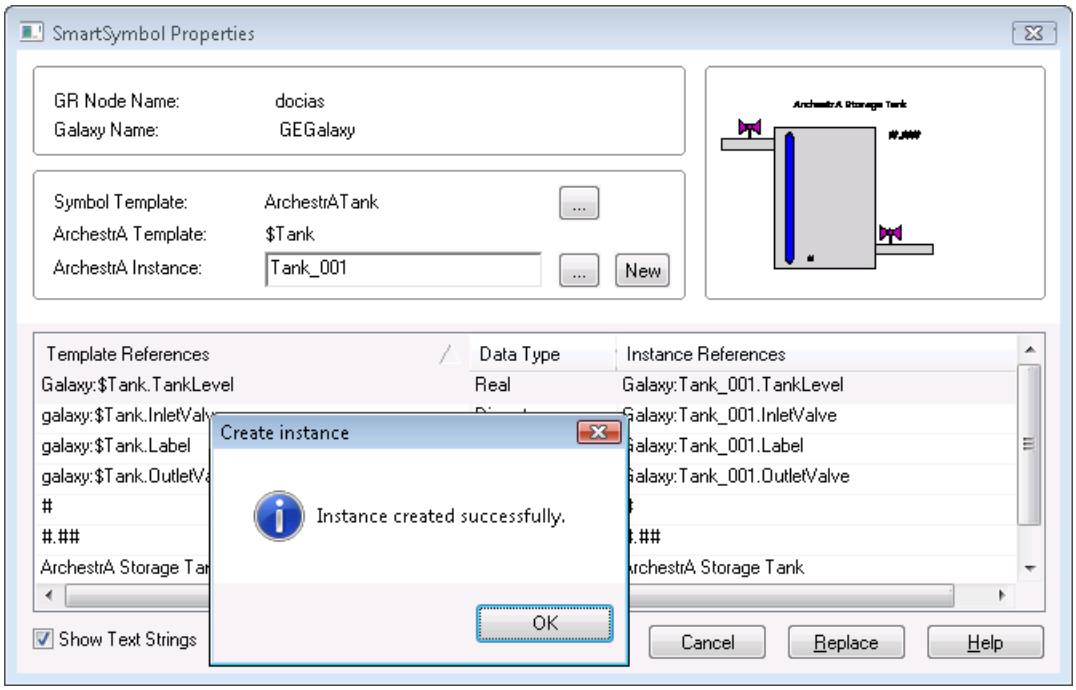
   手記： オブジェクトを初めて指定する場合は、ログインを促すメッセージが表示されます。有効なユーザー名、パスワード、およびドメイン名を入力します。Application Server のセキュリティが［なし］以外のモードに設定されている場合、OS ユーザーまたは OS グループ ベースのセキュリティに対してのみドメイン名が必要です。

4. ［**新規作成**］を選択します。新しいオブジェクトを作成する有効な Galaxy を選択するようメッセージが表示されたら、［**OK**］を選択します。［**Galaxy の設定**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. Galaxy を指定します。以下の手順を実行します。
   a. ［**GR ノード名**］ボックスで、Galaxy が稼動しているコンピュータの名前を入力します。
   b. ［**Galaxy 名**］リストで、Galaxy を選択します。
   c. **[OK] を**選択します。ArchestrA オブジェクト インスタンスが作成され、インスタンス参照は新しいインスタンスをポイントします。

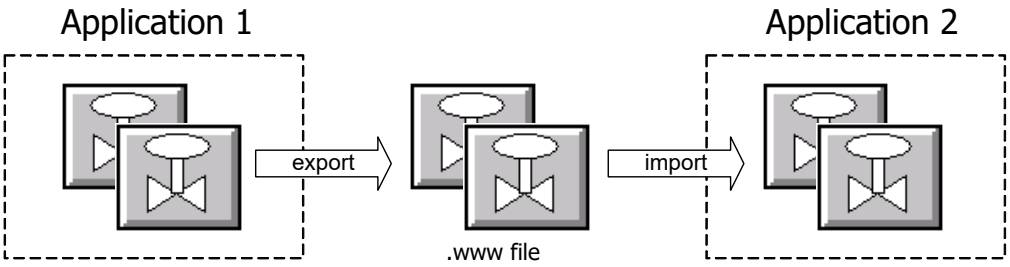


   a. 再び［**OK**］を選択して、［**インスタンスの作成**］ダイアログ ボックスを閉じます。
6. ［**OK**］を選択し、［**SmartSymbol プロパティ**］ダイアログ ボックスを閉じます。アプリケーション ウィンドウに新しい SmartSymbol インスタンスが表示されます。

SmartSymbol の管理

SmartSymbol マネージャを使用すると、複数の InTouch アプリケーション間および異なる物理システム間で SmartSymbol をインポートおよびエクスポートできます。SmartSymbol のエクスポートおよびインポートは、InTouch アプリケーション間で SmartSymbol を移動する最適な手段です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



SmartSymbol と共にウィンドウもインポートすることができ、グラフィックもインポートされますが、テンプレート情報はインポートされません。このため、SmartSymbol のインスタンスは孤立します。詳細については、「SmartSymbol の回復」を参照してください。

SmartSymbol マネージャを使用すると、SmartSymbol テンプレートの名前の変更、複製、削除、および保存を行うことができます。

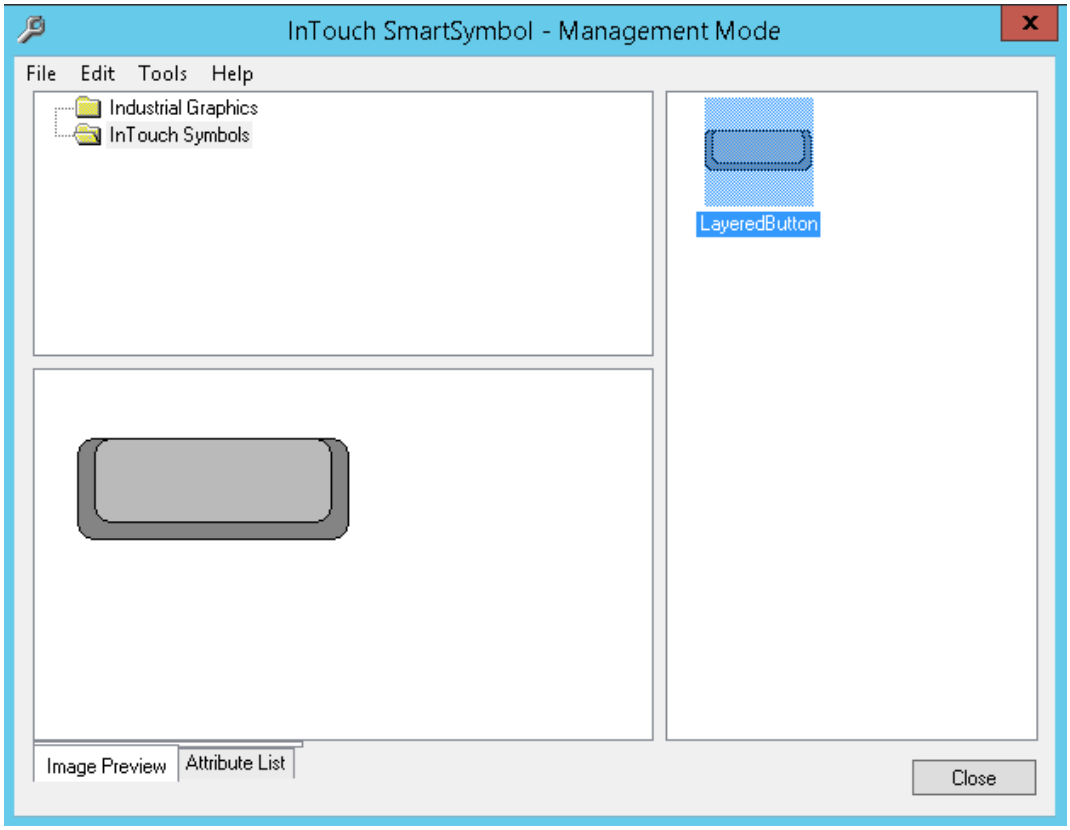
SmartSymbol のインポート

SmartSymbol は、他の InTouch アプリケーションからアプリケーションの SmartSymbol ライブラリにインポートできます。他のアプリケーションからシンボルをインポートすると、そのシンボルを再使用する代わりに、シンボルを再使用できます。

シンボル ライブラリに SmartSymbol をインポートするには:

1. すべてのアプリケーション ウィンドウを閉じます。
2. ［描画］メニューの［**SmartSymbol**］グループで［管理］を選択します。

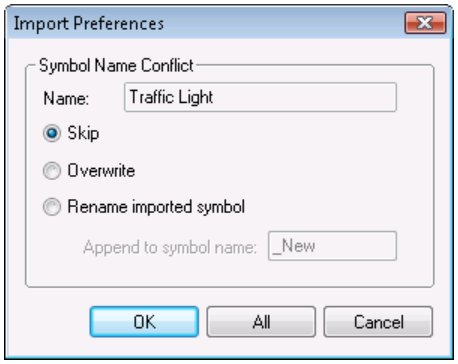
   ［**InTouch SmartSymbol - 管理モード**］ウィンドウが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［**ファイル**］メニューで、［**保存**］を選択します。［**シンボルのインポート**］ダイアログ ボックスが表示されます。
4. インポートする SmartSymbol を含むファイルを参照します。シンボルのエクスポート ファイルはファイル拡張子 .www を持ちます。
5. ファイルを選択し、 **[OK] を選択します**。［SmartSymbol - 管理モード］ウィンドウに、そのファイルの SmartSymbol が表示されます。

   名前が競合する場合、［**インポート環境設定**］ダイアログ ボックスが表示されます。



6. 以下のいずれかまたは複数の手順を実行します。
   - このシンボルのインポートを省略するには、［**スキップ**］を選択します。複数のシンボルをインポートしている場合、残りのシンボルがインポートされます。
   - 既存のシンボルを新しいシンボルで上書きするには、［**上書き**］を選択します。
   - 新しいシンボルを未使用の名前に変更するには、［**インポートするシンボルの名前を変更**］を選択します。［**シンボル名に追加**］ボックスに名前を入力します。
7. 以下のいずれかを実行します。
   - ［**OK**］を選択して、選択したオプションを SmartSymbol に適用します。
   - ［**インポートするシンボルの名前を変更**］を選択した場合は、［**すべて**］を選択して、［**シンボル名に追加**］ボックス内のテキストをパッケージ ファイル内で名前が競合するすべての SmartSymbol に適用します。

   ［**InTouch SmartSymbol - 管理モード**］ウィンドウに、インポートした SmartSymbol が表示されます。

SmartSymbol のエクスポート

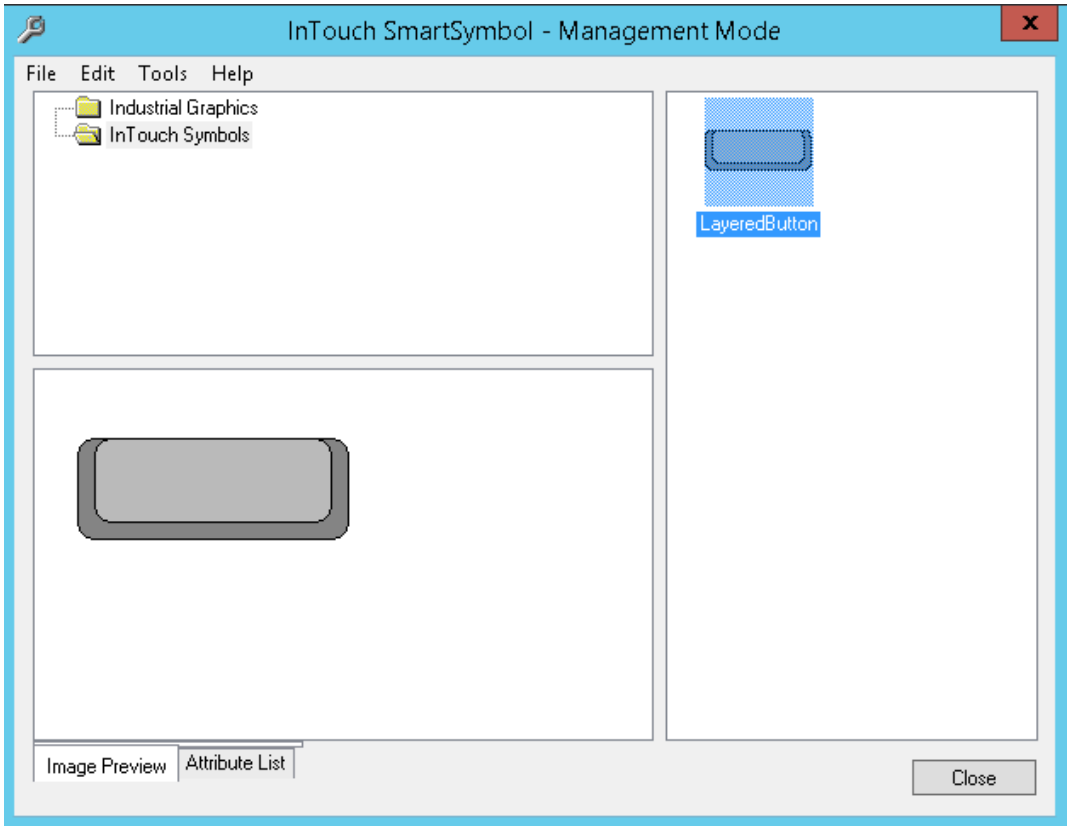
SmartSymbol をアプリケーションの SmartSymbol ライブラリで作成またはインポートすると、1 つまたは複数の SmartSymbol テンプレートを他の InTouch アプリケーションにエクスポートできます。SmartSymbol テンプレートのエクスポートは、SmartSymbol を InTouch アプリケーション間で移動する最適な手段です。

SmartSymbol をエクスポートするには

1. ［**描画**］メニューの［**SmartSymbol**］グループで［**管理**］を選択します。

   ［**InTouch SmartSymbol - 管理モード**］ウィンドウが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. SmartSymbol およびフォルダのリストから、エクスポートする SmartSymbol またはフォルダを選択します。
3. ［ファイル］メニューで、［開く］を選択します。［シンボルのエクスポート］ダイアログ ボックスが表示されます。
4. シンボルのエクスポート先であるフォルダを参照します。
5. 拡張子 .www を付けてファイル名を入力し、［保存］を選択します。SmartSymbol またはフォルダが、指定したフォルダにエクスポートされます。

SmartSymbol テンプレートの名前の変更

SmartSymbol マネージャを使用すると、SmartSymbol テンプレートの名前を変更できます。SmartSymbol テンプレートの名前の変更は、SmartSymbol インスタンスには影響を与えません。

SmartSymbol テンプレートの名前を変更するには：

1. SmartSymbol マネージャで、名前を変更する SmartSymbol テンプレートを選択します。
2. ［編集］メニューで［名前の変更］を選択します。
3. シンボルの新しい名前を入力して、**Enter** キーを押します。新しい名前の SmartSymbol テンプレートが表示されます。

SmartSymbol テンプレートの複製

SmartSymbol テンプレートを作成すると、そのコピーを作成できます。たとえば、テンプレートを複製して、そのテンプレートを変更したり編集したりして、それに似た機能を持つ新しいテンプレートを作成できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SmartSymbol テンプレートの編集の詳細については、「SmartSymbol の編集」を参照してください。

SmartSymbol テンプレートを複製するには：

1. SmartSymbol マネージャで、複製する SmartSymbol を選択します。
2. ［**編集**］メニューで［**コピー**］を選択します。
3. 新しい SmartSymbol 用のフォルダを選択します。
4. ［**編集**］メニューで、［**貼り付け**］を選択します。新しい SmartSymbol が表示されます。元の SmartSymbol と同じフォルダに配置する場合、新しい SmartSymbol の名前は「コピー～ <元の名前>」となります。

SmartSymbol テンプレートの削除

SmartSymbol テンプレートを削除すると、そのテンプレートに基づく SmartSymbol インスタンスのプロパティを開いたり、編集したり、参照したりすることはできなくなります。これらの SmartSymbol インスタンスのランタイムの状態は、削除によって影響されません。

削除した SmartSymbol はそのインスタンスから回復できます。詳細については、「SmartSymbol の回復」を参照してください。

SmartSymbol テンプレート削除するには：

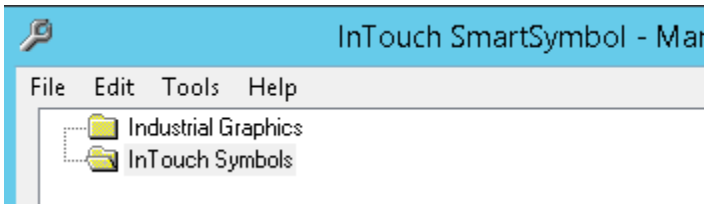
1. SmartSymbol マネージャで、削除する SmartSymbol を選択します。
2. ［**ファイル**］**メニュー**で、［**削除**］を選択します。メッセージが表示されたら、［**はい**］**を**選択します。

   SmartSymbol テンプレートが SmartSymbol ライブラリから削除されます。この SmartSymbol のすべてのインスタンスは孤立します。

フォルダ階層への SmartSymbol の保存

SmartSymbol は、標準の階層フォルダ構造で、SmartSymbol ライブラリに保存されます。SmartSymbol ライブラリの SmartSymbol の組織を簡素化するために、次の 2 つの標準フォルダが含まれています。

- 産業用グラフィック SmartSymbol テンプレートの最上位レベルのフォルダ
- InTouch SmartSymbol テンプレートの最上位レベルのフォルダ



SmartSymbol マネージャを使用して、テンプレートのサブフォルダを作成できます。作成されるときに関連付けられる必要のあるテンプレート フォルダに産業用グラフィック SmartSymbol テンプレートを保存します。たとえば、$Valve オブジェクトで使用するために SmartSymbol を作成した場合、シンボルテンプレートは「$Valve」テンプレート フォルダに保存します。

産業用グラフィック SmartSymbol を InTouch シンボル フォルダにドラッグすることはできません。また、InTouch SmartSymbol を産業用グラフィック シンボル フォルダにドラッグすることもできません。

SmartSymbol を別のフォルダに移動するには

1. 移動する SmartSymbol を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. SmartSymbol を新しいフォルダにドラッグします。

SmartSymbol と言語の切り替えのサポート

アプリケーションに SmartSymbol テンプレートが存在していれば、SmartSymbol の言語の切り替えは有効です。

SmartSymbol に翻訳可能な オブジェクトが含まれる場合、ディクショナリをエクスポートすると、たとえば「SSD_<シンボル名>_<言語 ID>_<ID>.xml」のように個別の XML が生成されます。この XML ファイルには、SmartSymbol に含まれるすべての翻訳可能な文字列が含まれています。このファイルを Excel で開き、他の InTouch アプリケーションに対して行うようにテキスト文字列を翻訳できます。

InTouch アプリケーションの翻訳をインポートすると、各 SmartSymbol の翻訳も同じようにインポートされます。

WindowViewer で言語を切り替えると、この方法で翻訳された翻訳可能な文字列を含む SmartSymbol が翻訳された状態で表示されます。

ディクショナリ ファイルを持つ SmartSymbol をエクスポートすると、ディクショナリ ファイルが .www ファイルと共にエクスポートされます。言語の切り替えの詳細については、『*AVEVA™ InTouch HMI アプリケーション ランタイム ガイド*』の「ランタイム時の言語の切り替え」を参照してください。

SmartSymbol の回復

ライブラリから SmartSymbol テンプレートを削除すると、その SmartSymbol のすべてのインスタンスは「孤立した」インスタンスと考えられます。削除した SmartSymbol は、孤立したインスタンスから回復できます。アプリケーション ウィンドウに孤立したインスタンスが存在しない場合、SmartSymbol は回復できません。

SmartSymbol テンプレートが削除された後にインスタンスのプロパティを開こうとすると、ライブラリには SmartSymbol が存在しないことを伝える警告メッセージが表示されます。

また、SmartSymbol が含まれるウィンドウをインポートすると、孤立したインスタンスを使用することもできます。孤立したインスタンスから SmartSymbol を回復して、その SmartSymbol の名前を変更する必要があります。

削除した SmartSymbol を回復するには:

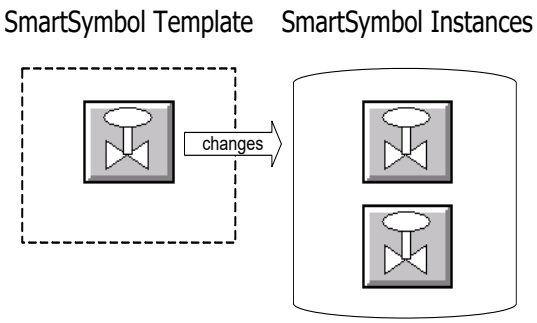
1. InTouch HMI アプリケーション ウィンドウで、削除した SmartSymbol の孤立したインスタンスを選択します。
2. ［**描画**］メニューの［**SmartSymbol**］グループで［**回復**］を選択します。

   SmartSymbol が［**新規シンボル**］の名前と共に、［**SmartSymbol - 管理モード**］ウィンドウに表示されます。
3. 必要に応じて、SmartSymbol の名前を変更します。

SmartSymbol の編集

SmartSymbol を作成した後で、テンプレートまたは SmartSymbol のインスタンスを変更したり修正したりすることによって、SmartSymbol を編集できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



SmartSymbol テンプレートの変更

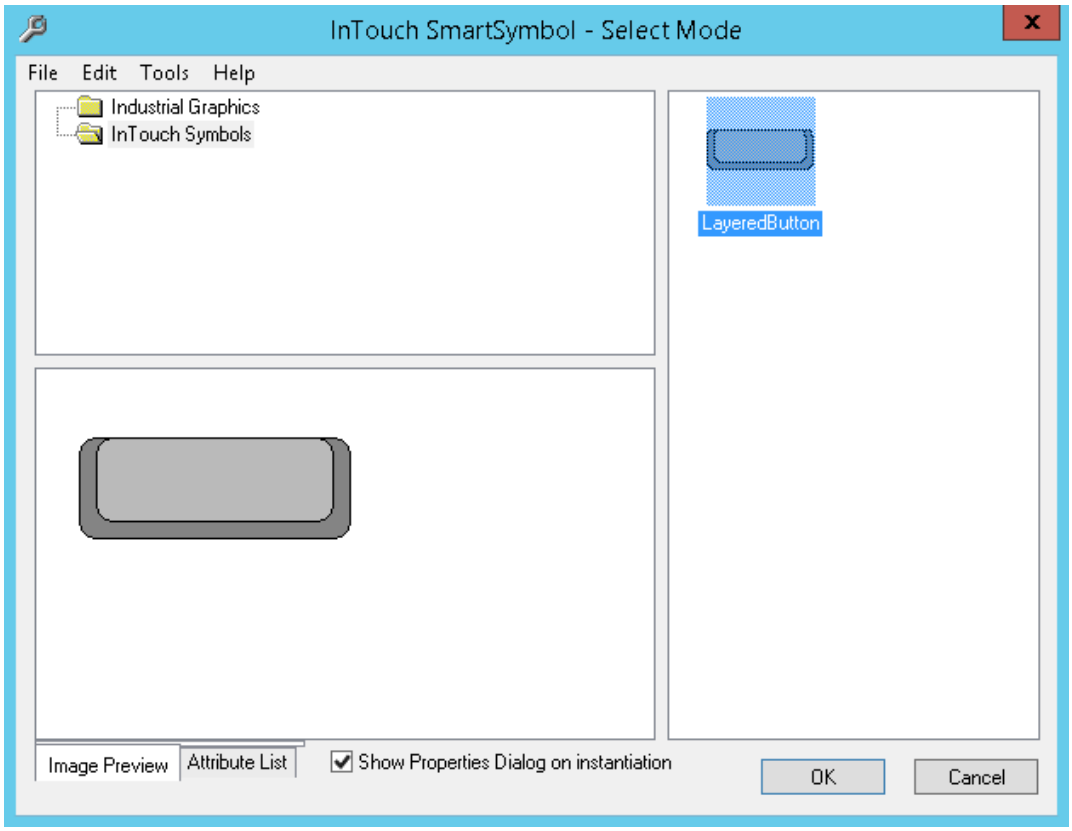
SmartSymbol を編集するには、セルを分解してから描画ツールを使用して変更を行います。SmartSymbol に関連付けられたアニメーションを変更することもできます。テンプレートを変更すると、SmartSymbol のすべてのインスタンスに影響します。

注意：SmartSymbol は、アプリケーション ウィンドウではなく一時ウィンドウで編集します。

既存の SmartSymbol テンプレートを編集するには

1. ［描画］メニューの［**SmartSymbol**］グループで［編集の開始］を選択します。
2. SmartSymbol を編集するウィンドウを選択します。

   ［**InTouch SmartSymbol - 選択モード**］ウィンドウが表示されます。



3. 編集する SmartSymbol を選択して、［**OK**］を選択します。

   SmartSymbol のインスタンスがアプリケーション ウィンドウに配置されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. ［**アニメーション**］メニューの［**セル**］グループで［**分解**］を選択します。

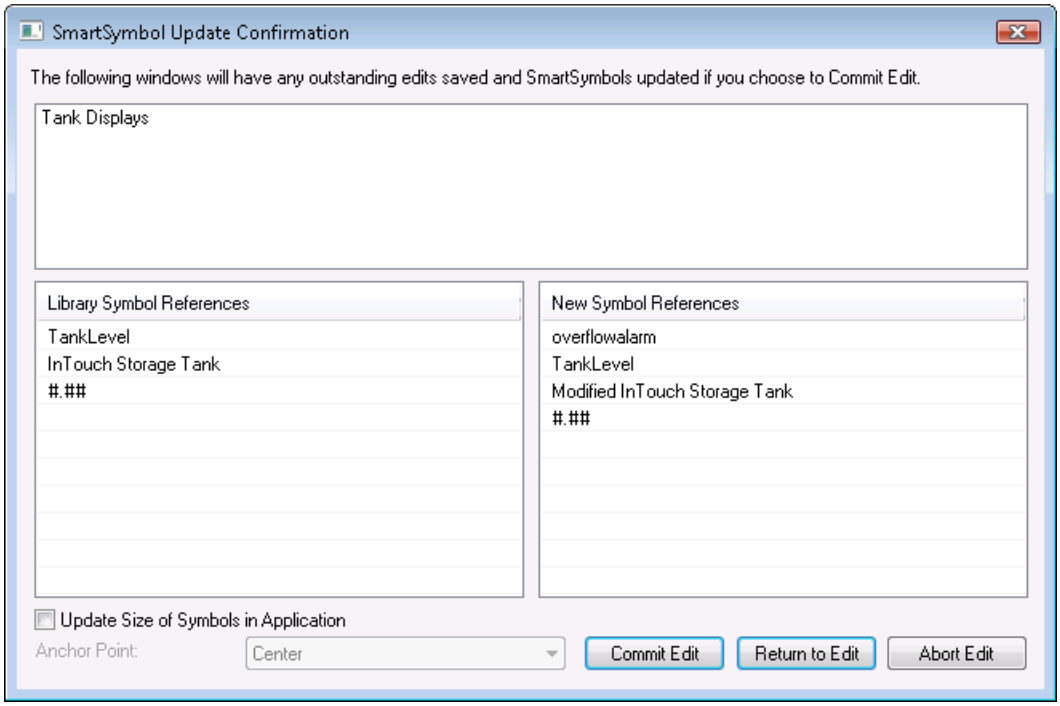
   シンボルがコンポーネント要素に分解されます。

5. これで、1 つまたは複数の要素を編集できるようになりました。

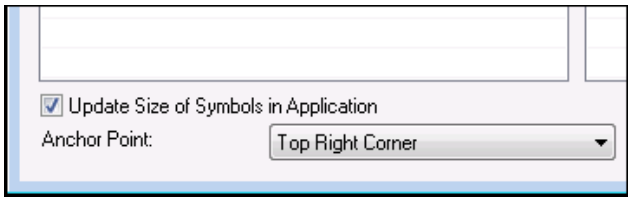
   注意：SmartSymbol の一部であるセルに要素を追加すると、より大きなセルとなります。SmartSymbol インスタンスに変更を反映する場合、サイズの変更を反映させるかどうかを選択できます。

6. 編集が完了したら、シンボルのすべての要素を選択します。
7. ［**アニメーション**］メニューの［**セル**］グループで［**セルの作成**］を選択します。
8. ［**描画**］メニューの［**SmartSymbol**］グループで［**編集の終了**］を選択します。

   ［**SmartSymbol 更新確認**］ダイアログ ボックスが表示されます。



9. 編集した SmartSymbol のサイズが変更された場合は、サイズの反映を設定できます。以下のいずれかを実行します。
   - 既存の SmartSymbol インスタンスのサイズに影響を与えないようにするには、［**アプリケーションでのシンボルの更新サイズ**］チェック ボックスをオフにします。
   - テンプレート サイズの変更を SmartSymbol インスタンスに反映させるには、［**アプリケーションでのシンボルの更新サイズ**］チェック ボックスをオンにして、［**アンカー ポイント**］リストで、サイズ変更が完了したときに画面に「アンカー」する SmartSymbol インスタンスの部分を選択します。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

10. 以下のいずれかを実行します。
    - 変更した内容を適用するには、［**編集のコミット**］を選択します。SmartSymbol マネージャにより、SmartSymbol テンプレートとすべてのインスタンスが更新されます。
    - SmartSymbol の編集を続けるには、［**編集に戻る**］を選択します。アプリケーション ウィンドウが再び表示され、編集を続けることができます。

SmartSymbol インスタンスの変更 SmartSymbol いんすたんすのへんこう

SmartSymbol インスタンスの参照および静的テキストは変更することができます。インスタンスの静的テキストは、検索および置換することができます。

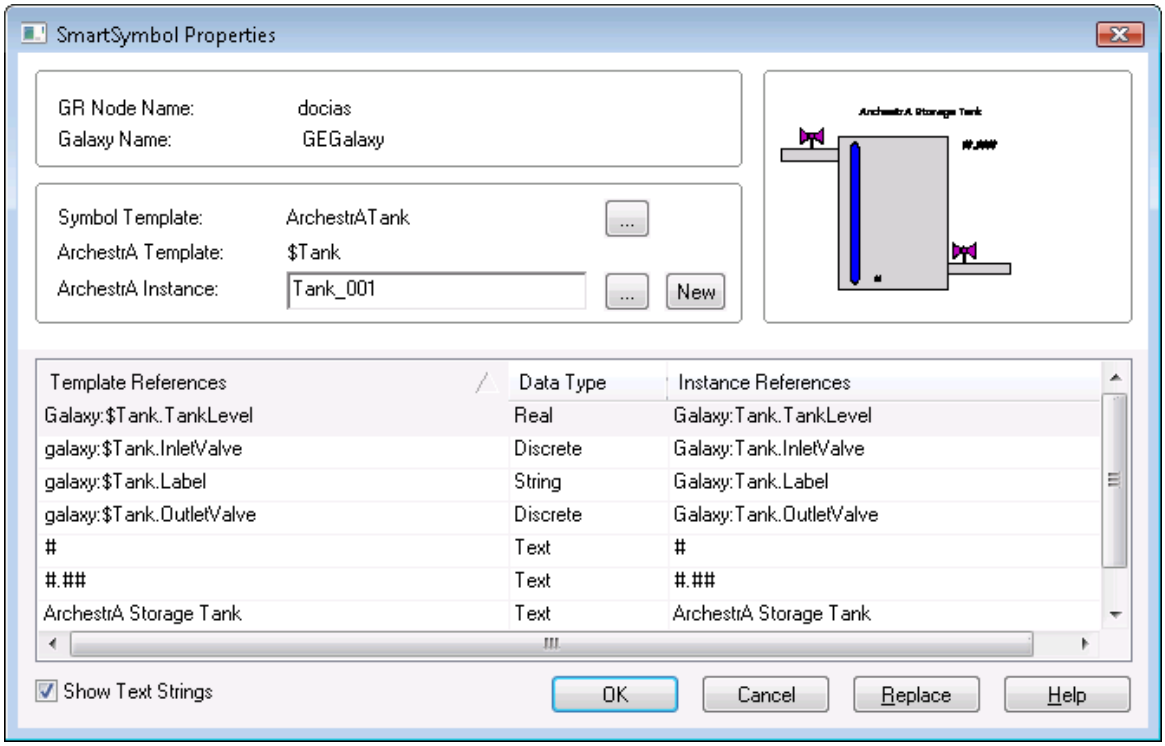
SmartSymbol インスタンスの別の参照の選択

SmartSymbol をインスタンスとしてウィンドウに配置すると、その参照先を別のオブジェクトや異なるタグ変数などに変更することができます。SmartSymbol テンプレートは影響されません。

ランタイムでは、SmartSymbol インスタンスが参照するタグ変数は、IOSetRemoteReferences() スクリプト関数を使用して変更できます。詳細については、「ランタイム中のリモート 参照のリダイレクト」を参照してください。

SmartSymbol インスタンスで参照を編集するには

1. SmartSymbol インスタンスをダブルクリックします。［**SmartSymbol プロパティ**］ウィンドウが表示されます。



2. 以下のいずれかを実行します。
    - ［**シンボル テンプレート**］の横にある［**参照**］ボタンを選択して、新しい SmartSymbol テンプレートを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**ArchestrA インスタンス**］テキスト ボックスの横にある［**参照**］ボタンを選択して、ArchestrA オブジェクト インスタンスを参照します。

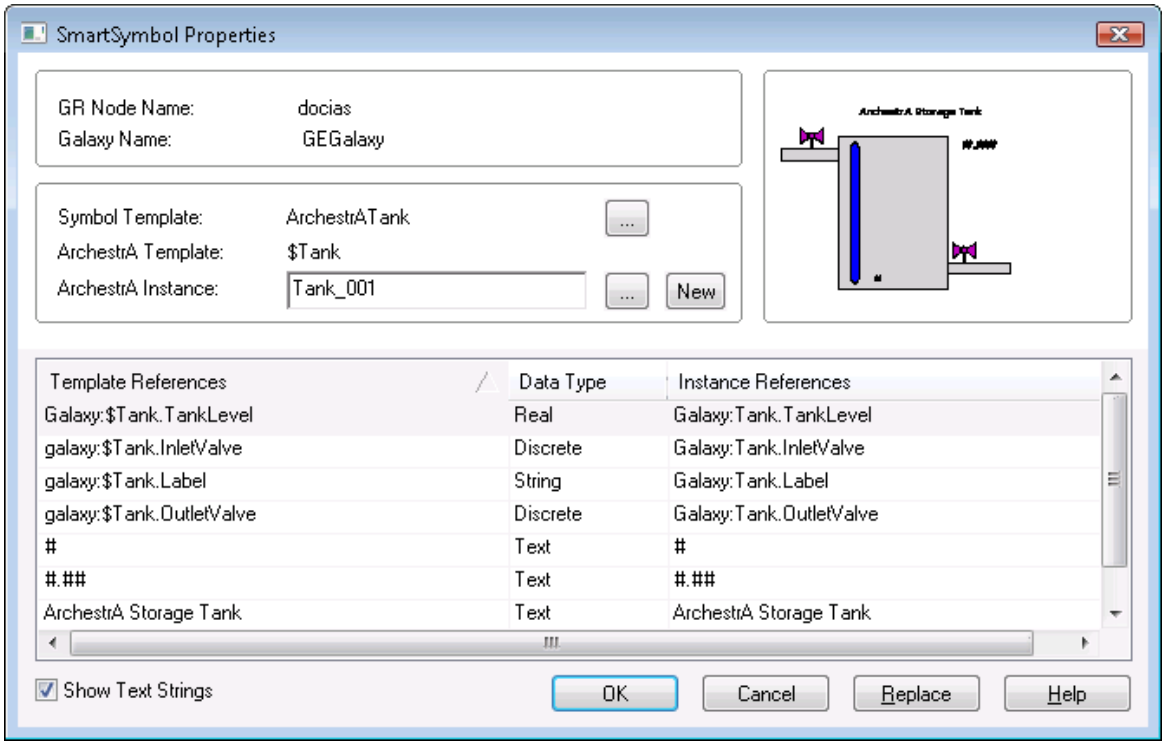
3. SmartSymbol にマップする別のオブジェクト インスタンスを選択し、［**OK**］を選択します。

SmartSymbol インスタンスのテキストおよび参照の手動での編集

アプリケーション ウィンドウで SmartSymbol インスタンスを作成すると、インスタンスの静的テキストを変更できます。

SmartSymbol インスタンスの静的テキストを変更するには

1. SmartSymbol インスタンスをダブルクリックします。［**SmartSymbol プロパティ**］ウィンドウが表示されます。



2. ［**インスタンス参照**］カラムで、テキスト ボックスを選択し、テキストを変更します。
3. **[OK]** を選択します。

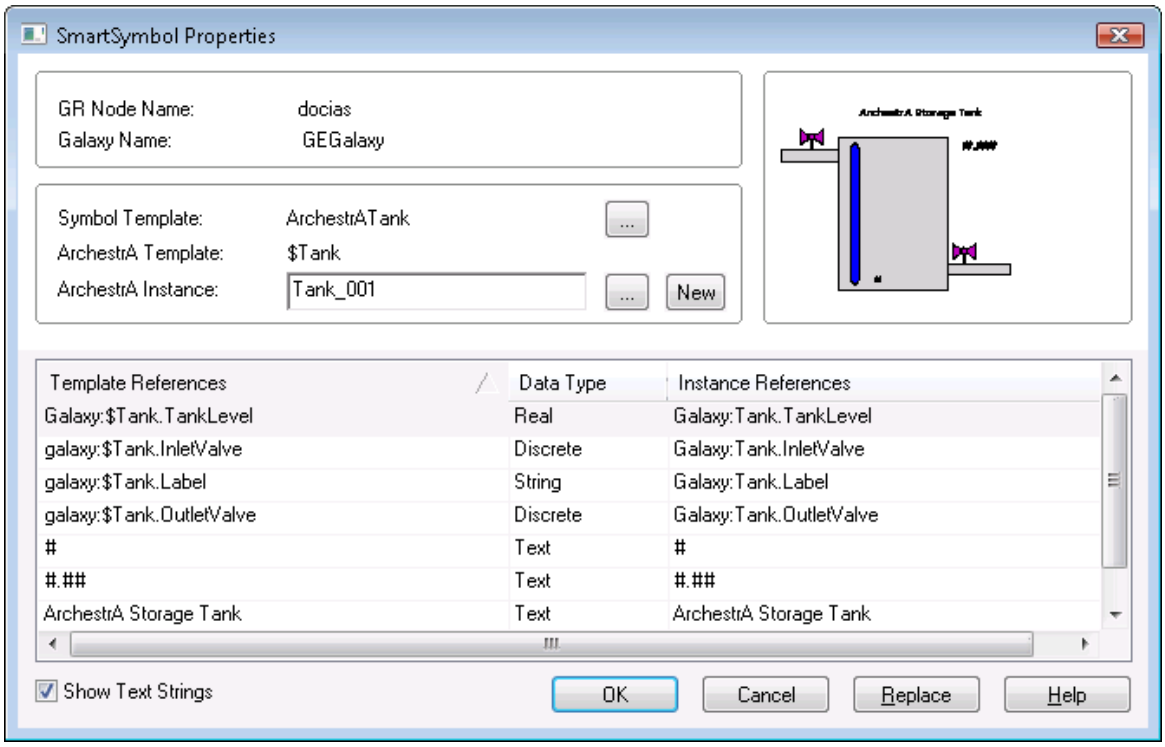
SmartSymbol インスタンスのタグ変数とテキスト文字列の置換

SmartSymbol インスタンスの複数の参照やテキスト文字列に同じ変更が必要な場合は、置換機能を使用できます。

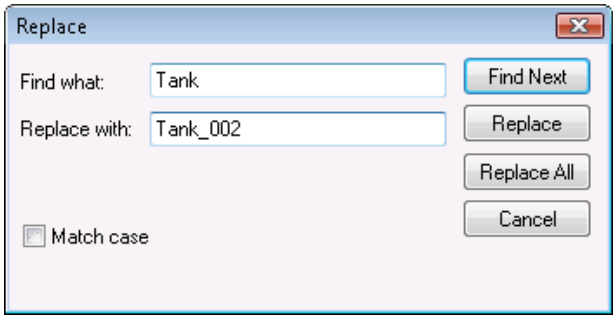
SmartSymbol インスタンスの参照を置換するには

1. SmartSymbol インスタンスをダブルクリックします。［**SmartSymbol プロパティ**］ウィンドウが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. **[置換]** を選択します。［**置換**］ダイアログ ボックスが表示されます。



3. 置換テキスト文字列を設定します。以下の手順を実行します。
   a. ［**検索する文字列**］ボックスに、置換するテキストを入力します。［**大文字と小文字を区別する**］チェック ボックスをオンにして、検索時に大文字小文字が区別されるように設定します。
   b. ［**置換後の文字列**］ボックスに、新しい置換テキストを入力します。置換テキストは常に入力されたとおりに使用されます。
4. 以下のいずれかを実行します。
   - すべてのテキストを置換するには、［**すべて置換**］を選択します。
   - 一度に 1 つのインスタンスのテキストを検索して置換するには、［**次を検索**］を選択して、［**置換して次に**］を選択し、そのインスタンスを置換します。
5. **[OK]** を選択します。アプリケーション ウィンドウに、タグ変数とテキスト文字列が変更された状態の SmartSymbol インスタンスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch SmartSymbol の移行

InTouch SmartSymbol は、インポート（移行）することによって産業用グラフィックで使用できます。SmartSymbol の外観および設定がインポートされ、アニメーション設定に変換されます。

インポートされた SmartSymbol では次のことが可能です。

- キャンバスの既存の要素に追加する。
- キャンバスの既存の要素を置換する。

通常、どの InTouch SmartSymbol でも産業用グラフィックにインポートできます。

注意：SmartSymbol には、インポートできないオブジェクト、またはインポートはできても機能が制限されているオブジェクトが含まれている場合があります。これらのオブジェクトの完全なリストについては、「SmartSymbol インポートの制限」を参照してください。

産業用グラフィックへの InTouch SmartSymbol のインポート

産業用グラフィックに InTouch SmartSymbol をインポートするには

1. 産業用グラフィック エディタを開きます。
2. ［**システム**］メニューで、［**InTouch SmartSymbol のインポート**］をクリックします。［**InTouch アプリケーション ウィザードの検索**］が表示されます。
3. InTouch アプリケーションがデフォルト以外のフォルダにある場合、［参照］ボタンをクリックしてアプリケーションのパスを参照します。
4. ［**アプリケーションの検索**］を選択します。［**検索ルート**］ボックスは、すべてのアプリケーションが検索されるパスを表示します。
5. 指定した**検索ルート** パスの下に InTouch アプリケーションがない場合、1 個を検索または参照するための新しい開始フォルダを入力して、**検索ルート** パスを変更します。
6. [**Find(検索)]** を選択します。指定した**検索ルート** フォルダに含まれるすべての InTouch アプリケーションが検索され、一覧表示されます。
7. SmartSymbol をインポートするアプリケーションを選択し、［**次へ**］をクリックします。［**InTouch SmartSymbol の選択**］ダイアログ ボックスが表示されます。
8. SmartSymbol 階層にある SmartSymbol の位置を参照し、インポートする SmartSymbol を選択し、［**OK**］をクリックします。

   キャンバス上に要素がすでにある場合、既存の要素を置換するかどうかを確認するダイアログ ボックスが表示されます。

   - ［**はい**］既存の要素を削除し、空のキャンバスに SmartSymbol をインポートする場合。
   - ［**いいえ**］既存の要素をそのままにして、SmartSymbol をインポートする場合。
9. SmartSymbol に現在オペレーティング システムにインストールされていないフォントが含まれる場合、［**フォント マッピングの編集**］ダイアログ ボックスが表示されます。

   ［**続行**］をクリックして提案されるフォント マッピングを受け入れるか、それぞれ個別のフォントのフォント マッピングを変更できます。そのためには、以下の手順を実行します。

   a. ［**マップされたフォント**］ 列のフォント名を選択します。［参照］ボタンが表示されます。

   b. [参照] ボタンを選択します。［**サポートされるフォントの選択**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

c. リストからフォントを選択して、［**OK**］をクリックします。選択したフォントが［**マップされたフォント**］列に表示されます。

d. フォント マッピングを変更する他のフォントに対して、この手順を繰り返します。

注意：次回 SmartSymbol をインポートするためのマッピングを保存する場合、［**Save mapping**］をチェックします。

10. SmartSymbol がインポートされ、キャンバスに表示されます。

SmartSymbol インポートの制限

SmartSymbol をインポートするとき、以下の設定がインポートされます。

- InTouch グラフィック
- グラフィック アニメーション
- スクリプト
- 参照

InTouch グラフィックのインポート

以下の表には、次のような InTouch グラフィックが示されています。

- 問題なくインポート可能です。
- インポートできますが、機能が変更されたり、プロセス中にいくつかの機能が失われます。
- インポートできません。

以下の InTouch グラフィックは、問題なくインポート可能です。

InTouch グラフィック	産業用グラフィック要素	注記
長方形	長方形	
角丸長方形	角丸長方形	
楕円形	楕円形	
線	線	
水平/垂直線	線	SmartSymbols は水平/垂直線を線に変換します。従って、ArchestrA はラインのみを生成します。
折れ線	折れ線	
多角形	多角形	
テキスト	テキスト	
ビットマップ	ビットマップ	
セル	グループ	ArchestrA プロパティ "アイコンとして扱う" = false。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch グラフィック	産業用グラフィック要素	注記
シンボル	グループ	ArchestrA プロパティ "アイコンとして扱う" = true。
ボタン	ボタン	

以下の InTouch グラフィックをインポートできますが、機能が変更されたり、プロセス中にいくつかの機能が失われます。

InTouch グラフィック	産業用グラフィック要素	注記
ウィザード	要素	SmartSymbol にグループ化するとき、要素のグループとして表示されます。
SmartSymbol	要素	別の Smart Symbol にグループ化されるとき、セルに分割し、SmartSymbol プロパティ自体を失います。

SmartSymbol に追加できないため、以下の InTouch グラフィックはインポートできません。

InTouch グラフィック	産業用グラフィック要素	注記
リアルタイムトレンド	該当なし	SmartSymbol に追加できません。
履歴トレンド	該当なし	SmartSymbol に追加できません。
ActiveX コントロール	該当なし	SmartSymbol に追加できません。これは、すべての ActiveX アラーム コントロール (Alarm DB View、Alarm Viewer など) を含む可能性があります。

グラフィック アニメーションのインポート

InTouch SmartSymbol をインポートするとき、InTouch アニメーションで設定されたすべてのデータは、ArchestrA アニメーションにインポートされます。InTouch アニメーションおよび ArchestrA アニメーションは別の名前を持ちますが、同じ機能を実行します。

以下の表は、どのアニメーションがそれぞれに対応しているのか示します。

InTouch アニメーション リンク	ArchestrA アニメーション
ユーザー入力 - 論理型	ユーザー入力

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アニメーション リンク	ArchestrA アニメーション
ユーザー入力 - アナログ型	ユーザー入力
ユーザー入力 - 文字列	ユーザー入力
スライダー - 垂直スライダー	垂直スライダー
スライダー - 水平スライダー	水平スライダー
タッチ押しボタン - 論理値	押しボタン
タッチ押しボタン - アクション	アクション スクリプト
タッチ押しボタン - ウィンドウの表示	サポートしていません
タッチ押しボタン - ウィンドウの非表示	サポートしていません
線の色 - 論理型	線スタイル
線の色 - アナログ型	線スタイル
線の色 - 論理型アラーム	線スタイル
線の色 - アナログ アラーム	線スタイル
塗りつぶしの色 - 論理型	塗りつぶしスタイル
塗りつぶしの色 - アナログ型	塗りつぶしスタイル
塗りつぶしの色 - 論理型アラーム	塗りつぶしスタイル
塗りつぶしの色 - アナログ アラーム	塗りつぶしスタイル
テキストの色 - 論理型	テキスト スタイル
テキストの色 - アナログ型	テキスト スタイル
テキスト色 - 論理型アラーム	テキスト スタイル
テキスト色 - アナログ アラーム	テキスト スタイル
オブジェクト サイズ - 高さ	高さ
オブジェクト サイズ - 幅	幅
位置 - 垂直	水平位置
位置 - 水平	垂直位置
塗りつぶしパーセント - 垂直	垂直塗りつぶしパーセント
塗りつぶしパーセント - 水平	水平塗りつぶしパーセント
その他 - 表示オン/オフ	表示オン/オフ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アニメーション リンク	ArchestrA アニメーション
その他 - 点滅	点滅
その他 - 向き	回転
その他 - 無効	無効化
その他 - ツールヒント	ツールヒント
値の表示 - 論理値	値の表示
値の表示 - アナログ値	値の表示
値の表示 - 文字列の値	値の表示

アクション スクリプトのインポートあくしょんすくりぷとのいんぽーと

SmartSymbol をインポートすると、オブジェクトに関連付けられている SmartSymbol 内のすべてのアクション スクリプトもインポートされます。SmartSymbol 内のアクション スクリプトは、産業用グラフィックのスクリプト アニメーションになります。

定義済み InTouch 関数 (QuickScripts) のほとんどはインポートされています。

数学関数

InTouch WindowMaker に含まれる次の数学関数が産業用グラフィック エディタでサポートされています。

```
Abs、ArcCos、ArcSin、ArcTan、Cos、Exp、Int、Log、LogN、Pi、Round、Sgn、Sin、Sqrt、Tan、Trunc
```

文字列関数

InTouch WindowMaker に含まれる次の文字列関数が産業用グラフィック エディタでサポートされています。

```
Dtext、StringASCII、StringChar、StringCompare、StringCompareNoCase、StringFromGMTTimeToLocal、
StringFromIntg、StringFromReal、StringFromTime、StringFromTimeLocal、StringInString、
StringLeft、StringLen、StringLower、StringMid、StringReplace、StringRight、StringSpace、
StringTest、StringToIntg、StringToReal、StringTrim、StringUpper、Text、wwStringFromTime
```

システム関数

InTouch WindowMaker に含まれる次のシステム関数が産業用グラフィック エディタでサポートされています。

```
[アプリケーションのアクティブ化]
```

その他の関数

InTouch WindowMaker に含まれる次のその他の関数が産業用グラフィック エディタでサポートされています。

```
DateTimeGMT、LogMessage、SendKeys、WWControl
```

参照のインポート

SmartSymbol をインポートするとき、以下の変更がタグと参照に行われます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch SmartSymbol	産業用グラフィック	例
ローカル タグ	"InTouch:" キーワードの頭字語。	リアル メモリ タグ "TankLevel1" は、"InTouch:TankLevel1" に変換されます。
ドット フィールドのローカル タグ	"InTouch:" キーワードの頭字語。	論理型メモリ タグ "TankLevel1.InAlarm" は、"InTouch:TankLevel1.InAlarm" に変換されます。
スーパータグ	"InTouch:" キーワードの頭字語。以下の構文で式を手動で囲う必要があります。 属性("...")	リアル SuperTag メンバー "Reactor1\Level" は "InTouch:Reactor1\Level" に変換されます。以下のように式を手動で変更する必要があります。 属性(「InTouch:Reactor1\Level」);
I/O 参照	"InTouch:" キーワードの頭字語。	整数 I/O タグ "Testprot:i00" は "InTouch:Testprot:i00" に変換されます。
Galaxy 参照	"Galaxy:" の頭字語が削除されます。	Galaxy 参照 "galaxy:Pump1.Valve1" は "Pump1.Valve1" に変換されます。

以下のアイテムは機能を変更した上でインポートされます。

InTouch SmartSymbol	産業用グラフィック	例
Galaxy:ObjectTagname。プロパティ.#VString	"Galaxy:" の頭字語が削除されますが、#VString はサポートされていません。#VString1、#Vstring2、#VString3、#VString4 にも適用されます。	"Galaxy:Tank.PV.#VString4" は "Tank.PV" に変換されます。
Galaxy:ObjectTagname。プロパティ.#ReadSts	"Galaxy:" の頭字語が削除されますが、#ReadSts はサポートされていません。	"Galaxy:Tank.PV.#ReadSts" は "Tank.PV" に変換されます。
Galaxy:ObjectTagname。プロパティ.#WriteSts	"Galaxy:" の頭字語が削除されますが、#WriteSts はサポートされていません。	"Galaxy:Tank.PV.#WriteSts" は "Tank.PV" に変換されます。
Galaxy:ObjectTagname。プロパティ.#EnumOrdinal	"Galaxy:" の頭字語が削除されますが、#EnumOrdinal はサポートされていません。	"Galaxy:Selection.Sel1.#EnumOrdinal" は "Selection.Sel1" に 変換されます。

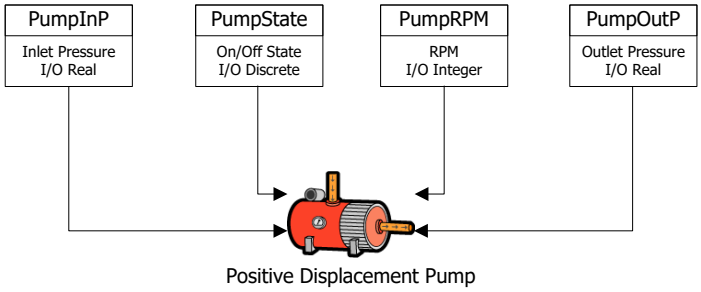
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 9 タグ

InTouch ヒューマン マシン インターフェイス（HMI）は、製造環境におけるコンポーネントをグラフィック表示するアプリケーションです。このグラフィカル インターフェイスを使用して、工場のオペレータは製造工程を監視および管理することができます。

以下の図は、製造工程のコンポーネントであるポンプの例を示しています。ポンプには、値に関連付けられているプロパティがあります。圧力、RPM、およびステータスはポンプのプロパティであり、その値は HMI から監視されます。



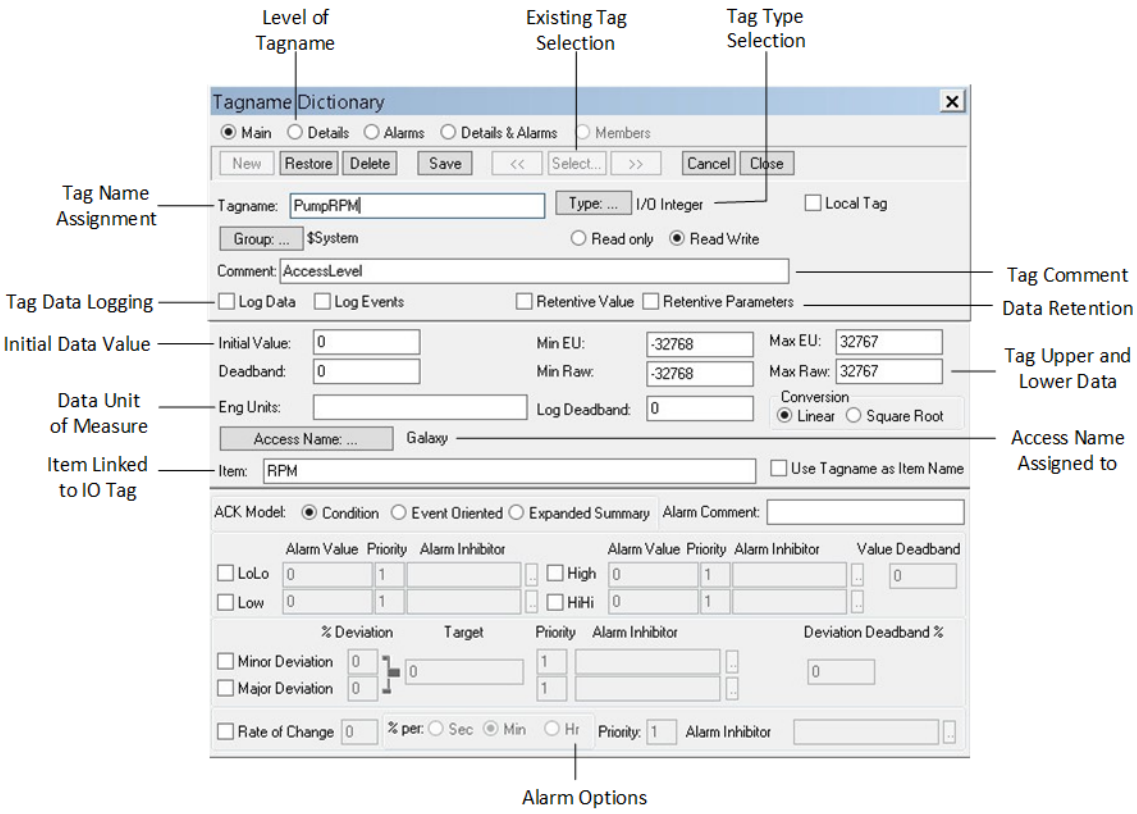
タグは InTouch HMI アプリケーションのデータ アイテムを表します。タグを使用すると、製造環境からの特定のデータ アイテムとして、特定のコンポーネント プロパティをアクセス可能にできます。上の図では、PumpState タグはポンプがオンであるかオフであるかを示しています。製造環境において、InTouch アプリケーションでプロパティを監視または制御するコンポーネントに対してタグを作成します。

製造コンポーネントから収集された異なるタイプのデータに対しては異なるタイプのタグを使用できます。たとえば、PumpState タグは、ポンプが実行中であるか停止中であるかを示すブール型のオン/オフ値を返します。アプリケーションの一部とするデータ型に対しては、適切なタイプの InTouch タグを割り当てます。

タグ名ディクショナリを使用したタグの管理

タグ名ディクショナリを使用して、InTouch アプリケーションのタグを作成できます。以下の図は、［**タグ名ディクショナリ**］ダイアログ ボックスと、I/O タグのプロパティを定義するすべてのオプションを示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



各フィールドの詳細については、「メモリ型タグ変数のプロパティ」および「I/O タグ変数のプロパティ」を参照してください。

タグ変数の使用法の計画

計画段階でアプリケーションのタグ変数の主要な必要条件を事前に指定することにより、開発時間を削減できます。また、すべての計画を実行すれば、InTouch アプリケーションの作成に要する時間を短縮できます。

タグ変数を作成する前に：

- InTouch アプリケーションの中で表す必要があるプロセスのすべての物理コンポーネントを特定します。

  アプリケーション内でデータ ソースとして表す必要があるコンポーネント属性のリストを作成します。

- 各コンポーネント属性に割り当てられるデータのタイプを特定します。

  コンポーネント属性に割り当てられたデータに基づいて、各タグ変数にタグ変数タイプを割り当てます。タグ変数へのデータ タイプの割り当てについては、「新しいタグの作成」を参照してください。

- InTouch アプリケーションの中に組み込む必要があるデータの特性を決定します。

  各タグ変数に対する以下のデータ特性を評価します。

  - データ値の予測範囲
  - データ値に割り当てる測定単位

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- データの初期値
- タグ変数の値に変更ありと認識する際のしきい値を設定するためのデッドバンド値
- タグ変数の値によって状況が変更された場合に表示されるメッセージ

タグ変数データの特性の定義については、「タグ変数プロパティの理解」を参照してください。

- タグ変数の命名の規則と標準を作成します。

  複雑なアプリケーションには通常、多くのタグ変数が必要です。命名規則を作成して標準化すれば、アプリケーションにおけるタグ変数の構成が分かるようになります。タグ変数の命名規則については、「タグ名規則」を参照してください。

- どの処理データを保存する必要があるかを決定します。

  選択されたデータはログ ファイルに保存されます。ログ済みデータを使用することにより、タグ変数の値の時系列変化を示す履歴トレンドを作成できます。タグ変数のログ記録の設定については、「タグ変数のログ記録」を参照してください。

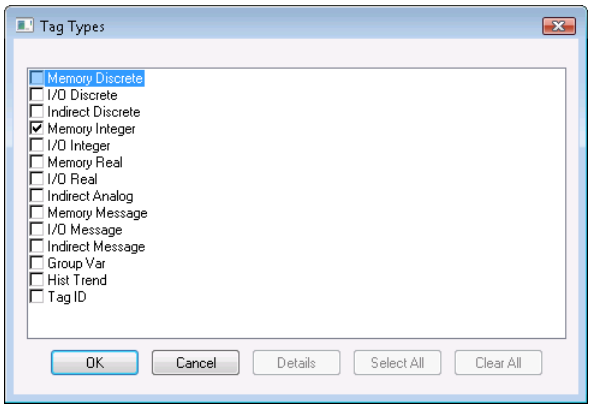
新しいタグの作成

WindowMaker **Tagname Dictionary** で作成を開始する前に、工場のプロセスを分析して InTouch アプリケーション内で作成する必要があるタグを決定します。

新しいタグを作成するには

1. WindowMaker で InTouch アプリケーションを開きます。
2. ［**ホーム**］メニューの［**タグ**］グループで［**タグ ディクショナリ**］を選択します。

   初めてタグ名ディクショナリを開くと、**$AccessLevel システム タグ**の定義が［**タグ名**］ボックスに表示されます。新しいタグを保存した後は、［**タグ名ディクショナリ**］には最後に保存されたタグの定義が表示されます。
3. 以下の手順を実行します。
   a. [**新規作成**] を選択します。［**タグ名**］ボックスが空白になります。
   b. 新しいタグの名前を入力します。タグの命名要件については、「タグ名規則」を参照してください。
   c. オプションとして、［**コメント**］ボックスに新しいタグに関するコメントを入力します。
4. **タイプ**を選択［**タグ タイプ**］ダイアログ ボックスが表示されて、サポートされている InTouch タグ タイプが表示されます。



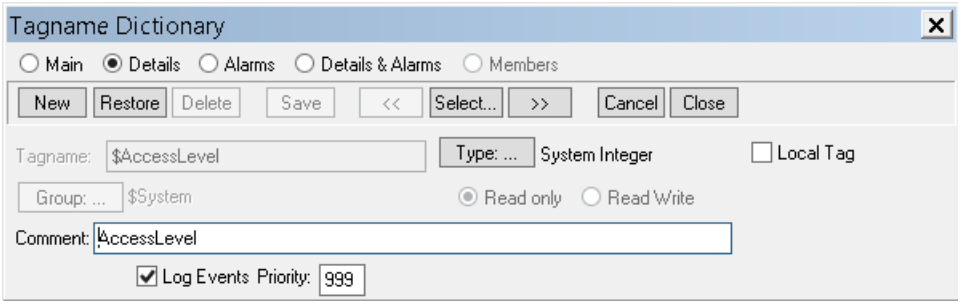
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. リストからタグ変数のタイプを選択して、［**OK**］を選択します。タグ名ディクショナリが再表示されて、選択したタグのタイプが表示されます。
6. 必要に応じて［**詳細設定**］を選択すると、選択したタグ変数タイプに対応するタグ変数ディクショナリの追加オプションが表示されます。
7. ［**タグ名ディクショナリ**］ダイアログ ボックスでさらにタグ オプションを指定します。

   タグ プロパティの指定については、「タグ プロパティの設定」を参照してください。
8. **保存**を選択します。［**閉じる**］選択して、［**タグ変数ディクショナリ**］ダイアログ ボックスを閉じます。

タグ プロパティの設定

［**タグ名ディクショナリ**］ダイアログ ボックスを使用して、すべてのタグ定義の一部である共通のタグ プロパティを指定できます。各タグに名前を割り当てる必要があります。オプションとして、コメントの追加が可能です。タグ名とコメントは、すべてのタグに共通するプロパティです。



InTouch タグの各タイプには独自のデータ プロパティがあります。タグ タイプを選択すると［**タグ名ディクショナリ**］ダイアログ ボックスが拡張されて、選択したタグ タイプに応じたオプションのセットが表示されます。

共通のタグ変数プロパティ

各タグ変数には独自の名前を割り当てる必要があります。オプションとしてコメントをタグ変数定義の 1 部とすることができます。定義した各タグ変数に応じて適切なコメントを指定すると良いでしょう。

すべてのタグ変数は、別の共通のタグ変数プロパティのアラーム グループに属しています。デフォルトでは、すべてのタグ変数は $System アラーム グループに属します。別のアラーム グループへのタグの割り当てについては、「アラームの設定」を参照してください。

タグ名規則

似通ったプロパティを持つ多くのタグ変数が必要な場合は、一貫した命名規則に従ってタグ変数に名前をつけます。

次に示す InTouch のタグ変数名の命名規則に従ってください。

- タグ変数名に使用できる最大文字数は 128 文字です。
- タグ変数名の先頭の文字には英数字（**A-Z**、**a-z**、**0-9**）を使用します。

  タグ変数名には英数字のみを使用することをお勧めします。
- タグ変数名には英数字を 1 文字以上の使用する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オプションとして、次に示す特殊文字を使用します。

- ダッシュ	! 感嘆符	# 番号記号
$ ドル記号	% パーセント	& アンパサンド
? 疑問符	@ アットマーク	_ 下線

タグ変数名への特殊文字の使用については、アプリーケーションでどうしても必要な場合を除き、できるだけ避けるようにしてください。

- タグ変数名へのダッシュ（-）の使用は避けてください。

  ダッシュは InTouch のタグ変数名に使用できる有効文字です。しかし InTouch では、ダッシュは論理演算を表す否定演算子または数値演算を表す減算演算子として評価されます。たとえば A=B-C と表記すると、A は B 引く C と解釈されるか、または B-C という名前のタグ変数がタグ変数 A に割り当てられます。

  タグ名が数字で始まる場合、ハイフンまたはマイナス記号（-）は使用できません。

- タグ変数名にスペースを使用しないでください。
- タグ変数名には指数として解釈される可能性がある数字を使用しないでください。

  たとえばタグ変数名を 125E4 とすると、基数に 4 乗が付いた指数表現と解釈されるため、使用できません。

- タグ変数名には 16 進数として解釈される可能性がある数字を使用しないでください。

  たとえば、タグ変数名を 0x123B とすると 16 進数として解釈されるため、使用できません。

自動命名タグ変数じどうめいめいたぐへんすう

タグ変数ディクショナリでタグ変数に名前を付ける場合は、使用している命名規則を InTouch HMI が追跡します。たとえばタグ変数名を Pump01、Pump02 とすると、InTouch HMI は次のタグ変数名として Pump03 を候補に挙げます。この名前は採用してもしなくても構いません。この命名支援機能はインデクシングと呼ばれています。

インデクシングは、タグ変数名の中の最後の続き番号に基づきます。たとえば、タグ変数名を PumpInP04LotB99A とした場合、InTouch HMI が次のタグ変数名として候補に挙げるのは PumpInP05LotB99A ではなく PumpInP04LotB100A です。

タグ コメント

タグ変数作成時のオプションとして、最大で 160 文字までのコメントを タグ変数ディクショナリの［**コメント**］ボックスに入力できます。

初めてタグ変数ディクショナリにアクセスすると、**$AccessLevel** システム タグ変数のデフォルト コメントが［**コメント**］ボックスに表示されます。これから定義するタグ変数にこのコメントを使用しない場合は、削除してください。

タグ変数プロパティの理解

共通のタグ変数プロパティを指定したら、作成中のタグ変数のタイプに固有なその他のプロパティを定義する必要があります。タグ変数タイプごとのメモリ タグ変数の基本的なプロパティを次の表に示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ タイプ	独自のプロパティ
論理型	初期値、オン メッセージ、オフ メッセージ、コメント
整数型	初期値、最小値、デッドバンド、工学単位、最大値、ログ デッドバンド、コメント
実数型	初期値、最小値、デッドバンド、工学単位、最大値、ログ デッドバンド、コメント
メッセージ型	最大文字数、初期値、コメント

I/O タグ変数には、ネットワーク通信を確立して、ネットワーク デバイスからの生データを InTouch アプリケーションで使用される正規化した値に変換するための追加のプロパティがあります。I/O タグ変数の定義については、「I/O タグ変数のプロパティの設定」を参照してください。

値の範囲、測定単位、および初期値

論理型、整数型、実数型、およびメッセージ型のタグ変数には、InTouch アプリケーションが WindowViewer で起動した際に初期値が割り当てられます。論理型タグ変数の場合の初期値は、とり得るバイナリ状態のいずれかです。整数型と実数型のタグ変数の場合、初期値はアプリケーションが起動したときにタグ変数に割り当てられた数値です。初期値はタグ変数ディクショナリで指定されます。

WindowViewer でアプリケーションの実行が停止した際に、タグ変数の最終値を初期値とするように指定できます。タグ変数ディクショナリから［**変数値保持**］オプションを選択すると、アプリケーションが再起動した際にタグ変数の初期値として最後の有効値が割り当てられます。

整数型と実数型のタグ変数には、タグ変数がとり得る数値範囲の下限と上限を設定するプロパティが含まれています。整数型と実数型のタグ変数にはどちらにも［**最小値**］および［**最大値**］プロパティがあり、範囲の下限と上限が定義されます。

また整数型と実数型のタグ変数には、タグ変数値の測定単位を表す工学単位を割り当てる［**工学単位**］プロパティも含まれています。たとえば、ポンプ圧に関連する整数型タグ変数の［**工学単位**］プロパティには PSI を割り当てることができます。

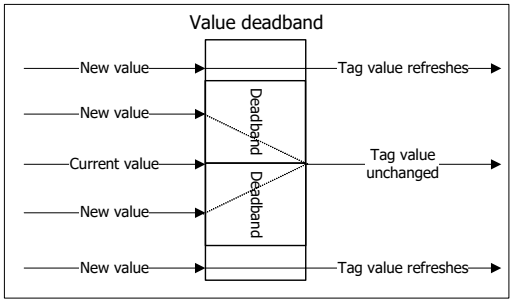
タグ変数デッドバンド

デッドバンドによりタグ変数値の感度が設定されます。デッドバンドは通常、絶えず値が変化する I/O タグ変数に割り当てられます。タグ変数の一時的な微小変化をデッドバンドによって取り除くことで、InTouch のデータ処理量が削減できます。

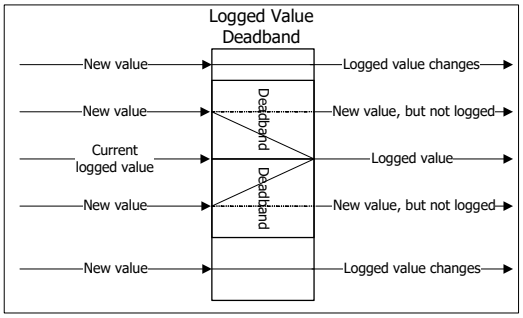
タグ変数ディクショナリには、整数データおよび実数データが割り当てられたタグ変数に対する 2 種類のデッドバンド プロパティが用意されています

- デッドバンド値プロパティにはしきい値が設定されています。WindowViewer では、このしきい値を超えた場合にのみランタイム メモリ内のタグ変数値がリフレッシュされます。以下の図は、現在のタグ変数値の前後にあるデッドバンドの領域を示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



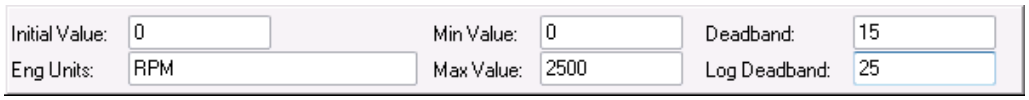
- ログ デッドバンドにはしきい値が設定され、このしきい値を超えなければタグ変数の値はログ ファイルに書き込まれません。以下の図は、ログ済みタグ変数の現在の値の前後にあるデッドバンドを示しています。



新しいタグ変数の値は、デッドバンドの外側の値のみがログ ファイルに書き込まれます。デッドバンド領域内の微小な変化は無視され、ログされません。

タグ変数デッドバンドを設定するには

1. ［**タグ変数ディクショナリ**］ダイアログ ボックスを開きます。
2. **[Select] を選択します。**［**タグの選択**］ダイアログ ボックスが表示されます。現在定義されているアプリケーションのタグ変数のリストが表示されます。
3. リストから整数型か実数型のタグ変数タイプを選択します。
4. **[OK]** を選択します。整数型か実数型のタグ変数を選択した場合には、［**タグ変数ディクショナリ**］ダイアログ ボックスの詳細設定に追加のオプションが表示されます。



5. 選択したタグ変数タイプに従い整数値か実数値を［**デッドバンド**］ボックスに入力して、デッドバンド値を設定します。

   デッドバンド値には、しきい値レベルの絶対値が工学単位で設定されます。
6. 選択したタグ変数タイプに従い整数値か実数値を［**ログ デッドバンド**］ボックスに入力して、ログ デッドバンドを設定します。

   ログ デッドバンドには、デッドバンド値と同様にしきい値の絶対値が工学単位で設定されます。
7. ［**保存**］を選択して、デッドバンドの変更を保存します。
8. ［**閉じる**］選択して、［タグ変数ディクショナリ］ダイアログ ボックスを閉じます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ変数値の保持

［**タグ変数ディクショナリ**］の詳細設定には、タグ変数の値とユーザーが加えたアラーム制限の変更を保持するための 2 種類のプロパティが用意されています。

［**変数値保持**］プロパティはすべてのタグ変数タイプに含まれています。アプリケーションが停止したときにタグ変数の現在の値を保持するには、［**変数値保持**］を選択してください。WindowViewer でアプリケーションを再起動すると、保持した値がこの変数の初期値として使用されます。

WindowViewer でアプリケーションが再起動されたときに、WindowViewer は I/O デバイスに対して保持した値を書き込みません。I/O の値は、I/O サーバーがデータを供給するデバイスを最初にスキャンしたあとで更新されます。

WindowViewer の実行中は、新規または既存のタグ変数に対する［**変数値保持**］オプションを選択／解除することはできません。このオプションを選択すると、タグ変数の初期値は常に更新されて現在の値が反映されます。WindowViewer が終了すると、初期値は最後に保持されていた値に設定されます。このオプションを後で解除すると、タグ変数の初期値は最後に保持されていた値に設定されます。

整数型と実数型のタグ変数には、［**パラメータ値保持**］プロパティが含まれています。アプリケーションの実行中にユーザーがタグ変数のアラーム制限に加えたすべての変更を保持するには、［**パラメータ値保持**］を選択してください。WindowViewer でアプリケーションが再起動されると、変更したアラーム制限が初期値として使用されます。

I/O 接続

I/O タグ変数のすべてのタイプには、外部データ ソースのアクセス名とアイテム名を指定する必要があります。I/O タグ変数のアクセス名とアイテム名の指定については、「I/O アクセス パラメータの設定」を参照してください。

タグ変数のログ記録

WindowViewer では、ランタイム中にログ変数のデッドバンドを超える変化がタグ変数の値に生じるたびに、履歴ログ ファイルに内容を書き込むようにすることができます。また WindowViewer では、現在のタグ変数の値に関わりなく一定間隔おきにログ ファイルへの書き込みを行います。デフォルトの間隔は 1 時間です。

注意：ログ記録の制御をさらに容易にしてその用途を拡大するには、Historian を使用して InTouch の履歴データを保存することをお勧めします。

［**タグ名ディクショナリ**］ダイアログ ボックスには、データとイベントをログ ファイルに記録するための個別のオプションがあります。変数値ログのオプションを設定することができます。イベント ログの設定については、「アラームの設定」を参照してください。

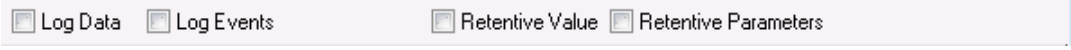
タグ変数の値を履歴ログ ファイルに書き込むには、履歴ログを有効にする必要があります。一般的なログ プロパティの設定については、「履歴ログの設定」を参照してください。

整数型と実数型のタグ変数に対しては、対応する詳細ダイアログ ボックスで［ログ デッドバンド］を設定します。［**ログ デッドバンド**］オプションでは、タグ変数の値が工学単位でいくつ変化したらログに内容が書き込まれるかを指定します。

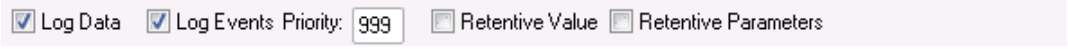
タグ変数のログを設定するには

1. **タグ名ディクショナリ**を開きます。
2. ログ ファイルに保存するデータのタグを選択します。
3. ［**データ ログ**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. オペレータ、I/O、QuickScript、またはオペレーティング システムに起因するタグ値の変更をログ記録するには、［**イベント ログ**］を選択します。［**イベント ログ**］を選択すると［**優先度**］ボックスが表示されます。



［**優先度**］の値によって、タグ変数に対するイベントの優先度が決定されます。有効な入力値は 1～999 で、1 は最高優先度、999 は最低優先度を示します。

5. ［**保存**］を選択して、［タグ変数ディクショナリ］を閉じます。

論理型タグの作成

InTouch アプリケーション内で実行されている内部プロセスのバイナリ状態を示す論理型タグを指定できます。論理型タグにはオン/オフの初期値を割り当てる必要があります。タグに関連するプロセスがアラーム状態になったときやアラーム状態から抜け出したときにアラーム イベント ウィンドウに表示するメッセージを指定することもできます。

メモリ論理型タグを定義する手順を以下に示します。I/O 論理型タグは、プログラマブル コントローラ、プロセス コンピュータ、およびネットワーク ノードのデータとのすべての入出力のバイナリ状態を示します。

I/O 論理型タグのプロパティの設定の詳細については、「論理型 I/O タグ変数の指定」を参照してください。

メモリ論理型タグの初期値とメッセージを定義するには

1. ［**タグ タイプ**］ダイアログ ボックスで、タグのタイプとして［**メモリ論理型**］を選択します。
2. ［**タグ名ディクショナリ**］ダイアログ ボックス上部の［**詳細**］を選択して詳細オプションを表示します。［**タグ名ディクショナリ**］ダイアログ ボックスの詳細部分が表示されます。

3. タグに関連付けられた初期値として［**オン**］または［**オフ**］を選択します。アプリケーションの起動時、タグがこの初期値に設定されます。
4. タグがアラーム状態になったときやアラーム状態から抜け出したときに表示するメッセージを［**オン メッセージ**］ボックスおよび［**オフ メッセージ**］ボックスに入力します。

   これらのメッセージは、タグにアラームの設定があるかどうかに関わらず、あらゆるアニメーション リンクやスクリプトで使用できます。

   - タグ値が 1（On、True）のときにアクティブな論理型アラームを定義した場合には、［**オン メッセージ**］ボックスに入力したメッセージが ActiveX アラーム表示の［**値**］列および［**しきい値**］列に表示されます。

     タグのアラーム状態が通常状態に戻ると、［**オフ メッセージ**］ボックスに入力したメッセージが［**値**］列に表示され、［オン メッセージ］は［**しきい値**］列に残ります。

   - タグ値が 0（Off、False）のときにアクティブな論理型アラームを定義した場合には、［**オフ メッセージ**］ボックスに入力したメッセージが ActiveX アラーム表示の［**値**］列および［**しきい値**］列に表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグのアラーム状態が通常状態に戻ると、[**オン メッセージ**] ボックスに入力したメッセージが [**値**] 列に表示され、[オフ メッセージ] は [**しきい値**] 列に残ります。

5. タグの変更を保存します。

整数型と実数型タグ変数の作成

整数型と実数型タグ変数を指定して InTouch アプリケーション内で実行されている処理の数値を表すことがきます。

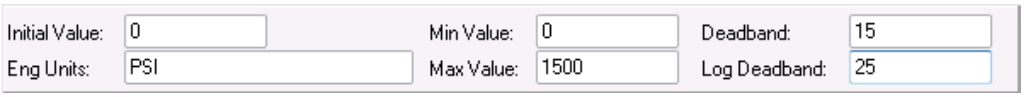
メモリと I/O の整数型と実数型のタグ変数を定義する方法を以下に示します。メモリ整数型とメモリ実数型のタグ変数には初期値を割り当てる必要があります。また、データの最大値と最小値を設定する必要もあります。

I/O 整数型と I/O 実数型のタグ変数の I/O プロパティの設定については、「整数型と実数型の I/O タグ変数の指定」を参照してください。

重要:InTouch の実数値は、8 桁の精度に制限されています。切り捨て、切り上げによる発生を避けるために、実数型タグ変数のプロパティを定義する際は 8 桁の精度を超えないようにしてください。詳細については、「IEEE 小数ユニット」を参照してください。

メモリ整数型とメモリ実数型のタグ変数の値を定義するには

1. [**タグ タイプ**] ダイアログ ボックスで、タグのタイプにメモリ整数型かメモリ実数型を指定します。[**タグ名ディクショナリ**] ダイアログ ボックスの詳細部分が表示されます。



2. 整数型と実数型タグ変数のプロパティを設定するには、以下の手順を実行します。
   - アプリケーション起動時にタグ変数に関連付けられる整数値または実数値を、[**初期値**] ボックスに入力します。
   - [**最小値**] ボックスに、タグ変数に対する最小整数値または最小実数値を入力します。

     [**最小値**] フィールドに設定されるのは、メモリ整数型やメモリ実数型に関連付けられた数値がとり得る最小値です。
   - [**最大値**] ボックスに、タグ変数に対する最大整数値または最大実数値を入力します。

     [**最大値**] フィールドに設定されるのは、メモリ整数型やメモリ実数型に関連付けられた数値がとり得る最大値です。
   - [**工学単位**] ボックスに、タグ変数名の工学単位を入力します。
3. タグの変更を保存します。

タグ変数のデッドバンドとログ デッドバンドのプロパティの設定については、「タグ変数デッドバンド」を参照してください。

メッセージ型タグ変数の作成

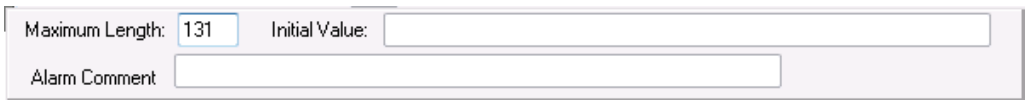
内部処理と外部処理の両方に対するメッセージ型タグ変数を指定できます。これらのタグ変数には、WindowViewer でアプリケーションが起動されときに表示される初期メッセージと、Alarm Viewer から読み込むことができるコメントを指定するプロパティがあります。

I/O メッセージ型タグ変数の定義については、「メッセージ型 I/O タグ変数の指定」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メモリ メッセージ型タグ変数を定義するには

1. ［**タグ変数タイプ**］ダイアログ ボックスで、タグ変数のタイプに［メモリ メッセージ型］を指定します。
2. 必要に応じて［**詳細設定**］を選択して、［**タグ変数ディクショナリ**］ダイアログ ボックスの詳細設定を表示させます。



3. メモリ メッセージ型タグ変数のプロパティを設定するには、以下の手順を実行します。
   - ［**最大文字数**］ボックスに、タグ変数のメッセージに表示できる最大文字数を整数値で入力します。あるいは、メッセージの最大文字数でデフォルト値の 131 文字をそのまま使用します。
   - ［**初期値**］ボックスには、WindowViewe でのアプリケーションの起動時にタグ変数に指定されるメッセージのテキストを入力します。
   - ［**アラーム コメント**］ボックスには、メッセージ型タグ変数の［**イベント ログ**］オプションが選択されている場合に AlarmViewer コントロールに読み込まれるメッセージを入力します。
4. タグの変更を保存します。

I/O タグ変数の作成

I/O タグ変数は、InTouch アプリケーションと外部処理間のネットワークの接続性を指定する共通プロパティのセットです。［**タグ変数ディクショナリ**］ダイアログ ボックスの詳細設定には、I/O タグ変数の外部プロパティを設定するためのオプションが含まれています。

I/O プロパティの設定については、「I/O を使用したデータ アクセス」を参照してください。

タグ変数の変更

タグ変数の変更は、タグ変数の作成に似ています。タグ変数ディクショナリから変更するタグ変数を選択します。次に、タグ変数の作成手順と同じように以下の手順に従ってタグ変数のプロパティを変更します。

重要：InTouch アプリケーションで使用された後のタグ変数のタイプを変更するのは容易ではありません。また、タグ変数のタイプの選択は、類似したデータ タイプに制限されることがあります。タグ変数を定義する際は、慎重に検討して正しいデータ タイプを選択してください。

タグ変数を変更するには

1. ［**タグ変数ディクショナリ**］ダイアログ ボックスを開きます。
2. **[Select]** を選択します。［**タグの選択**］ダイアログ ボックスが表示されます。アプリケーションに対して現在定義されているタグ変数のリストが表示されます。
3. 変更するタグ変数をリストから選択します。
4. **[OK]** を選択します。［**タグ変数ディクショナリ**］ダイアログ ボックスに、選択したタグ変数に対して指定されている設定値が表示されます。
5. タグ変数のプロパティを変更します。
6. ［**保存**］を選択して、変更したタグ変数を更新します。
7. ［**閉じる**］を選択して、タグ変数ディクショナリを閉じます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

OPC UA サーバーからの InTouch タグの作成

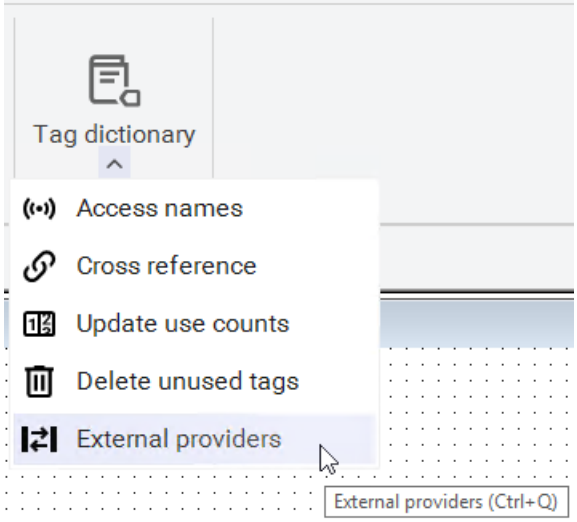
［タグ ディクショナリ］の［外部プロバイダ］オプションには、同じマシン上にある Gateway Communication Driver で設定されたすべての OPC UA サーバー接続が一覧表示されます。このオプションを使用すると、いくつかの手動ステップを回避して、OPC UA アイテム参照用のアクセス名と InTouch タグを簡単に作成できます。

Gateway Communication Driver で設定された OPC UA サーバーから InTouch を作成するには

前提条件:

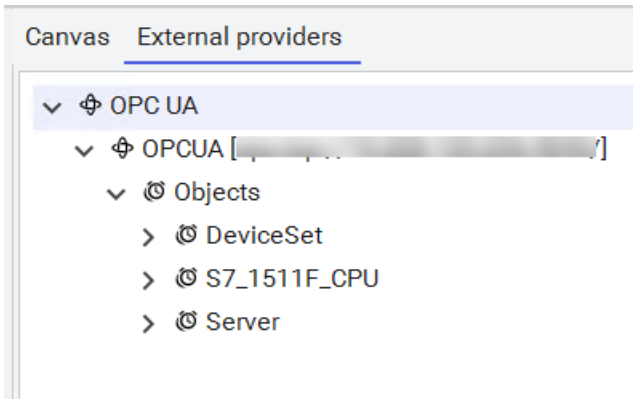
- Gateway Communication Driver で OPC UA サーバー接続を設定します。詳細については、Gateway Communication Driver ヘルプを参照してください。
- Gateway Communication Driver と InTouch WindowMaker は同じマシン上にある必要があります。

1. WindowMaker で InTouch アプリケーションを開きます。
2. ［ホーム］］メニューの［タグ］］グループで［タグ ディクショナリ］の下にあるドロップダウン矢印を選択して［外部プロバイダ］を選択します。または、キーボード ショートカット **Ctrl + Q** を使用します。



Gateway Communication Driver で設定されたすべての OPC UA サーバー接続が表示されます。

3. OPC UA ツリーを展開してアイテム参照を表示および参照できます。



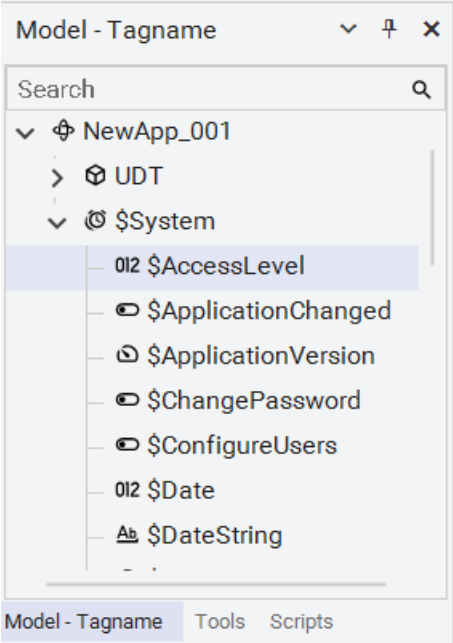
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注：
リモート Gateway Communication Driver からの OPC UA サーバーは表示されません。
OPC UA アイテムが表示された後に別の OPC UA サーバーを設定する場合、［**外部プロバイダ**］ウィンドウの下にある［**更新**］ボタンを選択して、新しく追加した OPC UA サーバー アイテムを表示します。

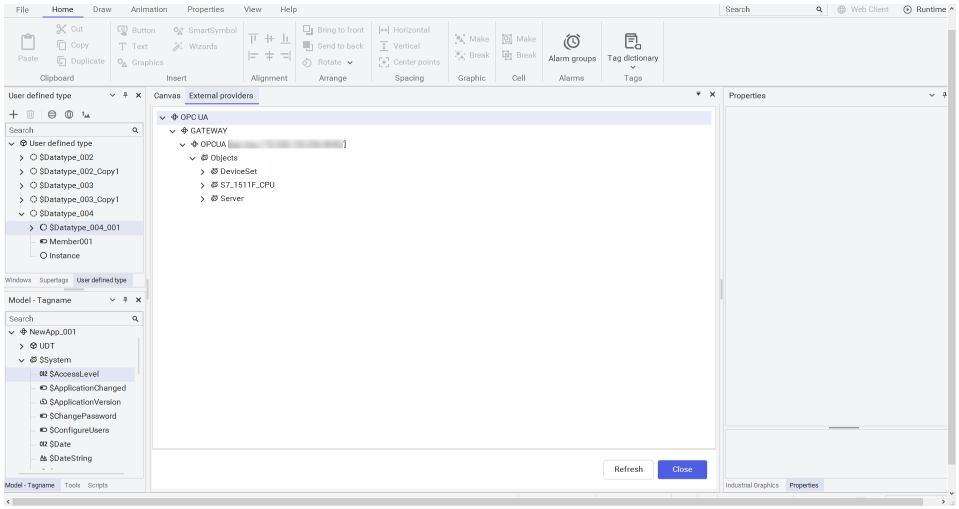
---

4. ［**モデル - タグ名**］ペインを開きます。

   ［**モデル - タグ名**］ペインに InTouch アプリケーションのすべてのアラーム グループとタグが表示されます。［モデル - タグ名］ペインの詳細については、「モデル - タグ名」トピックを参照してください。



5. ［**外部プロバイダ**］ウィンドウからアイテムを［モデル - タグ名］にドラッグ アンド ドロップします。**Ctrl** キーを押したままアイテムを選択することで複数のアイテムを選択できます。



- 目的のアラーム グループ ノードにドラッグ アンド ドロップして OPC UA タグを特定のアラーム グループに追加できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ランダムにドラッグ アンド ドロップすると、タグは **$System** ノードの下に作成されます。

  アイテムをドラッグ アンド ドロップすると OPC UA タグが作成されます。新しく作成された OPC UA タグは［**モデル - タグ名**］ペインに表示されます。

1. ［**外部プロバイダ**］ウィンドウを閉じるには、［**閉じる**］を選択します。

追加情報:

- 個々の OPC UA 接続には異なるアクセス名が追加されます。タグのアクセス名は、Gateway Communication Driver の OPC UA 接続のノード名です。

  注：タグを追加した後に Gateway Communication Driver の OPC UA ノードの名前を変更した場合、アクセス名を手動で変更する必要があります。アクセス名は自動的に変更されません。

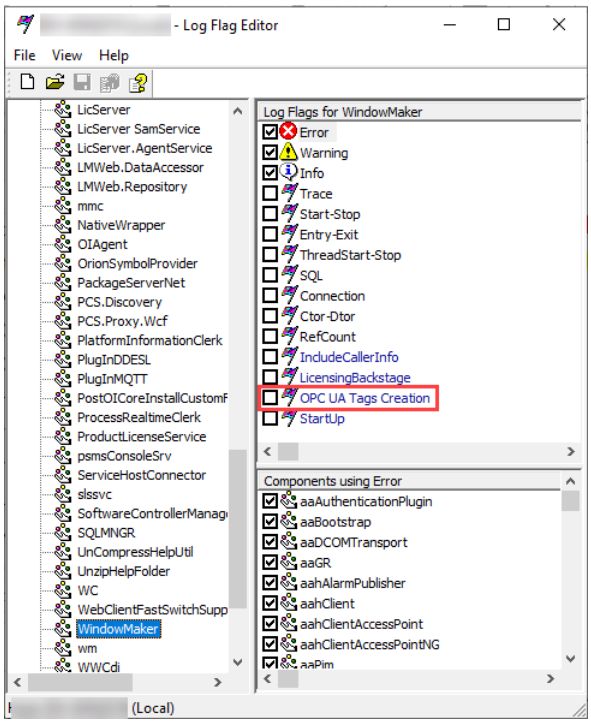
- 重複するタグ名にはインデックス番号が追加されます。たとえば、'Input' という名前の 2 つのタグがある場合、最初のタグの名前は 'Input' となり、2 番目のタグの名前は 'Input_1' になります。

  注：タグ名が最大タグ長の 128 文字を超える場合、128 文字になるまでタグ名の末尾の文字が削除されます。タグ名が 128 文字を超え、重複するタグ名がある場合、「タグ名 +_{ インデックス }」が 128 文字になるようタグ名の末尾の文字が削除されます。

- OPC UA タグのデータ型が不明な場合、対応する InTouch タグは I/O メッセージ型として作成されます。

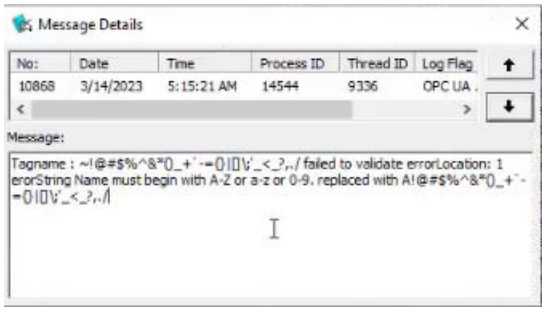
- アプリケーション マネージャで OPC UA サーバーとして設定することによって、InTouch を OPC UA サーバーとして使用することもできます。詳細については、「InTouch OPC UA サーバーを設定して使用する」トピックを参照してください。

- WindowMaker の**ログ フラグ エディタ**で［**OPC UA タグ作成**］チェック ボックスをオンにした場合、OPC UA タグに関連するログ メッセージを Log Viewer で表示できます。



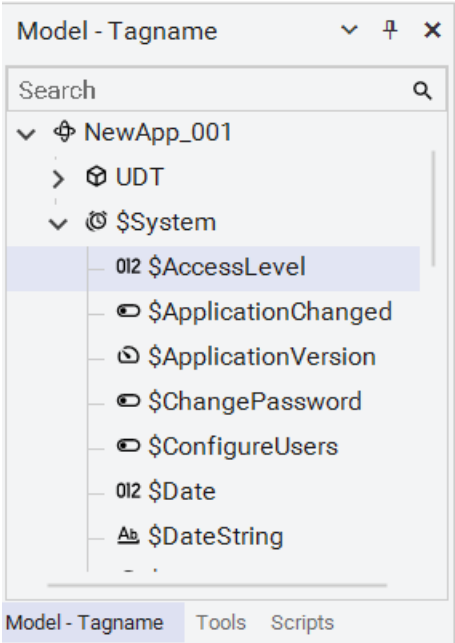
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- OPC UA アイテム名にサポートされていない特殊文字が含まれる場合、タグ名のサポートされない特殊文字は、対応する InTouch タグを作成する際に 'A' または '_' で置き換えらえます。サポートされない文字がタグ名の最初にある場合、その文字は 'A' で置き換えられます。その他の場所にあるサポートされない文字は '_' で置き換えられます。置換される文字の詳細については、Log Viewer の［**メッセージの詳細**］ウィンドウを参照してください。



モデル - タグ名

［**モデル - タグ名**］タブは、［ウィンドウ］タブと［スーパータグ］タブとともに WindowMakerWindowMaker の左側にあります。［**モデル - タグ名**］タブを選択して［モデル - タグ名］ペインを表示します。InTouch アプリケーションのすべてのアラーム グループとタグが表示されます。［**検索**］ボックスを使用して、目的のタグを検索できます。



タグを開く

1. ［**モデル - タグ名**］ペインで、開くタグを右クリックします。
2. **[開く]**を選択します。

   タグが［**タグ名ディクショナリ**］ウィンドウで開きます。

タグを 1 つのアラーム グループから別のアラーム グループに移動する

1. ［**モデル - タグ名**］ペインで、移動するタグを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. 目的のアラーム グループにドラッグ アンド ドロップします。

   選択したタグが目的のアラーム グループに移動します。

タグの削除

1. ［**モデル - タグ名**］ペインで、削除するタグを右クリックします。

   複数のタグを選択できます。

2. [**削除**]を選択します。
3. タグの削除を確認するメッセージが表示されたら［**はい**］を選択します。

   選択したタグが削除されます。

ﾀｸﾞ の削除

InTouch アプリケーションに対して定義されたすべてのタグ変数のカウントが管理されています。このタグ変数のカウントは、アニメーション リンクやスクリプトが含まれるウィンドウが削除されても自動的に減算されません。削除されたウィンドウに関連するタグ変数はそのまま使用可能とみなされ、削除することはできません。

今後使用しないタグ変数の削除を可能にするには、WindowViewer を閉じてローカル タグとリモート タグの使用カウントを更新する必要があります。タグ変数がどこで使用されているかは、InTouch クロス リファレンス ユーティリティで確認できます。クロス リファレンス ユーティリティの使用とタグ変数のカウントの更新に関して詳しくは、「タグの使用数の削減」を参照してください。

使用カウントが更新されればタグ変数を削除することができます。詳細については、「未使用タグの削除」を参照してください。

タグ変数リストと使用情報の印刷

WindowMaker の印刷ユーティリティを使用して、InTouch アプリケーションのデータベース、ウィンドウ、およびスクリプトの内容を WindowMaker で印刷することができます。

詳細については、「タグのクロス リファレンス リストの保存と印刷」を参照してください。

タグ ドットフィールドを使用したタグ プロパティの表示または変更

InTouch タグの各タイプには、タグに関連付けられているデータまたは可能な条件を記述する独自のプロパティ セットがあります。ドットフィールドはタグ プロパティを識別します。タグ名ディクショナリに表示されるほぼすべてのタグ プロパティにドットフィールドがあります。一部のドットフィールドは、すべての InTouch タグのタイプに共通しています。たとえば .Name ドットフィールドは、タグの名前に必ず関連付けられています。その他のドットフィールドは、特定のタイプのタグの一意のプロパティにのみ適用されています。

ドットフィールドをスクリプト、式、またはユーザー入力で使用すると、アプリケーションの実行中にタグのプロパティを監視および変更できます。以下の例は、スクリプトまたは式でタグ プロパティにアクセスするドットフィールドの構文を示します。

tag_name.property_dotfield

たとえば、アプリケーションの実行中にオペレータが HiHi アラームしきい値を変更できるように、アナログ - データ入力タッチ リンクを作成できます。その後、このリンクを Analog_Tag.HiHiLimit ドットフィールドの式で定義されたボタンに適用します。ランタイム時、オペレータはボタンを簡単に選択するだけで、タグの HiHi アラームしきい値に新しい値を入力できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールドはタグに関連付けられたデータの入出力に使用できます。履歴ドットフィールドは実行中のアプリケーションで現在表示されている履歴トレンドへの変更に使用できます。たとえば、ドットフィールドをスクリプトで使用すると、オペレータが履歴トレンドスクロールの変更、トレンドでのスクータのロックまたは再配置、または新しいタグへのペンの再割り当てを行うことができます。

利用可能なタグ タイプのドットフィールド

InTouch タグの各タイプには、タイプ独自のプロパティに関連付けられた一組のドットフィールドがあります。以下の表には、すべてのタイプのタグに対するドットフィールドがアルファベット順に一覧表示されています。

	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
ドットフィールド	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームメッセージ	履歴トレンド	分散アラームメッセージ/ヒストリカルアラーム	
.Ack	•	•	•		•	•	•		•	•		•			
.AckDev		•	•			•	•			•		•			
.AckDsc	•				•				•			•			
.AckROC	•	•	•			•	•			•		•			
.AckValue		•	•			•	•			•		•			
.Alarm	•	•	•		•	•	•		•	•		•			
.AlarmAccess														•	
.AlarmAckModel	•	•	•		•	•	•		•	•					
.AlarmClass														•	
.AlarmComment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
.AlarmDate														•	
.AlarmDev		•	•			•	•			•		•			
.AlarmDevCount		•	•			•	•			•		•			
.AlarmDevDeadband		•	•			•				•					
.AlarmDevUnAckCount		•	•			•	•			•		•			

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームタグ	履歴トレンド	分散アラームタグ/ヒストリカルアラーム	
.AlarmDisabled	•	•	•		•	•	•		•	•		•			
.AlarmDsc	•				•				•			•			
.AlarmDscCount	•				•				•			•			
.AlarmDscDisabled	•				•				•			•			
.AlarmDscEnabled	•				•				•			•			
.AlarmDscInhibitor ど	•				•				•			•			
.AlarmDscUnAckCount	•				•				•			•			
.AlarmEnabled	•	•	•		•	•	•		•	•		•			
.AlarmGroup														•	
.AlarmGroupSel														•	
.AlarmHiDisabled		•	•			•	•			•					
.AlarmHiEnabled		•	•			•	•			•					
.AlarmHiHiDisabled		•	•			•	•			•					
.AlarmHiHiEnabled		•	•			•	•			•					
.AlarmHiHiInhibitor		•	•			•	•			•					
.AlarmHiInhibitor ど		•	•			•	•			•					
.AlarmLimit														•	
.AlarmLoDisabled		•	•			•	•			•					
.AlarmLoEnabled		•	•			•	•			•					
.AlarmLoInhibitor		•	•			•	•								
.AlarmLoLoDisabled		•	•			•	•			•					

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ドットフィールド	タグ タイプ													
	メモリ型				I/O 型				間接型			その他		
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームグループ	履歴トレンド	分散アラーム/エクスプレッションコントロール
.AlarmLoLoEnabled		•	•			•	•			•				
.AlarmLoLoInhibitor		•	•			•	•			•				
.AlarmMajDevDisabled		•	•			•	•			•				
.AlarmMajDevEnabled		•	•			•	•			•				
.AlarmMajDevInhibitor		•	•			•	•			•				
.AlarmMinDevDisabled		•	•			•	•			•				
.AlarmMinDevEnabled		•	•			•	•			•				
.AlarmMinDevInhibitor		•	•			•	•			•				
.AlarmName														•
.AlarmOprName														•
.AlarmOprNode														•
.AlarmPri														•
.AlarmProv														•
.AlarmROC		•	•			•	•			•				
.AlarmROCCount		•	•			•	•			•				
.AlarmROCDisabled		•	•			•	•			•				
.AlarmROCEnabled		•	•			•	•			•				
.AlarmROCInhibitor		•	•			•	•			•				

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ドットフィールド	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームタグ	履歴トレンド	分散アラームタグ/エクスプロレーション	
.AlarmROCUnAckCount		•	•			•	•			•					
.AlarmState														•	
.AlarmTime														•	
.AlarmTotalCount	•	•	•		•	•	•		•	•		•			
.AlarmType														•	
.AlarmUnAckCount	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum1	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum1Set	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum2	•	•	•		•		•		•	•		•			
.AlarmUserDefNum2	•	•	•		•	•	•		•	•		•			
.AlarmUserDefStr	•	•	•		•	•	•		•	•		•			
.AlarmUserDefStrSet	•	•	•		•	•	•		•			•			
.AlarmValDeadban		•	•			•	•			•					
.AlarmValue			•				•			•		•		•	
.AlarmValueCount		•	•			•	•			•		•			
.AlarmValueUnAckCount		•	•			•	•			•		•			
.Caption														•	
.ChartLength													•		

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ドットフィールド	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームタグ型	履歴トレンド	分散アラーム/タイムベースロール	
.ChartStart													•		
.Comment.Comment	•	•	•	•	•	•	•	•	•	•	•	•	•		•
.DevTarget		•	•			•	•			•					
.DisplayMode													•		
.Enabled														•	
.EngUnits		•	•			•	•			•					
.Freeze														•	
.HiHiLimit		•	•			•	•			•					
.HiHiSet		•	•			•	•			•					
.HiHiStatus		•	•			•	•			•					
.HiLimit		•	•			•	•			•					
.HiSet		•	•			•	•			•					
.HiStatus		•	•			•	•			•					
.[ListChanged]														•	
.ListCount														•	
.ListIndex														•	
.LoLimit		•	•			•	•			•					
.LoLoLimit		•	•			•	•			•					
.LoLoSet		•	•			•	•			•					
.LoLoStatus		•	•			•	•			•					
.[LoSet]		•	•			•	•			•					
.LoStatus		•	•			•	•			•					

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ドットフィールド	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームタグ	履歴トレンド	分散アラーム/タイムベースアラーム	
.MajorDevPct		•	•			•	•			•					
.MajorDevSet		•	•			•	•			•					
.MajorDevStatus		•	•			•	•			•					
.MaxEU		•	•			•	•			•					
.MaxRange													•		
.MaxRaw		•	•			•	•								
.MinEU		•	•			•	•			•					
.MinorDevPct		•	•			•	•			•					
.MinorDevSet		•	•			•	•			•					
.MinorDevStatus		•	•			•	•			•					
.MinRange													•		
.MinRaw		•	•			•	•			•					
.Name	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
.NewIndex														•	
.NextPage														•	
.Normal	•	•	•		•	•	•		•	•		•			
.NumAlarms														•	
.OffMsg	•				•				•						
.OnMsg	•				•				•						
.PageNum														•	
.Pen1 through .Pen8													•		
.PendingUpdates														•	

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ドットフィールド	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームタグ	履歴トレンド	分散アラームタグ/コントロール	
.PrevPage														•	
.[PriFrom]														•	
.PriTo														•	
.Quality	•	•	•	•	•	•	•	•	•	•	•				
.QualityLimit	•	•	•	•	•	•	•	•	•	•	•				
.QualityLimitString	•	•	•	•	•	•	•	•	•	•	•				
.QualityStatus	•	•	•	•	•	•	•	•	•	•	•				
.QualityStatusString	•	•	•	•	•	•	•	•	•	•	•				
.QualitySubstatus	•	•	•	•	•	•	•	•	•	•	•				
.QualitySubstatusString	•	•	•	•	•	•	•	•	•	•	•				
.QueryState														•	
.QueryType														•	
.RawValue	•	•	•		•	•	•		•	•		•			
.ReadOnly														•	
.Reference	•	•	•	•	•	•	•	•	•	•	•				
.ReferenceComplete	•	•	•	•	•	•	•	•	•	•	•				
.ROCPct		•	•			•	•			•					
.ROCSet		•	•			•	•								
.ROCStatus		•	•			•	•			•					
.ScooterLockLeft													•		
.ScooterLockRight													•		
.ScooterPosLeft													•		

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ドットフィールド	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームメッセージ	履歴トレンド	分散アラームメッセージ/ヒストリカルアラーム	
.ScooterPosRight													•		
.Successful														•	
.SuppressRetain														•	
.TagID	•	•	•		•	•	•		•	•					
.TimeDate(時間 日付)	•	•	•	•	•	•	•	•	•	•	•				
.TimeDateStringTimeDateString	•	•	•	•	•	•	•	•	•	•	•				
.TimeDateTime(時間 日付・時刻)	•	•	•	•	•	•	•	•	•	•	•				
.TimeDay(時間 日)	•	•	•	•	•	•	•	•	•	•	•				
.TimeHour(時間 時)	•	•	•	•	•	•	•	•	•	•	•				
.TimeMinute(時間 分)	•	•	•	•	•	•	•	•	•	•	•				
.TimeMonth(時間モンス)	•	•	•	•	•	•	•	•	•	•	•				
.TimeMsec(時間 ミリ秒)	•	•	•	•	•	•	•	•	•	•	•				
.時間秒	•	•	•	•	•	•	•	•	•	•	•				
TimeTime(時間 時刻)	•	•	•	•	•	•	•	•	•	•	•				
.TimeTimeString (時間 文字列)	•	•	•	•	•	•	•	•	•	•	•				
.TimeYear(時間 年)	•	•	•	•	•	•	•	•	•	•	•				
.TopIndex														•	

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ドットフィールド	タグ タイプ														
	メモリ型				I/O 型				間接型			その他			
	論理型	整数型	実数型	メッセージ型	論理型	整数型	実数型	メッセージ型	論理型	アナログ型	メッセージ型	アラームグループ	履歴トレンド	分散アラームオブジェクト/コントロール	
.TotalPages(合計ページ数)														•	
未確認	•	•	•		•	•	•		•	•		•			
.UpdateCount(更新数)													•		
.UpdateInProgress(プロセスが実行中ですか)													•		
.UpdateTrend(更新トレンド)													•		
.Value（タグ名）	•	•	•	•	•	•	•	•	•	•	•	•			•
.Value（ウィンドウコントロール）														•	
.表示														•	

ドットフィールドは、目的とされる機能によって分類できます。ドットフィールドの機能分類に関する詳細については、後続のセクションを参照してください。

カテゴリ	参照先
値としきい値	タグ変数のしきい値の変更。
アラーム パラメータ	『AVEVA™ InTouch HMI アプリケーション ランタイム』の「ランタイム時のタグとグループのアラーム プロパティの制御」を参照してください。
I/O 型	「I/O を使用したデータ アクセス」
分散アラーム オブジェクト	*分散アラーム表示オブジェクトの操作*。
トレンド表示	タグ データのトレンド。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ	参照先
ウィンドウ コントロール	ウィザード

タグ変数のしきい値の変更

I/O サーバーからの生入力データは、InTouch アプリケーションに応じた値の範囲に変換されます。タグ変数の値は、タグ変数ディクショナリのプロパティの **生データ最小値**と**生データ最大値**で指定される最小値と最大値の範囲にクランプされます。

その後、これらの生データ値は、［**工学値最小値**］オプションと［**工学値最大値**］ オプションで設定された工学値範囲に変換されます。一連のドットフィールドを使用することにより、タグ変数の生データ値と工学単位範囲を監視して、変更できます。

以下の表は、アプリケーションの実行中にタグ変数の値を監視したり、変更したりするドットフィールドを一覧表示しています。

ドットフィールド	読み取り/書き込み	内容
.MinRaw	書き込み禁止	I/O サーバーから受け取った生データの下限クランプの設定
.MaxRaw	書き込み禁止	I/O サーバーから受け取った生データの上限クランプの設定
.MinEU	書き込み禁止	タグ変数に割り当てられた工学単位での最小値
.MaxEU	書き込み禁止	タグ変数に割り当てられた工学単位での最大値
.工学単位	読み取り/書き込み	タグ変数ディクショナリの**工学単位**オプションでアナログ型タグ変数に割り当てられたテキスト値
.生値	書き込み禁止	スケーリングが適用される前に I/O サーバーからタグ変数が受け取った実際の論理値かアナログ値
.価値	読み取り/書き込み	現在のタグ変数の値
.OnMsg	読み取り/書き込み	論理型タグ変数の値が True、On、または 1 を評価するときに割り当てられるメッセージ
.OffMsg	読み取り/書き込み	論理型タグ変数の値が False、Off、または 0 を評価するときに割り当てられるメッセージ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	読み取り/書き込み	内容
.Comment.Comment	読み取り/書き込み	タグ変数ディクショナリで指定されたタグ変数コメント

生値のしきい値の表示

I/O サーバーからの入力タグ変数の生データ値は、InTouch アプリケーションで使用する前にクランピングする必要があります。クランピングによって、定義済みの下限値と上限値の範囲に生データの値が制限されます。The.**MinRaw** および. および**.MaxRaw** ドットフィールドによって、生データの入力範囲の上限値および下限値が示されます。

.MinRaw ドットフィールド

.**MinRaw** ドットフィールドは、タグ変数に割り当てられた生データ最小値クランプ設定を示しています。.**MinRaw** ドットフィールドの値は、タグ変数ディクショナリで I/O タグ変数に対して割り当てられた**生データ最小値**です。この設定より小さい生データ値は、この最小値にクランプされます。

カテゴリ

タグ

使用法

```
tag_name.MinRaw;
```

パラメータ

Tag_name
任意の I/O 整数型、I/O 実数型、および間接アナログ型のタグ変数の名前。

備考

読み取り専用のドットフィールドで、生データ最小値の下限クランプ設定に割り当てられた値を示しています。

データタイプ

実数型または整数型（読み取り専用）

有効値

任意のアナログ値

例

以下のスクリプトは、ポンプの入力圧に関連付けられた生データ値が、このタグ変数の**生データ最小値**プロパティと**生データ最大値**プロパティで設定された上下限値の範囲外の場合にエラー ウィンドウを表示します。

```
IF ((PumpInP.RawValue > PumpInP.MaxRaw) OR
   (PumpInP.RawValue < PumpInP.MinRaw)) THEN
   Show "Instrument Failure Window";
ENDIF;
```

参照項目

.EngUnits、**.MinEU**、**.MaxEU**、**.MaxRaw**、**.RawValue**

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.MaxRaw ドットフィールド

.MaxRaw ドットフィールドは、タグ変数ディクショナリの**生データ最大値**プロパティによって I/O タグ変数に割り当てられた、生データ最大値の上限クランプを示します。この設定を超える生データ値は、この生データ最大値にクランプされます。

カテゴリ

タグ

使用法

```
Tag_name.MaxRaw
```

パラメータ

Tag_name
任意の I/O 整数型、I/O 実数型、および間接アナログ型のタグ変数の名前。

備考

読み取り専用のドットフィールドで、生データ最大値の上限クランプ設定に割り当てられた値を示しています。

データタイプ

実数型または整数型（読み取り専用）

有効値

任意のアナログ値

例

このスクリプトは、タグ変数の値が通常の動作範囲外であるかを判定し、範囲外と判定された場合にウィンドウを表示します。

```
IF ((Temp01.RawValue > Temp01.MaxRaw) OR (Temp01.RawValue <
   Temp01.MinRaw))THEN
   Show "Instrument Failure Window";
ENDIF;
```

参照項目

.EngUnits、**.MinEU**、**.MaxEU**、**.MinRaw**、**.RawValue**

タグ変数の生値の表示

The. .**RawValue** ドットフィールドは、I/O サーバーから受け取った監視されたプロパティの実際の論理値またはアナログ値を示します。この生データ値は、クランピングやスケーリングが適用されてタグ変数の工学単位に値が正規化される前の実際の入力値です。

.RawValue ドットフィールド

.RawValue ドットフィールドは、WindowViewer が I/O サーバーから 受け取った実際の値を示しています。.RawValue ドットフィールドによって、InTouch がスケーリングを適用する前の I/O タグ変数の値にアクセスできます。

カテゴリ

タグ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

```
Tag_name.RawValue
```

パラメータ

Tag_name
任意の I/O 整数型、I/O 実数型、I/O 論理型、間接論理型、および間接アナログ型のタグ変数の名前。

備考

この読み取り専用ドットフィールドは、InTouch がスケーリングを適用する前の実際の論理値またはアナログ I/O 値を表示するために使用されます。

データタイプ

.RawValue ドットフィールドに関連付けられているタグ変数のタイプに適したすべてのデータ。たとえば、実数型タグ変数には実数、論理型タグ変数には論理値など（読み取り専用）。

例

以下のスクリプトは、ポンプの入力圧の生データ値がタグ変数の最小値と最大値によるクランプ制限外になった場合に警告メッセージを発行します。

```
IF ((PumpInP.RawValue > PumpInP.MaxRaw) OR (PumpInP.RawValue < PumpInP.MinRaw)) THEN
   AlarmMessage = "Pump sensor is out of calibration or requires replacement.";
ENDIF;
```

参照項目

.EngUnits、**.MinEU**、**.MaxEU**、**.MinRaw**、**.MaxRaw**

工学値のしきい値の表示

I/O サーバーから WindowViewer で受け取った値は最初は生データと見なされます。生データ値にはスケーリングが必要な場合があります。InTouch HMI は、クランプされた生の入力データ値に対して演算による変換を行い、タグ変数の工学値範囲にスケールします。I/O 整数型と実数型のタグ変数には、工学値範囲の上下限値を示す**工学値最小値**プロパティと**工学値最大値**プロパティが用意されています。

.MaxEU ドットフィールド

.MaxEU ドットフィールドは、タグ変数ディクショナリで特定のタグ変数に対して割り当てられた工学単位の最大値を示しています。

カテゴリ

タグ

使用法

```
Tag_name.MaxEU
```

パラメータ

Tag_name
任意の整数型、実数型、または間接アナログ型のタグ変数。

備考

.MaxEU ドットフィールドは、タグ変数に対して定義された工学単位の範囲に生データ値をスケールするために使用されます。このドットフィールドにより、工学値範囲の上限値が定義されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データタイプ

実数型タグ変数には実数型、整数型タグ変数には整数型（読み取り専用）

有効値

指定したタグ変数のタイプにより異なります。

例

フィールドのプログラマブル ロジック コントローラ（PLC）によって、レベル ゲージが読み取られます。レベル送信機は、4～20 mA の範囲の信号を送信します。PLC はこの信号を 0 ～ 4095 までの整数値に変換します。この値が TankTwoLevel タグ変数に割り当てられます。

生データ値（0 ～ 4095）を表示しても、オペレータに対して役立つデータは提供されません。したがって、この値を対応する工学単位の範囲にスケールする必要があります。

これを達成するには、最小工学単位値フィールドと最大工学単位値フィールドを正確に設定する必要があります。この例では、生データ値 0（フィールドからは 4 mA）を「0 ガロン」に、値 4095（フィールドからは 20 mA）を「100 ガロン」に変換する場合、画面に正確な値を表示するには以下の設定が必要です。

```
TankTwoLevel.MinRaw = 0;
TankTwoLevel.MaxRaw = 4095;
TankTwoLevel.MinEU = 0;
TankTwoLevel.MaxEU = 100;
```

この設定では、フィールドの生データが 4095 の場合に、オペレータに対して表示される値は 100 となります。

参照項目

.EngUnits、**.MinEU**、**.MinRaw**、**.MaxRaw**、**.RawValue**

.MinEU ドットフィールド

.MinEU ドットフィールドは、タグ変数ディクショナリから特定のタグ変数に対して割り当てられた工学単位の最小値を示しています。

カテゴリ

タグ

使用法

```
Tag_name.MinEU
```

パラメータ

Tag_name
任意の整数型、実数型、または間接アナログ型のタグ変数。

備考

.MinEU ドットフィールドは、タグ変数に対して定義された工学単位の範囲に生データ値をスケールするために使用されます。このドットフィールドにより、工学値範囲の下限値が定義されます。

データタイプ

実数型タグ変数には実数型、整数型タグ変数には整数型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

指定したタグ変数のタイプにより異なります。

例

この例では、Tag1 タグ変数に対して定義された工学単位の範囲が、**AbsoluteTagRange** タグ変数に割り当てられます。

```
AbsoluteTagRange = (Tag1.MaxEU - Tag1.MinEU);
```

参照項目

.EngUnits、**.MaxEU**、**.MinRaw**、**.MaxRaw**、**.RawValue**

タグ変数の工学単位の変更

タグ変数にドットフィールドを関連付けて、タグ変数に割り当てられた工学単位のテキスト値を決定できます。

.EngUnits ドットフィールド

.**EngUnits** ドットフィールドは、**工学単位**プロパティでアナログ型タグ変数に割り当てられたテキスト値を示しています。.**EngUnits** ドットフィールドは、工学単位をテキスト値として示しています。

注: このフィールドに書き込まれた値は保持されません。

カテゴリ

タグ

使用法

```
Tag_name.EngUnits
```

パラメータ

Tag_name
任意の整数型、実数型、または間接アナログ型のタグ変数。

データ タイプ

メッセージ型（読み取り/書き込み）。

備考

.**EngUnits** ドットフィールドは、タグ変数に関連付けられた実際のデータのスケール、変換、または形式には影響を与えません。

有効値

0 ～ 31 文字を含む任意の文字列

例

以下のスクリプトは、タグ変数の工学単位が摂氏の場合に、華氏温度への変換関数を呼び出します。

```
IF Temperature.EngUnits == "Celsius" THEN
   CALL TempFConvert(Temperature);
ENDIF;
```

参照項目

.MinEU、**.MaxEU**、**.MinRaw**、**.MaxRaw**、**.RawValue**

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

工学単位でのタグ変数値の表示

アプリケーション内で明示的にドットフィールドが指定されていない場合、InTouch タグ変数にはデフォルトのドットフィールドが割り当てられます。

.Value ドットフィールド

.**Value** ドットフィールドは、指定したタグ変数の現在の値を工学単位で示しています。.**Value** ドットフィールドは、すべてのタグ変数に自動的に適用されるデフォルトの InTouch ドットフィールドです。タグ変数にドットフィールドの指定がない場合は、**.Value** ドットフィールドが使用されます。

カテゴリ

タグ

使用法

```
tag_name.Value
```

パラメータ

tag_name
履歴トレンド以外の任意のタイプのタグ変数

備考

.Value ドットフィールドの使用が必要になることはほとんどありません。ただし、場合によっては、このドットフィールドによって計算やパラメータの使用法が明確になることがあります。

データ タイプ

指定したタグ変数のタイプと同じタイプ（読み取り／書き込み）

有効値

指定したタグ変数のタイプにより異なります。

例

次のステートメントにより、メモリ整数型タグ変数 **PumpRPM** の値が 100 に設定されます。

```
PumpRPM.Value=100;
```

以下のように指定しても機能は同じです。

```
PumpRPM=100;
```

論理型タグ変数のメッセージの表示または変更

The.**OnMsg** と . と **.OffMsg ドットフィールド**は、論理型タグ変数のオンとオフの状態に対してタグ変数ディクショナリで指定したメッセージを示します。論理型タグ変数のオンとオフのメッセージは、15 文字以内の短い文字列です。

.OnMsg ドットフィールド

.**OnMsg** ドットフィールドを使用すると、タグ変数ディクショナリで論理型タグ変数に割り当てられたオン メッセージにアクセスできます。

カテゴリ

タグ

使用法

```
Tag_name.OnMsg
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

Tag_name
任意の論理型タグ変数

データタイプ

メッセージ型（読み取り／書き込み）。このドットフィールドに書き込まれた値は保持されません。

有効値

0 ～ 15 文字を含む任意の文字列

例

以下のステートメントは、間接型タグ変数 Ind**PumpState** のオン メッセージに「Pump1 running」という文字列が指定された場合にメッセージを発行します。

```
IF IndPumpState.OnMsg == "Pump1 running" THEN
   TypeOfTag = "The IndPumpState tag is assigned to Pump1.";
ENDIF;
```

参照項目

.OffMsg

.OffMsg ドットフィールド

.**OffMsg** ドットフィールドを使用すると、タグ変数ディクショナリで論理型タグ変数に割り当てられたオフ メッセージにアクセスできます。

カテゴリ

タグ

使用法

```
Tag_name.OffMsg
```

パラメータ

Tag_name
任意の論理型タグ変数

データタイプ

メッセージ型（読み取り／書き込み）。このドットフィールドに書き込まれた値は保持されません。

有効値

0 ～ 15 文字を含む任意の文字列

例

以下のステートメントでは、**MyDiscrete** タグ変数の状態に基づいて、**StateMessage** タグ変数に対して適切な文字列が割り当てられます。

```
StateMessage=Dtext (MyDiscrete, MyDiscrete.OnMsg, MyDiscrete.OffMsg);
```

参照項目

.OnMsg

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ変数のコメントの表示または変更

タグ変数コメントは、タグ変数ディクショナリでのみ永久的に変更できます。「 」を使用して別のコメントを割り当てることができます。アプリケーションの実行中にドットフィールドに**コメント**を付けます。このコメント変更が有効なのは、ランタイム セッション中だけです。タグ変数ディクショナリのタグ変数のコメントを永久的に変更することはできません。WindowViewer がシャットダウンされて再起動されると、元のコメントがタグ変数に割り当てられます。

.Comment ドットフィールド

.**Comment** ドットフィールドは、タグ変数ディクショナリでタグ変数に割り当てられたコメントを示しています。タグ コメントは、160 文字以内の文字列です。

カテゴリ

タグ

使用法

```
Tag_name.Comment
```

パラメータ

Tag_name
任意のタグ変数

備考

タグ変数に関連付けられたコメントは、InTouch アプリケーションの実行中に .Comment ドットフィールドを使用して変更できます。アプリケーションが停止されると、タグ変数ディクショナリで指定された元のコメントがタグ変数に割り当てられたまま残ります。

データ タイプ

メッセージ型

有効値

1 ～ 160 の文字を含む任意の文字列

例

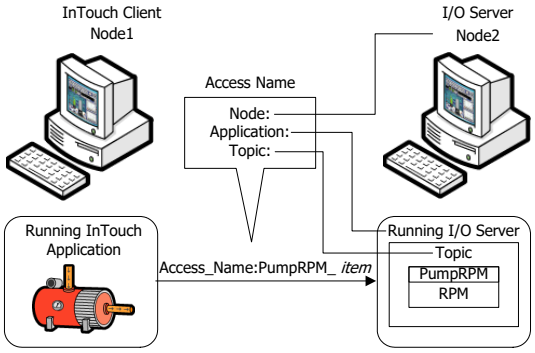
以下のステートメントでは、タグ変数に割り当てられたコメントをタグ変数の名前に組み合わせることによって、オペレータ メッセージが作成されます。

```
OperatorMessage=PumpRPM.Name + " has a comment of: " + PumpRPM.Comment;
```

I/O を使用したデータ アクセス

分散アプリケーションを開発して、InTouch システムの複数の機能コンポーネントを別のノード上に配置することができます。次の図に、別のノードに保存されているデータへの I/O リクエストの設定方法を示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3 部構成のアドレス指定規則を使用して、別のノードに保存されているデータの要素を識別するように InTouch アプリケーションを設定できます。この形式には、ノード名、アプリケーション名、トピック名が含まれています。リモート ノードからデータを取得するには、これらの 3 アイテムを指定した InTouch アプリケーションに対してアクセス名を設定する必要があります。

たとえば、別のノードで稼動しているリモート I/O サーバーのデータにアクセスする場合、アクセス名は以下の要素で構成されます。

アクセス名オプション	説明
ノード名	I/O サーバー プログラムを実行しているコンピュータのノード名です。
アプリケーション名	ノードで実行されている I/O サーバー プログラム名です。 たとえば、"UA_SampleServer" は OPC UA サーバーの名前である可能性があります。 Operations Integration（OI）サーバーに関連付けられたアプリケーション名の詳細については、OI サーバーのマニュアルを参照してください。
トピック名	I/O サーバー デバイス グループに割り当てられるラベルです。

InTouch データ ソースに Excel のスプレッドシートを使用する場合は、アクセス名を以下のように定義できます。

アクセス名オプション	説明
ノード名	Excel プログラムを実行しているコンピュータのノード名です。
アプリケーション名	アプリケーション名に Excel を指定します。
トピック名	要求データが格納されている Excel のブック名とスプレッドシート名です。たとえば、[Book1]Sheet1 と指定します。

ノード、アプリケーション、トピック、アイテムに加えて、リモート ノードにあるデータのタイプを指定する必要があります。この情報に基づいて、タグ名ディクショナリで定義されるタグの I/O タイプが決定されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Gateway Communications Driver の操作

Gateway Communications Driver は、ポータルおよびプロトコル コンバータとして機能し、2 つのコンピュータ システムまたはプログラムが相互に通信できるようにするアプリケーションです。OI Gateway を使用すれば、OPC などの異なるプロトコルを使用して通信するクライアントとデータ ソースをリンクさせることができます。InTouch から Gateway Communication Driver にアクセスしてフィールド デバイスを接続します。

InTouch は Gateway Communication Driver を活用して、オートメーションと制御アプリケーションの間、フィールド システムとデバイスの間、およびビジネスとオフィス アプリケーションの間の通信に OPC、OPC UA、およびその他のプロトコルを使用します。

Gateway Communication Driver は WindowViewer にバンドルされており、InTouch をインストールする際に自動的にインストールされます。スタンドアロン アプリケーションとしてインストールすることもできます。

Gateway Communication Driver について

Gateway Communication Driver は通信プロトコル コンバータとして機能します。Gateway Communication Driver を使用して、複数の異なるプロトコルを使用して通信するクライアントとデータ ソースをリンクさせることができます。

このユーザー支援マニュアルでは、Gateway Communication Driver コンポーネントの設定と実行に必要な情報のみが記載されています。その操作については、関連コンポーネントに付属するマニュアルを参照してください。

Operations Control Management Console（OCMC）へのスナップインである Log Viewer を使用して、Gateway Communication Driver の問題のトラブルシューティングを行うことができます。以下の詳細については、Log Viewer ヘルプを参照してください。

- エラー メッセージの表示
- 表示するメッセージの決定
- エラー メッセージのブックマーク設定

InTouch HMI ソフトウェアなどのクライアント アプリケーションを使用して問題のトラブルシューティングを行うこともできます。クライアント アプリケーションは、システム デバイス アイテムを使用して、ノードのステータスおよび一部のパラメータの値を特定できます。

Gateway Communication Driver の基本ルールを以下に示します。

- ノードあたり Gateway Communication Driver の 1 つのインスタンスを実行できます。
- Gateway は、OI サーバー マネージャ スナップインを使用してアクティブおよび非アクティブにすることができます。
- Gateway は、標準の COM アクティブ化メカニズムを使用して、COM サーバー（OPC サーバー）としてアクティブ化することができます。
- Gateway は OPC クライアント内で out-of-proc としてのみ実行できます。
- Gateway が通信できるのは、Application Server v2.0 以降で提供される ArchestrA データ ソース コンポーネントだけです。以前のバージョンの Application Server はサポートされません。

Gateway Communication Driver の使用を開始する

以下の手順は、Gateway Communication Driver の使用を開始するためのワークフローを説明します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

1. InTouch をインストールします。
   - Gateway Communication Driver は InTouch インストールの一部として含まれています。
   - OPC サーバーと OPC UA サーバーは別にインストールする必要があります。
2. OPC サーバーまたは OPU UA サーバーをインストールし、フィールド デバイスとの通信を設定します。
3. 必要に応じて、その他のプロトコルを設定します。
4. InTouch でアプリケーションを作成します。
5. Gateway Communication Driver を設定して、インストール済みの OPC サーバー、OPC UA サーバー、またはその他のデータ ソースをポイントします。
6. Gateway Communications Driver をアクティブ化します。

   Gateway Communication Driver のセットアップの詳細については、GatewayCommunications Driver ヘルプの「*初めての Gateway Communication Driver の設定*」を参照してください。

サポートされている InTouch 通信プロトコル

InTouch HMI を設定することにより、DDE か SuiteLink を使用することができます。InTouch HMI では、メッセージ交換通信プロトコルもサポートされています。

InTouch アプリケーション内での Message Exchange の使用の詳細については、「InTouch から Application Server データへのアクセス」を参照してください。

動的データ交換

動的データ交換（DDE）通信プロトコルによって、Windows アプリケーション間での通信が可能になります。DDE は、同時に実行されている 2 つのアプリケーション間に、クライアント／サーバー関係を構築します。サーバー アプリケーションはデータを提供し、このデータを必要とする他のアプリケーションからの要求を受け付けます。要求元のアプリケーションは、クライアントと呼ばれます。InTouch アプリケーションは、クライアントにもサーバーにも同時になることができます。

SuiteLink

SuiteLink は、産業用アプリケーション向けに特に設計された TCP/IP ベースのプロトコルです。SuiteLink は、データの整合性、高スループット、およびシンプルな診断手順を提供します。SuiteLink プロトコルをサポートしているのは、Microsoft Windows NT 4.0 以降です。

SuiteLink は DDE または NetDDE に取って代わるものではありません。クライアントとサーバー間の接続はそれぞれ、ご使用のネットワーク要件によって異なります。

SuiteLink では、以下のことが行えます。

- VTQ（Value Time Quality）は、VTQ 対応のクライアントに届けられるデータ値のすべてに時間の目盛りと品質表示を挿入します。
- データのスループット、サーバーのロード、コンピュータ リソースの消費、およびネットワークの転送に対する包括的な診断を、Microsoft Windows オペレーティング システムのパフォーマンス モニタから行うことができます。
- アプリケーションが 1 つのノード上にある場合や、数多くのノードに分散されている場合にかかわらず、アプリケーション間で一貫した大容量のデータを維持できます。
- ネットワーク転送プロトコルは、Microsoft の標準 Winsock インターフェイスを使った TCP/IP です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- TLS 1.2 プロトコルを使用して、Suitelink ノードを安全に設定できます。

標準ユーザーとしてのセキュアな Suitelink 通信の設定

TLS 1.2 プロトコルによるセキュアな Suitelink 接続では、クライアントとサーバーの間の通信チャネルが暗号化されます。リモート データにアクセスするために暗号化された SuiteLink を使用する場合、コンフィグレータから System Management Server（SMS）を設定する必要があります。暗号化は、SMS によって提供される証明書を介して行われます。SuiteLink サーバーとクライアントで暗号化された通信を使用するには、SuiteLink 3.0 をインストールする際に ASB ランタイム コンポーネント機能を選択する必要があります。

セキュアな SuiteLink（V3）の導入以降、SuiteLink はフォールバックを提供しているので、SuiteLink サーバーは、セキュア（V3、暗号化）または非セキュア（V2、非暗号化）モードの受信接続を受け入れることができます。System Platform 2023 リリースから、V2 接続を受け入れるフォールバック モードはデフォルトで無効化されています。フォールバック モードを有効にして非セキュアな接続を受け入れる場合、System Management Server で設定を変更する必要があります。SuiteLink 接続のモード (セキュアまたは非セキュア) は、Log Viewer で WWSLS コンポーネントをチェックすることによって確認できます。

注：System Platform 2023 の時点で、SuiteLink 接続はデフォルトで TLS 暗号化によって保護されます。非セキュア フォールバック モードが有効化されておらず、セキュアな接続の要件が満たされていない場合、SuiteLink 接続は失敗します。詳細については、「コンフィグレータでの System Management Server の設定」を参照してください。

レジストリ エディタを使用してフェールバック（非セキュア）Suitelink 接続を有効化または無効化するには

1. レジストリ エディタを開きます。
2. パス[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ArchestrA\WebApplications\Default\SuiteLink] に移動します。
3. 次のエントリを見つけます。

   "V2Server"=dword:00000001

   "V3Server"=dword:00000001
4. フォールバック モードを有効にするには、V2Server の値を 1 に設定します。
5. フォールバック モードを無効にするには、V2Server の値を 0 に設定します。
6. コンピュータを再起動します。

コンフィグレータでの System Management Server の設定

System Management Server（SMS）は、System Platform 共通プラットフォーム サービスの一部で、System Platform 2023 の重要なセキュリティ対策を実装するために使用されます。以下の内容が含まれます。

- ノード間通信のポート番号の設定。
- 暗号化された（セキュアな）通信だけを使用するか、セキュア（TLS）通信を確立できない場合に暗号化されていない通信を許可するよう SuiteLink 接続を介した通信を設定することができます。SuiteLink は、System Platform のさまざまな多くのアプリケーションで使用されます。SuiteLink セキュリティの設定の詳細については、［詳細設定］ボタンを選択して［通信］タブを参照してください。
- 証明書の管理。
- 外部 ID プロバイダを介したシングル サインオン（SSO）が許可される OpenID Connect 標準によるユーザー認証。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

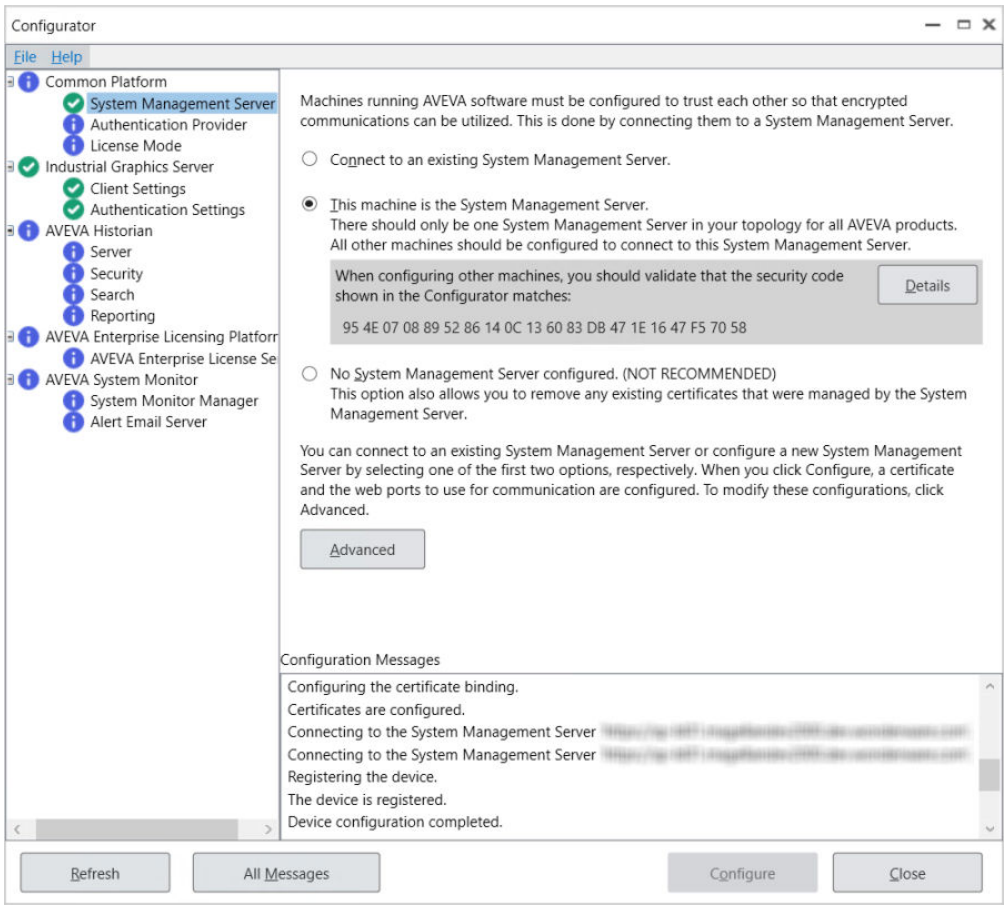
セキュリティを有効にするには、各 System Platform ノードが System Management Server と通信する必要があります。System Platform トポロジ内の System Management Server は 1 つだけである必要があります。そうでない場合、通信途絶が発生する可能性があります。System Management Server は、共有セキュリティ証明書を格納し、複数のマシン間で信頼関係を確立します。1 つの追加ノードを冗長 SSO サーバーとして設定できます。この冗長 SSO サーバーは、System Management Server にアクセスできない場合のシングル サインオンのバックアップとして機能します。

System Platform 2017 Update 3 以降にアップグレードされていないノードがある場合、それらのノードとの通信には非セキュアな通信を使用する必要があることがあります。ただし、System Management Server と通信するよう設定されている場合、System Platform 2017 Update 3 以降を実行するノード間の通信は暗号化されます。

System Management Server の設定:

1. コンフィグレータで［**共通プラットフォーム**］を展開して［**System Management Server**］を選択します。

注: System Management Server のユーザー資格情報の入力が求められた場合、**DomainName\UserName** の形式でユーザー名を入力します。ドメイン管理特権があってもローカル マシンの管理者でない場合は、ユーザー資格情報が要求されることがあります。System Management Server を設定するには、**Administrators** または **aaAdministrators** OS グループのメンバーである必要があります。詳細については、AVEVA System Platform ヘルプの「System Management Server を設定するためのユーザー資格情報」を参照してください。

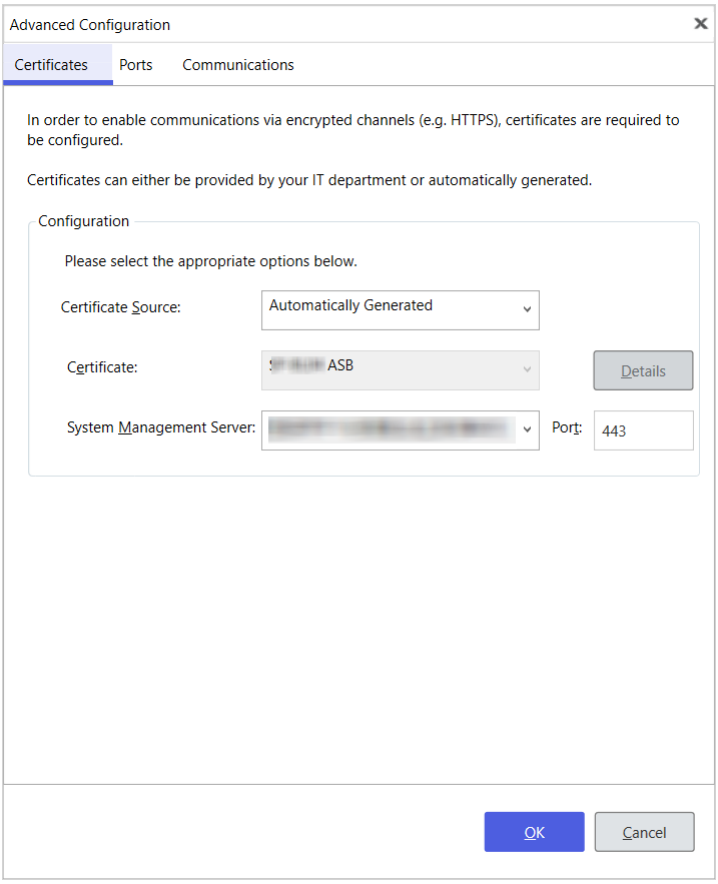


© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注：インストールが完了するとコンフィグレータが自動的に起動します。コンフィグレータは、任意の System Platform ノードの Windows［スタート］メニューから起動することもできます。

2. 以下の 3 つのオプションが提示されます。
   - **既存の System Management Server に接続:**これがデフォルトのオプションです。System Platform 探索サービスによってネットワーク上の既存の System Management Server が検索されます。検出された System Management Server はドロップダウン リストに表示されます。使用するサーバーを選択するか、サーバーのマシン名を入力します。System Platform トポロジのすべてのコンピュータは同じサーバーに接続する必要があります。
   - **このコンピュータを System Management Server にする場合、**このオプションを選択します。System Platform トポロジ内の他のすべてのコンピュータで［**既存の System Management Server に接続**］オプションを使用して、このサーバーに接続するように設定する必要があります。
   - **System Management Server を設定しない(非推奨):**暗号化とセキュア通信なしでコンピュータをセットアップする場合、このオプションを選択します。トポロジ内の他のコンピュータは System Management Server を使用するように設定できます。
3. **詳細**ボタンを選択します。

   ［**詳細設定**］ウィンドウが開きます。



   a. ［**証明書**］タブで証明書パラメータを設定します。
      - **証明書ソース: 自動生成** (デフォルト) または **IT によって提供**のいずれかを選択します。IT 部門が証明書を提供している場合、［**インポート**］ボタンをクリックして証明書ファイルを参

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

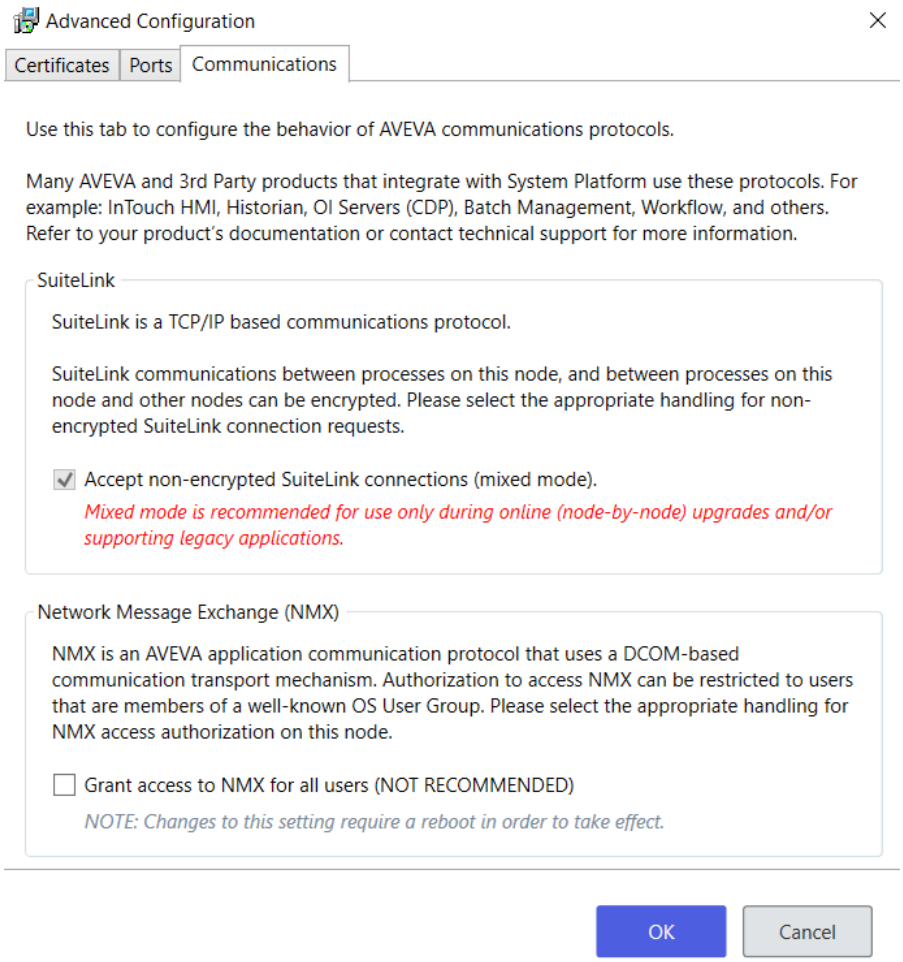
照します。詳細については、『System Platform インストール』の「証明書のインポート」を参照してください。

- **証明書ソース:**証明書名が表示されます。**詳細** を選択すると、インポート済みの証明書が表示されます。証明書は自動更新プロセスを介してサーバー ノードとリモート ノードの両方で定期的に更新されます。
- **System Management Server:**既存の System Management Server に接続する場合、選択したサーバーの名前とポート番号が表示されます。
- **共通プラットフォームのポート:**共通プラットフォームのポートは、特定の AVEVA ソフトウェア（Sentinel System Monitor など）との通信に使用されます。一般的に、デフォルト設定を使用できます。リモート ノードは、以下に示すポート番号で設定する必要があります。
  デフォルトの HTTP ポート:80
  デフォルト HTTPS ポート:443

詳細については、『AVEVA System Platform インストール』の「詳細設定オプション」セクションを参照してください。

b. ［**通信**］タブで通信パラメータを設定します。

- ［**暗号化されていない SuiteLink 接続を受け入れる（混合モード）**］チェック ボックスをオンにして［**OK**］を選択します。混合モードが有効になり、暗号化された接続と暗号化されていない接続の両方が受け入れられるようになります。

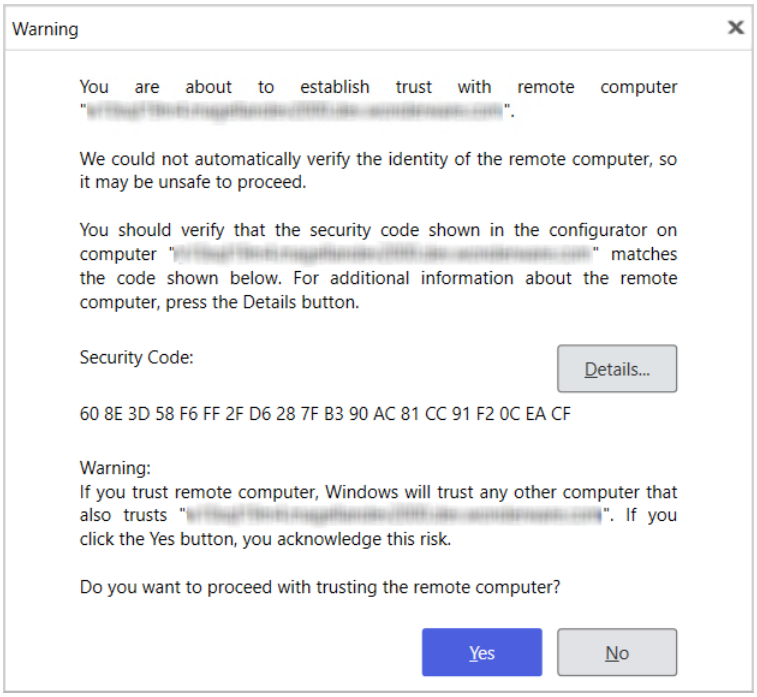


© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注: 暗号化された接続（V3）のみを使用する場合は、**暗号化されていない SuiteLink 接続を受け入れる（混合モード）** チェック ボックスをオフにして **OK** をクリックします。

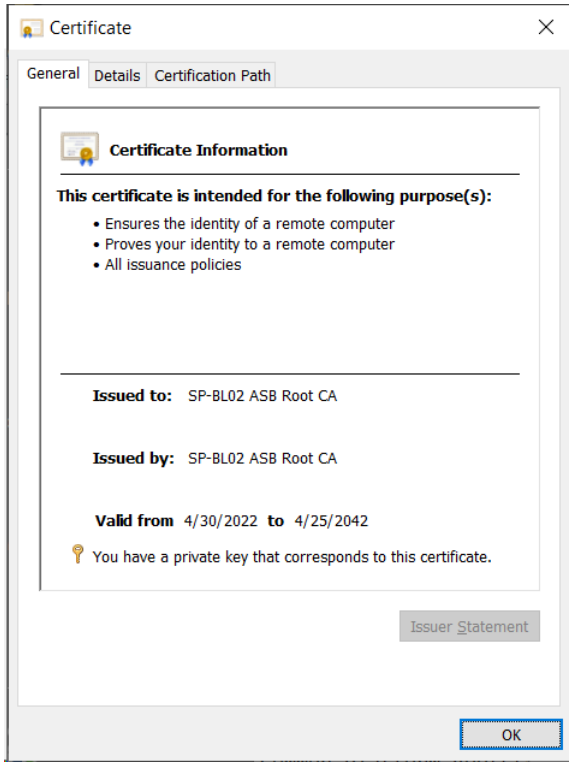
4. **OK** を選択して変更を保存し、SMS 設定ウィンドウに戻ります。
5. **設定**を選択します。

   ［セキュリティ警告］ウィンドウが表示されます。



   マシン間の信頼を確立することによって通信は自由に通過できますが、リモート コンピュータの ID が不明な場合はセキュリティの懸念が生じます。接続先のコンピュータに懸念がある場合、［詳細設定］ダイアログの［**詳細...**］ボタンを選択して証明書を開き、セキュリティ コードと証明書の詳細を確認します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. 設定が必要な次の項目を左ペインで選択します。必要なすべてのアイテムを設定したら［閉じる］ボタンをクリックしてインストールを完了します。『Application Server ユーザー文書』の「設定後のシステムの再起動」を参照してください。

System Management Server 設定の詳細については、『System Platform インストール』の「System Management Server の設定」セクションを参照してください。

標準ユーザーとしてのサーバーへのアクセス

標準ユーザーとしてサーバーにアクセスする場合、セキュアな SuiteLink チャネルを確立することはできません。暗号化されたセキュアな通信ワークフローを確立するために、次のいずれかを行ってください。

- **標準ユーザーをユーザー グループに追加する：**標準ユーザーは、サーバー側の 'ArchestrAWebHosting' ユーザー グループに追加する必要があります。ユーザー グループへのユーザーの追加の詳細については、Windows 固有のマニュアルを参照してください。
- **WindowViewer をサービスとして実行する：**タグ サーバー アーキテクチャ内の InTouch タブ ベースのアプリケーションの場合、タグ サーバー ノードで InTouch WindowViewer をサービスとして実行することによって暗号化されたセキュアな Suitelink 通信を確立できます。
- **管理者権限で WindowViewer を対話モードで実行する：**こうすると、SuiteLink サーバーとクライアント間の通信を暗号化するプライベートキーにアクセスするために必要なユーザー特権を持つ SuiteLink サーバーとして動作する非対話型ユーザーの下で WindowViewer を実行できます。この設定は、対話型ユーザーを使用した場合には使用できません。WindowViewer が実行されているユーザーは、SuiteLink サーバーがセキュアな通信を行うために必要なプライベート キーを取得するために使用されるセキュリティ コンテキストを提供するので、これは必要です。WindowViewer が対話型アプリケーションとして実行されている場合、（前に説明した通り）ユーザーは、暗号化された SuiteLink 通信をセキュリティを強化するプライベート キーにアクセスできない対話型ユーザーです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch をサービスとしてのタグ サーバーとして実行する場合、クライアント ノードの InTouch Tag Server クライアント ライセンスと組み合わせて、InTouch の通信をタグ サーバーに制限することができます。このシナリオを使用する場合、多くのユーザーは HyperV や VMWare などの PC 仮想化テクノロジーを使用して、その仮想マシン（VM）内のサービスとして WindowViewer を実行し、ソリューションの総保有コストを削減しています。

標準ユーザーとしての WindowViewer の実行

以下のシナリオでは、WindowViewer は管理特権のない標準ユーザーとして実行していることになります。

- 管理特権のない標準ユーザーとして System Platform IDE を起動し、WindowMaker で $InTouchViewApp を編集した後に WindowViewer への高速切り替えを行う。
- AVEVA アプリケーション マネージャから WindowViewer を起動するか、管理特権のない標準ユーザーとして WindowViewer を起動する。

SuiteLink の通信障害のトラブルシューティング

SuiteLink の通信に遭遇した場合:

- InTouch HMI がインストールされているコンピュータ上で、Microsoft TCP/IP が動作していることを確認します。
- コンピュータのノード名が 15 文字以内であることを確認します。
- SuiteLink が、InTouch HMI がインストールされているコンピュータのサービスとして実行していることを確認します。
- SuiteLink は、InTouch のインストール中に自動的にインストールされます。また、SuiteLink のサービスは自動的に起動されます。SuiteLink のサービスが停止した場合は、再起動する必要があります。

標準ユーザーとしてサーバーにアクセスする際の Suitelink 通信の問題

以下の表には、さまざまなバージョンの System Platform での非管理者ユーザー（標準ユーザー）の Suitelink 接続の動作、および該当するトラブルシューティング手順が示されています。

シナリオ	**System Platform 2020 R2 SP1 まで**	システム プラットフォーム **2023**	トラブルシューティング
SMS が設定されていない状態で WindowViewer を実行	ロガー メッセージなし	ロガー メッセージなし	該当なし
SMS が設定されていない状態でクライアントを WindowViewer に接続	WindowViewer への非セキュアな接続が確立されています。ロガー内に警告メッセージがあります。*	接続に失敗します。 ロガー内に警告メッセージがあります。*	• コンフィグレータで非セキュアな接続を許可するオプションを選択します。 非セキュアな接続が確立され、ロガーに警告メッセージが記録されます。 • セキュアな Suitelink 接続を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SMS が設定された状態で WindowViewer を実行	接続に失敗します。 ロガー内にエラー メッセージがあります。*	接続に失敗します。 ロガー内にエラー メッセージがあります。*	• WindowViewer サービスとして実行 • 管理者特権で WindowViewer を対話型モードで実行するか、 • WindowViewer ユーザーを "ArchestrAWebHosting" グループに追加します。
SMS が設定された状態でクライアントを WindowViewer に接続	WindowViewer への非セキュアな接続が確立されています。ロガー内に警告メッセージがあります。*	接続に失敗します。 ロガー内に警告メッセージがあります。*	• コンフィグレータで非セキュアな接続を許可するオプションを選択します。 非セキュアな接続が確立され、ロガーに警告メッセージが記録されます。 • セキュアな Suitelink 接続を設定します。
SMS とセキュアな Suitelink 接続（V3）が設定されておらず、フォールバックとしての非セキュアな Suitelink 接続（V2）が無効化	• Suitelink クライアントが WindowViewer に接続します。 • Web Client が InTouch タグにバインドします。 • ロガー内に警告メッセージがあります。*	• Suitelink クライアントが WindowViewer に接続しない。 • Web Client が InTouch タグにバインドしない。 WindowViewer の起動時にエラーメッセージがロガーに記録される。**	• コンフィグレータで SMS を設定します。 • セキュアな Suitelink 接続を設定します。 • フォールバックとして非セキュアな接続を許可するオプションを選択します。

警告メッセージ* - 警告 WWSLS 暗号化されていない通信チャネルが 127.0.0.1(VIEW) から確立されました。System Management Server を両方のノードにインストールして、クライアントとサーバー製品の両方を最新のバージョンにアップグレードして、このチャネルを保護してください。

エラー メッセージ** - エラー WWSLS [複数行メッセージ] - この設定では暗号化されたインバウンド SuiteLink 接続を確立できません。

OPC

OPC 自体は通信プロトコルではありませんが、ドライバとして機能します。これは専有されていない標準インターフェイスのセットで、Microsoft の OLE/COM テクノロジに基づいています。この標準インターフェイスにより、オートメーション/制御アプリケーション、フィールド システム/デバイス、ビジネス/オフィス アプリケーション間の相互運用性が可能になります。

ソフトウェア アプリケーション開発者にとって従来の要件だったフィールド デバイスとのデータ通信用のカスタム ドライバを作成する必要性を省くため、OPC は汎用的な高性能インターフェイスを定義することで、簡単に HMI、SCADA、制御、およびカスタム アプリケーションを簡単に再利用できるようにしています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ネットワーク上では、OPC は DCOM（分散 COM）を使用してリモート通信を行います。

OPC UA

OPC Unified Architecture（OPC UA）は、相互運用性を目的としたマシン間の通信プロトコルです。OPC UA は、強化されたセキュリティに加え、高度な通信、セキュリティ、および情報モデルを備えたプロセス制御、そして情報モデル、およびクロスプラットフォームの接続性を提供します。

OPC UA は、OI Gateway のクライアントとして実装されています。

OPC UA は、OPC とは大きく異なります。ここでは、クラシック OPC と OPC UA の主な違いについて説明します。

クラシック OPC	OPC UA
通信に Microsoft の COM/DCOM 技術を使用します。設定可能なタイムアウトがありません。OPC UA はシステムで設定される DCOM タイムアウトに依存します。	データのエクスポートにサービス アーキテクチャを使用し、通信と接続を容易にします。
Windows オペレーティング システムに依存します。	プラットフォームに依存し、広範なデバイスとプラットフォームに接続します。
セキュリティが制限されています。	セキュリティが組み込まれています。
メッセージの損失などの問題を処理する機能は組み込まれていません。	メッセージの損失などの問題を処理する機能が組み込まれています。

MQTT

MQTT（Message Queuing Telemetry Transport）は、TCP/IP を介して使用する Publish／Subscribe メッセージング プロトコルです。MQTT は、必要な電力と帯域幅を最小限にしながら、デバイスが相互に通信できるようにします。MQTT は、低速または不安定なネットワークに依存するデバイスでの使用に適したシンプルなメッセージング プロトコルです。

MQTT プロトコルはアプリケーション レイヤー仕様で、標準の ISO/IEC PRF 20922 として公開されています。MQTT は、仲介ブローカーを必要とする Publish／Subscirbe メカニズムです。パブリッシャーがブローカーにデータを送信し、サブスクライブ クライアントはブローカーにパブリッシュされたデータを受信します。特定のトピックをサブスクライブ（購読）するクライアントだけが、そのトピックに関するメッセージを受信します。プロトコルは、双方向通信をサポートするので、パブリッシャーであるデバイスが更新を受信することも可能です。

アクセス名の設定

InTouch I/O タグまたはリモート タグ 参照には、アクセス名を関連付ける必要があります。アクセス名によって、別の I/O データ ソースとの通信リンクが定義されます。各アクセス名には、ノード名、アプリケーション名、およびトピックで構成される I/O アドレスが指定されます。

分散アプリケーションでは、I/O 参照をネットワーク I/O サーバーのグローバル アドレスとして、またはローカル I/O サーバーのローカル アドレスとして設定できます。

InTouchView には、Application Server 環境で使用するために特に設計された HMI アプリケーションの視覚的なインターフェイスが表示されます。InTouchView アプリケーションはクライアントとして実行し、ほとんどの HMI 機能を提供するサーバーとして機能する Application Server も同時に実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouchView アプリケーションでは、フル機能の InTouch アプリケーションで使用できる標準的な関数の一部のみが提供されています。InTouchView アプリケーションは、Application Server Galaxy 以外の I/O ソースに接続できません。InTouchView アプリケーションで使用できるのはデフォルトの Galaxy アクセス名だけで、アクセス名を作成することはできません。

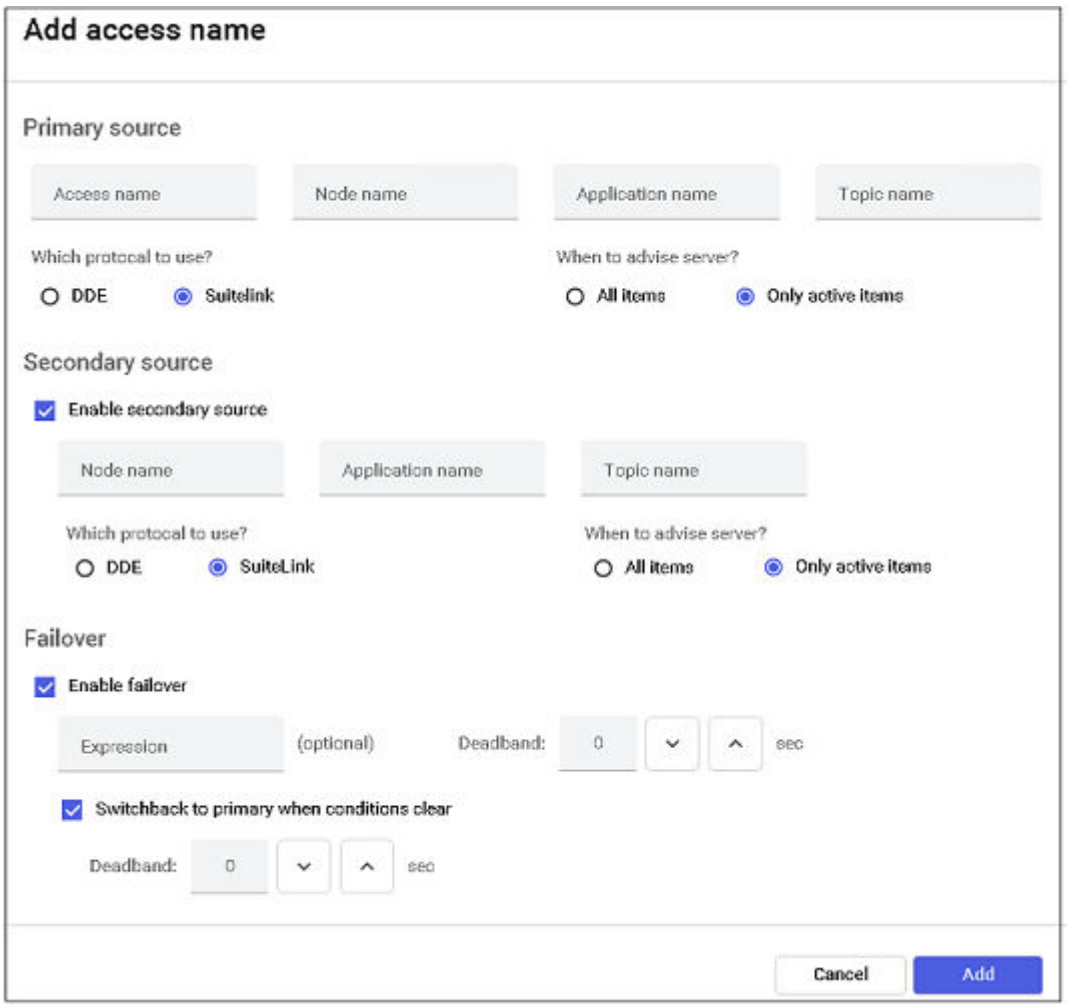
アクセス名を作成するには

1. ［ホーム］メニューの［タグ］グループで［アクセス名］をクリックします。

   ［アクセス名］ダイアログ ボックスが表示されます。

2. 追加を選択します。

   ［アクセス名の追加］ダイアログ ボックスが表示されます。



3. ［アクセス名の追加］ダイアログ ボックスのプロパティを設定します。以下の手順を実行します。
   - ［アクセス名］ボックスに、このアクセス名を識別する名前を入力します。アクセス名の最大文字数は 16 文字です。
   - データがネットワーク I/O サーバーに保存されている場合は、［ノード名］ボックスにリモートサーバーのノード名を入力します。ノード名の最大文字数は 71 文字です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**アプリケーション名**］ボックスに、データの取得元となる I/O サーバー プログラムの実際のプログラム名を入力します。

  I/O データ ソースが DAServer の場合は、DAServer プログラムの名前をプログラム ファイル名の拡張子 .exe を含めずに入力します。アプリケーション名の最大文字数は 100 文字です。

- ［**トピック名**］ボックスに、アクセスするトピック名を入力します。トピック名の最大文字数は 179 文字です。

  トピック名はアプリケーション固有のデータ要素のサブグループです。DAServer プログラムから取得されるデータの場合、トピック名は、DAServer プログラムでそのトピックに対して設定された名前と同じです。Microsoft Excel との通信では、トピック名はファイルを保存したときに付けたブック名およびスプレッドシート名でなければなりません。たとえば、[Book1]Sheet1 と指定します。

- I/O サーバーとの通信プロトコルを選択します。
- サーバーに保存されている情報へのポーリング オプションを選択します。

オプション	定義
全てのアイテムを要求	データが表示されているウィンドウ内にあるかどうか、またはスクリプトで使用されているかどうか、あるいはアラーム、ログ、トレンド グラフが設定されているかどうかに関わらず、すべてのデータがポーリングされます。このオプションを選択するとパフォーマンスに影響が出るため、使用はお勧めできません。
有効アイテムのみ要求	表示されているウィンドウ内のデータ、およびスクリプトで使用されているデータまたはアラーム、ログ、トレンド グラフが設定されているデータだけをポーリングします。 **注 :** ボタン アクション スクリプトがポーリングされるのは、表示されているウィンドウにボタンが見えている場合だけです。

4. セカンダリ バック アップ サーバーを選択する場合は、［**セカンダリ ソースを有効にする**］を選択します。
5. アクセス名を指定して［**追加**］を選択します。

   新しいアクセス名がリストに追加されます。
6. **閉じる**を選択します。

セカンダリ サーバーのフェイルオーバー切り替えの設定については、「アクセス名によるフェイルオーバー機能の使用」を参照してください。

アクセス名の削除

今後使用しないアクセス名を削除することができます。アクセス名の削除する前に、以下を確認してください。

- アクセス名にタグが関連付けられていないこと。
- WindowViewer が停止していること。

アクセス名を削除するには

1. ［**ホーム**］メニューの［**タグ**］グループで［**アクセス名**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［**アクセス名**］ダイアログ ボックスが表示されます。現在のアクセス名のリストが表示されます。

2. アクセス名を削除するには、削除するアクセス名をリストで選択して**削除**をクリックします。

   アクセス名削除の確認を促すメッセージが表示されます。

3. **はい**を選択します。
4. **閉じる**をクリックするか、他のアクセス名を削除する場合はこの手順を繰り返します。

I/O タグ変数による I/O データへのアクセス

InTouch HMI では、I/O タグ変数を使用してローカルおよびリモートの Windows アプリケーションとのデータの送受信が可能です。各 I/O 型タグ変数は、I/O サーバー プログラムの有効アイテムを参照します。タグ変数ディクショナリで、異なるタイプの I/O タグ変数を定義できます。

I/O タグ変数のプロパティの設定

タグ変数ディクショナリで、異なるタイプの I/O タグ変数を定義できます。

論理型 I/O タグ変数の指定

I/O 論理型タグは、プログラマブル コントローラ、プロセス コンピュータ、およびネットワーク ノードのデータとのすべての入出力のバイナリ状態を示します。

I/O 論理型タグ変数にはオン／オフの初期値を割り当てる必要があります。論理型 I/O タグ変数をバイナリ ソースの反対値に切り替えるように設定することもできます。また、タグ変数に割り当てられたプロセスがアラーム状況になったときにアラーム イベント ウィンドウに表示されるメッセージを指定することができます。

タグ変数ディクショナリでタグ変数を作成する手順の全般については、「新しいタグの作成」を参照してください。

I/O 論理型タグ変数を定義するには

1. タグ変数ディクショナリを開いて新しいタグ変数名を指定します。
2. ［**タグ変数タイプ**］ダイアログ ボックスでタグ変数を I/O 論理型に指定します。［**タグ名ディクショナリ**］ダイアログ ボックスの詳細部分が表示されます。

3. 以下の手順を実行します。
   - タグに関連付けられた初期値として［**オン**］または［**オフ**］を選択します。

     アプリケーションの起動時に InTouch HMI はこの値をタグ変数に割り当てますが、I/O デバイスにはこの初期値を書き込みません。
   - リモート I/O タグ変数から受け取った値に適用される入力データの変換に、［**そのまま**］または［**反転**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

入力データの変換	説明
そのまま	入力値は、そのままの状態で I/O サーバー プログラムから直接読み取られます。
反転	I/O 論理値がサーバー プログラムから読み取られるときに、値が反転されます。たとえば、I/O 入力データが 0 の場合、値には自動的に 1 がセットされます。

4. タグがアラーム状態になったときやアラーム状態から抜け出したときに表示するメッセージを［オン メッセージ］ボックスおよび［オフ メッセージ］ボックスに入力します。

   これらのメッセージは、タグにアラームの設定があるかどうかに関わらず、あらゆるアニメーション リンクやスクリプトで使用できます。

   - タグ値が 1（On、True）のときにアクティブな論理型アラームを定義した場合には、［オン メッセージ］ボックスに入力したメッセージが ActiveX アラーム表示の［値］列および［しきい値］列に表示されます。

     タグのアラーム状態が通常状態に戻ると、［オフ メッセージ］ボックスに入力したメッセージが［値］列に表示され、［オン メッセージ］は［しきい値］列に残ります。

   - タグ値が 0（Off、False）のときにアクティブな論理型アラームを定義した場合には、［オフ メッセージ］ボックスに入力したメッセージが ActiveX アラーム表示の［値］列および［しきい値］列に表示されます。

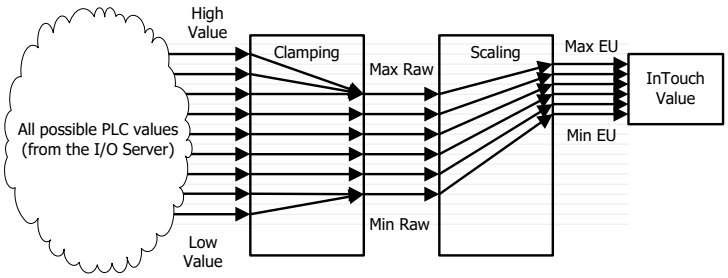
     タグのアラーム状態が通常状態に戻ると、［オン メッセージ］ボックスに入力したメッセージが［値］列に表示され、［オフ メッセージ］は［しきい値］列に残ります。

5. タグの変更を保存します。

整数型と実数型の I/O タグ変数の指定

I/O 整数型と I/O 実数型のタグ変数には、InTouch アプリケーションと外部プロセス間でやり取りされる数値データの特性を表す属性のセットを割り当てる必要があります。

InTouch HMI は、PLC から入力される生データを正規化します。以下の図は、生の I/O データ値がクランピングされてから、InTouch アプリケーションで表示できる工学単位にスケーリングされるプロセスを示しています。



I/O 整数型と I/O 実数型のタグ変数には、PLC から送られる生の入力データの最小値と最大値を設定する属性があります。InTouch HMI は、この生データ値の範囲外の I/O データ値をクランプします。クランピングによってこの範囲外の値には、生データの最小値か最大値が再割り当てされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

I/O 整数型と I/O 実数型のタグ変数には、クランプされた生データ値を工学単位の範囲にスケーリングするための属性があります。最小工学値と最大工学値の属性には、スケーリングされる値の上限値と下限値が設定されます。

I/O 整数型と I/O 実数型のタグ変数の定義では、工学値の算出の際に生データ値をスケーリングするために使用される変換タイプを指定します。線形または二乗根から選択できます。

線形スケーリングでは、最小と最大の両終点間の線形補完法を使用して結果が計算されます。入力の線形スケーリングのアルゴリズムは以下のとおりです。

```
EUValue = (RawValue - MinRaw) * ((MaxEU - MinEU) / (MaxRaw - MinRaw)) + MinEU
```

出力の線形スケーリングのアルゴリズムは以下のとおりです。

```
RawValue = (EUValue - MinEU) * ((MaxRaw - MinRaw) / (MaxEU - MinEU)) + MinRaw
```

二乗根スケーリングでは、生データの最小値と最大値が補完に使用されます。これは、圧力変換器などの非線形装置からの入力のスケーリングに役立ちます。入力の二乗根スケーリングのアルゴリズムは以下のとおりです。

```
EUValue = sqrt(RawValue - MinRaw) * ((MaxEU - MinEU) / sqrt(MaxRaw - MinRaw)) + MinEU
```

出力の二乗根スケーリングのアルゴリズムは以下のとおりです。

```
RawValue = square(((EUValue - MinEU)) * (sqrt(MaxRaw – MinRaw) / (MaxEU -MinEU))) + MinRaw
```

整数型と実数型の I/O タグ変数の定義

1. タグ変数ディクショナリを開いて新しいタグ変数名を指定します。
2. ［**タグ変数タイプ**］ダイアログ ボックスで、タグ変数のタイプに I/O 整数型か I/O 実数型を指定します。［**タグ名ディクショナリ**］ダイアログ ボックスの詳細部分が表示されます。

3. 以下の手順を実行します。
   - アプリケーション起動時にタグ変数に関連付けられる整数値または実数値を、［**初期値**］ボックスに入力します。

     アプリケーションによる、この初期値の外部プロセスへの書き込みは行われません。
   - ［**工学値最小値**］ボックスに、タグ変数に対する最小工学値を入力します。
   - ［**工学値最大値**］ボックスに、タグ変数に対する最大工学値を入力します。
   - ［**生データ最小値**］ボックスに、I/O 整数型または I/O 実数値の生データが取り得る最小クランプの値を入力します。
   - ［**生データ最大値**］ボックスに、I/O 整数型または I/O 実数値の生データが取り得る最大クランプの値を入力します。
   - ［**工学単位**］ボックスに、タグ変数の工学単位の名前を入力します。
   - 工学値の算出の際に生データ値をスケーリングするために使用する変換のタイプとして、［**線形**］または［**二乗根**］を選択します。
4. タグの変更を保存します。

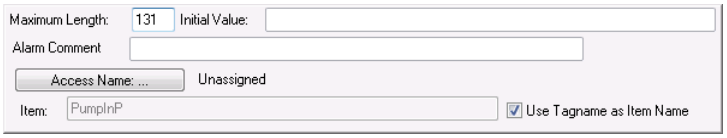
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メッセージ型 I/O タグ変数の指定

I/O メッセージ型タグ変数のオプションを指定して、リモート プロセスのネットワーク アドレスを特定することができます。このメッセージのプロパティは、メモリ メッセージ型タグ変数と同じです。

メモリ型と I/O メッセージ型のタグ変数値の定義

1. タグ変数ディクショナリを開いて新しいタグ変数名を指定します。
2. ［**タグ変数タイプ**］ダイアログ ボックスで、タグ変数のタイプに［**I/O メッセージ型**］を選択します。［**タグ名ディクショナリ**］ダイアログ ボックスの詳細部分が表示されます。



3. ［**最大文字数**］ボックスに、タグ変数のメッセージの最大文字数を入力します。

   入力できるメッセージの最大文字数は 131 文字です。

4. ［**初期値**］ボックスに、WindowViewer でアプリケーションが起動されたときに表示される文字列を入力します。
5. タグの変更を保存します。

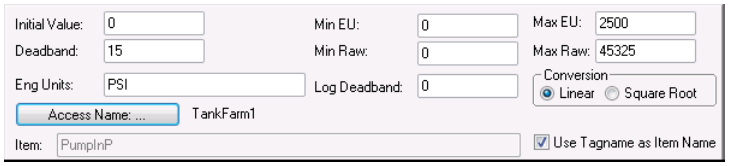
I/O アクセス パラメータの設定

タグ変数ディクショナリでタグ変数の I/O 属性を設定することができます。この属性により、タグ変数に関連付けられた外部データ ソースが識別されます。

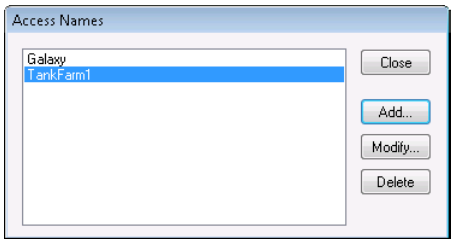
ここに示す手順では、タグ変数ディクショナリで I/O 属性を指定する方法だけを説明します。Galax y とアクセス名の設定については、「I/O を使用したデータ アクセス」を参照してください。

タグ変数の I/O 属性の設定

1. ［**タグ タイプ**］ダイアログ ボックスで、タグに I/O タグ タイプを指定します。［**タグ名ディクショナリ**］ダイアログ ボックスの詳細部分が表示されます。



2. **アクセス名**を選択して、タグ変数に割り当てるアクセス名を定義するか選択します。［**アクセス名**］ダイアログ ボックスに、InTouch HMI が認識している現状のアクセス名のリストが表示されます。（ArchestrA の接続に対するデフォルトのアクセス名は Galaxy です。）



3. アクセス名を追加するか、またはデフォルト値をそのまま使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. I/O タグ変数がデータを読み書きするサーバー プログラム内のデータ ポイントを選択します。
   - サーバー プログラム内のプロセス データ ポイントでデータを読み書きできるように、[**アイテム名**] ボックスにアイテム名を入力します。たとえば、PLC のレジスタから値を読み取るには、レジスタの識別名をアイテム名に入力します。以下に例を示します。

     Allen-Bradley PLC のレジスタ 1 をアイテム名として使用するには、[**アイテム名**] ボックスに「R1」と入力します。

     Allen-Bradley PLC のレジスタ 1 の最下位ビットをアイテム名として使用するには、[**アイテム名**] ボックスに「R1:0」と入力します。
   - タグ変数をアイテムとして使用するには、[**タグ変数名をアイテム名として使用**] を選択します。アイテム名には、最大 254 文字を入力できます。

ランタイム時の I/O タグ変数関連情報の取得

アクセス名の定義で指定したノード名、アプリケーション名、トピック名を返す関数を使用してスクリプトを記述できます。

IOGetNode() 関数

IOGetNode() 関数は、特定のアクセス名に対して定義されたノード アドレスを、スクリプトによってこの関数に関連付けられたタグ変数に返します。

カテゴリ

その他

構文

```
IOGetNode("AccessName");
```

引数

AccessName
ノード情報を取得する既存のアクセス名

備考

アクセス名はリテラル文字列で指定するか、他の InTouch タグ変数または関数が返す文字列の値で指定できます。

例

以下の例は、アクセス名 ModbusPLC1 のノード情報をタグ変数 **NodeName** に返します。

```
NodeName = IOGetNode("ModbusPLC1");
```

IOGetApplication() 関数

IOGetApplication() スクリプト関数は、特定のアクセス名に対して定義されたアプリケーション名を、関数の引数に指定したタグ変数に返します。

カテゴリ

その他

構文

```
IOGetApplication("AccessName");
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

AccessName
アプリケーションが定義されている既存のアクセス名

備考

アクセス名はリテラル文字列で指定するか、他の InTouch タグ変数または関数が返す文字列の値で指定できます。

例

以下の例は、アクセス名 ModbusPLC1 に対して指定されたアプリケーション名を、タグ変数 **AppName** に返します。

```
AppName = IOGetApplication ("ModbusPLC1");
```

IOGetTopic() 関数

IOGetTopic() 関数は、特定のアクセス名に対して定義されたトピック名を、スクリプトによってこの関数に関連付けられたタグ変数に返します。

カテゴリ

その他

構文

```
IOGetTopic("AccessName");
```

引数

AccessName
トピック名が返されるアクセス名

備考

アクセス名はリテラル文字列で指定するか、他の InTouch メッセージ型タグ変数または関数が返す文字列の値で指定できます。

例

以下の例は、アクセス名 ModbusPLC1 のトピック情報を、タグ変数 **TopicName** に返します。

```
TopicName = IOGetTopic("ModbusPLC1");
```

ランタイム時の I/O タグ リファレンスの動的変更

InTouch HMI では、ダイナミック リファレンス アドレスを使用して、診断アプリケーションのように値を一時的に必要とするデータ ポイントを表示します。ダイナミック リファレンス アドレスは、1 つのタグ変数で複数のデータ ソースのアドレス設定を可能にするものです。

1 つのタグ変数で複数のデータ ソースを動的に参照するには、いくつかの方法があります。

- によって、異なるアクセス名の特性を割り当てる**参照** I/O タグのドットフィールド
- IOSetItem() スクリプト関数を使用して、I/O タグ変数の .Reference ドットフィールドを設定する **参照** ドットフィールド
- **IOSetAccessName()** スクリプト関数を使用して、ランタイム中にアクセス名の特性を変更する。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.Reference ドットフィールド

I/O タグ変数の **.Reference** ドットフィールドに対して有効なリファレンスを割り当ることにより、ダイナミック リファレンス アドレスを使用できます。**.Reference** ドットフィールドを使用して I/O タグ変数に割り当てられたアクセス名の特性を修正することにより、データ ソースを動的に変更することができます。

.Reference の構文は以下の通りです。

tag.Reference="accessname.item"	タグ変数に割り当てられたアクセス名とアイテムを変更します。
tag.Reference="[.]item"	I/O タグ変数に割り当てられたアイテムを変更します。
tag.Reference="accessname."	I/O タグ変数のアクセス名を変更します。
tag.Reference=""	I/O タグ変数の使用を解除します。

各 I/O 型タグ変数には **.ReferenceComplete** ドットフィールドがあります。この論理型ドットフィールドの値は、**.Reference** ドットフィールドで要求したアイテムが、**.Value** ドットフィールドの内容に反映されたかどうかを示します。

WindowViewer でアプリケーションが起動されると、**.ReferenceComplete** フィールドに False（0）が設定されます。**.Reference** ドットフィールドに指定したソースにより **.Value** ドットフィールドが更新されたことが確認されると、**.ReferenceComplete** の値は True（1）に設定されます。**.Reference** ドットフィールドを変更すると、**.ReferenceComplete** ドットフィールドは自動的に False（0）に設定され、新しい値が更新された時点で True（1）になります。

IOSetItem() 関数

スクリプトに **IOSetItem()** 関数を使用して、ダイナミック リファレンス アドレスを使用できます。**IOSetItem()** には、I/O タグ変数の **.Reference** ドットフィールドに割り当てられた値をランタイム中に変更するための引数が含まれています。

カテゴリ

その他

構文

```
IoSetItem ("Tag", "AccessName", "Item");
```

引数

Tag
引用符で囲んだ任意の InTouch I/O タグ変数

AccessName
I/O タグ変数に割り当てられたアクセス名。

Item
I/O タグ変数に割り当てられたアイテム。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ変数、アクセス名、およびアイテムは、リテラル文字列として指定するか、その他の InTouch タグ変数または関数が返す文字列の値を指定できます。

例

以下の例では、タグ変数 **PumpInP1** の **.Reference** ドットフィールドが、アクセス名 excel と アイテム R1C1 をポイントするように変更されます。

```
IOSetItem("PumpInP1", "excel", "R1C1");
```

または、

```
Number = 1;
TagNameString = "PumpInP" + Text(Number,"#");
IOSetItem(TagNameString, "excel", "R1C1");
```

アクセス名とアイテムの値として空の文字列（""）が指定された場合、タグ変数は無効になります。たとえば、以下のように指定するとタグ変数 **PumpInP2** は解除されます。

```
IOSetItem("PumpInp2", "", "");
```

アイテムのみに空の文字列を指定すると、タグ変数の現在のアイテム値は保持され、アクセス名の値のみが更新されます。たとえば、以下ではタグ変数 **PumpInP3** のアクセス名は excel2 に変更されますが、現在のアイテム値は変更されません。

```
IOSetItem("PumpInP3", "excel2", "");
```

同様に、アクセス名の値のみに空の文字列を指定すると、タグ変数の現在のアクセス名の値が保持され、アイテム値が更新されます。たとえば、以下ではタグ変数 **PumpInP4** のアイテムは R1C2 に変更されますが、現在のアクセス名の値は変更されません。

```
IOSetItem("PumpInP4", "", "R1C2");
```

IOSetAccessName() 関数

スクリプトに **IOSetAccessName()** 関数を使用して、ダイナミック リファレンス アドレスを使用できます。**IOSetAccessName()** は、アプリケーションまたは I/O タグ変数のアクセス名のトピック名の特性をランタイム中に修正します。

注記: IOSetAccessName() 関数が処理されると、既存の通信を終了してから新しい通信を開始するまでの時間遅れが生じます。この間に新しいトピックに対して実行された POKE や書き込みは無効になります。

カテゴリ

その他

構文

```
IOSetAccessName("AccessName", "NodeName", "AppName", "TopicName");
```

引数

AccessName
新しい AppName と Topic Name の値を割り当てる既存のアクセス名。実際の文字列またはメッセージ型タグ変数です。

NodeName
割り当てる新しいノード名。実際の文字列またはメッセージ型タグ変数です。

AppName
割り当てる新しいアプリケーション名。実際の文字列またはメッセージ型タグ変数です。

TopicName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

割り当てる新しいトピック名。実際の文字列またはメッセージ型タグ変数です。

備考

AccessName、AppName、および TopicName の引数に指定する値は、リテラル文字列として指定するか、その他の InTouch タグ変数または関数が返す文字列の値を指定できます。

[**アクセス名の追加**] ダイアログ ボックスを使用してアクセス名のフェイルオーバーのセカンダリ ソースを設定した場合、IOSetAccessName() 関数を使用してランタイム時のセカンダリ ソースの設定を変更することはできません。

注記: WindowMaker でアクセス名を作成するとき、このアクセス名が SuiteLink タイプを使用している場合は、InTouch HMI は作成するアクセス名による同じノード、アプリケーションとトピックへのアクセスができないようにします。**IOSetAccessName** 関数がランタイム中にアクセス名をリダイレクトしてアクセス名を複製することがないようにしてください。ランタイム中に **IOSetAccessName()** 関数を使用すると、SuiteLink タイプのアクセス名がリダイレクトされてトピックが複製されます。ただし、リダイレクトしたアクセス名は作動しません。

例

以下のスクリプト関数の使用により、アクセス名 **MyAccess1** はアプリケーション **Excel** とトピック [**Book1]Sheet1** をポイントするように変更されますが、現在のノード名は変更されません。

```
IOSetAccessName("MyAccess1", "", "EXCEL", "[Book1]Sheet1");
```

トピックに空の文字列が指定されると、アクセス名の現在のアプリケーションの値は更新され、トピックの値は保持されます。

たとえば、以下のスクリプトではアクセス名 **MyAccess2** のアプリケーション名を EXCEL に変更しますが、現在のトピックの値は変更されません。

```
IOSetAccessName("MyAccess2", "", "EXCEL", "");
```

アプリケーション名のみに空の文字列が指定されると、タグ変数の現在のトピックの値は更新され、アプリケーションの値は保持されます。たとえば、以下のスクリプトではタグ変数 **MyAccess3** のトピックを [**Book2]Sheet1** に変更しますが、現在のアプリケーション名の値は変更されません。

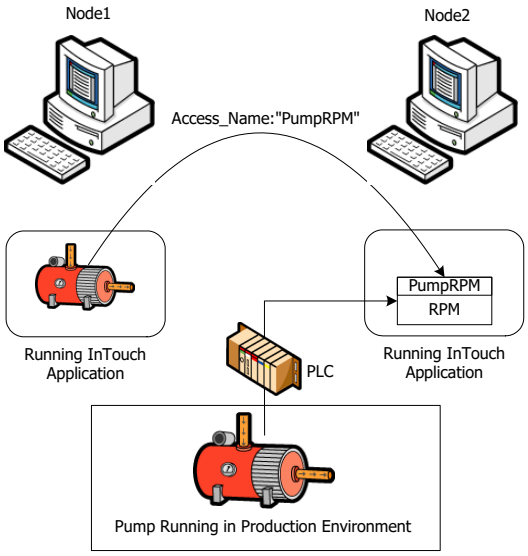
```
IOSetAccessName("MyAccess3", "", "", "[Book2]Sheet1");
```

この例は、PLC の冗長性が必要な場合に使用できます。

タグのリモート 参照への変換

クライアント/サーバー アーキテクチャ ベースの分散 InTouch アプリケーションを作成することができます。クライアント アプリケーションを、別のリモート ノードで定義されたタグを使用する 1 つのネットワーク ノード上で実行することができます。以下の図に、ノード 1 で稼動し、ノード 2 のタグ PumpRPM へのリモート 参照を作成する InTouch アプリケーションを示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



この例では、タグ PumpRPM の値をノード 2 から 2 つの方法で取得できます。

- I/O 型タグ変数に関連付けられているアクセス名のノード名としてノード 2 を使用する I/O 型タグ変数をノード 1 のタグ変数ディクショナリに作成します。
- タグ **PumpRPM** への直接リモート 参照を使用します。たとえば、**PLC1:"TempTag"** を使用します。

  次のような形式でリモート タグ名の前にアクセス名を追加することにより、ウィンドウや QuickScript でリモート タグを参照できます。

  access_name:"tag_name"

ウィンドウまたは QuickScript をインポートすると、プレースホルダ タグをリモート タグ 参照に変換できます。たとえば、ウィンドウをインポートしたアプリケーションをポイントするようにプレースホルダ タグを変換できます。このタグは、ローカルのタグ名ディクショナリで定義する必要はありません。

ローカル タグをリモート タグ 参照に変換するには、いくつかの方法があります。

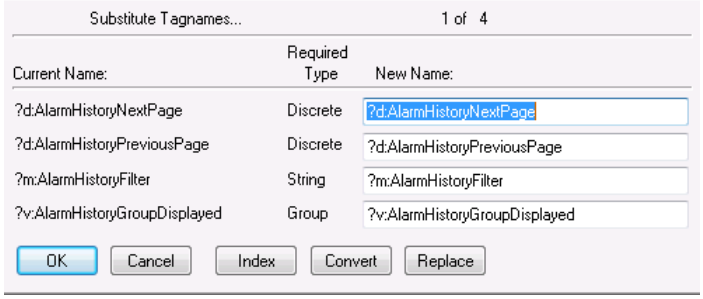
- リモート タグ 参照を追加します。
- インポートしたウィンドウに関連付けられたプレースホルダ タグを変換します
- タグ ブラウザを起動してタグ ソースのタグ名ディクショナリを開き、リモート タグ 参照を選択します。

タグをリモート タグ 参照に手動で変換するには

1. WindowMaker でアプリケーション ウィンドウを開きます。
2. リモート タグ 参照に変換するローカル タグに関連付けられたオブジェクトを選択します。
3. ［**アニメーション**］タブの［**置換**］グループで［**タグ**］をクリックします。

   ［**タグ名の置換**］ダイアログ ボックスが開き、オブジェクトに関連付けられたタグのリストが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

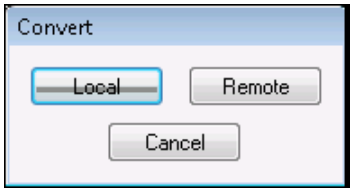


4. 各タグ名にインデックスを追加するには、**インデックス**を選択します。
5. **変換**を選択します。タグをローカルまたはリモート 参照 タグに変換するオプションが付いた、[**変換**] ダイアログ ボックスが表示されます。
6. **リモート**を選択します。[**アクセス名**] ダイアログ ボックスが表示されます。ローカルの InTouch アプリケーションで定義されているすべてのアクセス名が表示されます。
7. リストからアクセス名を選択します。
8. **閉じる**を選択します。[**タグの置換**] ダイアログ ボックスにリスト表示されているすべてのタグは、タグ名にアクセス名が付けられたリモート タグ 参照に自動的に変換されます。
9. **[OK]** を選択します。

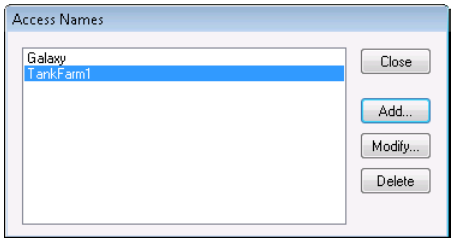
インポートしたウィンドウのタグをリモート 参照に変換するには

1. インポートしたウィンドウを開いてすべてのオブジェクトを選択します。
2. [**アニメーション**] タブの [**置換**] グループで [**タグ**] をクリックします。

   [**タグ名の置換**] ダイアログ ボックスが表示されます。
3. **変換**を選択します。[**変換**] ダイアログ ボックスが表示されます。



4. **リモート**を選択します。[**アクセス名**] ダイアログ ボックスが表示されます。ローカルの InTouch アプリケーションで定義されているすべてのアクセス名が表示されます。



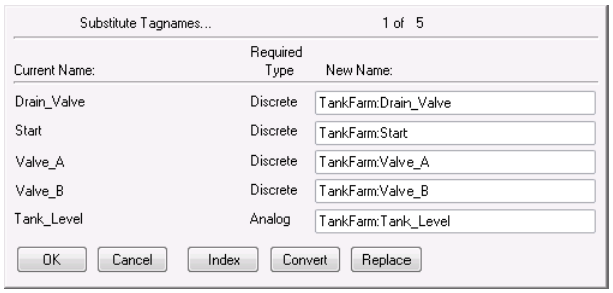
5. リストからアクセス名を選択します。

   アクセス名が正しく設定されたことを確認するには、**修正**をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ ソースにアクセスするためのアクセス名が現在定義されていない場合には、［**追加**］をクリックして定義します。アクセス名にはアプリケーションが常駐しているリモート ノードの名前を含める必要があります。

6. **閉じる**を選択します。［**タグの置換**］ダイアログ ボックスにリスト表示されているすべてのタグは、タグ名にアクセス名が付けられたリモート タグ 参照に自動的に変換されます。

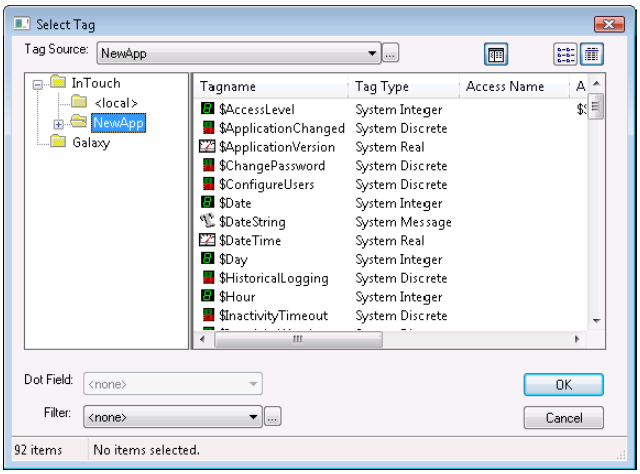


7. **[OK]** を選択します。

リモート タグ 参照をタグ ブラウザで選択する

1. リモート タグ 参照に変換するローカル タグに関連付けられたオブジェクトを選択します。
2. ［**アニメーション**］タブの［**置換**］グループで［**タグ**］をクリックします。

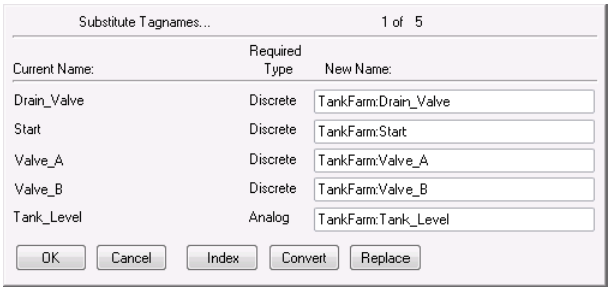
   ［**タグ名の置換**］ダイアログ ボックスが開き、選択したタグが表示されます。
3. リモート タグ 参照で置き換えるタグ名を［**新しい名前**］ボックス内で削除します。
4. ［**新しい名前**］ボックスをダブルクリックします。［**タグの選択**］ダイアログ ボックスが開き、アプリケーションに関連付けられたタグのリストが表示されます。
5. ツリー ビューを使用してリモート タグを選択します。
   a. ［**ツリー**］アイコンをクリックすると、左ウィンドウにローカルとリモートの全アクセス名の階層リストが表示されます。



   b. リモート アクセス名のフォルダを選択すると、フォルダに割り当てられたタグが右ペインに表示されます。
   c. リモート 参照として使用するリモート タグを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

d. **[OK]** を選択します。［**タグ名の置換**］ダイアログ ボックスが開き、選択したリモート タグの名前が［**新しい名前**］ボックスに表示されます。

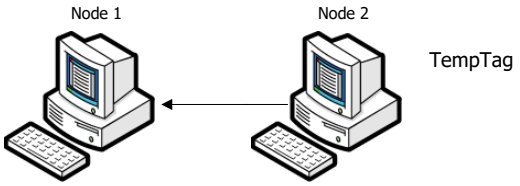


6. ［**OK**］をクリックしてダイアログ ボックスを閉じ、リモートタグを選択したオブジェクトに関連付けます。
7. リモート 参照に関連付ける各タグに対して以上の手順を繰り返します。

リモート 参照による I/O データ アクセス

InTouch HMI では、工場のオートメーション アプリケーションのクライアント/サーバー アーキテクチャをサポートします。InTouch アプリケーションが稼動しているノード上にあるローカル タグ名ディクショナリのタグを一切使用することなく、クライアント アプリケーションの開発が可能です。また、リモート タグの参照を使用することにより、リモート ノードのタグを使用する 1 つのノード上でアプリケーションを実行できます。

以下の図に、**TempTag** がノード 2 でローカルに定義されているシンプルな例を示します。



この例では、ノード 1 で稼動している InTouch アプリケーションは、2 つの方法によってノード 2 の TempTag の値を取得できます。

- 関連付けられているアクセス名のノード名にノード 2 を使用する I/O 型タグを、ノード 1 のタグ名ディクショナリで作成します。
- リモート 参照を使用して直接 **TempTag** を参照します。たとえば、Node2:"TempTag" のように使用します。

他のアプリケーションでリモート タグを直接参照する場合には、AccessName:item の記述だけが必要です。ローカル タグ名ディクショナリでリモートタグを定義する必要はありません。リモート 参照により、DAServer や Microsoft Excel などのあらゆる I/O データ ソースのデータにアクセスすることもできます。

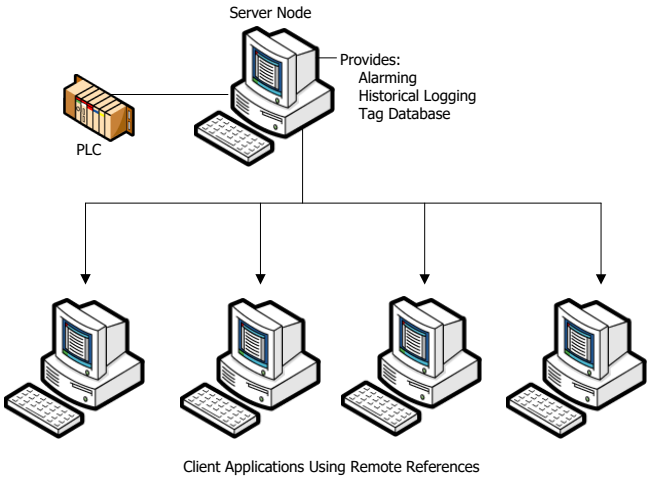
スーパータグへのリモート 参照を作成することもできます。スーパータグへのリモート タグ 参照に有効な構文を次に示します。

Access_name:Parent_Instance\ChildMember\SubMember.

スーパータグに対するリモート 参照については、「スーパータグ メンバーを参照する」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウまたは QuickScript をインポートすると、プレースホルダ タグをリモート タグ 参照に変換できます。ローカルのタグ名ディクショナリでタグを定義する必要はありません。リモート 参照は、ネットワーク上の任意のアプリケーションからアクセスできます。



ランタイム中のリモート 参照のリダイレクト

スクリプトを使用して、Application Server オブジェクト参照またはリモート 参照を、ランタイム時の InTouch タグにリダイレクトできます。これにより、想定される特定の状態に基いて、またはオペレータ アクションを直接介して、グラフィック シンボルのオブジェクト インスタンスを切り替えることができます。

IOSetRemoteReferences() 関数

IOSetRemoteReferences() スクリプト関数を使用して、InTouch アプリケーションの実行中に Application Server オブジェクト参照またはリモート 参照をタグにリダイレクトできます。**IOSetRemoteReferences()** は、指定した文字列に一致するすべてのリモート 参照を検索して、指定した引数の値に応じて参照を変更します。この関数をトリガするスクリプトを記述して、想定されるさまざまな状態に基き、またはユーザー アクションを介して参照をリダイレクトすることができます。

カテゴリ

その他

構文

```
IOSetRemoteReferences(BaseAccess, NewAccess, MatchString, SubstituteString, Mode)
```

引数

BaseAccess
この文字列引数は、参照での照合に使用する、元の設定されたアクセス名を指定します。

NewAccess
新しいアクセス名です。このアクセス名は、元のアクセス名が BaseAccess で指定された文字列に一致し、さらに MatchString の値が指定された場合には元のアイテム名がこの値に一致する、すべての参照に適用されます。

MatchString
参照の元の設定されたアイテム名での照合に使用する文字列です。MatchString の値が空の文字列の場合は、すべてのアイテム名が一致するとみなされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SubstituteString
元のアイテム名に置換される文字列です。この文字列は、MatchString の値と置き換わって参照に対する新しいアクティブなアイテム名を生成します。これが空の文字列の場合、置換は行われません。

Mode
元の設定されたアイテム名と MatchString 値との比較モードを定義します。マッチングは常にアイテム名の先頭から行われます。Mode の値が 0 の場合は、アイテム名全体の一致、またはピリオド（.）の位置までの一致が必要であることが指定されます。1 の場合は、次の文字がピリオドでなくても部分的な一致が可能であることが指定されます。

備考

IOSetRemoteReferences() は、オブジェクト参照の変更前に、新しいタグまたはアクセス名の有効性のチェックは行いません。

- **IOSetRemoteReferences()** はリモート 参照のみを変更します。この関数は、元の設定されたアクセス名が BaseAccess 引数に指定した値に一致し、さらに元のアイテム名が MatchString の値に一致する参照をリダイレクトします。
- **IOSetRemoteReferences()** の単一の呼び出しは、メモリ上のアクティブ ウィンドウ内にあり、元の設定された名前の文字列が BaseAccess および MatchString に指定した値に一致するすべてのリモート参照に影響します。
- BaseAccess 引数に値を指定しないと、**IOSetRemoteReferences()** はリモート 参照をリダイレクトしません。
- MatchString 引数が空の値の場合、**IOSetRemoteReferences()** は、元のアクセス名が BaseAccess 引数に指定した値に一致するすべてのリモート 参照をリダイレクトします。
- Mode 引数が 0 の場合、アイテム名での置換はオブジェクト名（またはタグ）全体、またはプロパティ名（またはドット フィールド）全体に対してのみ行われます。MatchString 引数の値は、元のアイテム名全体、またはピリオドの直前にある文字まで一致する必要があります。
- Mode 引数が 1 に設定されている場合は、アイテムの文字列と照合アイテムの文字列の開始が同じ場合に、アイテム文字列の部分的な置換が可能です。つまり、MatchString は元のアイテム文字列の一部と一致し、一致部分はアイテム文字列の先頭文字から一致している必要があります。一致した文字列の最終文字の次の文字が、ピリオドである必要はありません。
- 元の設定されたリモート 参照の名前は、変更されません。引き続き **IOSetRemoteReferences()** を呼び出す場合、現在アクティブな名前を指定する必要はありません。**IOSetRemoteReferences()** の呼び出しは、任意の順序で実行できます。
- 1 つのリモート 参照を参照するために複数のウィンドウが必要になる場合、リモート 参照は I/O タグのように機能します。リモート 参照をリダイレクトすると、すべてのウィンドウは同じリモート参照を表示します。2 つの異なる対象を同時に参照するために、1 つの名前を使用しないでください。

注記: たとえば、ウィンドウの「表示時」スクリプトなどで多くの参照を同時に変更する場合は、すべての参照が解決されるまでに時間を要することがあります。

例

以下の例は、元のアイテム名が pumpX に正確に一致した場合に、Galaxy アクセス名のアイテム名 pump001 にすべてのリモート 参照をリダイレクトします。

```
IOSetRemoteReferences("Galaxy","","pumpX", "pump001",0);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以下の例は、アイテム名が pumpX に正確に一致した場合に、アクセス名 Galaxy を TagServer1 に変更します。さらに、アイテム名を p2 に変更します。

```
IOSetRemoteReferences("Galaxy","TagServer1", "pumpX","p2",0);
```

以下の例は、アイテム名が pumpX の場合に、アクセス名 TagServer1 を TagServer2 に変更します。さらに、アイテム名を backpump3 に変更します。

```
IOSetRemoteReferences("TagServer1","TagServer2", "pumpX","backpump3",0)
```

以下の例は、アクセス名 TagServer1 のアイテム名 Tank を Plant に変更します。

```
IOSetRemoteReferences("TagServer1","","Tank", "Plant",1)
```

以下の例では、*BaseAccess* 引数の値が指定されていないため、リモート 参照はリダイレクトされません。

```
IOSetRemoteReferences("","Galaxy","pumpX", "pump001",0);
```

リファレンスの回復

NewAccess 引数に値が指定されず空白の場合、IOSetRemoteReferences は、アクティブなアクセス名を元のベース アクセス名に戻します。

MatchString 引数に値が指定されず空白の場合、IOSetRemoteReferen**ces()** は、アクティブなアイテム名を元のアイテム名に戻します。

注：SubstituteString が空白でない場合でも、MatchString が空白である場合は、アイテム名は元のアイテム名に戻されます。名前の最初にテキストを挿入することはできません。たとえば、IOSetRemoteReferences("Access1", "", "","Valve",0); というスクリプトを実行しても、すべての元のアイテム名の最初または最後に、「Valve」という文字列は付加されません。

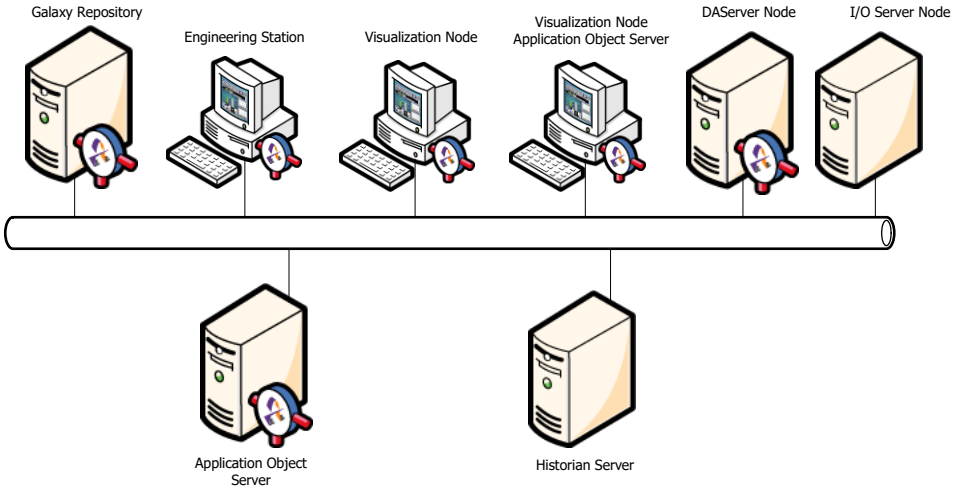
SubstituteString に値が指定されず空白の場合、IOSetRemoteReferences() はアクティブなアイテム名を元のアイテム名に戻します。空白でない MatchString と空白の SubstituteString を使用すると、指定した元のベース アクセス上のリモート リファレンスからサブセットを選択し、元のアイテム名に戻します。

InTouch から Application Server データへのアクセス

ArchestrA には、一連の製品のための共通サービスのセットと基本アーキテクチャが用意されています。このようなさまざまな製品群から製品を選択し、モジュール方式の ArchestrA コンポーネントを使用して工場の自動化システムや情報システムを構築できます。

Application Server によって、工場の自動化アプリケーションを構築するための一連のサービスが提供されます。Application Server のサービスは、システム内の複数のノードを介して配信されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



通常は、InTouch HMI が Application Server と連携して、工場のプロセス管理のためにユーザーがインタラクティブに使用するアプリケーションにビジュアル インターフェースを提供します。

InTouch と Application Server オブジェクトの使用

InTouch タグ変数を Application Server オブジェクト属性のやり取りに使用して、InTouch アプリケーションと Application Server データ レポジトリ間でデータ転送を行えます。

InTouch HMI の通信プロトコルには、メッセージ交換がサポートされています。WindowViewer で InTouch アプリケーションが実行されると、メッセージ交換は WindowViewer を匿名エンジンとみなします。

匿名性があるということは、InTouch アプリケーションには他のメッセージ交換クライアントからアクセスできる属性がないことを意味します。WindowViewer は、Application Server Galaxy ではオートメーション オブジェクトとしては設定されず、また管理も表示もされません。そのため InTouch HMI は、メッセージ交換のみを使用して Application Server の利用可能なアクティブ アイテムを要求します。

InTouch タグ ブラウザを使用してリモート タグのタグ ソースとして Galaxy を選択し、Galaxy のネームスペースを参照することができます。Application Server のオブジェクト属性や属性のプロパティは、リモート リファレンスで使用したり、InTouch I/O タグ変数のアイテムとして使用できます。

リモート タグ変数ソースとしての Application Server オブジェクトの使用の詳細については、「リモート タグ ソースとして Galaxy を使用するように InTouch HMI を設定する」を参照してください。.

あらかじめ設定されている Galaxy というアクセス名を、WindowMaker でのメッセージ交換アクセス用に利用することができます。アクセス名 Galaxy は、ArchestrA 環境の InTouch HMI に対してのみ関連性を持ちます。アクセス名 Galaxy にはユーザーが変更できるプロパティはありません。

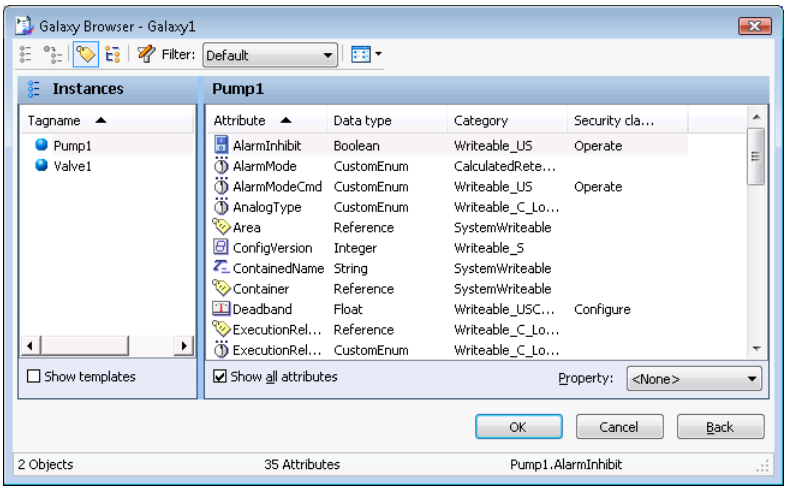
InTouch からの Application Server のオブジェクト属性の参照

InTouch HMI から Application Server 属性を参照して選択するには、最初に InTouch HMI で Galaxy のタグ ソースを設定する必要があります。詳細については、「リモート タグ ソースとして Galaxy を使用するように InTouch HMI を設定する」を参照してください。

InTouch の［**タグ変数を選択してください**］ダイアログ ボックスからタグ ソースを選択します。

InTouch の［Galaxy Browser］ダイアログ ボックスに、対象の Galaxy にあるすべてのオブジェクトのリストが表示されます。オブジェクトを展開すると、その配下のオブジェクトやランタイムでのアクセスが可能な属性を表示できます。［Galaxy Browser］ダイアログ ボックスには、「_」（隠しマーク）で始まる属性や、QualifiedStruct のタイプの属性は表示されません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



［Galaxy Browser］ダイアログ ボックスから、通常の InTouch の［タグ変数を選択してください］ダイアログ ボックスに戻るには、［**Back**］をクリックします。

Application Server ブラウザの制限事項

InTouch の［属性ブラウザ（Attribute Browser）］ダイアログ ボックスでの Application Server オブジェクトの表示には、次のような制限事項があります。

- ランタイム中に単一の Galaxy のネームスペース内で表示できる属性のみが表示されます。これには、オブジェクトの TagName と HierarchicalName 間を切り替える機能が含まれます。

  以下の要件に適合する場合に、［**属性ブラウザ**］ダイアログ ボックスからオブジェクト属性を選択できます。

  - ランタイム中に表示可能であること
  - オートメーション オブジェクト内で現在チェック済みであること
  - 属性名の「.」の後に「_」が続かないこと

- ［属性ブラウザ］ダイアログ ボックスには、InTouch HMI でサポートされている Application Server オブジェクト属性のデータ タイプのみを表示できます。サポートされるデータ タイプの詳細については、「Application Server データ タイプの InTouch データ タイプへのマッピング」を参照してください。

- InTouch の［属性ブラウザ］ダイアログ ボックスには、InTouch アイテム名が最大文字数の 95 文字以上の属性は表示されません。

- オートメーション オブジェクトの配列要素は、「TagName.AttributeName[index]」をリファレンスに使用することにより InTouch HMI で表示または取得されます。すべての配列要素の値を表示または取得するには、インデックスに -1 を使用してください。

- タグ ブラウザを使用して、オブジェクト属性のプロパティを選択できます。デフォルトでは、属性の選択時に Value プロパティが選択されます。

Application Server オブジェクトの特殊な拡張子

WindowMaker タグ ブラウザと WindowViewer のメッセージ交換クライアントは、各 Application Server オブジェクト属性に特別な拡張子を追加して認識します。この拡張子によって、InTouch HMI では利用できなかった情報にアクセスできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

この特別な拡張子は省略可能で、InTouch アプリケーションでは使用する必要はありません。ただし、ステータス情報や品質情報を頻繁に扱うアプリケーションでは、この拡張子を使用する必要があります。

これらの項目によって属性のネームスペースが拡張され、WindowViewer がアプリケーション スクリプトとウィンドウに対して公開できる追加プロパティが包含されます。たとえば、「TIC101.PV.#ReadSts」へのリファレンスによって、TIC101.PV への要求に関する MxStatus の情報にアクセスできます。この情報は、メッセージ交換によって公開される拡張情報の表示に非常に有用です。

これらのプロパティは、指定された要素としての Application Server 内のオブジェクト属性としては存在しません。このプロパティは、クライアント抽象レイヤーで提供されたクライアント側の拡張子であり、InTouch HMI でのオブジェクト属性の表示を可能にします。次の表で、InTouch HMI に対する属性の拡張子を説明します。

属性の拡張子	データ型	説明
なし	強制	デフォルトの拡張子。拡張子が提供されていないことを示します。アイテムは、強制的に与えられた InTouch に適合するデータ タイプで読み書きされます。クライアントが、次に説明する #ReadSts アイテムや #WriteSts アイテムを要求する場合、失敗した読み取り／書き込み情報を取得することができます。例:「Pump1.PV」。
.#VString float/double 属性の場合のみ： .#VString1 .#VString2 .#VString3 .#VString4	文字列型（読み取り／書き込み）	接尾辞として「.#VString」を持つリファレンスへの要求を設定します。この拡張子は基本のリファレンスです。読み取りと書き込みがどちらも正しく作動しているときには、基本リファレンスの現在の値を文字列として返します。UserGetAttribute が不良ステータスを返す場合、このアイテムは現在の値の代わりに MxStatus に基くステータスの短い説明文字列を返します。短いステータスの説明文字列には、以下のものがあります。 “?Pending” – 保留 “?Warning” - 警告 “?Comms” – 通信エラー “?Config” – 設定エラー “?Oper” – 操作エラー “?Security” – セキュリティ エラー “?Software” – ソフトウェア エラー “?Other” – その他のエラー

.#VString では、ステータスは良好でも品質が不良の場合、このアイテムは値の代わりに、メッセージ交換で利用可能な品質の説明文字列を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

UserGetAttribute に対する品質とステータスが良好、または品質が良好でステータスが不明な場合は、値が文字列として返されます。メッセージ交換で返されたデータ タイプが文字列でない場合は、強制機能が必要となることがあります。品質やステータスが不明な場合、値には接尾辞として「?」が表示されます。たとえば、「3.27?」または「True?」となります。

Application Server データ タイプの InTouch データ タイプへのマッピング

Application Server に含まれる一部の属性とデータ タイプは、InTouch タグ変数がサポートする主要な 4 つのデータ タイプに直接マップされません。

次の表で、読み取り／書き込み操作に対してクライアント抽象レイヤーがデータ タイプをマップする方法を説明します。また、Galaxy ディクショナリが InTouch に対して公開しているデータ タイプについても説明しています。

属性 プロパティ データ タイプ	InTouch データ タイプ	注記
浮動小数点	実数型 - 32 ビット	そのまま渡します。
倍精度	実数型 - 32 ビット	ダブル型が IEEE NAN の場合、浮動型 IEEE NAN に変換されます。オーバーフローする場合は、**Quality** を不良に設定して浮動型 IEEE NAN に渡します。ダブル型の値が最小浮動小数値 1.17549E-38 より小さい場合は、この値を 0.0 の浮動型として処理し、**Quality** を良好に設定します。
ブール型	論理型	False = 0、True = 1
整数型	整数型 - 32 ビット	そのまま渡します。
String（常に Unicode）	メッセージ型 - MBCS（コード化された複数バイト文字列セット）	InTouch の文字列として長すぎる場合は文字列を切り詰めて、品質を不明に設定します。各 Unicode 文字の両方のバイトは保持されます。
時間	メッセージ型 - MBCS	その地域に応じた文字列形式です。**MxValue** を使用して文字列を変換します。
経過時間	実数型	浮動型の秒数として渡します。**MxValue** はこのタイプへの強制機能をサポートしています。
MxDataType	メッセージ型 - MBCS	文字列を渡します。
MxSecurityClassification	メッセージ型 - MBCS	文字列を渡します。
MxQuality	メッセージ型 - MBCS	文字列を渡します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性 プロパティ データ タイプ	InTouch データ タイプ	注記
MxReference	メッセージ型 - MBCS	Unicode としてのみ参照文字列を渡します。
MxCategorizedStatus	メッセージ型 - MBCS	文字列を渡します。
MxQualifiedStruct	サポートされていません。	サポートされていません。
MxQualifiedEnum	メッセージ型 - MBCS	Enum 文字列を渡します。順番を示す整数値には、.#EnumOrdinal を参照することによりアプリケーションからアクセスできます。たとえば、“Pump1.PV.#EnumOrdinal” でアクセスできます。
文字列の配列	メッセージ - MBCS（読み取り専用）	配列の各要素を、以下のようなカンマ区切り文字列に配置します。 “String1,String2,String3” 配置できるのは、InTouch 文字列値の上限までです。この文字列が切り詰められた場合、InTouch HMI に送られる品質は不明になります。文字列配列の全体に対する書き込はできませんが、配列の個々の要素に対する書き込みは可能です。
すべての配列	整数型、実数型、メッセージ型、または論理型	配列の単一要素に対してのみの要求をサポートします。この場合、上で説明した変換が適用されます。それ以外の場合、返り値は不良品質の空の文字列です。
6964MxInternationalizedText	メッセージ型	ランタイム時に文字列タイプとしてアクセスされます。

Application Server 属性の読み取り/書き込み動作

システムがオートメーション オブジェクト属性を書き込む場合、初期の書き込みステータスは "? Pending" に設定されます。

書き込みが完了すると、#WriteSts 文字列が書き込み結果で更新されます。書き込みが正常に終了した場合は、#WriteSts の値には空の文字列がセットされます。書き込み処理がエラーを返して保留状態になると、要求の更新が引き続き読み取り時に発生しても、#WriteSts アイテムは最後の書き込みステータスをそのまま表示します。

#VString1 ～ #VString4 のアイテムを使用して、浮動値やダブル値を文字列形式に変換することもできます。N の値は、返される値の小数点以下の桁数を示します。たとえば、「3.1234」は #VString4 を使用し

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

た文字列です。N の値を使用しない #VString アイテムを使用すると、浮動型やダブル型の値を整数型に四捨五入して、文字列型の値として返すことができます。

属性の拡張子	データ型	説明
.#EnumOrdinal	整数型（読み取り/書き込み）	Qualified Enum 型の属性を読み取っている現在の序数を示す値です。この場合、列挙型に対して文字列ではなく整数型が返されます。
.#ReadSts	文字列（読み取り専用）	この文字列が連結されるアイテムに対して要求された現在の読み取りステータスです。 「TIC101.PV.#ReadSts」のように表します。このステータスは、メッセージ交換によって文字列で供給されます。以下のいずれかの値をとります。 “?Config” – 設定エラー “?Comms” – 通信エラー “?Oper” – 操作エラー “?Pending” – 保留 “?Warning” - 警告 “?Security” – セキュリティ エラー “?Software” – ソフトウェア エラー “?Other” – その他のエラー **注 :** 関連付けられたアイテム（たとえば、TIC101.PV）が要求を受けていない場合、返される文字列は空白になります。
.#WriteSts	文字列（読み取り専用）	この文字列が連結されるアイテムの最後の書き込みステータスで、たとえば Pump1.Cmd.#WriteSts のように表します。このステータスは、メッセージ交換によって文字列で供給されます。この文字列が空白の場合は、このアイテムへの最後の書き込みは正常です。空白以外には、以下のいずれかの値をとります。 “?Config” – 設定エラー “?Comms” – 通信エラー “?Oper” – 操作エラー “?Pending” – 保留 “?Warning” - 警告 “?Security” – セキュリティ エラー “?Software” – ソフトウェア エラー “?Other” – その他のエラー

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性の拡張子	データ型	説明
		注 : 関連付けられたアイテム（たとえば、TIC101.PV）が要求を受けていない場合、返される文字列は空白になります。

リモート タグ ソースとして Galaxy を使用するように InTouch HMI を設定する

InTouch タグ ブラウザを使用して、タグ ソースとして Application Server オブジェクトを選択し、Galaxy データベースを参照することができます。Application Server オブジェクト属性または属性プロパティは、リモート 参照で使用したり、InTouch I/O タグのアイテムとして使用できます。

InTouch アプリケーションが、Application Server アプリケーションのビジュアル インターフェースを提供するクライアントとして実行されている場合、InTouch アプリケーションと同じノードに Application Server Bootstrap と WinPlatform オブジェクトをインストールする必要があります。

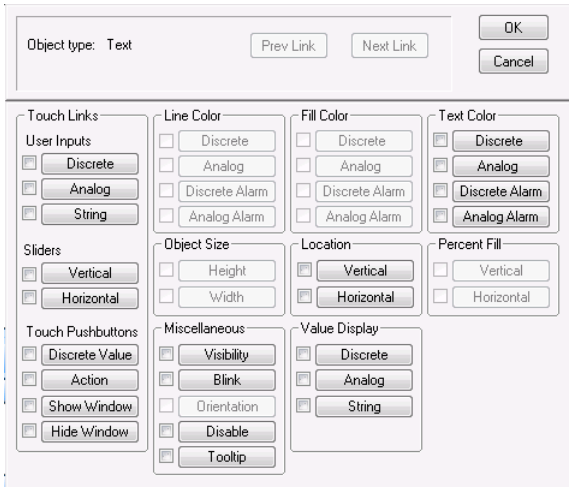
また、InTouch HMI から Galaxy ネームスペースを参照するには、Integrated Development Environment（IDE）をインストールする必要があります。

InTouch HMI では、プラットフォームのメッセージ交換機能を使用して、Galaxy ネームスペースの表示や優れた通信機能を実現しています。

リモートタグ ソースとして Galaxy を使用するように InTouch を設定するには

1. WindowMaker でアプリケーション ウィンドウを開きます。
2. テキスト オブジェクトをダブルクリックします。

   ［アニメーション リンク］ダイアログ ボックスが表示されます。

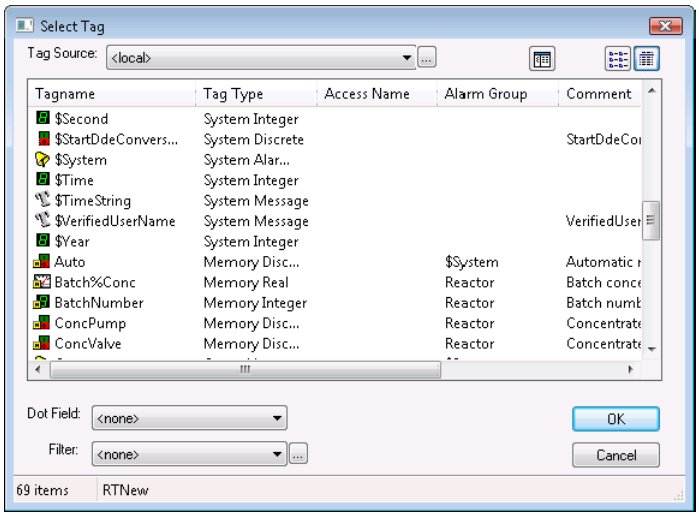


3. ［値の表示］領域で、［アナログ］を選択します。

   式を挿入するダイアログ ボックスが表示されます。
4. ［式］ボックスに表示されている式をすべて削除します。
5. ［式］ボックス内をダブルクリックします。

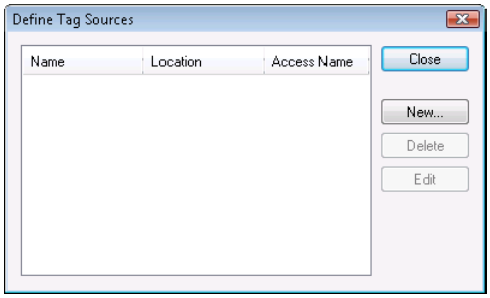
   ［タグの選択］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

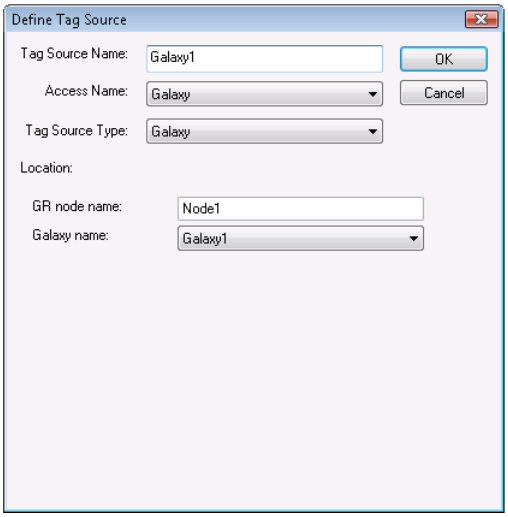


6. ［タグ ソース］ボックスの右のボタンを選択します。

   ［タグ ソースの一覧］ダイアログ ボックスが表示されます。



7. ［新規］をクリックすると［タグソースの一覧］ダイアログ ボックスが表示されます。

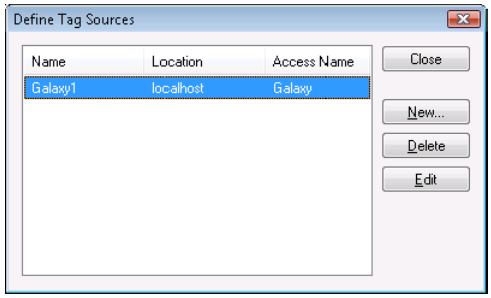


8. ［タグ ソースの一覧］ダイアログ ボックスに値を入力します。以下の手順を実行します。
   a. ［タグ ソース名］ボックスに、リモート Galaxy タグ ソースの名前を入力します。
   b. ［アクセス名］ボックスのリストから Galaxy を選択します。
   c. ［タグ ソース タイプ］ボックスのリストから Galaxy を選択します。

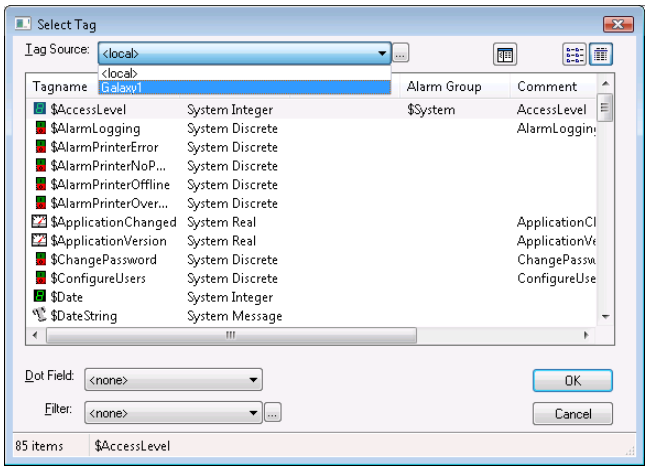
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

d. ［場所］領域では Galaxy レポジトリ ノード名を入力してから、リストから Galaxy を選択します。

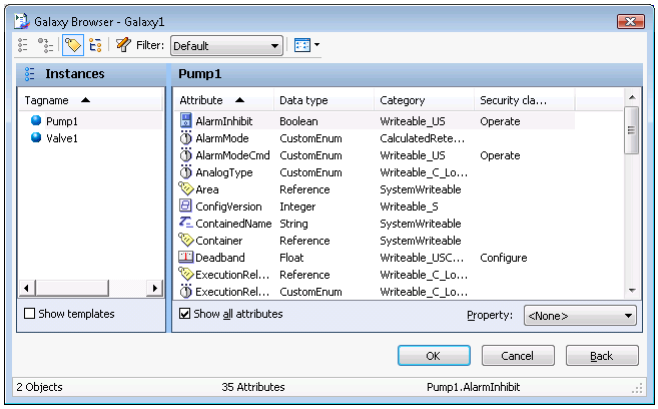
e. **[OK]** を選択します。［タグ ソースの一覧］ダイアログ ボックスに、リストで定義したリモート タグ ソースが表示されます。



9. ［閉じる］をクリックして［タグ ソースの一覧］ダイアログ ボックスを閉じます。［タグの選択］ダイアログ ボックスの［タグ ソース］ボックスのリストに、新しいタグ ソースが表示されます。



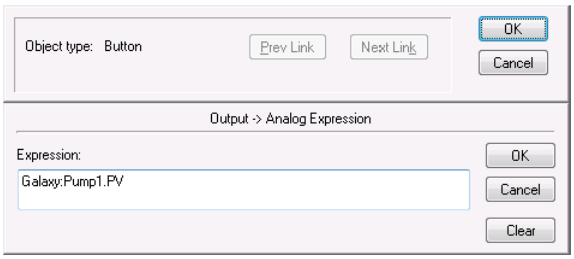
10. ［タグ ソース］ボックスで、作成した新しいタグ ソースを選択します。［**Galaxy ブラウザ**］ダイアログ ボックスが開き、左ペインにタグのリストが表示されます。



11. ［**Galaxy ブラウザ**］ダイアログ ボックスの左ペインでタグを選択します。［**Galaxy ブラウザ**］ダイアログ ボックスの右ペインが、選択したタグの属性で更新されます。

12. 使用する属性を強調表示して、**[OK]** を選択します。［アナログ値表示リンク］ダイアログ ボックスが開き、［式］ボックスに式が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



13. 文字列式が正しいことを確認します。

    式には次の形式が使用されます。

    Galaxy:tag_name.attribute_name

    例：

    ```
    Galaxy:Pump1.PV
    ```

14. ［**OK**］を選択して、［**アナログ値表示リンク**］ダイアログ ボックスを閉じます。
15. 必要に応じて、残りのオブジェクト リンクを設定します。
16. アニメーション リンクのダイアログ ボックスで、［OK］をクリックします。
17. **ランタイム**を選択します。テキスト オブジェクトに、設定したタグの属性に対する値が表示されます。

I/O タグ変数の時間スタンプと品質情報の表示

InTouch HMI は、VTQ 対応のクライアントに配信されるすべてのデータ値に値、時間、および品質（VTQ）のインジケータを挿入します。InTouch は、トラブルシューティングに有用なデータ品質のインジケータとしてドットフィールドのセットを使用しています。

- .Value ドットフィールドには、指定したタグ変数の値が格納されています。また、このドットフィールドは、すべての InTouch タグ変数のデフォルトのドットフィールドです。他のドットフィールドが指定されない場合は、.Value ドットフィールドが使用されます。
- 一連の **Time** ドットフィールドは、タグ変数が最後に更新された時間を示す時間スタンプです。
- **Quality** ドットフィールドは、I/O タグ変数に割り当てられたデータ値の信頼性を表します。

I/O タグ変数の時間スタンプ情報の表示

Time ドットフィールドのセットは、次の形式でタグ変数に割り当てられます。

Tag_name.Time_Dotfield

.TimeDate ドットフィールド

.**TimeDate** ドットフィールドは、1970 年 1 月 1 日から I/O サーバーのタグ変数の値が最後に更新された日付までの日数を示します。

カテゴリ

タグ

使用法

```
Tag_name.TimeDate
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

Tag_name
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データ タイプ

整数型（読み取り専用）

参照項目

.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeDateString ドットフィールド

.TimeDateString ドットフィールドは、I/O サーバーのタグ変数の値が最後に更新された日時を示します。

カテゴリ

タグ

使用法

```
Tag.TimeDateString
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

メッセージ型（読み取り専用）

参照項目

.TimeDate、.TimeDay、
TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeDateTime ドットフィールド

.TimeDateTime ドットフィールドは、1970 年 1 月 1 日から I/O サーバーのタグ変数の値が最後に更新された日付までの日数を小数点以下を含めて表します。

カテゴリ

タグ

使用法

```
Tag.TimeDateTime
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データタイプ

実数型（読み取り専用）

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeDay ドットフィールド

.TimeDay ドットフィールドは、I/O サーバーのタグ変数の値が最後に更新された日付からのその月の経過日数を表します。

カテゴリ

タグ

使用法

```
Tag.TimeDay
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

有効値

値の範囲は 1 ～ 31 です。

参照項目

.TimeDate、.TimeDateString、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeHour ドットフィールド

.TimeHour ドットフィールドは、I/O サーバーのタグ変数の値が最後に更新された時刻からのその日の経過時間を表します。

カテゴリ

タグ

使用法

```
Tag.TimeHour
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

値の範囲は 0 ～ 23 です。

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeMinute ドットフィールド

.TimeMinute ドットフィールドは、I/O サーバーのタグ変数の値が最後に更新された時刻の分の部分を示します。

カテゴリ

タグ

使用法

```
Tag.TimeMinute
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

有効値

値の範囲は 0 ～ 59 です。

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMsec、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeMonth ドットフィールド

.TimeMonth ドットフィールドは、I/O サーバーのタグ変数の値が更新された時点の日付の月数（1 ～ 12）を表します。

カテゴリ

タグ

使用法

```
Tag.TimeMonth
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

値の範囲は 1 ～ 12 です。

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeMsec ドットフィールド

.TimeMsec ドットフィールドは、I/O サーバーのタグ変数の値が最後に更新された時刻のミリ秒の部分を示します。

カテゴリ

タグ

使用法

```
Tag.TimeMsec
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

有効値

値の範囲は 0 ～ 999 です。

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString、.TimeYear

.TimeSecond ドットフィールド

.TimeSecond ドットフィールドは、I/O サーバーのタグ変数の値が最後に更新された時刻の秒の部分を示します。

カテゴリ

タグ

使用法

```
Tag.TimeSecond
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

値の範囲は 0 ～ 59 です。

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.TimeTimeString、.TimeYear

.TimeTime ドットフィールド

.TimeTime ドットフィールドは、I/O サーバーのタグ変数の値が更新された時点のタイムスタンプを表し、午前 0 時から経過したミリ秒の数で示されます。

カテゴリ

タグ

使用法

```
Tag.TimeTime
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

有効値

値の範囲は 0 ～ 86399999 です。

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.TimeTimeString、.TimeYear

.TimeTimeString ドットフィールド

.TimeTimeString ドットフィールドは、I/O サーバーのタグ変数の値が最後に更新された時刻を示します。

カテゴリ

タグ

使用法

```
Tag.TimeTimeString
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

メッセージ型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.TimeTimeString、.TimeYear

.TimeYear ドットフィールド

.TimeYear ドットフィールドは、I/O サーバーのタグ変数の値が更新された年を 4 桁の数字で表します。

カテゴリ

タグ

使用法

```
Tag.TimeYear
```

パラメータ

Tag
論理型、整数型、実数型、メッセージ型、間接アナログ型、間接論理型、または間接メッセージ型の任意のタグ変数

データタイプ

整数型（読み取り専用）

有効値

4 桁の数字で表される年。

参照項目

.TimeDate、.TimeDateString、.TimeDay、.TimeDateTime、.TimeHour、.TimeMinute、.TimeMsec、.TimeMonth、.TimeSecond、.Time.Time、.TimeTimeString。

I/O タグ変数の品質情報の表示

品質ドットフィールドのセットを使用して、I/O サーバーと InTouch アプリケーション間でやり取りされるデータの完全性を確保できます。品質ドットフィールドはアイテムのデータ値の品質状況を表します。この品質属性により、ネットーワーク ノード間で転送される InTouch データの完全性の監視が極めて容易になります。

データ品質基準は、OLE for Process Control（OPC）で提案された標準に基づいており、同様にフィールドバス データ品質規格にも基づいています。

ランタイム時の数値の書式設定は、データ タイプとデータの品質に基づいて設定することができます。

品質データ形式

I/O サーバーは、6 種類の相互に独立したデータ品質状況を報告することができ、InTouch .Quality ドットフィールドに値を割り当てることによりクライアントに送信されます。品質ドットフィールドの下位 8 ビット (最下位バイト) は現在、品質 (Q)、サブステータス (S)、制限 (L) の 3 つのビット フィールドの形式で定義されており、形式は **QQSSSSLL** です。クライアント アプリケーションがサーバーと通信できない場合、。**品質**ドットフィールドの値は 0 です。

.Quality ドットフィールドを使用して I/O サーバーが報告するデータ品質状況を以下の表に示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

品質状況	10 進数値	16 進数値	最上位バイト XXXXXXXX	最下位バイト QQSSSSLL	品質	品質サブステータス	しきい値
良好	192	0x00C0	00000000	11000000	Q=3	S=0	L=0
クランプされた上限（範囲外）	86	0x0056	00000000	01010110	Q=1	S=5	L=2
クランプされた下限（範囲外）	85	0x0055	0000000	01010101	Q=1	S=5	L=1
変換不可	64	0x0040	0000000	01000000	Q=1	S=0	L=0
通信に失敗	24	0x0018	0000000	00011000	Q=0	S=6	L=0
ポイントへのアクセス不可	4	0x0004	0000000	00000100	Q=0	S=1	L=0

データ Quality ドットフィールドについて

.Quality ドットフィールドは、データが最後に受信されたときのデータ値の品質を表します。SuiteLink および DDE プロトコルは、I/O サーバーによってデータ変更が行われたときのみ、更新された品質をクライアント（たとえば、WindowViewer）に送ります。そのため、新しいデータ値を受け取ったときだけ、品質が変化したことが確認されます。一部の I/O サーバーは、データに関連付けられた品質が変化したときに、現在のそのデータ値と共に更新された品質を送ることができます。

SuiteLink および DDE プロトコルを使用して、サーバー アイテムの値の品質を直接参照できないことがあります。参照するには、I/O サーバーは Item.Quality を直接サポートする必要があります。このサポートがなければ、アイテムは通知を続けることができず、.Quality ドットフィールド値はずっと 0 のままです。

TestProt I/O サーバー シミュレータは、Item.Quality を直接サポートしていません。また、このシミュレータは、品質メニュー コマンドを使用して品質を修正したときに、新しいデータ値を送り出しません。

I/O アイテムのデータ品質を観察しようとするときに、I/O サーバーが Item.Quality のアドレス指定を直接サポートしていない場合は、サーバー アイテムを観察できるように InTouch I/O タグ変数を定義してから、InTouch タグ変数の .Quality を監視します。タグ変数の制限を超える場合には、スクリプトで IOSetRemoteReferences() 関数を使用して動的に I/O ポイントを調整する方法を検討してください。

SuiteLink および DDE プロトコルは、接続状況や I/O サーバー ステータスにおけるその他の変化をクライアントへ送る品質アイテムとはみなしません。結果として、アイテムの品質が必ずしもデータ サーバーやクライアント／サーバー接続の現在のステータスを表すとは限りません。I/O サーバーのプロセスは停止できまが、.Quality ドットフィールドの値は変化しません。また、通信リンクが切断されても、.Quality フィールドの値が変わらないこともあります。

I/O サーバーの接続を監視するには、DDE または SuiteLink の内部ステータス アイテムを使用します。

.Quality ドットフィールド

.Quality ドットフィールドは、I/O サーバーが提供するデータの品質に関する数値評価を表します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

タグ

使用法

```
Tag.Quality
```

パラメータ

Tag
論理型、整数型、実数型、間接アナログ型、またはメッセージ型の任意のタグ変数タイプ

データタイプ

整数型（読み取り専用）

有効値

値の範囲は 0 ～ 255 です。

参照項目

.QualityLimit、.QualityStatus、.QualitySubstatus

例

```
IF I0Tag.Quality <> 192 THEN
   LogMessage("このデータは適切ではありません。2 バイトの高バイトの品質がゼロであることが想定されています");
   LogMessage(".QualityStatus を確認し、この想定を無効にしてください");
ENDIF;
```

.QualityLimit ドットフィールド

.QualityLimit ドットフィールドは、接続されている I/O サーバーが提供するデータ値の品質しきい値を表します。

カテゴリ

タグ

使用法

```
Tag.QualityLimit
```

パラメータ

Tag
論理型、整数型、実数型、間接アナログ型、またはメッセージ型の任意のタグ変数タイプ

データタイプ

整数型（読み取り専用）

有効値

0 = 限度なし

1 = 下限

2 = 上限

3 = 一定

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.Quality

.QualityLimitString ドットフィールド

.QualityLimitString ドットフィールドは、接続されている I/O サーバーが提供するデータ値の品質しきい値文字列を表します。

カテゴリ

タグ

使用法

```
Tag.QualityLimitString
```

パラメータ

Tag
論理型、整数型、実数型、間接アナログ型、またはメッセージ型の任意のタグ変数タイプ

データタイプ

メッセージ型（読み取り専用）

参照項目

.Quality、.QualityLimit

.QualityStatus ドットフィールド

.QualityStatus ドットフィールドは、I/O サーバーが提供するデータ値の整数型の品質ステータスを表します。

カテゴリ

タグ

使用法

```
Tag.QualityStatus
```

パラメータ

Tag
論理型、整数型、実数型、間接アナログ型、またはメッセージ型の任意のタグ変数タイプ

データタイプ

整数型（読み取り専用）

有効値（SSSS）

0 = 不良

1 = 不明

3 = 良好

例

```
IF I0Tag.QualityStatus <> 3 THEN
   LogMessage("このデータは適切ではありません。");
ENDIF;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.Quality、.QualitySubStatus

.QualityStatusString ドットフィールド

.QualityStatusString ドットフィールドは、I/O サーバーが提供するデータ値の品質ステータス文字列を表します。

カテゴリ

タグ

使用法

```
Tag.QualityStatusString
```

パラメータ

Tag
論理型、整数型、実数型、間接アナログ型、またはメッセージ型の任意のタグ変数タイプ

データタイプ

メッセージ型（読み取り専用）

参照項目

.QualityStatus、.QualitySubStatus、.Quality

.QualitySubstatus ドットフィールド

.QualitySubstatus ドットフィールドは、I/O サーバーが提供するデータ値の品質サブステータスを表します。

カテゴリ

タグ

使用法

```
Tag.QualitySubstatus
```

パラメータ

Tag
論理型、整数型、実数型、間接アナログ型、またはメッセージ型の任意のタグ変数タイプ

データタイプ

整数型（読み取り専用）

有効値（SSSS）と（QQ）

不良品質（QQ=0）のサブステータス（SSSS）。

0 = 非特定

1 = 設定エラー

2 = 未接続

3 = デバイスの故障

4 = センサーの故障

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5 = 最後に知られている値

6 = 通信エラー

7 = 動作不能

不明品質（QQ=1）のサブステータス（SSSS）。

0 = 非特定

1 = 最後に知られている値

4 = センサーが不正確

5 = 工学値の超過

6 = サブノーマル

良好品質（QQ=3）のサブステータス（SSSS）。

0 = 非特定

6 = ローカル オーバーライド

参照項目

.QualityStatus、.QualitySubStatus、.Quality

.QualitySubstatusString ドットフィールド

.QualitySubstatusString ドットフィールドは、I/O サーバーが提供するデータ値の品質ステータス文字列を表します。

カテゴリ

タグ

使用法

```
Tag.QualitySubstatusString
```

パラメータ

Tag
論理型、整数型、実数型、間接アナログ型、またはメッセージ型の任意のタグ変数タイプ

データタイプ

メッセージ型（読み取り専用）

参照項目

.QualityStatus、.QualitySubstatus、.Quality

ランタイム時の I/O 接続の初期化と再設定らんたいむじの IO せつぞくのしょきかとさいせってい

WindowViewer は、InTouch アプリケーションが起動されたときにすべての I/O 通信を初期化します。また、InTouch アプリケーションの動作中に手動で I/O 通信を再起動することもできます。ランタイム中に I/O 通信の初期化と再設定を行うには、WindowViewer のコマンドまたはスクリプトを使用します。

また、デフォルト値に基づいて I/O 接続を再初期化する必要があることを指定することもできます。このオプションを選択すると、アクセス名の再初期化時にはデフォルトの設定が使用されて現在の設定は

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

無視されます。アクセス名によって I/O 通信を再初期化するには、InTouch アプリケーションのアクセス名を事前に定義しておく必要があります。

コマンドによる I/O 接続の再初期化

WindowViewer の［**システム**］メニューには、すべての I/O 通信を再初期化したり、特定の I/O 通信を選択するためのコマンド セットが用意されています。

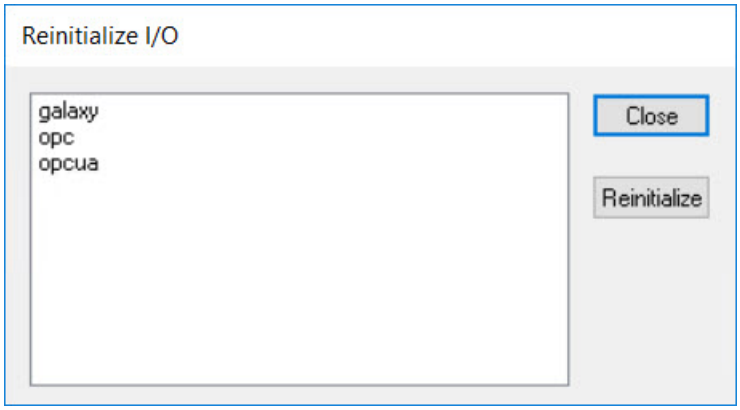
InTouch のデフォルト設定を使用して、アクセス名を再初期化することができます。デフォルトの再初期化を行うとアクセス名は、ノード名、アプリケーション名、およびトピックに現在指定されている値を無視します。アクセス名は、元のアクセス名の設定で再初期化されます。

ランタイム時にすべてのアクセス名を再初期化するには

1. ［システム］メニューで、［連結中の I/O 通信の再初期化］をクリックします。
2. **すべて再初期化** を選択します。全アクセス名が再初期化されます。

ランタイム時にアクセス名を選択して再初期化する

1. ［**特別**］メニューで、［**I/O 通信の再初期化**］をクリックして、［**選択**］をクリックします。［**I/O の再初期化**］ダイアログ ボックスに、アクセス名のリストが表示されます。



2. 再初期化するアクセス名を 1 つまたは複数選択して、［再初期化］をクリックします。 選択したアクセス名が再初期化されます。

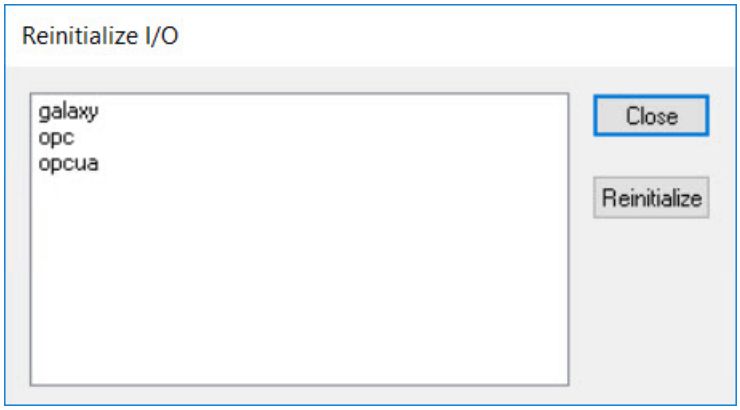
デフォルト設定を使用してアクセス名を再初期化する

1. WindowMaker 内でアプリケーションを開きます。
1. ［**ファイル**］メニューで、［**設定**］をポイントし、［**WindowViewer**］をクリックします。**WindowViewer の設定**画面が表示されます。
2. ［**環境設定**］タブの［**I/O**］領域で、［**デフォルトの再初期化**］チェック ボックスをオンにします。

3. **[OK]** を選択します。
4. WindowViewer でアプリケーションを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［特別］メニューで、［**I/O 通信の再初期化**］をクリックして、［選択］をクリックします。［**I/O の再初期化**］ダイアログ ボックスが表示されます。



6. 再初期化するアクセス名を 1 つまたは複数選択して、［再初期化］をクリックします。ノード名、アプリケーション名、およびトピックの現在の設定は無視されます。アクセス名は、元のアクセス名の設定で再初期化されます。

スクリプトによる I/O 接続の再初期化

以下に示す関数を使用したスクリプトを記述することにより、1 つまたは複数のアクセス名への I/O 接続を再初期化できます。

- **IOReinitAccessName()**
- **IOReinitialize()**
- **IOStartUninitConversations()**

IOReinitAccessName() 関数

IOReinitAccessName() 関数は、特定のアクセス名への I/O 接続を再初期化します

カテゴリ

I/O 通信

構文

```
IOReinitAccessName("AccessName", Default);
```

引数

AccessName
再初期化するアクセス名

Default
Default = 1。WindowMaker から割り当てられた元のデフォルト アクセス名の値を再初期化に使用します。
Default = 0。アクセス名に割り当てられた現在のノード、アプリケーション、トピックの値を再初期化に使用します。

備考

デフォルト設定は、アクセス名設定パネルおよび WindowViewer の設定（I/O 通信リトライ時間、ローカルのサーバーを起動、デフォルトの再初期化）の設定内容によって決まります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下の例は、ノード、アプリケーション、およびトピックに割り当てられたデフォルト値を使用して、AccessName1 への I/O 接続を再初期化します。

```
IOReinitAccessName("AccessName1", 1);
```

以下の例は、ノード、アプリケーション、およびトピックに割り当てられた現在の値を使用して、AccessName2 への I/O 接続を再初期化します。

```
IOReinitAccessName("AccessName2", 0);
```

IOReinitialize() 関数

IOReinitialize() 関数は、InTouch アプリケーションに対して定義されたすべてのアクティブな I/O 接続を終了してから再起動します。

カテゴリ

その他

構文

```
IOReinitialize()
```

引数

なし。

備考

IOReinitialize() 関数が実行する処理は、WindowViewer の［**システム**］メニューの［**連結中の I/O 通信の再初期化**]コマンドと同じです。

WindowViewer がサービスとして実行している場合、**IOReinitialize()** 関数と WindowViewer の［**システム**］メニューの［**すべて再初期化**］コマンドではすべてのアクセス名を初期化することはできません。［**I/O 通信の再初期化**］ダイアログ ボックスにリストされているアクセス名を選択して［**再初期化**］を選択すると、選択したアクセス名が再初期化されます。

［**連結中の I/O 通信の再初期化**］ダイアログ ボックスへのナビゲーションの詳細については、「コマンドによる I/O 接続の再初期化」を参照してください。

例

以下の例では、InTouch アプリケーションに対して定義されたすべてのアクティブな I/O 接続を終了してから再起動します。

```
IOReinitialize();
```

IOStartUninitConversations() 関数

WindowViewer で InTouch アプリケーションが起動されると、すべての I/O 通信を開始するための初期化要求が自動的に処理されます。プログラムが I/O サーバーの初期化要求に応答しない場合は、**IOStartUninitConversations()** スクリプト関数を使用して WindowViewer に I/O 通信の起動を再試行させることができます。

カテゴリ

その他

構文

```
IOStartUninitConversations();
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

なし。

備考

IOStartUninitConversations() 関数が実行する処理は、WindowViewer の［**システム**］メニューの［**未連結の I/O 通信の再初期化**]コマンドと同じです。

例

以下の例は、WindowViewer アプリケーションに対して定義されたすべての I/O 接続を起動するための初期化要求を再発行するように InTouch に指示します。

```
IOStartUninitConversations();
```

アクセス名によるフェイルオーバー機能の使用

メイン I/O サーバーに通信障害が発生した場合に、自動的にセカンダリ I/O サーバーに切り替わるように InTouch HMI を指定できます。この機能は、I/O フェイルオーバーと呼ばれています。

フェイルオーバーの設定

InTouch アプリケーションがメイン I/O サーバーと通信できなくなった場合に、フェイルオーバー用のセカンダリ I/O サーバーに切り替わるように指定できます。

フェイルオーバーの設定では、フェイルオーバー デッドバンドの指定を行います。フェイルオーバー デッドバンドは、メイン アクセス名からセカンダリ アクセス名に切り替わるまでの秒単位の遅延時間です。InTouch HMI は、式の評価が True の状態または I/O 通信障害の状態がデッドバンドの期間続いた場合にフェイルオーバーをトリガします。フェイルオーバー デッドバンドを 0 か空白に設定した場合、I/O 通信障害が検出されると直ちにフェイルオーバーがトリガされます。

アクセス名に対してフェイルオーバーを設定する

1. 必要に応じて、WindowViewer を停止します。
2. ［**ホーム**］メニューの［**タグ**］グループで［**アクセス名**］をクリックします。

   ［**アクセス名**］ダイアログ ボックスが開き、すべての定義済みアクセス名のリストが表示されます。
3. リストからアクセス名を選択してフェイルオーバー サーバーを追加します。
4. **編集**を選択します。

   ［**アクセス名の追加**］が表示されます。
5. ［**セカンダリ ソース**］セクションで［**セカンダリ ソースを有効にする**］チェック ボックスを選択します。
6. 以下の手順を実行します。
   - ［**ノード名**］ボックスに、セカンダリ I/O サーバーのノード名を入力します。
   - ［**アプリケーション名**］ボックスに、データの取得元となるセカンダリ I/O サーバー プログラムの名前を入力します。
   - ［**トピック名**］ボックスに、セカンダリ I/O ソースからアクセスするトピック名を入力します。
   - ［**プロトコル名**］領域で、**DDE** または SuiteLink のいずれかをセカンダリ I/O サーバー通信プロトコルとして選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［サーバーへの通知］領域で、セカンダリ I/O ソースに対して［すべてのアイテムを要求］ または ［有効アイテムのみ要求］ を選択します。

7. ［フェイルオーバー］セクションで［ファイルオーバーを有効化］チェック ボックスを選択します。
8. 以下の手順を実行します。
   - オプションのフェイルオーバー式を入力するか、［フェイルオーバー式］ボックス内をダブルクリックしてタグを選択します。フェイルオーバー式の詳細については、「バックアップ アクセス名へのフェイルオーバーの強制」を参照してください。
   - ［フェイルオーバー デッドバンド］ボックスで、フェイルオーバー デッドバンドの長さを秒単位で入力します。
   - フェイルオーバー状態が解除された後で、セカンダリ アクセス名からメイン アクセス名に戻す切り替え機能を有効にする場合は、［フェイルオーバー条件の解除時にメインへ切り替える］をオンにします。

     デフォルトでは、メイン アクセス名に戻す機能は無効です。［フェイルオーバー条件の解除時にメインへ切り替える］をオンにすると、［フェイルオーバー設定］ダイアログ ボックスでの ［フェイルバック デッドバンド］オプションの選択が可能になります。
   - ［フェイルバック デッドバンド］に、フェイルバック デッドバンドの長さを秒単位で入力します。

     InTouch HMI は、式の評価と関連するすべての I/O 通信障害の解除がデッドバンドの期間継続するとフェイルバックをトリガします。式が空白または 0 のままの場合は、I/O 通信障害の状態が解除されると直ちにフェイルバックが発生します。
9. 更新を選択します。

フェイルオーバー ペアのアクセス名パラメータを編集する

フェイルオーバー ペアの一部であるアクセス名パラメータを編集するには、セカンダリ I/O ソースを持つフェイルオーバーにアクセス名が設定されている必要があります。

フェイルオーバー ペアのアクセス名パラメータを編集する

1. 必要に応じて、WindowViewer を停止します。
2. ［ホーム］メニューの［タグ］グループで［アクセス名］をクリックします。

   ［アクセス名］ダイアログ ボックスが表示されます。
3. アクセス名ペアを強調表示して、［編集］を選択します。

   ［アクセス名の追加］ダイアログ ボックスに、メインとセカンダリのアクセス名のパラメータが表示されます。
4. メインとセカンダリのアクセス名パラメータを変更します。
5. 更新を選択します。

アクセス名へのフェイルオーバーの削除

アクセス名へのフェイルオーバーを削除するには、セカンダリ I/O ソースを持つフェイルオーバーにアクセス名が設定されている必要があります。

アクセス名ペアへのフェイルオーバーの削除

1. ［ホーム］メニューの［タグ］グループで［アクセス名］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. アクセス名ペアを選択して、［編集］をクリックします。

   ［アクセス名の追加］ダイアログ ボックスが表示されます。

3. ［セカンダリ ソースを有効にする］ チェック ボックスをオフにします。

4. **[OK]** を選択します。

   アクセス名ペアへのフェイルオーバーが無効になります。

バックアップ アクセス名へのフェイルオーバーの強制

フェイルオーバーが発生しなくても、アクセス名のソースをメインとセカンダリ間で手動で切り替えることができます。この機能は、フェイルオーバーの強制と呼ばれています。フェイルオーバーを強制するには、セカンダリ I/O ソースを持つフェイルオーバーにアクセス名が設定されている必要があります。

フェイルオーバー式または **IOForceFailover()** スクリプト関数を使用して、フェイルオーバーを強制することができます。

フェイルオーバー式

［フェイルオーバー設定］セクションに、フェイルオーバーをトリガするタグや式を指定するための［フェイルオーバー式］オプションが表示されます。［フェイルオーバー式］の値として入力されたフェイルオーバー メモリ論理型タグを次の図に示します。

フェイルオーバー タグを True に設定するなどによりフェイルオーバー式が True に設定されると、アクセス名の I/O データ ソースはメイン（False）からセカンダリ（True）に切り替わります。

IOForceFailover() 関数

IOForceFailover() スクリプト関数は、アクセス名のデータ ソースをメインとセカンダリ間で切り替えます。また、このスクリプト関数が呼び出されるたびに、アクティブな I/O ノードはメイン ノードとセカンダリ ノード間で切り替わります。

IOForceFailover() 関数は通常、ボタンや他のウィンドウ オブジェクトに関連付けられたスクリプトの一部として記述されます。ユーザーがアプリケーション ウィンドウからこのオブジェクトを選択すると、フェイルオーバーが強制されます。再度このオブジェクトを選択すると、**IOForceFailover** 関数によって I/O 接続は以前のアクティブ I/O ノードに戻されます。

カテゴリ

I/O 通信

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
IOForceFailover("AccessName");
```

引数

アクセス名
フェイルオーバーに設定されているアクセス名

例

アクセス名 Acc1 はメインとセカンダリのデータ ソースを持ち、メインがアクティブです。以下のスクリプトを実行すると、Acc1 はセカンダリ データ ソースにフェイルオーバーします。

```
IOForceFailOver("Acc1");
```

フェイルオーバー機能の一時的な無効化

アクセス名の I/O ノードをメインとセカンダリ間で切り替えるフェイルオーバーを、手動で無効にすることができます。通常は、InTouch システムのコンポーネントが起動されてから準備完了状態になるまでの短期間に、フェイルオーバーの一時的な無効化が利用されます。コンポーネントが安定化した後で、フェイルオーバーの切り替え機能を回復できます。

アクセス名へのフェイルオーバーを無効化するには、セカンダリ I/O ソースを持つフェイルオーバーにアクセス名が設定されている必要があります。

手動でフェイルオーバーの切り替え機能を無効にするには、2 つの方法があります。

- ［**フェイルオーバー**］セクションで［**フェイルオーバー不可**］オプションを選択します。
- **IODisableFailover()** 関数を記述したスクリプトを実行します。

［フェイルオーバー不可］設定オプション

アクセス名がプライマリとセカンダリ I/O ノードの間で切り替わらないようにするには、［**フェイルオーバー設定**］セクションで、［**フェイルオーバーを有効化**］オプションをオフにします。

［**フェイルオーバー不可**］オプションを有効に設定するには、アクセス名の定義を編集しなければなりません。このオプションがアクセス名に対して有効である間は、フェイルオーバーは無効です。

IODisableFailover() スクリプト関数

IODisableFailover() 関数をスクリプトに使用して、特定のアクセス名に対するフェイルオーバーを無効にできます。**IODisableFailover()** は、**IOForceFailover()** スクリプト関数による方法を除く、すべてのフェイルオーバー手段による切り替えを無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

I/O 通信

構文

```
IODisableFailover("AccessName",Option);
```

引数

AccessName
フェイルオーバーに設定されているアクセス名

Option
1 = フェイルオーバー不可
0 = フェイルオーバー可

備考

アクセス名はリテラル文字列で指定するか、他の InTouch タグ変数または関数が返す文字列の値で指定できます。

例

以下の例は、アクセス名 ModbusPLC1 に対するフェイルオーバーを無効にします。

```
IODisableFailover ("ModbusPLC1",1)
```

以下の例は、アクセス名 ModbusPLC1 に対するフェイルオーバーを有効にします。

```
IODisableFailover ("ModbusPLC1",0)
```

スクリプトによるフェイルオーバー ペア関連情報の取得

スクリプトを記述して、アクセス名のメイン、セカンダリ、およびアクティブな I/O ソースのステータスを返す関数を使用することができます。ユーザーは通常、フェイルオーバーの強制を実行する前にアクセス名のセカンダリ I/O ソースのステータスを確認するスクリプトを実行します。

アクセス名情報を返すスクリプトを記述するには、セカンダリ I/O ソースを持つフェイルオーバーにアクセス名が設定されている必要があります。

IOGetAccessNameStatus() 関数

IOGetAccessNameStatus() スクリプト関数は、アクセス名のメイン、セカンダリ、またはアクティブな I/O ソースの接続ステータスを示す整数値を返します。

IOGetAccessNameStatus() の戻り値は、通常は整数型タグ変数に関連付けられます。このタグ変数の値により、アクティブ、メイン、およびセカンダリの、アクセス名の I/O ソースのステータスをユーザーに示す論理値表示アニメーション リンクを動作させることができます。

カテゴリ

その他

構文

```
Result=IOGetAccessNameStatus("AccessName", Mode);
```

引数

AccessName
ステータスを返す既存のアクセス名。

Mode

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

この引数に割り当てる値に応じて、どのフェイルオーバー ペアのアクセス名に対して現在のステータスに関するクエリーを実行するかが決まります。
0 - アクティブなアクセス名 I/O ソースのステータス
1 - アクセス名のメイン I/O ソースのステータス
2 - アクセス名のセカンダリ I/O ソースのステータス

結果

返り値	説明
-1	アクセス名に設定エラーがあります。アクセス名が存在しないか、またはアクセス名にセカンダリ I/O ソースが定義されていません。
0	要求した I/O ソースとの接続に失敗しました。
1	要求した I/O ソースとの接続に成功しました。

備考

IOGetAccessNameStatus() 関数は通常、現在非アクティブなセカンダリ I/O ソースのステータスを確認するスクリプトに使用されます。ユーザーはこのスクリプトを実行して、フェイルオーバーの強制を実行する前にセカンダリ接続のステータスを確認します。

例

以下の例は、アクセス名 ModbusPLC1 のセカンダリ I/O ソースのステータスを返します。返り値は **ANStatus** タグ変数に関連付けられます。

```
ANStatus = IOGetAccessNameStatus ("ModbusPLC1",2)
```

IOGetActiveSourceName() 関数

IOGetActiveSourceName() スクリプト関数は、アクセス名が現在プライマリ データ ソースまたはセカンダリ データ ソースを使用しているかどうかを返します。

IOGetActiveSourceName() 関数は通常、ボタンや他のウィンドウ オブジェクトに関連付けられたスクリプトに記述されます。ユーザーは、アプリケーション ウィンドウからこのオブジェクトを選択して、アプリケーションの I/O Servers のステータスを要求します。

カテゴリ

その他

構文

```
Result=IOGetActiveSourceName("AccessName");
```

引数

AccessName
ソース名を返す既存のアクセス名

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

IOGetActiveSourceName() は、アクセス名のメインまたはセカンダリのノードがアクティブにポーリングされているかどうかを示す文字列を返します。**IOGetActiveSourceName()** 関数の返り値を以下に示します。

メイン	アクセス名のメイン ノードがアクティブにポーリングされています。
セカンダリ	アクセス名のセカンダリ ノード、つまりフェイルオーバー ノードがアクティブにポーリングされています。
Null	アクセス名のメイン ノードおよびセカンダリ ノードとも非アクティブです。

例

以下の例では、アクセス名 ModbusPLC1 の現在のアクティブ ノードを示す返り値（メイン、セカンダリ、または Null）が、メッセージ型タグ変数 **ActiveServer** に割り当てられます。

```
ActiveServer = IOGetActiveSourceName ("ModbusPLC1");
```

I/O 接続ステータスの監視

WindowViewer には IOStatus というビルトインのトピックが用意されており、PLC と通信している I/O サーバーと InTouch アプリケーション間の特定の I/O 通信のステータスを監視できます。

注：InTouch の 7.0 以前のバージョンでは、トピック名は DDEStatus でした。

I/O 通信を監視するように **IOStatus** トピックを設定することができます。

I/Ostatus トピック名の使用

IOStatus トピックを準備して、WindowViewer と I/O サーバー間の I/O 通信を監視できます。この例では、WindowViewer は、I/O サーバー内でトピック名として PLC1 に定義されている PLC に対するシミュレーション I/O サーバーと通信します。

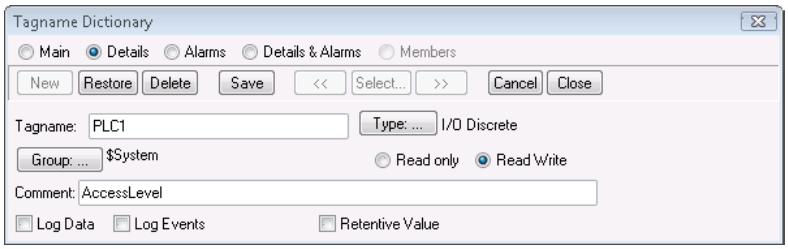
注：Simulation Server サーバーは、トレーニング用ツールとして使用できる汎用 DAServer で、c:\program files\common files\ArchestrA フォルダ内にあります。

I/O 通信のステータスを監視するには

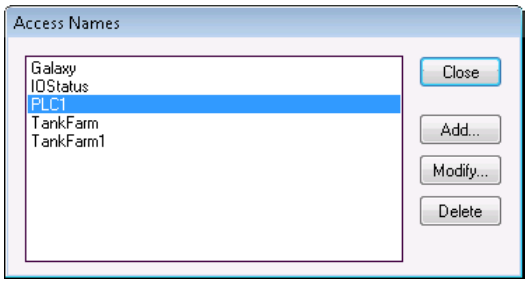
1. WindowMaker でアプリケーションを開きます。
2. タグ名ディクショナリを開きます。
3. I/O 論理型タグを作成します。

   IOStatus を使用して I/O 通信を監視する場合、監視対象のアクセス名に対して少なくとも 1 つの I/O 型タグを定義する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

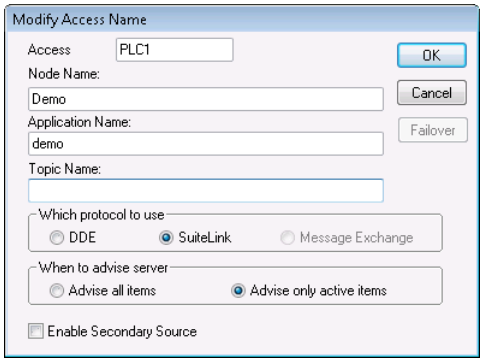


4. トピック名として IOStatus を定義しているアクセス名の定義にタグ変数を割り当てるために、[**アクセス名**] を選択します。



この時点で、**PLC1** のアクセス名の定義は存在しています。

5. PLC1 を強調表示し、**[Modify]**を選択 します。



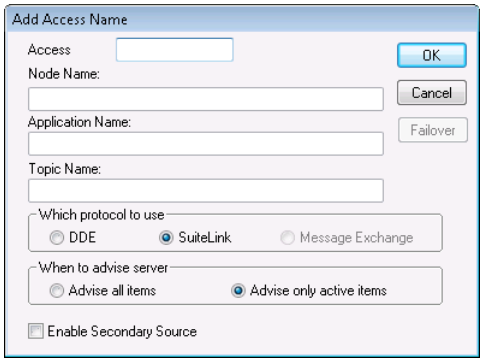
この例では、タグとトピック名が同じであるため、正しいトピック名を含むアクセス名を容易に見つけ出すことができます。

6. [**キャンセル**] を選択してダイアログ ボックスを閉じ、 [**アクセス名**] ダイアログ ボックスに戻ります。

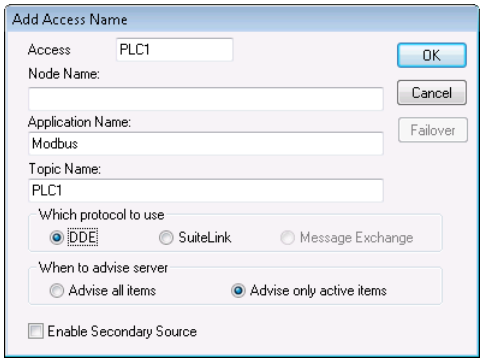
7. **追加**を選択します。

[**アクセス名の追加**] ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



8. 以下の手順を実行します。
   a. ［**アクセス名**］ボックスに、IOStatus と入力します。
   b. ［**アプリケーション名**］ボックスに、**View** と入力します。これにより WindowViewer でステータスが監視されます。
   c. ［**トピック名**］ボックスに、InTouch の内部トピックとして **IOStatus** と入力します。
   d. ［**有効アイテムのみ要求**］を選択します。
9. ［**OK**］をクリックして、ダイアログ ボックスを閉じます。［**アクセス名**］ダイアログ ボックスが再表示され、新しいアクセス名、IOStatus がリストに表示されます。



10. ［**閉じる**］をクリックしてダイアログ ボックスを閉じると、新しいアクセス名が I/O 論理型タグ変数に関連付けられます。

    ［アイテム名］ボックスに、監視する実際のトピック名のアクセス名を入力します。

11. タグがトピック名と同じであるため、［タグ名をアイテム名として使用］を選択すると、トピック名がアイテム名として自動的に入力されます。

注：ビルトインのトピック IOStatus（InTouch 7.0 以前のバージョンでは DDEStatus）を使用して I/O 通信を監視する場合、［アクセス名］ボックスに入力する名前が常にアイテム名としても使用されます。

アイテム名は、最大 254 文字の文字列です。アイテム名の長さは、トピック名、アイテム名、および 1 の区切り文字の合計として計算されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Excel での IOStatus トピックの使用

Excel のスプレッドシート セルの式に同じ情報を入力することにより、I/O ステータスを監視できます。たとえば、前記手順の説明と同じトピックを監視するには、セルに以下の式を入力します。

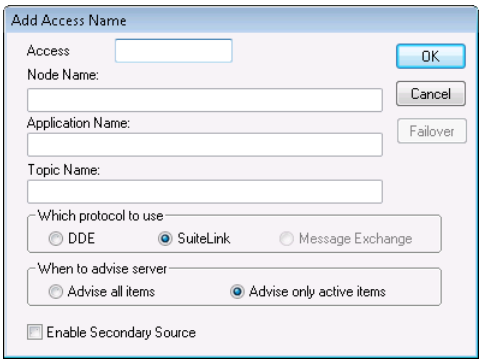
```
=view|IOStatus!PLC1'
```

I/O 通信ステータスの監視

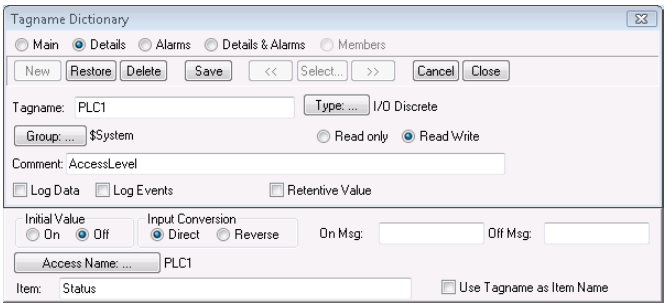
使用されている各トピック名に対してビルトインの論理型アイテム **Status** を使用して、I/O サーバー プログラムとの通信のステータスを監視できます。通信障害が発生すると、**Status** アイテムは 0 に設定されます。I/O サーバー プログラムとの通信が正常の場合は、**Status** アイテムは 1 に設定されます。

注：IOStatus アイテムを使用してトピックのステータスを監視する場合、監視対象のトピックに対して少なくとも 1 つの I/O ポイントがアクティブであることが必要です。

タグ変数を定義して、アイテム名に **Status** という語を使用してデバイス用に設定されたトピックをそのタグ変数に関連付けることにより、InTouch HMI からサーバー通信のステータスを読み取ることができます。たとえば、WindowViewer が Modbus DAServer を使用して PLC と通信している場合、アクセス名の定義は以下の例に類似したものになります。



トピック PLC1 へのすべての通信ステータスを監視するには、タグ変数の定義を次のように作成します。



Excel 使用時には、以下の式をセルに入力して PLC 通信ステータスを読み出すことができます。

```
=シミュレーション|PLC1!ステータス
```

他のアプリケーションからの InTouch タグ変数データへのアクセス

他のアプリケーションが InTouch HMI のデータ値を要求する場合、3 つの I/O アドレスのアイテムを認識する必要があります。以下に InTouch の I/O アドレス 形式について説明します。

VIEW（アプリケーション名）は、データ要素を含む InTouch のランタイム プログラムを識別します。

TAGNAME（トピック名）は、タグ変数の読み取り／書き込み時に常に使用する名前です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ActualTagname（アイテム名）は、InTouch タグ変数ディクショナリでアイテムに対して定義された実際のタグ変数です。

たとえば、Excel から InTouch HMI のデータ値にアクセスするには、データを書き込むセルに DDE リモート リファレンスの式を指定します。

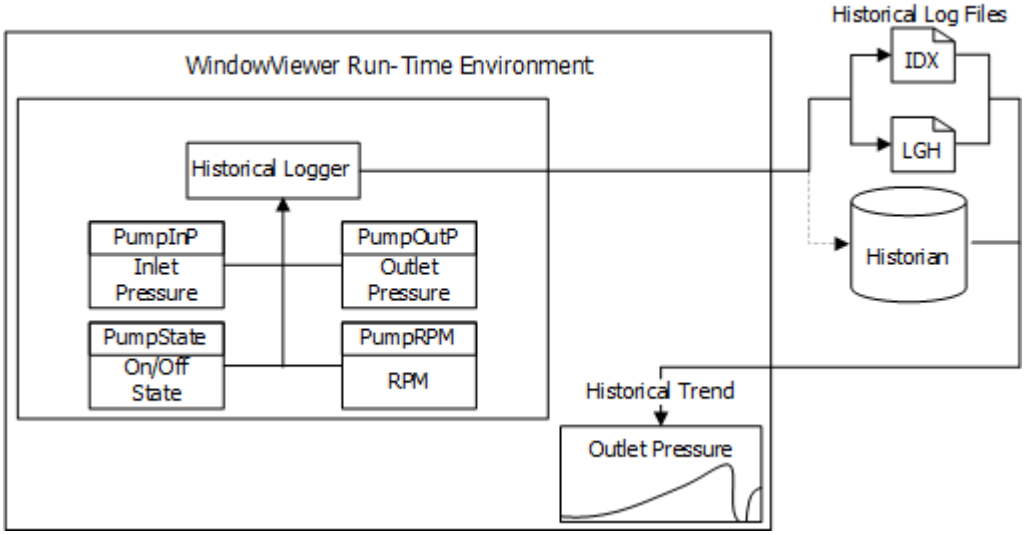
=VIEW|TAGNAME!ActualTag_name

タグ値の記録

InTouch アプリケーションの実行中に、タグをログ記録し、永続的に保存できます。このログ データを使用すると、一定期間に行われるプラント プロセスのいくつかの側面を表示する履歴トレンド グラフを作成できます。

注：大きなアプリケーションで InTouch の履歴データを保存する場合や詳細なレポートを作成する場合は、Historian Server を使用する必要があります。履歴ログ記録の設定の詳細については、Historian Server のマニュアルを参照してください。

以下の図は、履歴ログ ファイルおよび Historian Server データベースに保存されたポンプからのタグ データを示しています。InTouch 履歴ロガーは、タグの値が変化して、指定したログのデッドバンド範囲を超えるたびにログ エントリを書き込みます。



履歴ログ ファイルにデータを保存すると、InTouch HMI によってログ ファイルが作成されます。1 つのファイルには、専用の形式で保存されるログ データが含まれます。もう 1 つのファイルは、そのデータへのインデックスです。

この 2 つのログ ファイルには、以下の形式で名前が割り当てられます。

YYMMDD00.LGH および *YYMMDD00.IDX*

ここで、

YY	ログ ファイルが作成された年の最後の 2 桁
MM	ファイルが作成された月の 2 桁の数（01 ～ 12）
DD	ファイルが作成された日付の 2 桁の数（01 ～ 31）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

00 ログ ファイル名の定数 00

1 日のログ サイクルは深夜に開始および終了します。履歴ロガーは、深夜に最後のエントリをアクティブなログ ファイルに書き込み、そのファイルをアーカイブします。翌日には、2 つの新しいファイルが作成され、データはそのファイルにログ記録されます。

ログ ファイルは指定された日数の間保存されます。保存期間を過ぎたログ ファイルは削除されます。ログ ファイルを保持するための日数の設定の詳細については、「一般的なログ プロパティの設定 - 履歴ログ ファイル」を参照してください。

履歴ログの設定

InTouch アプリケーションで履歴ログを設定するには、主に 3 つの作業を行う必要があります。

- 履歴ログのタグ変数を設定する
- InTouch アプリケーションの一般的なログ プロパティを設定する
  - タグ変数値を InTouch 履歴ログ ファイルに保存するためのプロパティを設定する
  - または、タグ値を Historian Server に保存するためのプロパティを設定する
- （オプション）履歴ログの頻度を設定する

注：LGH ファイルおよび Historian Server へのログ記録を設定できます。

履歴ログのタグ変数を設定する

履歴ログのタグは、タグ名ディクショナリから選択します。選択したタグの値が変更されると、履歴ロガーは、各タグのログ デッドバンドとその現在の値に基づいてエントリをログに書き込むかどうかを決定します。

タグをログ記録有効からログ記録無効に変更すると、そのタグに関連付けられたデータはログ ファイルに保存されなくなります。ログ記録を再び有効にすると、ログ記録は再開されます。ただし、履歴トレンドには、ログ記録が無効になっていた間のギャップが示されます。

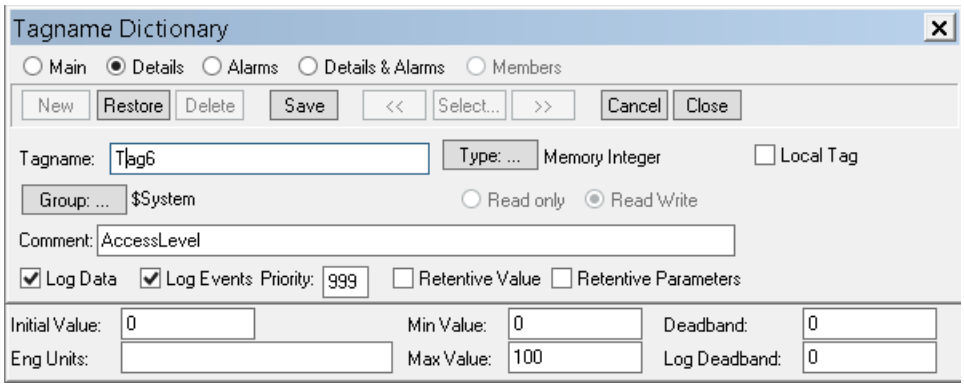
WindowViewer がアプリケーションを実行している間は、タグのログ記録への変更は無視されます。タグに対してログ記録を変更しても、稼動しているアプリケーションを停止して再起動するまでは有効になりません。

各タグのログ記録は、タグ名ディクショナリから設定します。

タグの履歴ログを有効にする

1. 必要であれば、WindowViewer がアプリケーションを実行しないようにします。
2. WindowMaker でアプリケーションを開きます。
3. タグ名ディクショナリを開きます。
4. タグ名ディクショナリ リストから、データをログに記録するタグを選択します。
5. ［**データ ログ**］チェック ボックスをオンにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



［タグ名ディクショナリ］ダイアログ ボックスには、ログ記録に密接に関連付けられたその他のタグの属性が含まれています。

- ［ログ デッドバンド］は、超過するとタグの値がログ ファイルに書き込まれる工学値のしきい値を設定します。デッドバンドの範囲外の新しい値だけが、ログ ファイルに書き込まれます。デッドバンド範囲内でのわずかな値の変化は無視されます。
- ［工学値最小値］プロパティおよび［工学値最大値］プロパティは、クランピングされた生データ値を工学値の範囲内で調整します。［工学値最小値］プロパティおよび［工学値最大値］プロパティでは、調整値の上限と下限を設定します。

  工学値最小値および工学値最大値は、トレンドに表示されるログ値の範囲の境界を決定します。デフォルトでは、InTouch 履歴トレンドにはログ データが工学値範囲の 0 ～ 100% で表示されます。

6. 保存を選択します。
7. これらの手順を繰り返して、ログ記録するデータが含まれる各タグのログ記録を有効にします。
8. 完了したら、［閉じる］をクリックして、タグ変数ディクショナリを閉じます。

一般的なログ プロパティの設定 - 履歴ログ ファイル

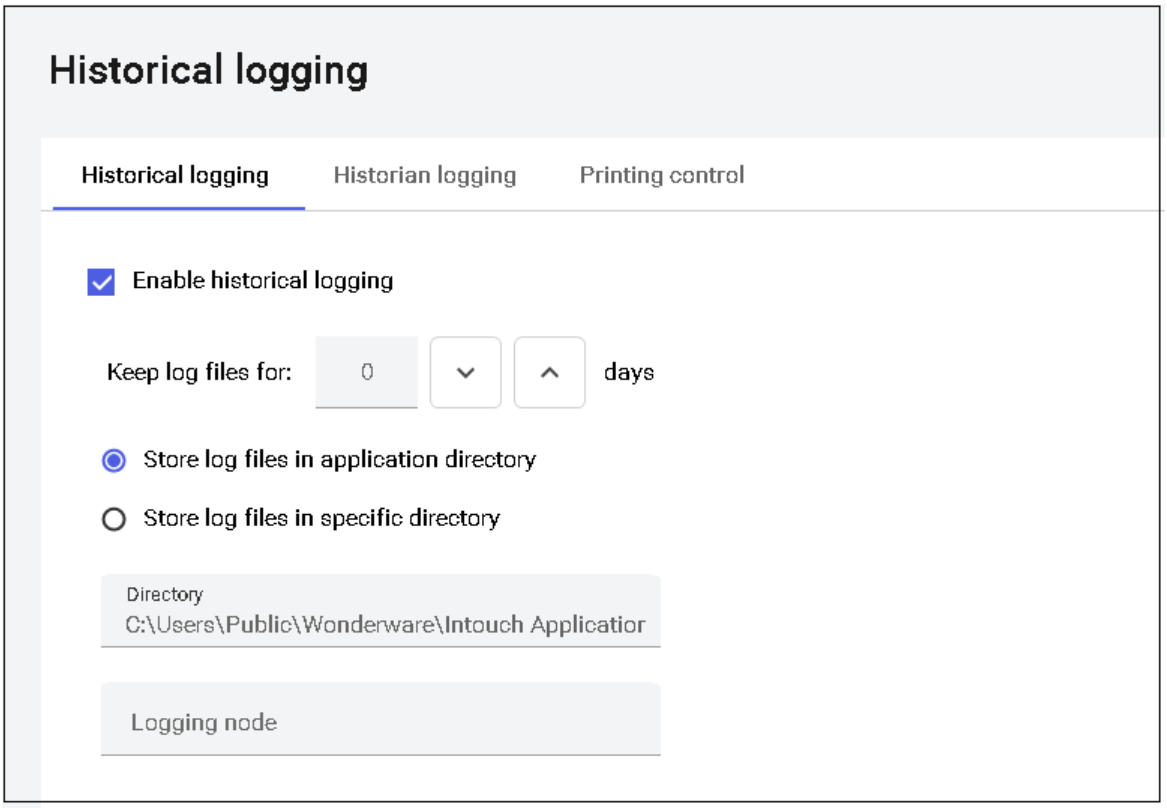
選択したアプリケーションに適用する一般的なログ記録プロパティを設定できます。

履歴ログの設定

1. 必要であれば、WindowViewer で稼動している InTouch アプリケーションを閉じます。
2. WindowMaker を開きます。
3. ［ファイル］メニューの［設定］をクリックし、［履歴ログ］をクリックします。

   ［履歴ログ］設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**履歴ログ実行**］チェック ボックスをオンにします。
5. ［**ログ ファイルの保存期間**］ボックスに、ログ ファイルを現在の日付から保持する日数を入力します。

   ログ ファイルは、現在の日付から指定した保存期間だけ保存されます。保存期間を過ぎたログ ファイルは削除されます。この値を 0 に設定すると、すべてのログ ファイルは無期限に保存されます。

   例：

   保存期間を 5 日間に設定し、月の最初の日からログ記録を開始しました。その月の 7 日の時点で、その前の 5 日分とその日の分（02 ～ 07）のログ ファイルが保持されています。月の最初の日に作成されたログ ファイルは削除されます。

   ログ データの保存日数を設定する際は、ディスク容量の使用率を考慮してください。ハード ディスクの空き容量が不足すると、履歴ログは停止します。ログ記録を再開するには、ディスク容量を解放する必要があります。
6. ログ ファイルを保存するフォルダの場所を選択します。

   注：ログ データを保存するファイルのフォルダ パスと名前は最大 55 文字です。

   ［**アプリケーション ディレクトリにログ ファイルを保存**］を選択すると、ログ データを作成している InTouch アプリケーションと同じフォルダにログ ファイルを保存できます。

   ［**特定のディレクトリにログ ファイルを保存**］を選択すると、ログ ファイルの保存先として別のフォルダを指定できます。ログ ファイルを保存するフォルダは、以下のように指定できます。

   - Windows のフォルダ パス（C:\History Log Files など）
   - UNC（汎用名前付け規則）パス（\\node\share\directory など）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ログ ファイルを分散ノードに保存している場合は、UNC パスでディレクトリを指定する必要があります。

NAD リポジトリ アプリケーション ノードのアプリケーション ディレクトリに履歴データが書き込まれるように設定されている場合、すべての NAD ノードは、履歴データを NAD リポジトリ アプリケーションに書き込もうとします。この問題を避けるには、各 NAD ノードで、NAD リポジトリ アプリケーション ノードではなくローカル ディレクトリに書き込まれるように履歴データを設定します。

7. ［**ロギング ノード**］ボックスに、ログ データを作成している InTouch アプリケーションを実行しているコンピュータのノード名を入力します。
8. **保存**を選択します。

   ログ記録設定の変更内容は、すぐに有効になります。次回アプリケーションを起動すると、ログ記録が開始されます。

トレンドの印刷の詳細については、「ランタイム中のトレンド印刷」を参照してください。

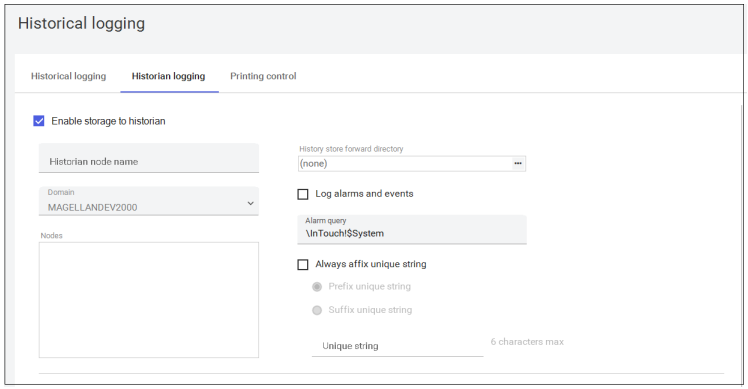
一般的なログ プロパティの設定 - Historian への保存

選択したアプリケーションに適用する一般的なログ記録プロパティを設定できます。

Historian への保存を設定する

1. 必要であれば、WindowViewer で稼動している InTouch アプリケーションを閉じます。
2. WindowMaker を開きます。
3. ［**ファイル**］メニューの［**設定**］をクリックし、［**履歴ログ**］をクリックします。

   ［**履歴ログ**］設定画面が表示されます。
4. ［**Historian ログ**］タブを選択します。



5. ［**Historian へのストレージを有効にする**］チェック ボックスをオンにします。
6. ［**Historian ノード名**］ボックスに、Historian Server を実行しているコンピュータのノード名を入力します。IP アドレスを指定することもできます。サーバーが同じマシンにインストールされている場合は、localhost を使用できます。使用できる特殊文字は、ピリオド（.）、アンダースコア、（_）、およびハイフン（-）だけです。
7. ストア フォワードに関連するファイルを保存するローカル フォルダの場所へのパスを［**履歴ストア フォワード ディレクトリ**］ボックスに入力します。これらのファイルを使用すると、Historian Server への接続が失われた場合に履歴データを一時的に保存できます。接続が確立されたら、Historian

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Server は、このストア フォワード ディレクトリのファイルと同期し、すべての情報が維持されます。

8. ［**アラームとイベントをログ**］を選択してアラームとイベントのログを有効にします。［**アラーム クエリー**］テキスト ボックスでアラーム クエリーを指定します。

   注：異なる InTouch アプリケーションがタグ データを同じ Historian Server に保存し、同じタグ名を使用した場合、そのようなシナリオは InTouch アプリケーションで検出されず、Historian データが重複することがあります。複数のアプリケーションを区別するには、一意の接頭辞または接尾辞を使用します。詳細については、「追加文字列の設定」を参照してください。

9. **保存**を選択します。

   ログ記録設定の変更内容は、すぐに有効になります。次回アプリケーションを起動すると、ログ記録が開始されます。

トレンドの印刷の詳細については、「ランタイム中のトレンド印刷」を参照してください。

注意：tagname.x ファイルを使用して、OCMC 経由で InTouch タグを Historian に手動でインポートする場合、データを履歴化するには WindowViewer を管理者モードで実行する必要があります。管理モード以外では、この方法によるデータ履歴化はサポートされていません。tagname.x ファイルを使用して InTouch タグをインポートする方法については、Historian ヘルプの「ディクショナリのインポートまたは再インポート（Import or reimport a dictionary）」トピックを参照してください。

追加文字列の設定

同じアプリケーションを複数のノードで使用し、同じ Historian Server に接続するシナリオでは、一意の接尾辞または接頭辞を使用してノードの異なるタグを区別できます。文字列は、WindowMaker から設定するか、各ノードの 'dhistcfg.ini' ファイルを手動で編集することによって設定することができます。

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］をクリックし、［**履歴ログ**］をクリックします。［履歴ログ］設定画面で［**Historian ログ**］タブを選択します。
3. ［**Historian へのストレージを有効にする**］チェック ボックスをオンにします。
4. ［**常に一意の文字列を追加する**］チェック ボックスをオンにします。
5. ［**一意の接頭辞を追加**］または［**一意の接尾辞を追加**］を選択します。
6. ［**一意の文字列**］フィールドに文字列を入力します。文字列の最大長は 6 文字まで指定できます。
7. **[OK]** を選択します。

手動での .ini ファイルの追加文字列の編集

追加文字列は、各ノードのアプリケーション フォルダ内にある dhistcfg.ini を変更して手動で更新できます。

1. アプリケーション フォルダに移動します。
2. dhistcfg.ini ファイルを編集します。

   例 1:「aa」という一意の接頭文字列の割り当て

```
szHistorianNode=<MachineName>
bStorageLoggingEnabled=1
bAffixEnabled=1
bPrefixEnabled=1
bSuffixEnabled=0
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
szHistUniqueString=aa
```

例 2:「xx」という一意の接尾文字列の割り当て

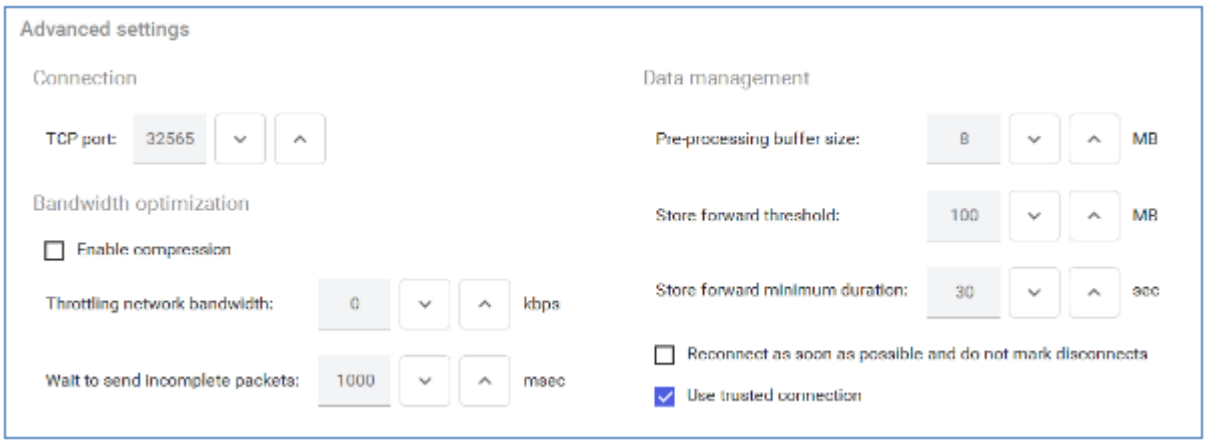
```
szHistorianNode=<MachineName>
bStorageLoggingEnabled=1
bAffixEnabled=1
bPrefixEnabled=0
bSuffixEnabled=1
szHistUniqueString=xx
```

3. .ini ファイルを保存して閉じます。

変更内容は WindowViewer を再起動した後に適用されます。

詳細設定の構成

［詳細設定］ セクションを更新して、Historian の接続に関連する設定を指定できます。



1. ［接続］では［**TCP ポート**］を指定できます。Historian Server ノードの TCP ポートには、履歴データが送信されます。TCP ポートは、Historian Server のインストール時に設定されます。デフォルトポートは 32565 です。デフォルトの Historian TCP ポートは設定可能で、データ レプリケーションに加えて、リモート IDAS バージョン 2023 R2 以降（gRPC 通信を暗示）との通信に使用されます。

   注：WCF 通信をサポートする従来の Historian TCP ポート 32568 はデータ レプリケーションと Historian バージョン 2023 以前とのリモート IDAS 通信に使用され、以前のバージョンとの互換性の目的で維持されています。InTouch HMI 2023 以前で作成されたアプリケーションのデフォルト ポート番号は 32568 です。アップグレード中、移行したアプリケーションの TCP ポートは自動的に 32565 に更新されます。ただし、ポート番号を手動で変更した場合、変更したポート番号は、アップグレードしたアプリケーションで維持されます。

2. ［帯域幅の最適化］セクションでは、［圧縮を有効にする］チェック ボックスをオンにすることができます。このチェック ボックスをオンにすると、以下を指定できます。

   - **ネットワーク帯域幅の調整**：Historian と通信するときに HCAL が使用するネットワーク通信の帯域幅の制限を kbps 単位で指定します。0 の値を設定すると、この機能が無効になります（デフォルト）。帯域幅のニーズの推定に関する詳細については、『System Platform インストール』に記載されている Historian Server のパフォーマンスおよびサイズ設定に関する推奨事項を参照してください。指定できる値は、0 kbps ～ 65535 kbps です。
   - **不完全なパケットの送信を待つ**：Historian Server に送信するまでに、部分的に充填された Historian クライアント アクセス レイヤー（HCAL）バッファを維持する最大時間をミリ秒で指定

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

します。いっぱいになったバッファは、このフィールドで設定された値に関係なく直ちに送信されます。変化する頻度が高いデータの場合、この設定は関係ありません。データの変更頻度がが遅く、ネットワーク帯域幅が制限されている場合、この値を増やしてネットワーク接続の頻度を下げることを検討してください。指定できる値は、1000 ミリ秒 ～ 30000 ミリ秒です。

3. ［**データ管理**］セクションでは、以下のオプションを指定できます。
   - **前処理バッファのサイズ**：Historian クライアント アクセス レイヤー（HCAL）で使用するすべてのバッファの合計サイズ（MB）。デフォルト値および最小値は 8 です。非常に高いデータ バーストがある場合、この値を増やしてください。値が低すぎると、データ損失が発生し、バッファのオーバーフローに関連するエラー メッセージが Logger に表示されます。この値は、10 MB 単位で増加できます。指定できる値は 8 MB ～ 65535 MB です。
   - **保存転送しきい値**：HCAL 保存転送ディスクで維持する空き容量のサイズ（MB）。指定した容量は、保存転送中に使用されません。負の値を指定することはできません。指定できる値は 8 MB ～ 65535 MB です。
   - **保存転送最小時間**：HCAL が保存転送モードで機能する最小時間（秒）。HCAL が保存転送モードで機能する条件が存在しなくなっても、この時間、HCAL は保存転送モードで機能します。指定できる値は、30 秒 ～ 3600 秒です。
   - **[できるだけ早く再接続し、切断をマークしない]** チェックボックスをオンにします。Application Server と Historian の間の通信が切断されたときにトレンドを表示する方法を指定します。これは、通信が回復した後の履歴データのトレンドの表示方法には影響しません。TRUE の場合、切断中のクライアント側のトレンドにギャップは表示されません。切断中のギャップは切断前に取得された値で補完されます。FALSE の場合、切断中のクライアント側のトレンドにギャップが表示されます。切断時に NULL 値が挿入されてギャップが作成されます。どちらの場合でも、再接続後は保存転送データでギャップが埋められます。
   - 「**信頼できる接続を使用する**」チェックボックスを選択します。これにより、Historian Server とのセキュアな通信のみが保証されます。
     ［**信頼された接続を使用する**］チェック ボックスは、すべての新規アプリケーションと移行したアプリケーションに対してデフォルトで有効になります。
     - ［**信頼された接続を使用する**］チェック ボックスをオンにすると、信頼された接続のみが Historian Server と通信できます。クライアントと Historian ノードの間に共通の証明書がない場合、接続は失敗します。
     - ［**信頼された接続を使用する**］チェック ボックスがオフの場合、ロガーに SSL エラーがある場合でも Historian との接続が確立されます。

注 : コンフィグレータで SMS を設定し、このチェック ボックスをオンにしてセキュアな接続を確立することが推奨されます。

4. **保存**を選択します。

WCF および gRPC 通信をサポートするサービス

Historian のバージョンが 2023 R2 以降の場合、下位互換性をサポートするために以下のサービスが実行されていることを確認してください。

- aahClientAccessPoint（WCF 通信）- バージョン 2023 以前のクライアントをサポートします。
- aahClientAccessPointNG（gRPC 通信）- バージョン 2023 R2 以降のクライアントをサポートします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

履歴ログの頻度の制御

オプションで、エントリをログ記録する 2 つの条件を指定できます。

- タグ変数値がログ デッドバンド値より大きい工学単位値によって変更されると、すぐにログ エントリが書き込まれます。
- ログ記録されるすべてのタグ変数の現在の値が、固定間隔で書き込まれます。また、現在の値に関わらず、記録されるすべてのタグ変数のエントリが書き込まれます。デフォルトの固定間隔は 60 分です。

デフォルトのログ記録間隔をそのまま使用するか、間隔を変更するには、intouch.ini ファイルに 2 つのパラメータを追加します。

- *ForceLoggingForceLogging*

  ForceLogging は、ログ記録間隔の長さを分単位で指定します。*ForceLogging* の値は、5 ～ 120 分の間で設定できます。デフォルトは、*ForceLogging=60* です。

- *ForceLogCurrentValueForceLogCurrentValue*

  ForceLogCurrentValue は、最後にログに記録された値のログ デッドバンド内にあっても一定のログ間隔で現在のタグ変数値をログするよう指定します。0 に設定すると、最後にログに記録された値が再度ログに記録されます。デフォルトは、*ForceLogging=0* です。

以下の例は、2 つのログ記録パラメータが含まれる intouch.ini ファイルの例を示しています。

```
WinFullScreen=1
WinWidth=808
AlarmBufferSize=5000
ForceLogging=5
ForceLogCurrentValue=1
```

この例では、タグ変数は 5 分ごとに履歴ログ ファイルに書き込まれます。

履歴ログの頻度を変更する

1. WindowMaker と WindowViewer を閉じます。
2. intouch.ini ファイルを InTouch アプリケーションと同じフォルダから見つけます。
3. intouch.ini ファイルを編集します。
4. *ForceLogging* ステートメントに 5 ～ 120 の間の値を挿入します。
5. *ForceLogCurrentValue=1* ステートメントを挿入します。
6. 変更を保存し、intouch.ini ファイルを閉じます。
7. WindowViewer を再起動します。

ランタイムでの履歴ログの開始と停止

アプリケーションの稼動中は、WindowViewer の［**システム**］メニューのコマンドを実行して、履歴ログを手動で停止したり、再起動したりすることができます。

- ［**履歴データ ログの停止**］コマンドを実行すると、アプリケーションの現在のセッション中はログ記録を停止します。アプリケーションを手動で再起動するまで、現在のセッション中はログ記録は停止したままとなります。

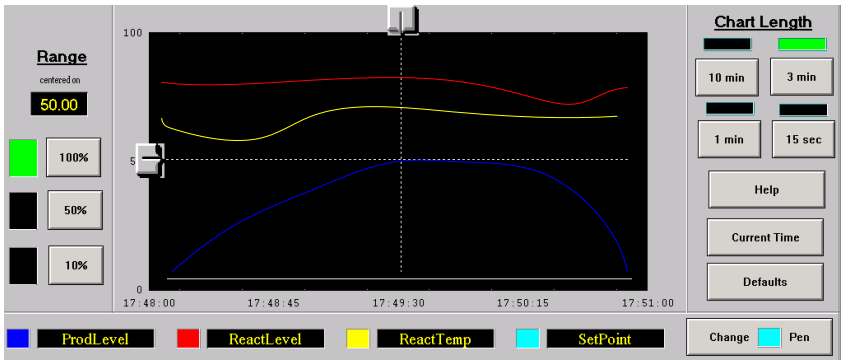
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**履歴データ ログの停止**］オプションを使用して手動で停止した後で、［**履歴データ ログの再起動**］コマンドを実行すると、ログ記録が再開されます。

また、アプリケーションにボタンを追加し、**$HistoricalLogging** システム タグ変数が含まれる QuickScript を記述することによっても、履歴ログを起動したり、停止したりすることができます。**$HistoricalLogging** に値 1 が割り当てられると、ログ記録が開始されます。**$HistoricalLogging** に値 0 が割り当てられると、ログは停止します。**$HistoricalLogging** システムタグの詳細については、 システム タグ を参照してください。

タグ データのトレンド

トレンドを作成して、InTouch アプリケーションから収集されたデータを自動表示できます。WindowMaker には、履歴トレンドやリアルタイム トレンドを作成できるユーティリティとウィザードのセットが用意されています。以下の図は、InTouch 平均/散布トレンドの例を示しています。



また、複数のトレンド コントロールを使用することもできます。トレンド コントロールを使用すれば、トレンドに表示するデータやトレンドのデータ表示方式を選択できます。

リアルタイム トレンドや履歴トレンドを設定できます。どちらのトレンド タイプにも、トレンド データの収集間隔と表示方式の設定オプションが用意されています。

InTouch トレンドのタイプ

履歴トレンドには、過去に収集されたログ データと InTouch データ レポジトリに保存されたログ データが表示されます。

分散履歴システムを使用すれば、アクセス可能なネットワーク ノードにあるすべての InTouch 履歴ログ ファイルから履歴データを検索することができます。履歴トレンドの検索機能を拡張する分散履歴システムでは、リモート ログ データベースまでカバーされます。

リアルタイム トレンドは、比較的短い間隔で生じるデータの表示を連続して更新します。WindowMaker のリアルタイム トレンド ツールを使用することにより、ウィンドウ内にトレンド オブジェクトを作成できます。またオプションの 16 ペン トレンド ツールをインストールすれば、最大で 16 のタグ変数のデータを表示するリアルタイム トレンドを作成することもできます。

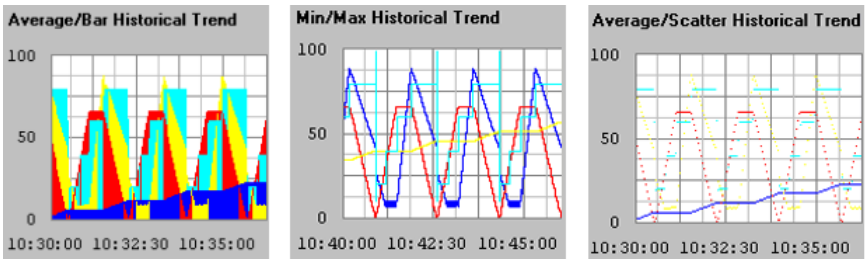
履歴トレンドの理解

履歴トレンドには、過去のデータの連続したセグメントが表示されます。リアルタイム トレンドとは異なり、履歴トレンドはスクリプトまたはユーザー操作によってのみ更新されます。

履歴トレンドには、最大で 8 タグ変数のデータのグラフが表示できます。トレンド ペンに対してタグ変数を割り当てることにより、履歴トレンドに表示するデータが指定されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以下の図に 3 タイプの InTouch 履歴トレンドを示します。



- 平均／棒履歴トレンドには、一定時間ごとのデータ ポイントの平均値が棒グラフで表示されます。
- 最小／最大履歴トレンドには、表示時間全体にわたる値の変化が工学値のパーセントを縦軸に取って表示されます。変化量よりも時間の流れと変化率に重点が置かれています。
- 平均／散布履歴トレンドには、各トレンド間隔ごとのデータ ポイントの平均値が表示されます。

スクータと呼ばれるグラフィック スライダを作成することにより、トレンド内の現在のスクータの位置に応じた詳細なトレンド データにアクセスできます。たとえば、データが表示されているトレンドの領域にスクータを移動すると、その位置のトレンド グラフに表示されているすべてのデータベース値の時刻と値を表示できます。

また、スクータ間や最大値と最小値などのデータ間隔を拡大／縮小表示するボタンやスライダを作成することもできます。グラフ全体、またはスクータ間の領域の平均や標準偏差を表示することもできます。

履歴トレンドをスクロールする時間も、任意に設定できます。カスタムスケールを作成し、 **MinEU** と.**MaxEU** ドットフィールドにリンクすることにより、すべてのデータ セットをそのデータの工学単位で表示するトレンドを作成することができます。

リアルタイム トレンドの理解

リアルタイム トレンドには、現在実行中の InTouch アプリケーションのデータが表示されます。リアルタイム トレンドは継続的に更新されます。リアルタイム トレンドには、最大で 4 つのローカル タグ変数か式に関連する現在のデータ値が表示されます。

以下の操作を実行できます。

- リアルタイム トレンドの作成
- トレンドに対してタグを選択する
- トレンドの時間範囲と更新間隔の指定
- トレンドの表示オプションの設定

保存済みタグの値の履歴トレンド表示

以下のいずれかの WindowMaker ユーティリティを使用して、履歴トレンドを作成できます。

- 履歴トレンド ツール
- 履歴トレンド ウィザード
- 16 ペン トレンド ウィザード（オプション）

さらに、ActiveFactory トレンドを組み込んで Historian データベースに保存された InTouch 履歴トレンド データを表示できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

履歴トレンド オブジェクトの使用

WindowMaker 履歴トレンド ツールを使用して、トレンドを作成して設定することができます。以下の操作を実行できます。

- 履歴トレンドの作成
- トレンドに対してタグを選択する
- トレンドの時間範囲と更新間隔の指定
- トレンドの表示オプションの設定

履歴トレンドの作成

履歴トレンド ツールを使用することにより、ウィンドウ内にトレンド オブジェクトを作成できます。新しく作成した履歴トレンド オブジェクトには、InTouch のデフォルト設定値が適用されます。

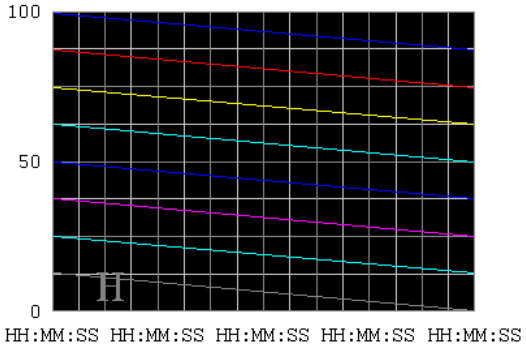
履歴トレンドの設定を変更した後は、WindowMaker では、最後の設定値が新しいトレンド オブジェクトの初期設定値として使用されます。

ウィンドウ枠の中であれば、どんなサイズのトレンド グラフでも表示できます。

履歴トレンドの作成

1. WindowMaker で、履歴トレンドを作成するウィンドウを開きます。
2. ［描画］メニューの［**トレンド**］グループで［**履歴**］をクリックします。
3. 履歴トレンドを作成するウィンドウの領域にマウスを移動します。マウスを斜めにドラッグして長方形を作成し、所望のトレンド サイズにします。

   ウィンドウに履歴トレンド オブジェクトが表示されます。



4. 必要に応じて、オブジェクトのハンドルを使用してトレンドの高さと幅を調整します。

履歴トレンドに表示するタグの設定

履歴トレンド ペンによって、指定期間のログ データのグラフが作成されます。トレンド ペンは、履歴データを収集するタグに割り当てられます。

履歴トレンドがサポートするペンの数は 8 本までです。

注：WindowViewer は必ず閉じてください。そうでないと、ペン ボックスを選択できません。

プロバイダが設定されている場合は、リモート履歴プロバイダからタグを選択できます。リモート履歴プロバイダの設定の詳細については、『*AVEVA™ InTouch HMI アプリケーション配置*』の「アプリケーションの分散」を参照してください。

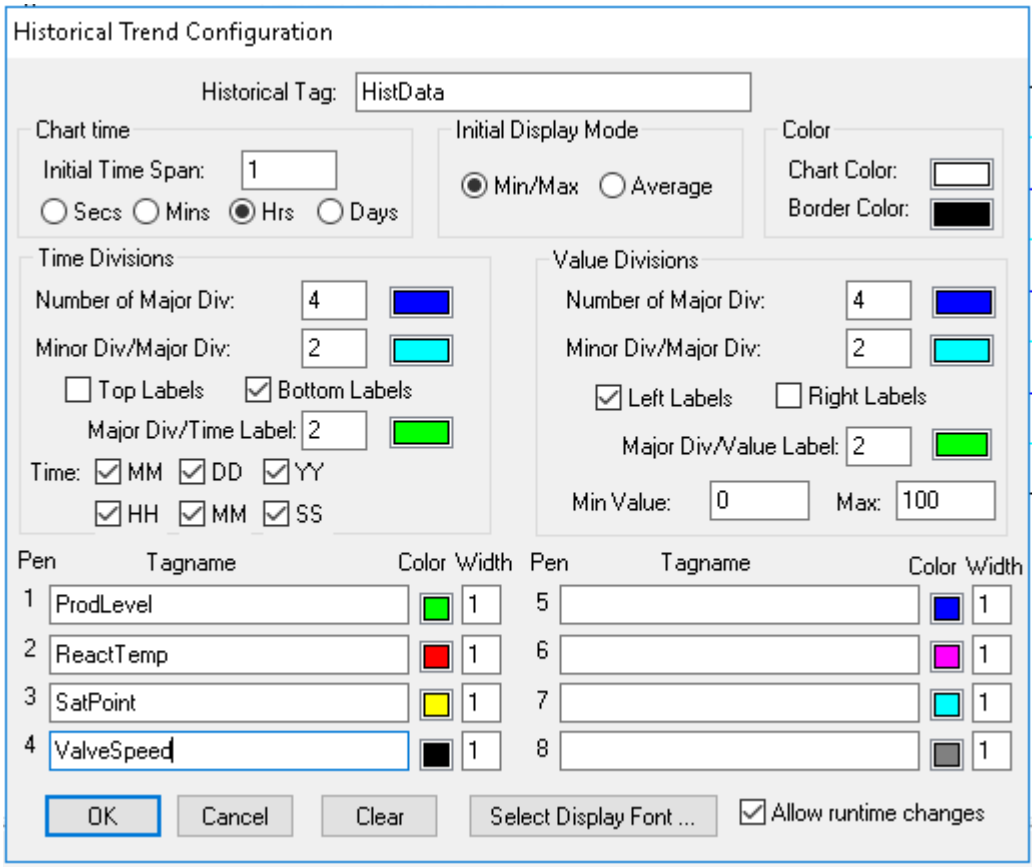
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注 : IndustrialSQL Server の履歴プロバイダを設定して履歴トレンド データを視覚化することもできます。他のチャート オプションを使用してさらに用途を広げるには、ActiveFactory トレンド ツールを使用して IndustrialSQL Server データベースに保存された InTouch 履歴データのトレンドを作成してください。

履歴トレンドに表示するタグを設定する

1. ウィンドウ内のトレンド オブジェクトをダブルクリックします。

   ［**履歴トレンドの設定**］ダイアログ ボックスが表示されます。



2. ［**履歴タグ**］ボックスに、トレンドに使用するタグを入力します。

   タグは履歴トレンド型として定義する必要があります。InTouch アプリケーションの各履歴トレンドに対して、異なる履歴トレンドのタグを割り当ててください。

   タグ名ディクショナリに現在定義されていないタグを入力した場合は、タグの作成を確認するメッセージがダイアログ ボックスに表示されます。［**OK**］を選択してタグを定義すると、［**タグ名ディクショナリ**］ダイアログ ボックスが表示されます。このタグは 128 文字をサポートしません。

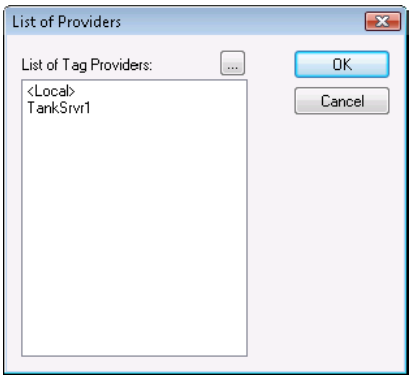
3. ［**タグ名**］領域で、1 つ以上の［**ペン**］ボックスに既存のローカル タグまたはリモート タグを指定します。

4. 既存のローカル タグかリモート タグを直接指定するには、［**ペン**］フィールド内をクリックしてタグ変数の名前を入力します。

5. タグを参照して指定するには

   a. ［**ペン**］ボックス内をダブルクリックします。［**プロバイダのリスト**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



b. このペンに使用するタグ プロバイダを選択します。

c. ［**OK**］をクリックすると、選択されたプロバイダのタグ変数リストのダイアログ ボックスが表示されます。

d. リストのタグをダブルクリックして選択します。

6. タグが割り当てられた各ペンの横にあるカラー ボックスをダブルクリックすると、カラー パレットが開きます。ペンの色を選択します。

7. ［**幅**］ボックスに、トレンドに表示される各ペンの線幅をピクセル単位で入力します。

8. 履歴トレンド ペンに割り当てるタグごとに、手順 3 から 7 を繰り返します。

9. ［**実行中の変更可能**］チェック ボックスをオンにすると、アプリケーションの実行中にオペレータが履歴トレンドの設定を変更できるようになります。

   ランタイム時の履歴トレンドの更新の詳細については、「ランタイム時のトレンド設定の変更」を参照してください。

履歴トレンドの時間範囲を設定する

履歴トレンドの時間範囲を設定できます。

履歴トレンドの時間範囲を設定する

1. トレンド オブジェクトをダブルクリックして［**履歴トレンドの設定**］ダイアログ ボックスを表示します。

2. ［**チャート時間**］領域の［**初期時間範囲**］に、トレンドの横軸（X 軸）に表示する時間長を入力します。

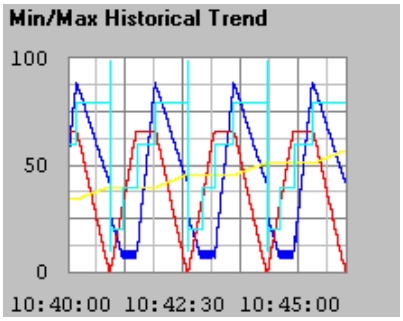
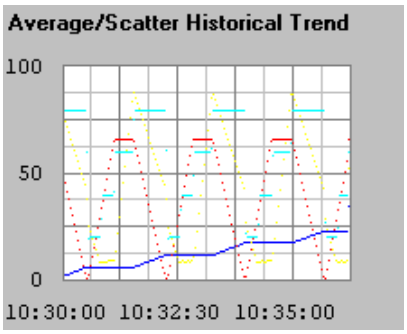
3. 時間の単位を、［秒］、［分］、［時］、［日］の中から選択します。

   たとえば、［**初期時間範囲**］に 8 を入力してから［**時**］を選択すると、トレンドに表示される時間範囲は 8 時間になります。

4. ［**初期表示モード**］領域では、トレンドを表示するウィンドウが WindowViewer で最初に表示されたときに表示する履歴トレンドのタイプを選択します。

初期表示モード	説明

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

最小/最大	チャートには、時間範囲全体にわたる工学単位スケールのパーセンテージの変化が垂直線で表示されます。 
平均	トレンド時間セグメント内の各ピクセルは、セグメントの全期間におけるタグの平均値を示したものです。 

5. 履歴トレンドの表示方式を設定するには、「履歴トレンドの表示オプションを設定する」に進んでください。

履歴トレンドの表示オプションを設定する

履歴トレンドの表示方式を設定することができます。

履歴トレンドの表示オプションを設定する

1. トレンド オブジェクトをダブルクリックします。［**履歴トレンドの設定**］ダイアログ ボックスが表示されます。
2. 色オプションの設定には、以下の手順を実行します。
   - ［**配色**］領域で［**背景**］フィールドをクリックして、カラー パレットを開きます。
   - パレット内でトレンドの背景色を選択します。

     デフォルトの背景色は白です。背景色を別の色にすると、トレンドの印刷速度が遅くなります。
   - ［**輪郭色**］を選択してカラー パレットを開きます。
   - パレット内でトレンドの輪郭色を選択します。
3. 時間軸の目盛りオプションを設定するには、以下の手順を実行します。
   - ［**時間軸の目盛り**］領域の［**主目盛り数**］に、トレンドの主目盛り数を入力します。

     主目盛りはトレンドの横軸（時間軸）に表示されます。主目盛りを分割する間隔は最長で 65536 秒、つまり 18 時間 12 分 16 秒までです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 主目盛り線の色を選択します。
- ［**主目盛り間の分割数**］に、各主目盛り内の小目盛りの数を入力します。

  時間軸の小目盛りの数は主目盛り内で均等に割り切れる必要があります。たとえば、主目盛りを 60 秒に設定して［**主目盛り間の分割数**］に 2 を入力すると、小目盛りは 30 秒に設定されます。
- 小目盛り線の色を選択します。
- ［**上ラベル**］または［**下ラベル**］を選択して、トレンドに表示する時間ラベルの位置を指定します。
- 時間ラベルを使用する場合は、［**主目盛り/時間ラベル**］ボックスに時間ラベルあたりの時間軸の主目盛り線の数を入力します。
- 時間軸目盛りのラベルの色を選択します。
- 主目盛り線のラベルに表示する時間単位を選択します。

月（MM）	時（HH）
日（DD）	分（MM）
年（YY）	秒（SS）

4. ［**縦軸の目盛り**］領域で、トレンドの縦軸の表示を設定します。

   ［**縦軸の目盛り**］オプションはの設定方法は、［**時間軸の目盛り**］オプションと同じです。縦軸では、すべてのタグの工学単位に基づいて、トレンドに表示するデータ値の範囲が指定されます。
5. ［**OK**］をクリックして設定変更を保存してから、［**履歴トレンドの設定**］ダイアログ ボックスを閉じます。

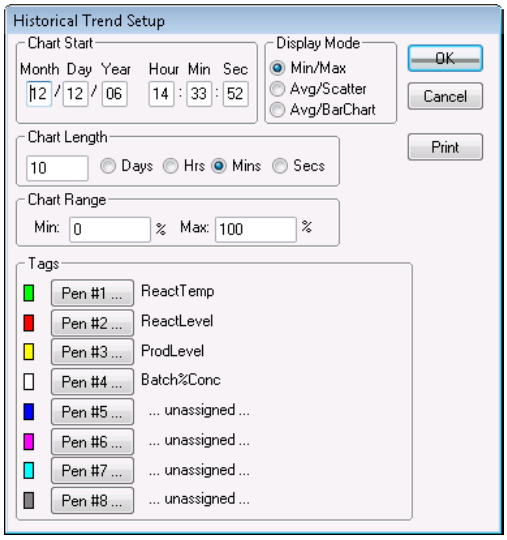
ランタイム時のトレンド設定の変更

履歴トレントの設定で［**実行中の変更可能**］オプションを選択すると、履歴トレンドの実行中に設定を変更できます。オペレータは、表示されているウィンドウからトレンドを選択してダイアログ ボックスを表示させてからトレンドの設定を変更します。

ランタイム中に履歴トレンドを設定するには

1. 実行中に履歴トレンドをクリックします。［**履歴トレンドの設定**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［**表示開始時間**］領域で、履歴トレンド データの収集期間の開始日時を入力します。
3. ［**表示モード**］領域で、履歴トレンド チャートのタイプを選択します。

   トレンドの表示モードは描画スピードに影響を及ぼします。トレンドの描画スピードを決める主要な要素は、トレンドに表示される線の長さです。線が長くなるほどトレンドの生成時間が長くなります。

   線幅も同様に描画スピードに影響します。線幅が広いと描画速度は著しく遅くなります。［最大／最小］と［平均／散布］トレンドは、［平均／棒］トレンドよりも短い時間で生成されます。
4. ［**表示時間**］領域で、トレンド表示される時間間隔を入力してから時間単位を選択します。

   たとえば、2 を入力してから［**時**］を選択すると、トレンドの時間間隔は 2 時間になります。
5. ［**縦軸表示**］領域に、履歴トレントの縦軸に表示される工学値の範囲をパーセント単位で入力します。

   トレンドのスケールは、トレンド表示されているタグの工学値の範囲をパーセントの範囲で定義したものです。値の範囲は 0 から 100 までです。たとえば、選択したタグの工学値の範囲に対して 40 から 60 パーセントの範囲で変化するトレンドが必要な場合は、［**最小値**］と［**最大値**］のそれぞれの範囲フィールドに 40 と 60 を入力します。
6. ［**タグ変数**］領域で、タグ変数を割り当てるペン番号をクリックします。

   ［**タグの選択**］ダイアログ ボックスに、履歴トレンド ペンへの割り当てが可能なタグのリストが表示されます。
7. 以下のステートメントを QuickScript またはボタンに使用することにより、オペレータはチャートを更新できます。

```
Hist_TrendTag.UpdateTrend = 1
```

8. 以下のいずれかの関数を QuickScript またはボタンに使用します。

   HTUpdateToCurrentTime(Hist_Tag);

   HTScrollLeft(Hist_Tag,Percent);

   HTScrollRight(Hist_Tag,Percent);

   HTZoomIn(Hist_Tag,LockString);

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

HTZoomOut(Hist_Tag,LockString);

HTSetPenName(Hist_Tag、PenNum、Tagname);

トレンド関数を含むスクリプトの使用の詳細については、「スクリプトの使用による履歴トレンドウィザードのコントロール」を参照してください。

9. 以下のトレンド タグ ドットフィールドを変更します。

.ChartStart(チャート開始)

.ChartLength(チャートの長さ)

.MaxRange(最大範囲)

.MinRange(最小範囲)

.Pen1-.Pen8(ペン 1-ペン 8)

履歴トレンドを含むドットフィールドの使用の詳細については、「ドットフィールドの使用による履歴トレンドのコントロール」を参照してください。

ドットフィールドの使用による履歴トレンドのコントロール

ドットフィールドを使用すれば、ランタイム実行中に履歴トレントを管理できます。

.DisplayMode ドットフィールド

.DisplayMode ドットフィールドでは、タグ変数の値の表示に使用されるトレンド形式を指定します。

カテゴリ

履歴

使用法

```
tag_name.DisplayMode
```

パラメータ

tag_name
任意の履歴トレンドのタグ変数

データタイプ

アナログ型（読み取り／書き込み）

有効値

1 = 各サンプル期間の最大値と最小値を表示します（デフォルト）。

2 = 散布履歴トレンドの各サンプル期間の平均値を表示します。

3 = 棒グラフ履歴トレンドの各サンプル期間の平均値を表示します。

例

このステートメントでは、「HistTrend_Tag」で表される履歴トレンドの値に棒グラフ履歴トレンドの形式が指定されます。

```
HistTrend_Tag.DisplayMode=3;
```

参照項目

.ChartLength、.ChartStart

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.MinRange ドットフィールド

.MinRange ドットフィールドでは、各タグ変数の履歴トレンド表示に使用される、タグ変数の工学値範囲に対する最小パーセントが指定されます。

カテゴリ

履歴

使用法

```
tag_name.MinRange
```

パラメータ

tag_name
任意の履歴トレンドのタグ変数

備考

履歴トレンドには、複数のタイプのタグ変数を同時に表示することができます。異なるタイプのタグ変数には別の工学単位が割り当てられるため、工学単位で値の範囲の下限値と上限値を指定するのは困難です。そのため、最小値と最大値は各タグ変数の工学値範囲に対するパーセントで表示しています。したがって履歴トレンドは、タグ変数の実際の工学値範囲にかかわらず、タグ変数に固有の工学値のパーセント表示で示されます。

データタイプ

実数型（読み取り／書き込み）

有効値

.MaxRange と .MinRange ドットフィールドの範囲は 0 から 100 です。.MinRange は .MaxRange よりも小さい値にする必要があります。これらのドットフィールドに 0 未満か 100 を超える値が割り当てられると、自動的に 0 または 100 に変更されます。.MinRange が .MaxRange 以上の場合は、トレンドにデータは表示されません。

例

以下のドットフィールド ステートメントは、履歴トレンドの最小パーセントを履歴トレンドのタグ変数がとり得る工学値範囲の 25 パーセントに設定します。

```
HistTrend.MinRange=25
```

参照項目

.ChartStart、.ChartLength、.DisplayMode、.EngUnits、.MinEU、.MaxEU、.MaxRange、.MinRaw、.MaxRaw、.RawValue

.MaxRange ドットフィールド

.MaxRange ドットフィールドでは、各タグ変数の履歴トレンド表示に使用される、タグ変数の工学値の範囲に対する最大パーセントが指定されます。

カテゴリ

履歴

使用法

```
tag_name.MaxRange
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

tag_name
任意の履歴トレンドのタグ変数

備考

履歴トレンドには、複数のタイプのタグ変数を同時に表示することができます。タグ変数には異なる工学単位が割り当てられるため、工学単位で値の範囲の下限値と上限値を指定するのは困難になります。そのため、最小値と最大値は各タグ変数の工学値範囲に対するパーセントで表示しています。したがって、タグ変数の実際の工学値範囲にかかわりなく、履歴トレンドはタグ変数の工学値のパーセント表示で示されます。

データタイプ

実数型（読み取り／書き込み）

有効値

.MaxRange と .MinRange ドットフィールドの範囲は 0 から 100 です。.MinRange は .MaxRange よりも小さい値にする必要があります。これらのドットフィールドに 0 未満か 100 を超える値が割り当てられると、自動的に 0 または 100 に変更されます。.MinRange が .MaxRange 以上の場合は、トレンドにデータは表示されません。

例

以下のドットフィールド ステートメントは、履歴トレンドの最大パーセントを履歴トレンドのタグ変数がとり得る工学値範囲の 75 パーセントに設定します。

```
HistTrend.MaxRange=75
```

参照項目

.ChartStart、.ChartLength、.DisplayMode、.EngUnits、.MinEU、.MaxEU、.MinRange、.MinRaw、.MaxRaw、.RawValue

.UpdateCount ドットフィールド

.UpdateCount ドットフィールドでは、履歴トレンドの更新回数がカウントされます。**.UpdateCount** ドットフィールドは、関数のトリガとして使用できます。

カテゴリ

履歴

使用法

```
HistTrendTag.UpdateCount
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

データタイプ

整数型（読み取り専用）

有効値

任意の正の整数値

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下の例では、右スクータの現在位置で Pen1 の値を取得するために **HTGetValueAtScooter()** 関数を使用しています。関数の引数に対して変更が行われると、その関数の値が再度求められます。更新が終了するたびに **.UpdateCount** が増分し、このステートメントも更新されます。

```
MyRealTag=HTGetValueAtScooter( MyHistTrendTag,MyHistTrendTag.UpdateCount, 2,
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

参照項目

.UpdateInProgress、.UpdateTrend

.UpdateInProgress ドットフィールド

.UpdateInProgress ドットフィールドには、履歴トレンド更新処理の現在の状況が示されます。履歴取得処理中にはこのドットフィールドに 1 がセットされ、それ以外の場合は 0 がセットされます。

カテゴリ

履歴

使用法

```
HistTrendTag.UpdateInProgress
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

備考

履歴トレンドが新しいデータを要求するたびに、このドットフィールド値は 1 にセットされます。処理が終了すると、**.UpdateInProgress** は 0 にリセットされます。**.UpdateInProgress** は、履歴トレンドに関連する関数の中で使用することができます。

ユーザーが、現在の表示期間外にトレンドをスクロールした場合には、履歴データの取得に時間を要することがあります。**.UpdateInProgress ドットフィールド**を使用すれば、要求したデータの取得中であることをユーザーに知らせることができます。応答がなければ、トレンドを更新中であることがユーザーにはわかりません。

データタイプ

論理型（読み取り専用）

有効値

0 = 更新は行われていません

1 = 更新中

例

.UpdateInProgress ドットフィールドは通常、履歴トレンドの文字オブジェクトやスクロール ボタンの表示可能リンクの式としてよく使用されます。以下のメッセージ型値の表示アニメーション リンクを指定して .UpdateInProgress ドットフィールドを使用すると、データの取得中にウィンドウに「ビジー」を表示することができます。

```
DText(HistTrend1.UpdateInProgress,"Busy","Ready")
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.UpdateCount、.UpdateTrend

.UpdateTrend ドットフィールド

.UpdateTrend ドットフィールドは、履歴トレンドの更新をトリガします。ボタン アクション スクリプトで**.UpdateTrend** ドットフィールドを使用すると、ランタイム中にユーザーは手動でトレンドを更新できます。

カテゴリ

履歴

使用法

```
HistTrendTag.UpdateTrend
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

備考

通常、履歴トレンドは自動的には更新されません。特定のタグ変数に対するチャートを更新して現在の値を表示するには、**.ChartStart** か **.ChartLength** のいずれかのドットフィールドを変更する必要があります。このドットフィールドをボタン アクション スクリプトで使用すれば、ユーザーはランタイム中にチャートを更新することができます。また、履歴トレンドに関連付けられている別のドットフィールドが変化する場合には、QuickScript でこのドットフィールドを使用できます。

.UpdateTrend ドットフィールドには 1 だけをセットできます。

データタイプ

論理型（書き込み専用）

有効値

1

例

この例では、**MyHistTrendTag** タグ変数に関連付けられた履歴トレンドを現在の全パラメータ値で更新するようにトリガします。

```
MyHistTrendTag.UpdateTrend=1;
```

.ChartLength ドットフィールド

.ChartLength ドットフィールドでは、履歴トレンドに表示される時間長が指定されます。

カテゴリ

履歴

使用法

```
HistTrendTag.ChartLength
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

.ChartLength の値により、チャートの長さが秒単位で指定されます。この長さは、履歴トレンドのチャートに現在表示されている時間として定義されます。具体的には、履歴トレンドのチャートから表示時間を求めるには次の計算式が使用されます。

```
表示時間 = （チャートの右端の日付／時間スタンプ） - （チャートの左端の日付／時間スタンプ）
```

日付／時間スタンプは 1970 年 1 月 1 日の午前 0 時からの経過時間を秒数で表しているため、計算結果はチャートの左端と右端の間に表示される秒単位の時間になります。

.ChartLength の値を増減させる場合には、必ず秒単位で時間を表します。したがって、現在の **.ChartLength** を 2 時間減らすには、計算を実行する前に時間単位を秒単位に変換する必要があります。次に例を示します。

```
（2 時間）*（ 60 分／時間）*（60 秒／分） = 7200 秒
```

データタイプ

整数型（読み取り／書き込み）

有効値

任意の正の整数値

例

以下の例では、履歴トレンドの長さが 1 時間に設定されます。

```
HtTag.ChartLength=3600 {60 分 * 60 秒／分};
```

以下の例では、トレンドが 50 パーセント分だけ左にスクロールされます。

```
HtTag.ChartStart=HtTag.ChartStart - HtTag.ChartLength / 2;
```

以下の例では、トレンドが 10 パーセント分だけ左にスクロールされます。

```
HtTag.ChartStart=HtTag.ChartStart - (.10 * HtTag.ChartLength);
```

参照項目

.ChartStart

.ChartStart ドットフィールド

.ChartStart ドットフィールドは、履歴トレンドの開始（左端）日付／時間スタンプの値を設定または確認するために使用されます。

カテゴリ

履歴

使用法

```
HistTrendTag.ChartStart
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

備考

この読み取り／書き込みドットフィールドは、履歴トレンドの開始（左端）日付／時間スタンプの値を設定または確認するために使用されます。**.ChartStart** ドットフィールドは、1970 年 1 月 1 日の午前 0 時

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

からの経過時間を秒数で表します。この開始時点は、履歴トレンドの先頭の日付／時間スタンプとして定義されます。

データタイプ

整数型（読み取り／書き込み）

有効値

任意の正の整数値

例

次のステートメントは、チャートを右側に 1 分（60 秒）スクロールします。

```
HtTagname.ChartStart=HtTagname.ChartStart + 60;
```

参照項目

.ChartLength

.Pen1-8 ドットフィールド

.Pen1-8 ドットフィールドでは、履歴トレンド ペンにログ済みタグが割り当てられます。

カテゴリ

履歴

使用法

```
HistTrendTag{.Pen1 | .Pen2 | .Pen3 | .Pen4 | .Pen5 | .Pen6 | .Pen7 | .Pen8};
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ

備考

以下の形式で .Pen1-8 ドットフィールドを使用して、トレンド ペンにタグを割り当てます。

HistTrend.Pen*X* = Tag_Name.TagID

ここで、*X* は 1 から 8 までの整数です。

可能であれば、HTSetPenName() または HTGetPenName() 関数の使用をお勧めします。

注記: .PenX ドットフィールドに割り当てられるのは、ローカル タグだけです。Provider.tag のような表記法は使用できません。この表記法は、HTSetPenName() 関数でのみ使用できます。
このドットフィールドの機能については、パートに分かれて画面表示される履歴トレンド ウィザードが参考になります。

データ タイプ

タグ ID（読み取り/書き込み）

有効値

このドットフィールドのデータ タイプは **TagID** です。そのため、**.Pen1-8** ドットフィールドにはタグのハンドルだけが割り当てられます。**.Pen1-8** ドットフィールドにタグの名前を直接割り当てることはできません。以下の構文を使用して、**.Pen1-8** ドットフィールドに対してタグの **.TagID** ドットフィールドを割り当てる必要があります。

```
HistTrendTag.Pen1=LoggedTag.TagID;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

一般的に、TagID 型のタグは、別の TagID 型のタグとしか等式化できません。そのため **.TagID** ドットフィールドの拡張を他のタグに追加しない限り、他のタグ タイプとは混合できません。

.Pen1-8 ドットフィールドは読み取り/書き込みですが、ドットフィールドの値は画面に直接表示できません。

例

以下の例では、履歴トレンドのタグに関連付けられた履歴トレンドの **.Pen5** ドットフィールドに対して、新しいタグが割り当てられます。**.TagID** ドットフィールドを **.Pen5** ドットフィールドに割り当てるには、.TagID ドットフィールトをログ済みタグの名前に追加しなければなりません。

```
HistTrendTag.Pen5=PumpPress.TagID;
```

上記の例によって、HistTrendTag.Pen5 に割り当てられたタグの名前が明確になりました。各トレンド ペンに割り当てられたタグを示す凡例を作成しておくと、オペレータへの有用な情報になります。

メッセージ表示リンクでは、HistTrendTag.Pen5 に割り当てられたタグを示すことはできません。.Pen5 ドットフィールドの実際の値は WindowViewer 内のメモリ ロケーションを表す整数値で、表示の目的にはあまり有用でないためです。そのため、**Pen05** という新しい TagID 型のタグを作成する必要があります。前のステートメントに以下のステートメントを追加します。

```
Pen05=HistTrendTag.Pen5;
```

最初の例では、PumpPress タグが HistTrendTag の ペン 5 に割り当てられます。この例では、PumpPress タグの TagID になった HistTrendTag の Pen5 の値に Pen05 が割り当てられます。

.Pen1-8 ドットフィールドは、トレンド表示に選択されたペンに関連付けられているタグへのポインタです。.Pen1-8 ドットフィールドのデータ タイプは、.TagID という特殊なものです。割り当てを作成した後、TagID タグの .Name ドットフィールドを使用して、タグの名前を表示できます。

.TagID ドットフィールド

.TagID ドットフィールドを **Pen1** ～ **Pen8** ドットフィールドと共に使用して、履歴トレンド ペンにタグ変数を割り当てることができます。

カテゴリ

履歴タグ変数。

使用法

```
tag_name.TagID
```

パラメータ

tag_name
論理型、整数型、実数型、間接論理型、または間接アナログ型の任意のタグ変数

備考

.TagID ドットフィールド はタグ変数のハンドルを可能にするもので、主に履歴トレンドのペンにタグ変数を割り当てるために使用されます。

データタイプ

タグ変数 ID 型（読み取り専用）

例

以下は、**.TagID** ドットフィールドを使用して履歴トレンドのペン 6 に **PumpRPM** タグ変数を割り当てる例です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
HistTrendTag.Pen6=PumpRPM.TagID;
```

参照項目

.Pen1-.Pen8

.ScooterLockLeft ドットフィールド

.ScooterLockLeft ドットフィールドでは、ユーザーが履歴トレンド上で右スクータを左スクータの現在位置よりもさらに左側に移動できるかどうかが指定されます。

カテゴリ

履歴

使用法

```
HistTrendTag.ScooterLockLeft
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

備考

一般的に、ユーザーには右スクータを左スクータの現在位置よりも左側に移動させないようにする必要があります。左スクータがロックされている場合、右スクータを左スクータよりさらに左側に移動すると、右スクータの位置は強制的に左スクータの位置と等しくなります。

データタイプ

論理型（読み取り／書き込み）

有効値

0 = False 履歴トレンド上で右スクータを左スクータの現在位置よりもさらに左側に移動できます。

1 = True 履歴トレンド上で右スクータを左スクータの現在位置よりもさらに左側に移動できません。

例

以下の例では、履歴トレンド上で右スクータを左スクータの現在位置よりもさらに左側に移動できないようにします。

```
HistTrendTag.ScooterLockLeft=1;
```

参照項目

.ScooterPosRight、.ScooterPosLeft、.ScooterLockRight

.ScooterLockRight ドットフィールド

.ScooterLockRight ドットフィールドでは、ユーザーが履歴トレンド上で左スクータを右スクータの現在位置よりもさらに右側に移動できるかどうかが指定されます。

カテゴリ

履歴

使用法

```
HistTrendTag.ScooterLockRight
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

備考

一般的にユーザーには、左スクータを右スクータの現在位置よりも右側に移動させないようにする必要があります。右スクータがロックされている場合、左スクータを右スクータよりさらに右側に移動すると、左スクータの位置は強制的に右スクータの位置と等しくなります。

データタイプ

論理型（読み取り／書き込み）

有効値

0 = False 履歴トレンド上で左スクータを右スクータの現在位置よりもさらに右側に移動できます。

1 = True 履歴トレンド上で左スクータを右スクータの現在位置よりもさらに右側に移動できません。

例

以下の例では、履歴トレンド上で左スクータを右スクータの現在位置よりもさらに右側に移動できないようにします。

```
HistTrendTag.ScooterLockRight=1;
```

参照項目

.ScooterPosRight、.ScooterPosLeft、.ScooterLockLeft

.ScooterPosLeft ドットフィールド

.ScooterPosLeft は、左スクータの位置を履歴トレンド上で動的にコントロールします。

カテゴリ

履歴

使用法

```
HistTrendTag.ScooterPosLeft
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

備考

この読み取り／書き込みのドットフィールドは、左スクータの位置を動的にコントロールします。このドットフィールドを QuickScript 関数に使用すると、左スクータの現在位置を取得したり、このドットフィールドに新しい値を割り当てて左スクータをトレンド上の別の位置に動かしたりできます。

このドットフィールドは通常、**HTGetValue()** 関数と組み合わせて使用されます。この関数では、履歴トレンドのスクータの位置の指定のほかに、どの履歴トレンドをクエリーするのかを指定する必要があります。

データタイプ

実数型（読み取り／書き込み）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

0.0～1.0; ここで、0.0 は履歴トレンド チャートの一番左端で、1.0 は履歴トレンド チャートの一番右端。

例

以下の例では、左スクータの位置が変更されます。MyHistTrendTag タグ変数に現在関連付けられている履歴トレンド チャートで、チャートの左端からチャート全体の長さの 34 パーセントの位置に左スクータを移動します。

```
MyHistTrendTag.ScooterPosLeft=.34;
```

以下のステートメントでは、QuickScript の **HTGetValueAtScooter()** 関数によって左スクータの現在位置におけるペン 1 の値が取得されます。関数の引数リスト内で値が変更されると、その関数の値が再度求められます。また、左スクータの位置が変わるたびに、このステートメントは更新されます。

```
MyRealTag=HTGetValueAtScooter (MyHistTrendTag,MyHistTrendTag.UpdateCount, 1,
MyHistTrendTag.ScooterPosLeft, 1, "PenValue");
```

参照項目

.ScooterPosRight、.ScooterLockLeft、.ScooterLockRight

.ScooterPosRight ドットフィールド

この読み取り／書き込みの **.ScooterPosRight** ドットフィールドは、右スクータの位置を動的にコントロールします。

カテゴリ

履歴

使用法

```
HistTrendTag.ScooterPosRight
```

パラメータ

HistTrendTag
トレンド名が割り当てられた履歴トレンドのタグ変数

備考

この読み取り／書き込みのドットフィールドは、右スクータの位置を動的にコントロールします。このドットフィールドを QuickScript 関数に使用すると、右スクータの現在位置を取得できます。また、このドットフィールドに新しい値を割り当てて、右スクータをトレンド上の別の位置に動かすこともできます。

ドットフィールドは通常、 **HTGetValue()** 関数と組み合わせて使用されます。この関数では、履歴トレンドのスクータの位置の指定のほかに、どの履歴トレンドをクエリーするのかを指定する必要があります。

データタイプ

実数型（読み取り／書き込み）

有効値

0.0～1.0; ここで、0.0 は履歴トレンド チャートの一番左端で、1.0 は履歴トレンド チャートの一番右端。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のステートメントでは、右スクータに新しい位置が指定されます。MyHistTrendTag タグ変数に関連付けられている履歴トレンド チャートで、チャートの左端からチャート全体の長さの 34 パーセントの位置に右スクータを移動します。

```
MyHistTrendTag.ScooterPosRight=.34;
```

以下のステートメントでは、**HTGetValueAtScooter()** QuickScript 関数を使用して右スクータの新たな位置におけるペン 1 の値が取得されます。関数のパラメータ リスト内で変数が変更されると、その関数の値が再度求められます。また、右スクータの位置が変わるたびに、このステートメントは更新されます。

```
MyRealTag=HTGetValueAtScooter(MyHistTrendTag, MyHistTrendTag.UpdateCount, 2,
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

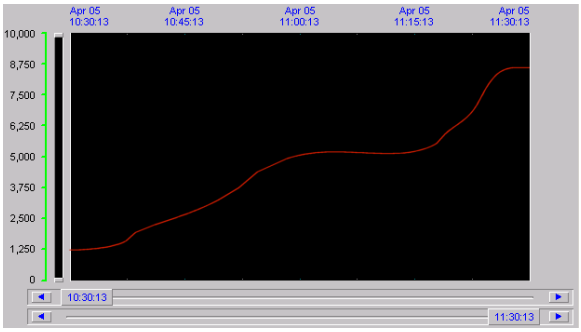
参照項目

.ScooterPosLeft、.ScooterLockLeft、.ScooterLockRight

履歴トレンド ウィザードの使用

履歴トレンド ウィザードを使用して、自動的に履歴トレンドを作成することができます。手動で履歴トレンド ペンにタグ変数を割り当てなくても、ウィザードが標準値を使用して自動的に履歴トレンドを設定します。

履歴トレンド ウィザードで作成した標準のトレンドを以下に示します。このトレンドには、トレンドの特定位置のデータを表示したり、トレンド データの範囲を選択してズームインするために使用するスクータと呼ばれるスライダが含まれています。



履歴トレンドにズームや移動機能またはペン コントロールを追加するには、［トレンド ズーム / スクロール パネル］ウィザードと［トレンド ペン凡例］ウィザードを使用します。

履歴トレンドを作成して設定できます。以下の操作を実行できます。

- ウィザードで履歴トレンドを作成する
- トレンドに対してタグを選択する
- 履歴トレンドの表示時間を設定する
- QuickScript でトレンドをコントロールする

履歴トレンド ウィザードによるトレンドの作成

履歴トレンド ウィザードの自動機能を使用して標準の履歴トレンドを作成できます。ウィザード オプションを使用して手動で履歴トレンドを設定する方法については、別のセクションに記載します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

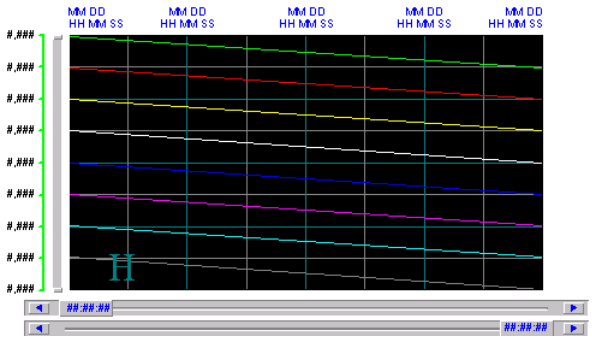
ウィザードで履歴トレンドを作成する

1. WindowMaker で履歴トレンドを作成するウィンドウを開きます。
2. ［**描画**］メニューの［挿入］グループで［**ウィザード**］を選択します。

   ［**ウィザードの選択**］ダイアログ ボックスが表示されます。
3. ウィザードのリストから［**トレンド**］を選択します。

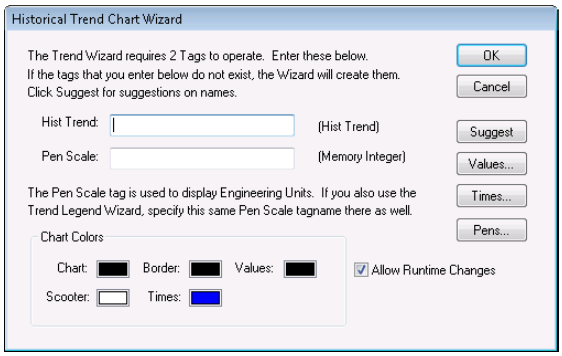
   ［**ウィザードの選択**］ダイアログ ボックスの右ペインに、トレンド ウィザードのアイコンのセットが表示されます。
4. ［**スクータとスケール付き履歴トレンド**］ウィザードを選択して、［**OK**］を選択します。

   ［**ウィザードの選択**］ダイアログ ボックスが閉じて目的のウィンドウが再度表示されます。
5. 履歴トレンドの左上端を配置するウィンドウ上の位置までカーソルを移動します。トレンドをウィンドウに配置する場合に選択します。



6. トレンドをダブルクリックします。

   ［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスが表示されます。



7. [**Suggest**]を選択します。 履歴トレンド グラフ ウィザードによって、デフォルトの設定値が自動的にトレンドに割り当てられます。

   残る設定作業は、トレンド ペンへのタグの割り当てだけです。

トレンド グラフに表示するタグの設定

履歴トレンド グラフ ウィザードでトレンド ペンにタグを割り当てる作業は、履歴/リアルタイム ツールの手順と似ています。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

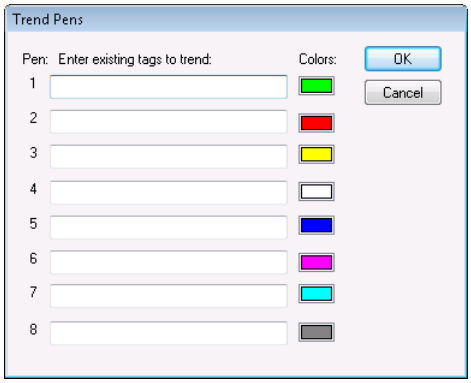
履歴トレンド グラフ ウィザードでタグを割り当てる

1. 履歴トレンドをダブルクリックすると、

   ［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスが表示されます。

2. **[ペン]** を選択します。

   ［**トレンド ペン**］ダイアログ ボックスが表示されます。



3. ［**ペン**］ボックスに既存のローカル タグの名前を入力します。入力できる最大文字数は 49 文字です。

   注：WindowViewer は必ず閉じてください。そうでないと、ペン ボックスを選択できません。

   ［**ペン**］ボックス内をダブルクリックすると、［**タグの選択**］ダイアログ ボックスが表示され、アプリケーションの［**ログ データ**］オプションが割り当てられたタグのリストが表示されます。［**タグの選択**］ダイアログ ボックスでタグを選択することにより、タグをペンに割り当てることができます。

4. ペンの色をデフォルト色から変更する場合は、各ペンの横にあるカラー ボックスをクリックして別の色を選択します。変更しない場合は、この手順を行わずにデフォルト色のままとします。

5. ［**OK**］をクリックして、［**トレンド ペン**］ダイアログ ボックスを閉じます。

6. ［**OK**］をクリックし、［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスを閉じます。

履歴トレンドの表示時間を設定する

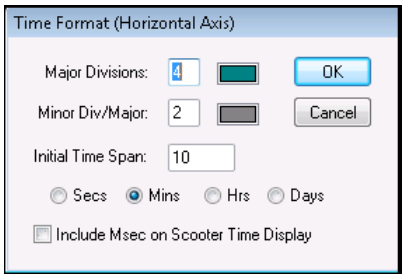
［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスには、履歴トレンド ウィザードで作成されたトレンドに表示される表示時間を手動で設定するオプションが用意されています。履歴トレンド ウィザードによるデフォルト設定をそのまま使用しない場合は、手動でトレンド時間の設定を変更できます。

履歴トレンドの時間範囲を設定するには

1. 履歴トレンドをダブルクリックすると、［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスが表示されます。

2. **時間**を選択します。［**時間の形式**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 時間の形式を設定するには、以下の手順を実行します。
   a. ［**主目盛り**］**ボックス**に、トレンドの横軸（時間軸）に表示する主目盛りの数を入力します。
   b. ［**主目盛り間の分割数**］ボックスに、各主目盛り内の小目盛りの数を入力します。
   c. ［**初期値**］ボックスに、トレンドの横軸の表示時間長を入力します。履歴トレンド ウィザードで作成されたトレンドは、WindowViewer でのアプリケーションの実行中に更新することができます。ユーザーは、トレンドの時間長を変更できますが、履歴トレンドの開始時には常に［**時間の形式**］ダイアログ ボックスで設定した時間間隔が使用されます。
   d. トレンドの時間単位を［秒］、［分」、［時」、［日］の中から選択します。
   e. オプションにより、スクータの時間表示をミリ秒単位まで含めることができます。現在時刻にミリ秒単位を追加して表示するスクータ スライダの例を以下に示します。

4. ［**OK**］をクリックして、［**時間の形式**］ダイアログ ボックスを閉じます。
5. ［**OK**］をクリックし、［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスを閉じます。

表示オプションの設定

［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスには、履歴トレンドの縦軸の単位を設定するオプションが用意されています。トレンドの縦軸に表示される主目盛りと小目盛りを手動で設定することができます。

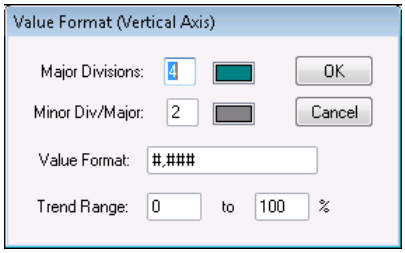
履歴トレンド グラフ ウィザードで表示オプションを設定する

1. 履歴トレンドをダブルクリックします。

   ［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスが表示されます。

2. [**Values** を選択します。

   ［**値の形式**］ダイアログ ボックスに、トレンドの縦軸（値軸）の設定オプションが表示されます。



3. 値の形式を設定するには、以下の手順を実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. ［**主目盛り**］ボックスに、トレンドの縦軸に表示する主目盛りの数を入力します。カラー ボックスを選択してカラー パレットを表示して、縦軸の主目盛り線に指定する色を選択します。

b. ［**主目盛り間の分割数**］ボックスに、縦軸の各主目盛り内に表示する小目盛りの数を入力します。カラー ボックスをクリックしてカラー パレットが表示して、横軸の小目盛り線に指定する色をクリックします。

c. ［**値の形式**］ボックスに、トレンドの縦軸（値軸）に表示する数値の形式を入力します。デフォルトの数値形式は **#,###** です。

d. ［**トレンドの範囲**］ボックスに、トレンドに表示するタグの工学単位に対する下限と上限のパーセントを入力します。

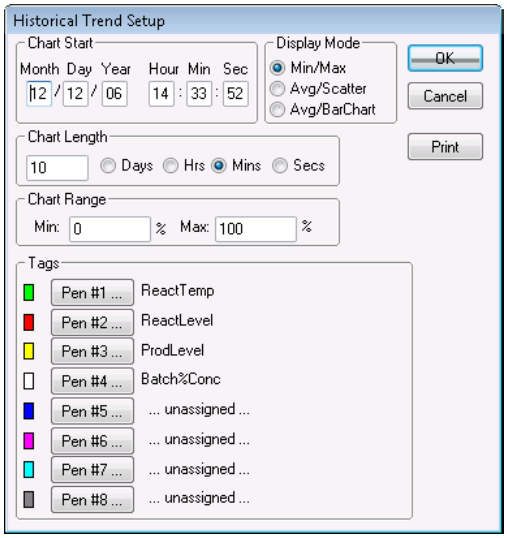
4. ［**OK**］をクリックして、［**値の形式**］ダイアログ ボックスを閉じます。
5. ［**OK**］をクリックし、［**履歴トレンド グラフ ウィザード**］ダイアログ ボックスを閉じます。

ランタイム中の設定変更

履歴トレントの設定で［**実行中の変更可能**］オプションを選択した場合は、ユーザーはランタイム中に履歴トレンドの設定をさまざまに変更できます。

ランタイム中に履歴トレンドを設定する

1. WindowViewer でトレンドをクリックします。［**履歴トレンドの設定**］ダイアログ ボックスが表示されます。



()

2. ［**表示開始時間**］領域で、表示を開始する日時を入力します。
3. ［**表示モード**］領域で、履歴トレンドのタイプを選択します。
4. ［**表示時間**］領域で、トレンドに表示する時間の長さを入力してから時間の単位を選択します。
5. ［**縦軸表示**］領域で、トレンドの縦軸に表示される工学値のスケールをパーセント単位で入力します。
6. ［**タグ変数**］領域で、各［**ペン #**］をクリックしてトレンド ペンにタグ変数を割り当てます。［**タグ変数を選択してください**］ダイアログ ボックスが表示されて、ログが可能なタグ変数が表示されます。
7. トレンド ペンに割り当てるタグ変数名をダブルクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

8. ［**OK**］をクリックして、ランタイム中のトレンド変更を保存します。

スクリプトの使用による履歴トレンド ウィザードのコントロール

QuickScript の関数をトレンド オブジェクトやアニメーション リンクの式と共に使用して、ランタイム中に履歴トレンドをコントロールできます。たとえば、QuickScript を使用して、トレンドを現在時刻に更新したり、タグ変数をトレンド ペンに再割り当てしたり、チャートにペンを接続したり、グリッドを再描画したり、スクータを削除または再表示したりすることができます。

トレンドを現在時刻に更新

最新のタグ変数データを表示するために履歴トレンドを更新するスクリプトを作成することができます。

HTUpdateToCurrentTime() 関数

HT**UpdateToCurrentTime()** 関数は、現在の時刻を終了時刻に一致させてデータを取得して表示します。開始時刻は、終了時刻からチャートの幅を引いたものとなります。

カテゴリ

履歴

構文

```
HTUpdateToCurrentTime(Hist_Tag);
```

引数

Hist_Tag
履歴トレンド名が割り当てられた履歴トレンドのタグ変数

例

以下のステートメントでは、現在の時刻の **Trend1** 履歴タグ変数のデータを取得して表示します。

```
HTUpdateToCurrentTime("Trend1");
```

現在時刻が午後 3 時 4 分でトレンドの幅が 60 秒の場合、更新された終了時刻は午後 3 時 4 分になります。また、更新された開始時刻は午後 3 時 3 分になります。

トレンド設定の変更

以下のスクリプト関数を使用して、履歴トレンドのペンに割り当てられたタグ変数を変更できます。

- HTSelectTag() 関数
- HTSetPenName() 関数

HTSelectTag() 関数

HTSelectTag() 関数を使用すると［**タグ変数を選択してください**］ダイアログ ボックスが開き、ユーザーはトレンド ペンに別のタグ変数を割り当てられます。

注記: タグ変数を選択してください］ダイアログ ボックスには、タグ変数ディクショナリで［**ログ データ**］オプションが選択されて、履歴ログ用に定義されたタグ変数のみがリスト表示されます。

カテゴリ

履歴

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
HTSelectTag();
```

備考

HTSelectTag() 関数は、タグ変数ディクショナリで［**ログ データ**］オプションが選択されているタグ変数のみを表示します。ただし、タグ ブラウザのフィルタを使用すれば、表示するタグのセットを絞り込むことができます。たとえば「A」で始まるすべてのタグ変数を指定できます。選択されたタグ変数を戻り値とするこの関数を、ペンにタグ変数を割り当てる関数のパラメータとして使用することができます。

例

以下の QuickScript は、［**タグ変数を選択してください**］ダイアログ ボックスを WindowViewer に表示します。ユーザーはリストからタグ変数を選択できます。選択したタグ変数は、HistTrend という名前の履歴オブジェクトのペン 1 に割り当てられます。

```
HTSetPenName("HistTrend",1,HTSelectTag());
```

参照項目

HTSetPenName()

HTSetPenName() 関数

HTSetPenName() 関数は、トレンド ペンに別のタグ変数を割り当てます。

カテゴリ

履歴

構文

```
HTSetPenName(Hist_Tag,PenNum,Tagname);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

PenNum
トレンドのペン番号（1～8）を示す整数値か整数型タグ変数。

Tagname
ペンに割り当てる新しいタグ変数名

備考

この QuickScript 関数は、ランタイム中に分散履歴プロバイダのタグ変数を追加する唯一の方法です。

ペン名のリファレンスに入力できる最大文字数は 49 文字です。

トレンド ペンの割り当てを解除しようとすると、以下のエラー メッセージが表示されることがあります。

```
VIEW /UpdateData:Invalid DBS.TAGNAME handle – 0
```

このエラーは、*histprovider.tag_name* の形式でリモート タグにすでに割り当てられたペンを解除しようとしたことが原因です。この問題を解決するには、［**ログ データ**］オプションが選択されたローカル タグ変数を作成する必要があります。さらに、以下のスクリプトを使用してペンの割り当てを解除します。

```
HTSetPenName( "HistTrend", 1, "localtag" );
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
{ローカルでログ済みのタグ変数---localtag にペンを割り当てる}
HistTrend.Pen1=None;
{ペンの割り当てを解除する}
```

ここで、None はタグ変数 ID 型のタグ変数です。

例

以下のステートメントは、Trend1 のペン 3 に OutletPressure タグ変数を割り当てます。

```
HTSetPenName("Trend1",3,"OutletPressure");
```

以下のステートメントは、Trend1 の TrendPen4 に **HistPrv1.Tag1** タグ変数を割り当てます。

```
HTSetPenName("Trend1",TrendPen4,"HistPrv1.Tag1");
```

参照項目

HTSelectTag()

トレンドと履歴データ関連情報の取得

履歴トレンドの情報を履歴トレンドの実行中に取得するスクリプトを作成することができます。次の関数を使用します。

- HTGetPenName() 関数
- HTGetTimeAtScooter() 関数
- HTGetTimeStringAtScooter() 関数
- HTGetValue() 関数
- HTGetValueAtScooter() 関数
- HTGetValueAtZone() 関数
- HTScrollLeft() 関数
- HTScrollRight() 関数
- HTZoomIn() 関数
- HTZoomOut() 関数

HTGetPenName() 関数

HTGetPenName() 関数は、履歴トレンドの特定のペン番号に現在割り当てられているタグ変数名を返します。

カテゴリ

履歴

構文

```
MessageResult=HTGetPenName(Hist_Tag,UpdateCount, PenNum);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

UpdateCount
トレンドの**.UpdateCount** ドットフィールドを示す整数値。この引数の値は、関数の値を再度求めるためのデータ変更トリガとしての機能を果たします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

PenNum
トレンドのペン番号（1～8）を示す整数値か整数型タグ変数。

例

以下のステートメントでは、Trend1 トレンドのペン 2 に割り当てられたタグ変数名が取得され、メッセージ型タグ変数の **TrendPen** にその名前を格納します。

```
TrendPen=HTGetPenName("Trend1", Trend1.UpdateCount,2);
```

HTGetTimeAtScooter() 関数

HTGetTimeAtScooter() は、*ScootNum* と *ScootLoc* 引数で指定したスクータ位置のサンプルの時刻を、1970 年 1 月 1 日午前 0 時から数えた秒数で返します。

カテゴリ

履歴

構文

```
IntegerResult=HTGetTimeAtScooter(Hist_Tag, UpdateCount,ScootNum,ScootLoc);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

UpdateCount
トレンドの**.UpdateCount** ドットフィールドを示す整数値。

ScootNum
左または右スクータを示す整数値：
1 = 左スクータ
2 = 右スクータ

ScootLoc
トレンド上の .ScooterPosRight または .ScooterPosLeft の位置の値を示す実数値。

備考

引数の UpdateCount、ScootNum、および ScootLoc に割り当てられた値が変わると、この式の値が再度求められます。これにより、新しいデータの検索後やスクータの移動後にこの式が評価されて値が更新されます。

例

以下のステートメントでは、Trend1 トレンドの現在の左スクータ位置の値に対応する時刻が秒単位で取得されます。

```
HTGetTimeAtScooter("Trend1",Trend1.UpdateCount,1, Trend1.ScooterPosLeft);
```

HTGetTimeStringAtScooter() 関数

HTGetTimeStringAtScooter() 関数は、特定のスクータの位置のサンプルに対応する時刻／日付を含む文字列を返します。

カテゴリ

履歴

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
MessageResult=HTGetTimeStringAtScooter(Hist_Tag, UpdateCount, ScootNum, ScootLoc,
Format_Text);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

UpdateCount
トレンドの**.UpdateCount** ドットフィールドを示す整数値。

ScootNum
左または右スクータを示す整数値：
1 = 左スクータ
2 = 右スクータ

ScootLoc
トレンド上の .ScooterPosRight または .ScooterPosLeft の位置の値を示す実数値。

Format_Text
時刻／日付に使用する形式を指定する文字列。以下の Format_Text 文字列を受け入れることができます。
"Date"、"Time"、"DateTime"、"DOWShort"（たとえば Wed）、および "DOWLong"（たとえば Wednesday）

備考

引数の UpdateCount、ScootNum、および ScootLoc に割り当てられた値が変わると、この式の値が再度求められます。これにより、新しいデータの検索後やスクータの移動後にこの式が評価されて値が更新されます。文字列の形式によって返り値の内容が決定されます。

例

以下のステートメントでは、Trend1 トレンドの現在の右スクータ位置の値に対応する日付と時刻が取得されます。この値は、メッセージ型タグ変数の NewRightTimeString に "Time" 形式で格納されます。

```
NewRightTimeString=HTGetTimeStringAtScooter ("Trend1",Trend1.UpdateCount,2,
Trend1.ScooterPosRight,"Time");
```

HTGetValue() 関数

HTGetValue() 関数は、トレンドの指定ペンに対して要求したタイプの値を返します。

カテゴリ

履歴

構文

```
RealResult=HTGetValue(Hist_Tag,UpdateCount, PenNum,ValType_Text);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

UpdateCount
トレンドの**.UpdateCount** ドットフィールドを示す整数値。

PenNum
トレンドのペン番号（1～8）を示す整数値か整数型タグ変数。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ValType_Text
戻り値のタイプを示す文字列：

PenAverageValue = トレンド全体の平均

PenMaxValue = トレンド全体の最大値

PenMinValue = トレンド全体の最小値

PenMaxEU = トレンド全体の工学値最大値

PenMinEU = トレンド全体の工学値最小値

PenStdDev = トレンド全体の標準偏差

備考

この関数は、要求した値を実数として返します。

例

以下のステートメントでは、PumpPress トレンドから取得された ペン 2 データ の標準偏差が得られます。この値は、メモリ実数型タグ変数 LeftHemisphereSD に保存されます。

```
LeftHemisphereSD=HTGetValue("PumpPress", PumpPress.UpdateCount,2,"PenStdDev");
```

HTGetValueAtScooter() 関数

HTGetValueAtScooter() 関数は、指定したスクータ位置、トレンド、およびペン番号のサンプルに対して要求したタイプの値を返します。UpdateCount 引数により、関数処理の終了後に式が評価されて値が求められます。

カテゴリ

履歴

構文

```
RealResult=HTGetValueAtScooter(Hist_Tag, UpdateCount,ScootNum,ScootLoc,PenNum,
ValType_Text);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

UpdateCount
トレンドの**.UpdateCount** ドットフィールドを示す整数値。

ScootNum
左または右スクータを示す整数値：
1 = 左スクータ
2 = 右スクータ

ScootLoc
トレンドの .ScooterPosRight または .ScooterPosLeft ドットフィールドを示す実数値。

PenNum
ペン番号（1～8）を示す整数値か整数型タグ変数。

ValType_Text
戻り値のタイプを示す文字列：

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

PenValue = スクータ位置の値。
PenValid = 無効の場合は 0、有効の場合は 1。

HTGetValueAtScooter() 関数に ValType_Text 引数を使用する場合は、上記の有効なタイプのいずれかを使用します。

例

以下の関数の戻り値は、Trend1 トレンドのペン 3 の右スクータ現在位置に対応する値が実際のサンプル値の場合は 1 で、無効値の場合は 0 です。

```
HTGetValueAtScooter("Trend1",Trend1.UpdateCount, 2,Trend1.ScooterPosRight,3, "PenValid");
```

HTGetValueAtZone() 関数

HTGetValueAtZone() 関数は、指定ペンのトレンドの左右スクータ間のデータに対する要求したタイプの値を返します。

カテゴリ

履歴

構文

```
RealResult=HTGetValueAtZone(Hist_Tag,UpdateCount,
Scoot1Loc,Scoot2Loc,PenNum,ValType_Text);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

UpdateCount
トレンドの**.UpdateCount** ドットフィールドを示す整数値。関数を評価して値を求めるためのトリガとしてのみ使用されます。

Scoot1Loc
トレンドの .ScooterPosLeft ドットフィールドを示す実数値。関数を評価して値を求めるためのトリガとしてのみ使用されます。

Scoot2Loc
トレンドの .ScooterPosRight ドットフィールドを示す実数値。関数を評価して値を求めるためのトリガとしてのみ使用されます。

PenNum
トレンドのペン番号（1～8）を示す整数値か整数型タグ変数。

ValType_Text
戻り値のタイプを示す文字列
PenAverageValue = スクータ間ゾーンの平均値
PenMaxValue = スクータ間ゾーンの最大値
PenMinValue = スクータ間ゾーンの最小値
PenMaxEU = スクータ間ゾーンの工学値最大値
PenMinEU = スクータ間ゾーンの工学値最小値
PenStdDev = スクータ間ゾーンの標準偏差

備考

戻り値は、指定したタイプの計算値を示す実数です。Scoot1Loc と Scoot2Loc 引数に指定した定数は結果に影響を与えるものではなく、関数を評価して値を更新するためのトリガに使用されるだけです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Scoot1Loc と Scoot2Loc 引数に指定した値とは関係なく、この関数ではトレンド タグ変数の .ScooterPosLeft と .ScooterPosRight ドットフィールドの値が直接使用されます。

例

以下のステートメントでは、Trend1 トレンドの ペン 1 の左右スクータ間のデータに対する平均値が計算されます。この値は、メモリ実数型タグ変数の AvgValue に格納されます。

```
AvgValue=HTGetValueAtZone("Trend1", Trend1.UpdateCount,Trend1.ScooterPosLeft,
Trend1.ScooterPosRight,1,"PenAverageValue");
```

トレンドのパンとズームを行う

ランタイム中に履歴トレンドの特定のデータを選択する関数を含む QuickScript を作成できます。

HTScrollLeft() 関数

HTScrollLeft() 関数は、トレンドの表示時間全体に対するパーセントを指定して、トレンドの開始時刻を現在の開始時刻より早い時刻に設定します。その結果、トレンドの表示時間全体に対するパーセントで指定分だけ早い時刻にチャートが左スクロールされます。

カテゴリ

履歴

構文

```
HTScrollLeft(Hist_Tag,Percent);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

Percent
左にスクロールする間隔をチャートの表示時間に対するパーセント（0.0 ～ 100.0）で示した実数値。

例

以下のステートメントは、PumpPress トレンドの幅全体の 10 パーセント分だけ時刻／日付を左スクロールします。

```
HTScrollLeft("PumpPress",10.0);
```

現在の表示開始時刻が 午後 12 時 00 分 00 秒で表示幅が 60 秒の場合、この関数の実行後の新しいトレンドの開始時刻は午前 11 時 59 分 54 秒になります。

HTScrollRight() 関数

HTScrollRight() 関数は、トレンドの表示幅に対するパーセントを指定して、トレンドの開始時刻を現在の開始時刻より遅い時刻に設定します。その結果、トレンドの表示幅に対するパーセントで指定分だけチャートの時刻／日付が右スクロールされます。

カテゴリ

履歴

構文

```
HTScrollRight(Hist_Tag,Percent);
```

引数

Hist_Tag

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

トレンド名が割り当てられた履歴トレンドのタグ変数

Percent
チャートを右にスクロールするパーセント（0.0 ～ 100.0）を示す実数値

例

以下のステートメントは、PumpPress トレンドを 20 パーセント分だけ右スクロールします。

```
HTScrollRight("PumpPress",20.0);
```

現在の表示開始時刻が 午後 12 時 00 分 00 秒で表示幅が 60 秒の場合、この関数の実行後の新しいトレンドの開始時刻は午後 12 時 00 分 12 秒になります。

HTZoomIn() 関数

HTZoomIn() 関数は、新しいチャート幅と開始時刻を計算します。トレンドのスクータがトレンドの左右の端にある場合、新しいチャート幅は古いチャート幅の半分になります。新しい開始時刻は、*LockString* 引数の値に基づいて計算されます。

スクータがトレンドの左右の端にない場合は、HTZoomIn() はスクータで定義されるゾーンに対してトレンドをズームして、*LockString* 引数は無視されます。

カテゴリ

履歴

構文

```
HTZoomIn(Hist_Tag,LockString);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

LockString
ズームのタイプを示す文字列：

StartTime	ズーム前の開始時刻を維持する
Center	ズーム前の中央時刻を維持する
EndTime	ズーム前の終了時刻を維持する

備考

スクータがトレンドの左右の端にない場合は、新しいチャート幅は .ScooterPosLeft と .ScooterPosRight の位置の間の時間になります。この場合、LockString の値は使用されません。チャート幅の最小値は 1 秒間です。ズーム後のスクータ位置の設定は、.ScooterPosLeft=0.0 および .ScooterPosRight=1.0 になります。

例

下記のステートメントは、Trend1 トレンドの表示を 2 倍にズームして、トレンドの開始時刻を維持します。Trend1.ScooterPosRight は 1.0、Trend1.ScooterPosLeft は 0.0 になります。ズーム前の開始時刻が午後 1 時 25 分 00 秒でチャート幅が 30 秒の場合、新しい開始時刻は午後 1 時 25 分 00 秒のままで、新しいチャート幅は 15 秒になります。

```
HTZoomIn("Trend1","StartTime");
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

HTZoomOut() 関数

HTZoomOut() 関数は、新しいチャート幅と開始時刻を計算します。新しいチャート幅は、古いチャート幅に 2 を掛けたものです。新しい開始時刻は、*LockString* 引数の値に基づいて計算されます。

カテゴリ

履歴

構文

HTZoomOut(*Hist_Tag*,*LockString*);

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

LockString
ズームのタイプを示す文字列：
StartTime = ズーム前の開始時刻を維持する
Center = ズーム前の中央時刻を維持する
EndTime = ズーム前の終了時刻を維持する

備考

現在のスクータの位置は HTZoomOut() に影響しません。この関数のズームが終了すると、新しいスクータ位置の設定は .ScooterPosLeft=0.0 および .ScooterPosRight=1.0 になります。

例

以下のステートメントは、Volume トレンドの表示を 2 倍にズーム アウトして、トレンドの中央時刻を維持します。ズーム前の開始時刻が午後 2 時 15 分 00 秒でチャート幅が 30 秒の場合、新しい開始時刻は午後 2 時 14 分 45 秒になります。チャート幅は 60 秒で、トレンドの中央時刻は 午後 2 時 15 分 15 秒のままです。

```
HTZoomOut("Volume","Center");
```

トレンドの印刷

PrintHT() 関数をスクリプトで使用して、WindowViewer の画面に現在表示されている履歴トレンドを印刷できます。

PrintHT() 関数

PrintHT() 関数は、現在画面に表示されている履歴トレンドを印刷します。**PrintHT()** 関数は通常、履歴トレンド ウィンドウに配置されている画面ボタンと関連付けられます。ユーザーはこのボタンを選択して、表示されている履歴トレンドを現在の値で印刷します。

カテゴリ

履歴

構文

PrintHT(*Trend_Tag*);

引数

Trend_Tag
履歴トレンド タグ変数。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例では、現在画面に表示されている PumpPress 履歴トレンドが印刷されます。

```
PrintHT(PumpPress);
```

トレンドのトラブルシューティング

正しく取得されたデータが履歴トレンドに表示されているかを確認する QuickScript を作成できます。トレンドのトラブルシューティングには、HTGetLastError() 関数を使用します。

HTGetLastError() 関数

HTGetLastError() 関数をスクリプトに使用して、特定の履歴トレンド ペンの最終データの取り込み中にエラーが発生したかを確認できます。

カテゴリ

履歴

構文

```
[Result=]HTGetLastError(Hist_Tag,UpdateCount, PenNum);
```

引数

Hist_Tag
トレンド名が割り当てられた履歴トレンドのタグ変数

UpdateCount
トレンドの **.UpdateCount** ドットフィールドを示す整数値。

PenNum
トレンドのペン番号（1～8）を示す整数値か整数型タグ変数。

Result
特定のペンに対する最終のスクリプト関数呼び出しの結果を示す、タグ変数に割り当てられた整数値。

0 = エラーなし

1 = 一般サーバー エラー

2 = 古い要求

3 = ファイル エラー

4 = サーバーの未ロード

5 = 関数に指定したトレンド／ペンが存在しない

6 = トレンド タグ変数がサーバーに存在しない

7 = 関数に指定したペン番号が無効（1～8 の範囲外）

8 = ペン番号にタグ変数が割り当てられていない、またはログ済みでないタグ変数が割り当てられている

例

以下のステートメントでは、Trend1 トレンドのペン 3 に対する最終のデータ取り込みのステータスが取得され、整数型タグ変数 ResultCode にその結果が割り当てられます。

```
[ResultCode=]HTGetLastError("Trend1", Trend1.UpdateCount,3);
```

アニメーションのアナログ値の表示 QuickScript では、以下のステートメントを使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
HTGetLastError("Trend1",Trend1.UpdateCount,3);
```

トレンドでのリアルタイム値の表示

リアルタイム トレンドを作成するには 2 つの方法があります。リアルタイム トレンド オブジェクトには、データの選択、時間範囲の設定、グラフの外観の指定を行うためのコントロールの標準セットがあります。さらに InTouch には、リアルタイム トレンドおよび履歴トレンドを作成するためのオプション コントロールとして 16 ペン トレンド ウィザードも用意されています。16 ペン トレンド ウィザードでのリアルタイム トレンドの作成の詳細については、『*AVEVA™ InTouch HMI 管理ガイド*』の「16 ペン トレンドの作成」を参照してください

リアルタイム トレンド オブジェクトの使用

リアルタイム トレンドを作成して、アプリケーションで現在の値を表示できます。

リアルタイム トレンドの作成

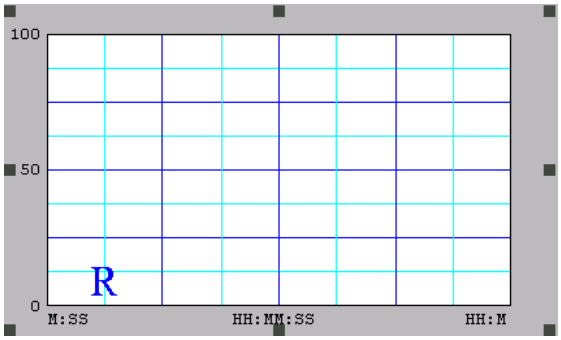
リアルタイム トレンド ツールを使用して、ウィンドウ内にトレンド オブジェクトを作成できます。WindowMaker では、ウィンドウに新しく貼り付けたリアルタイム トレンド オブジェクトにはデフォルト設定値が適用されます。リアルタイム トレンドの設定を行うとその後は、新しいリアルタイム トレンド オブジェクトの初期設定として WindowMaker は最後の設定値を使用します。

ウィンドウ枠の中であれば、どんなサイズのトレンド グラフでも表示できます。

リアルタイム トレンドの作成

1. ［**描画**］メニューの［**トレンド**］グループで［**リアルタイム**］をクリックします。
2. リアルタイム トレンドを作成するウィンドウの領域の上にマウスを移動します。マウスを斜めにドラッグして、目的のトレンドのサイズの長方形を作成します。

   ウィンドウにリアルタイム トレンド オブジェクトが表示されます。



3. 必要に応じて、オブジェクトのハンドルを使用してトレンドの高さと幅を調整します。

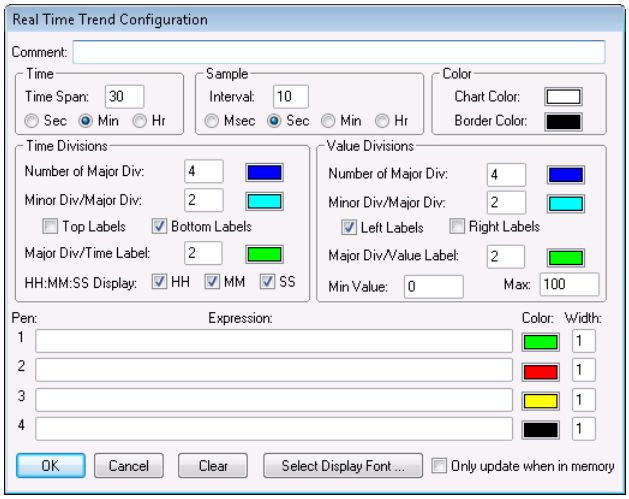
リアルタイム トレンドに表示するタグ変数の選択

リアルタイム トレンド ペンによって、ローカル タグ変数や 1 つ以上のローカル タグ変数が含まれる式の現在データのグラフが作成されます。リアルタイム トレンドにタグ変数データを表示するペンの設定を行います。

リアルタイム トレンド タグ変数を設定する

1. ウィンドウ内のトレンド オブジェクトをダブルクリックします。［**リアルタイム トレンド グラフの設定**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［式］領域に、ローカル タグ変数か 1 つ以上のローカル タグ変数を含む式の名前を入力します。

   ［ペン］ボックス内をダブルクリックすると、［**タグ変数を選択してください**］ダイアログ ボックスが表示され、アプリケーションに定義されたタグ変数のリストが表示されます。［**タグ変数を選択してください**］ダイアログ ボックスでタグ変数を選択することにより、タグ変数をペンに割り当てることができます。

3. タグ変数が割り当てられた各ペンの横にあるカラー ボックスを選択すると、カラー パレットが開きます。

4. ペンに割り当てる色を選択します。

5. ［**幅**］ボックスに、トレンドに表示される各ペンの線幅をピクセル単位で入力します。

   線幅に 1 よりも大きい値を選択すると、トレンドの更新や印刷に要する時間が長くなります。

6. トレンドがアクティブ ウィンドウに表示されている間だけトレンドを更新するには、［**メモリ内常駐時のみ更新**］ チェック ボックスをオンにします。

   このオプションを選択しないと、ウィンドウが閉じている場合でも常にトレンドは更新されます。トレンドが常時更新されていると、システムのパフォーメンスが低下します。

7. ［**リアルタイム トレンド グラフの設定**］ダイアログ ボックスを開いたまま、「リアルタイム トレンドの時間範囲と更新率の設定」に記載されている次の手順に進みます。

リアルタイム トレンドの時間範囲と更新率の設定

リアルタイム トレンドの時間範囲と更新率を設定できます。

リアルタイム トレンドの時間範囲と更新率を設定する

1. トレンド オブジェクトをダブルクリックします。［**リアルタイム トレンドの設定**］ダイアログ ボックスが表示されます。

2. ［**時間**］領域の［**時間範囲**］ ボックスに、トレンドの横軸（X 軸）に表示する時間長を入力します。

3. トレンド時間の単位を選択します。

   - 秒
   - 分
   - 時

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

たとえば、[時間範囲] ボックスに 30 と入力してから [分] を選択すると、チャートに表示される時間範囲は 30 分になります。

4. [サンプル] 領域の [間隔] ボックスに、トレンドの式を評価してチャートを更新する間隔の値を入力します。
5. 間隔の単位を選択します。
   - ミリ秒
   - 秒
   - 分
   - 時

   たとえば、[間隔] に 10 を入力してから [秒] を選択すると、リアルタイム トレンドは 10 秒ごとに更新されます。
6. [リアルタイム トレンドの設定] ダイアログ ボックスを開いたまま、「リアルタイム トレンド 表示オプションを設定する」に記載されている次の手順に進みます。

リアルタイム トレンド 表示オプションを設定する

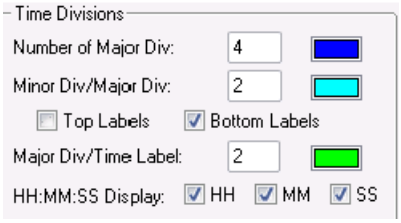
リアルタイム トレンドの表示方式を設定することができます。

リアルタイム トレンド 表示オプションを設定する

1. トレンド オブジェクトをダブルクリックします。[リアルタイム トレンドの設定] ダイアログ ボックスが表示されます。
2. [配色] 領域で色を設定します。以下のいずれかを実行します。
   - **チャート色**ボックスを選択し、カラーパレットを開きます。トレンドの背景色を選択します。

     デフォルトの背景色は白です。背景色を別の色にすると、トレンドの印刷速度が著しく遅くなります。
   - **輪郭色**ボックスを選択してカラー パレットを開きます。トレンドの輪郭色を選択します。
3. [時間軸の目盛り] 領域で、時間軸の目盛りを設定します。以下の手順を実行します。
   - [主目盛り数] ボックスに、トレンドの主目盛り数を入力します。主目盛りは、トレンドの横軸（時間軸）に表示されます。

     時間軸の主目盛り数は、[主目盛り間の分割数] の値で割り切れる必要があります。たとえば、目盛り数 20 は、[主目盛り間の分割数] の値 4 で割り切れます。



   - 主目盛り線の色を選択します。
   - [主目盛り間の分割数] ボックスに、各主目盛り内に表示される小目盛りの数を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

時間軸の小目盛りの数は主目盛り間隔内で均等に割り切れる必要があります。たとえば、主目盛り間隔を 60 秒に設定して［**主目盛り間の分割数**］に 2 を入力すると、時間軸の小目盛り間隔は 30 秒になります。

- 小目盛り線の色を選択します。
- ［**上ラベル**］か［**下ラベル**］チェック ボックスをオンにして、トレンドの時間ラベルの表示位置を指定します。

  時間ラベルをトレンドの上下に表示するように両方のオプションをオンにできます。両方のオプションをオフのままにすると、トレンドの横軸に時間ラベルは表示されません。
- 時間ラベルを使用する場合は、［**主目盛り／時間ラベル**］に時間ラベルあたりの時間軸の主目盛り線の数を入力します。主目盛り数は、［**主目盛り間の分割数**］の値で割り切れる必要があります。

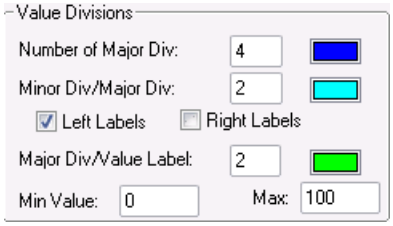
  時間軸目盛りのラベルの色を選択します。
- 時間軸の主目盛りラベルに表示される時間単位を選択します。

  時（HH）

  分（MM）

  秒（SS）

4. ［**縦軸の目盛り**］領域で、トレンドの縦軸の表示を設定します。



［**縦軸の目盛り**］オプションの設定は、［**時間軸の目盛り**］オプションの設定と似ています。y- 軸の主目盛りと小目盛りは、時間ではなくデータ値の大きさを表します。縦軸では、すべてのタグの工学単位に基づいて、トレンドに表示するデータ値の範囲が指定されます。

主目盛り値と小目盛り値に小数点を表示するには、［**最小値**］と［**最大値**］オプションに実数値を入力します（例：0.00 ～ 100.00）。

5. ［**表示フォントの選択**］を選択します。 ［**フォント**］ダイアログ ボックスが表示されて、トレンドに表示されるテキストのフォント、スタイル、サイズの設定オプションが表示されます。
6. **[OK]** を選択します。

ランタイム中のトレンド印刷

トレンドを高速印刷できるようにするにはいくつかの要素があります。第 1 の要素は、印刷ページ上のトレンドのサイズです。また印刷速度は、選択した表示モードによっても異なります。［最大／最小］と［平均／散布］トレンドは、［平均／棒］トレンドよりも短い時間で印刷されます。また、線が長く太いほど印刷速度が遅くなります。

印刷速度に影響する別の要因には、トレンドの背景色があります。ほとんどの場合、背景に色をつけずに白の背景で印刷するほうが高速です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

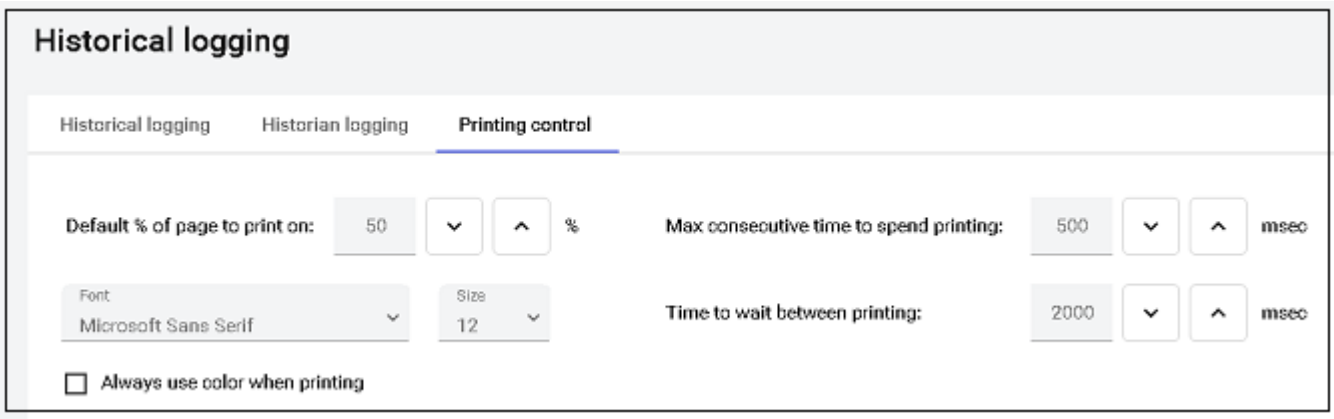
トレンド印刷オプションの設定

トレンドの印刷方法を決めるオプションを設定できます。

履歴トレンドの印刷を設定する

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］をクリックし、［履歴ログ］をクリックします。

   ［履歴ログ］設定画面が表示されます。
3. ［印刷制御］タブを選択します。



4. トレンドを印刷するページの割合を［履歴トレンド グラフの印刷割合］ボックスで指定します。

   たとえば、50 と入力すると、縦横ともページの半分に印刷されます。50 パーセントでトレンドを印刷すると、全ページ印刷よりもはるかに高速になります。

   PrintWindow() QuickScript 関数を使用して印刷することもできます。
5. ［連続印刷処理時間］ ボックスに、ミリ秒単位の処理時間を入力します。

   タイム スライスは、印刷モジュールをバックグラウンドで実行してトレンドを印刷するためにコンピュータ処理に割り当てられる時間を表します。トレンド印刷に利用できるタイム スライスの時間を長くすれば高速印刷できますが、コンピュータ上で実行されている他の処理が遅くなります。
6. ［印刷処理の実行間隔］ ボックスに、次のタイム スライスまでの印刷モジュールの待ち時間をミリ秒の単位で入力します。

   プロセッサ タイム スライス間の待機期間が短くなると、トレンドをより迅速に印刷できます。
7. ［フォント］ドロップダウンリストから、トレンドに表示する［印字フォント］を選択します。
8. カラー印刷の［印刷時に常に色を使用］チェック ボックスをオンにします。
9. 保存を選択します。

他の InTouch ノードや Historian Server の履歴タグ値の表示

リモート保存されたデータを使用して履歴トレンドを作成するには、InTouch 履歴プロバイダ リストにリモート プロバイダを登録する必要があります。このリストには、各履歴プロバイダの名前とネットワーク位置が指定されます。履歴トレンド ペンがリモート履歴プロバイダのタグを指し示すと、そのたびにこの名前が参照されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

リモート履歴プロバイダを定義して、リモート位置に保存されたタグのデータに履歴トレンド ペンを割り当てることができます。以下の操作を実行できます。

- リモート履歴プロバイダを設定する
- リモート履歴プロバイダのデータを表示するようにペンを設定する
- タグ ブラウザを使用して、リモート履歴プロバイダに保存されたタグのデータにペンを割り当てる
- QuickScript を使用して、リモート タグにペンを割り当てる

リモート履歴プロバイダのデータの使用の詳細については、『*AVEVA™ InTouch HMI アプリケーション配置*』の「アプリケーションの分散」を参照してください。

Historian Server での InTouch HMI の使用

Historian Server は、産業用アプリケーション向けに特に設計されたリアルタイムのリレーショナル データベースです。Historical Logger を設定して、InTouch 履歴データを AVEVA 履歴サーバー データベースの Historian に保存できます（オプション）。

注：リモート履歴プロバイダでの InTouch HMI の設定の詳細については、『*AVEVA™ InTouch HMI アプリケーション配置*』の「アプリケーションの分散」を参照してください。

Historian を使用して履歴データを保存する場合、データベースへの接続を指定するために WindowMaker の分散名前マネージャを使用する必要があります。

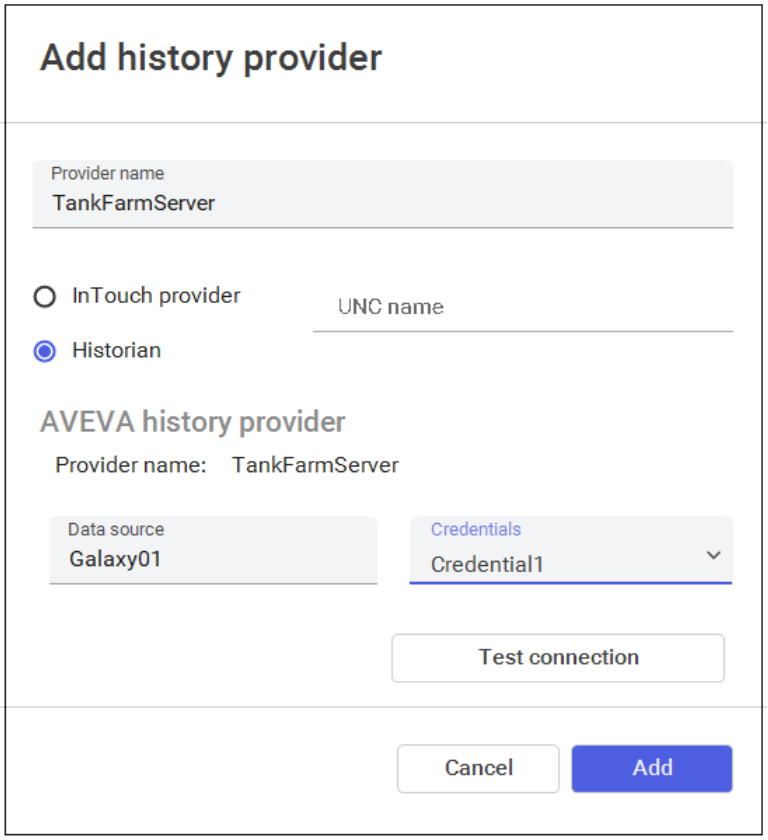
Historian Server データベースへの 接続を設定する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］を強調表示し、［分散名前マネージャ］を選択します。

   ［**分散アラーム**］のタブおよび［**分散履歴**］のタブを含む［分散名前マネージャ］設定画面が表示されます。
3. ［**分散履歴**］タブで［**+**］アイコンを選択して履歴プロバイダを追加します。または、Alt+A を押します。

   ［**履歴プロバイダの追加**］ダイアログが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**プロバイダ名**］ボックスに、新しい履歴プロバイダの名前を入力します。

   プロバイダ名には、16 文字以下の英数字を使用できます。

5. ［**Historian**］を選択します。

   a. ［**データ ソース**］ボックスに、Historian Server がインストールされているサーバーのノード名を入力します。

   b. ［**資格情報**］ドロップダウンで認証に使用する資格情報を選択します。

   注：スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「資格情報マネージャの操作」を参照してください。

   a. ［**接続テスト**］を選択して、Historian Server データベースへの接続を確認します。データベースへの接続が成功したかどうかを通知するメッセージが表示されます。

6. ［**追加**］を選択して、ダイアログ ボックスを閉じます。［履歴プロバイダ］リストに Historian Server ノードが表示されます。

7. **保存**を選択します。

リモート トレンド データを表示するためのペンの設定

履歴トレンドには、ローカルとリモートの両方の履歴プロバイダからのタグ変数データを表示できます。リモート履歴プロバイダのデータを表示するようにトレンド ペンを指定することができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

リモート履歴プロバイダのタグ変数を表示する

1. 履歴トレンドをダブルクリックして［**履歴トレンド グラフの設定**］ダイアログ ボックスを表示します。
2. 各ペンの［**タグ変数**］ボックスに、リモート履歴プロバイダへのリファレンスを入力します。リモート履歴プロバイダへのリファレンスの形式は以下のとおりです。

   history_provider_name.tag_name

   例：
   ```
   TankFarm1.Pump1RPM
   ```

   履歴トレンドの各ペンは、異なるリモート履歴プロバイダを参照することができます。
3. **OK** を選択して 、設定変更を保存します。

注：.TagID ドットフィールドは、リモート履歴プロバイタのタグ リファレンスには使用できません。

タグ ブラウザを使用してリモート履歴プロバイダにペンを割り当てる

タグ ブラウザを使用してリモート履歴プロバイダのタグ変数データにトレンド ペンを割り当てるための手順を以下に説明します。タグ ブラウザを使用してタグ変数を選択することにより、各タグ変数名を手入力する必要がなくなり、エラー発生の確率を低減できます。

アクセス名に指定するリモート ノード名は、タグ変数を含むノードの実際の名前である必要はありません。ただし、アクセス名を作成して、リモート履歴プロバイダをタグ ソースとして定義する必要があります。アクセス名の作成の詳細については、アクセス名の設定を参照してください。

リモート履歴プロバイダをタグ ソースとして定義する

1. 履歴プロバイダのあるノード名を指定するアクセス名を作成します。
2. 履歴トレンドをダブルクリックして［**履歴トレンド グラフの設定**］ダイアログ ボックスを開きます。
3. ペンの［**タグ変数**］入力ボックスをダブルクリックすると、［**タグ変数を選択してください**］ ダイアログ ボックスが表示されます。
4. ［**タグ ソースの定義**］を選択して、リモート履歴プロバイダをタグ ソースとして定義します。
5. ［**タグ ソース**］の矢印を選択して新しいリモート履歴プロバイダのタグ ソースをリストから選択するか、［**ツリー ビュー**］ボタンを選択してツリー ビュー ペインからタグ ソースを選択します。［**タグ変数を選択してください**］ダイアログ ボックスが更新されて、選択したリモート履歴プロバイダのタグ変数が表示されます。
6. 履歴ペンに割り当てるタグ変数を選択して、［**OK**］を選択します。［**履歴トレンド グラフの設定**］ダイアログ ボックスに戻り、選択したタグ変数がペンの［**タグ変数**］ボックスに *AccessName*:*Item* として表示されます。
7. アクセス名の部分を［**分散名前マネージャ**］で定義した履歴プロバイダ名で置き換えます。

   たとえば、HistPrv1.*tag_name*

   この手順は面倒なように見えますが、タグ ブラウザで履歴プロバイダをタグ ソースとして定義しておくと、別のタグ変数入力フィールドをダブルクリックした際に、**タグ ブラウザ**でタグ変数名をダブルクリックしてアクセス名の部分を履歴プロバイダ名で置き換えるだけで済むようになります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

履歴トレンドをランタイム中に変更できるよう設定されている場合は、WindowViewer でユーザーがペン ボタンを選択してタグ変数を変更しようとするとタグ ブラウザが表示されますが、変更できるのはローカル アプリケーションのタグ変数だけです。

QuickScript を使用してリモート履歴プロバイダにペンを割り当てる

InTouch アプリケーションの実行中に、リモート履歴プロバイダのタグ変数データを表示するようにトレンド ペンを設定できます。HTSetPenName() 関数を使用して、リモート履歴プロバイダのタグ リファレンスを指定する QuickScript を作成してください。以下に例を示します。

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

この例では、HistPrv1 リモート履歴プロバイダの **Boiler1** タグ変数の値を表示する履歴トレンドのペン番号に 1 が指定されます。

ランタイム中の［**履歴トレンド グラフの設定**］ダイアログ ボックスと **Pen** ドットフィールドは、リモート履歴プロバイダに対してはサポートされていません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザー定義データ型

ユーザー定義データ型（UDT）は、個々のユーザー定義メンバーで構成される階層データ構造です。各メンバーは、整数、ブール、文字列、その他の一般的なデータ型、またはその他の UDT として独自のデータ型を持つことができます。

UDT を使用すると、複数のレベルの入れ子構造を作成するオプションを提供することによってエンジニアリングの労力を削減できます。例えば、反応炉を表現する個々のタグのセットを作成する代わりに InTouch で 1 つの反応炉 UDT を作成できます。

UDT の概要

InTouch HMI は、以下の UDT サポートを提供します。WindowMakerWindowMaker の左側に［**ユーザー定義型**］ペインが表示されます。これには以下が含まれます。

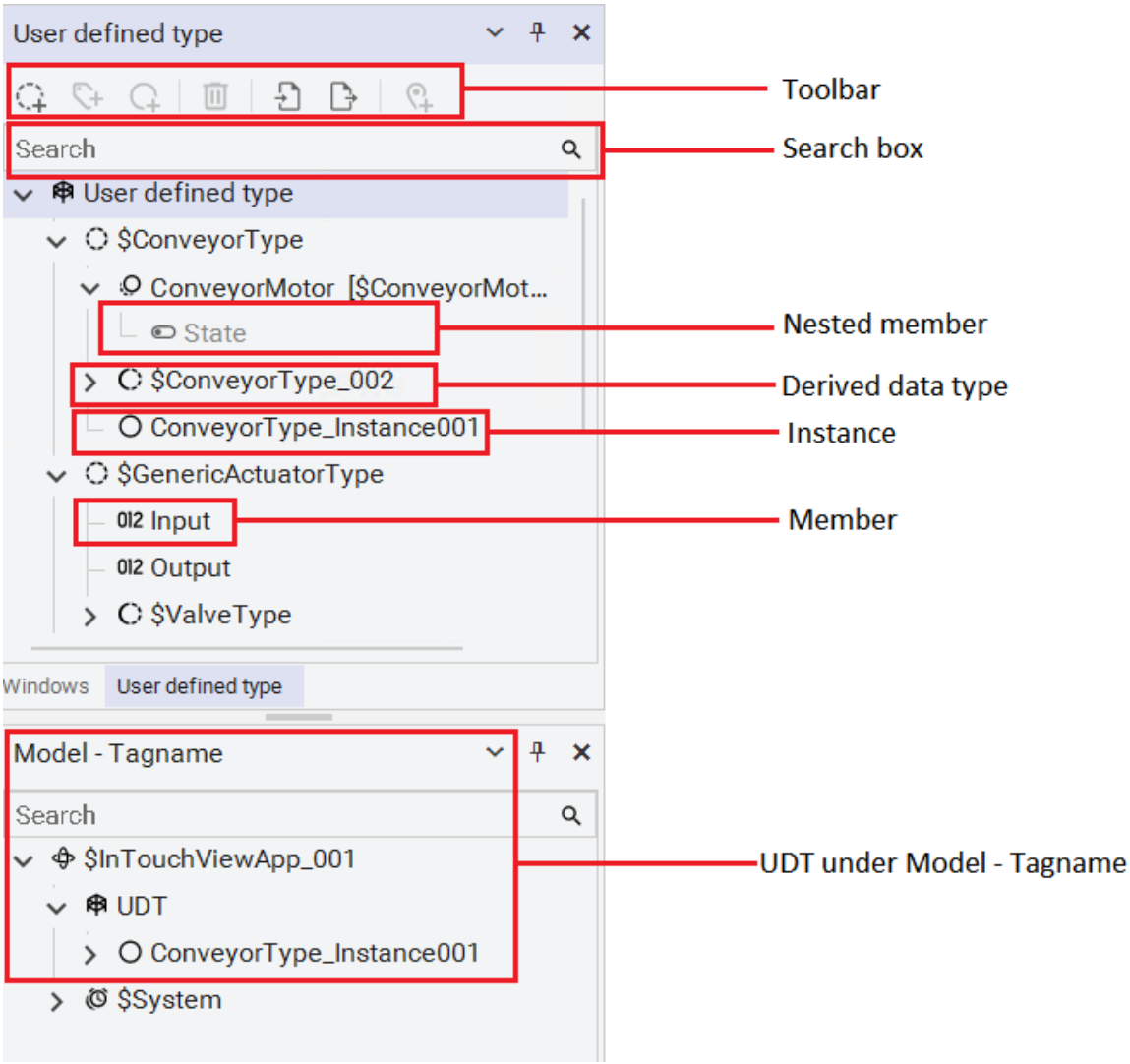
- ［**検索**］ボックス - 目的のデータ型、派生データ型、インスタンス、またはメンバーを検索できます。
- **ツールバー** - ツールバーには以下のオプションが含まれます。
  - **新規データ型を追加**
  - **新規メンバーを追加**
  - **新規インスタンスを追加**
  - **削除**
  - **データ型をインポート**
  - **データ型のエクスポート**
  - **場所メンバーを追加**

ユーザー定義型階層には、これらのアイテムを以下の順序で含めることができます。

1. **直接メンバー** - 現在のユーザー定義型内で定義されたメンバー。
2. **継承メンバー** - その他のデータ型から継承されたメンバー。これらはグレーアウトされて表示されます。
3. **派生データ型** - ベース データ型から派生したデータ型（ベース データ型は読み取り専用で InTouch で事前定義されています）。これには、派生元のベース データ型の配下のすべてのメンバーが含まれます。既存のアイテムとは別に、メンバーを派生データ型に追加できます。ベース データ型と派生データ型には、デフォルトで $ 記号の接頭辞が付けられています。派生型の名前を変更した場合、名前の最初の文字として $ が引き続き使用されます。
4. **インスタンス** - インスタンスはユーザー定義型の物理または仮想オブジェクトを表します。グラフィック アニメーションまたはスクリプトでユーザー定義型を使用するには、インスタンスを作成する必要があります。インスタンス タグ メンバーは、InTouch 基本タグです。

注 : [**プロパティ**] グリッド内の TBT を編集できます。タグ名辞書の UDT は表示または編集できません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



UDT の仕様

ユーザー定義データ型（UDT）は、以下の仕様に準拠する必要があります。

名前の長さ

- データ型、派生データ型、およびインスタンスの名前の最大長は 32 文字です。
- メンバー データ型とメンバーの名前の最大長は 32 文字です。

有効な名前

- インスタンス名には、英数字に加えて、$（ドル記号）、#（ポンド記号）、_（アンダースコア）の特殊文字を含めることができます。
- $ 文字は最初の文字として使用できません。

最大のネストレベル

- 最大で 6 つのネストされた UDT レベルを使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

たとえば、"Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6.Member1" の場合を考えてみます。

ここで、

- "Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6" は 6 つのネストされた UDT レベルです。
- "Member1" は、メモリ整数型などの基本 InTouch タグ タイプです。

注：循環参照はサポートされません。

手動でのユーザー定義データ型の作成

ユーザー定義データ型の作成および管理は、WindowMakerWindowMaker の［**ユーザー定義型**］ペインで行うことができます。独立した複数の操作のセットを使用して、個々の UDT を作成し、複数の UDT で構成される階層構造を構築できます。以下のトピックでは、これらの操作について説明します。

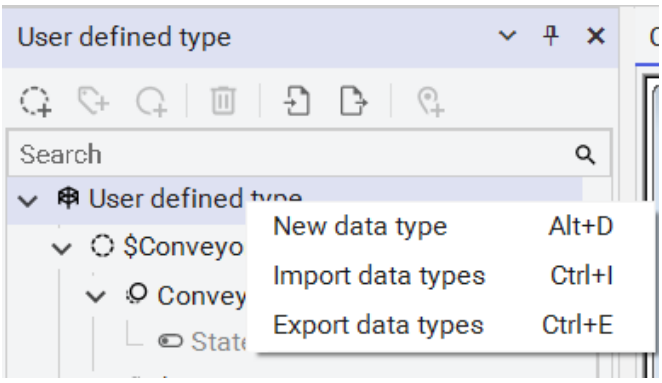
- 新しいデータ型の作成
- 新しいメンバーの作成
- 新しい派生データ型の作成
- 新しいインスタンスの作成
- 別のデータ型の下でのメンバーの入れ子
- データ型をインポート
- データ型のエクスポート
- ユーザー定義データ型の管理
- ユーザー定義データ型の一括編集

注意：このセクションで示される UDT 名、メンバー名、およびプロパティ名は説明のみを目的としています。型とメンバーには実際のアプリケーションに適切な名前を付けてください。

新しいデータ型の作成

新しいデータ型の作成

1. ［**ユーザー定義型**］ペインで、［**ユーザー定義型**］を右選択し、コンテキスト メニューから［**新規データ型**］を選択するか、ツールバーの［**新規データ型**］アイコンを選択します。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

または

ツールバーで [新規データ タイプ] を選択します。

または

ショートカットキー **Alt+D** を使用します。

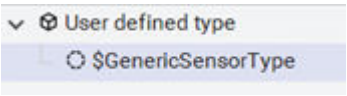
データ型が作成されます。

デフォルトのデータ型名は $Data type_00*n* です。*n* の初期値は 1 です。データ型の名前の先頭には $ 記号が付きます。$ 記号を削除することはできません。データ型の名前には、英数字に加えて $、#、および _（アンダースコア）文字を使用できます。名前の最大長は 32 文字です。

2. データ型の名前を変更するには、データ型を右クリックして［**名前の変更**］を選択します。

例：

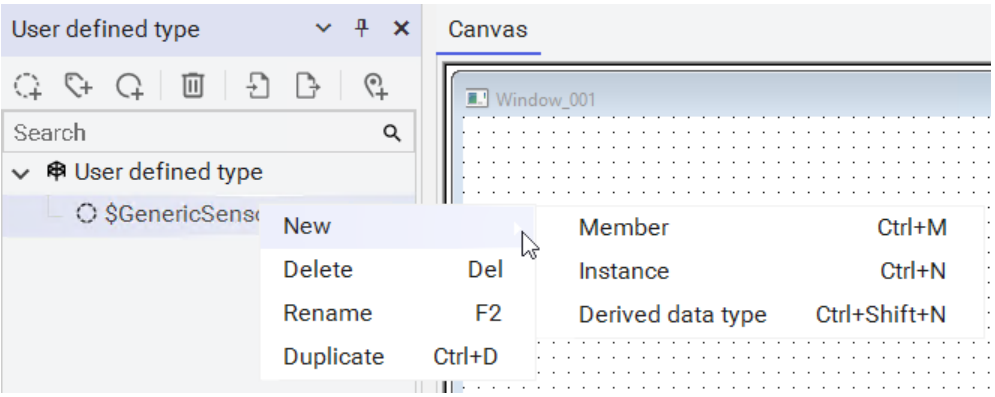
新しいデータ型の名前を GenericSensorType に変更します。



新しいメンバーの作成

新しいメンバーの作成

1. ［**ユーザー定義型**］ペインでデータ型を右選択し、コンテキスト メニューから［**新規作成**］して［**メンバー**］を選択するか、［**新規メンバー**］アイコンを選択します。



または

ツールバーで [New member] を選択します。

または

ショートカットキー **Ctrl+M** を使用します。

新しいメンバー **Member001** が作成されます。メンバーの名前には、英数字に加えて $、#、および _（アンダースコア）文字を使用できます。メンバー名の最大長は 32 文字です。

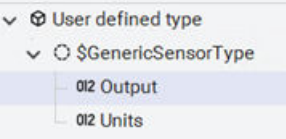
2. メンバーの名前を変更するには、メンバーを右クリックして［**名前の変更**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例：

次のような基本 InTouch タグ タイプの 2 つのメンバーを追加できます。

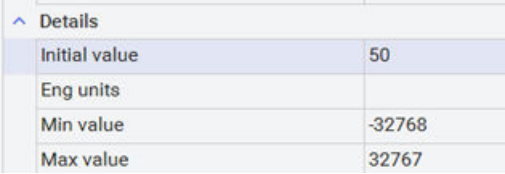
- 出力 - 型: 整数
- 単位 - 型: メッセージ（文字列）



右側のペインの［**プロパティ**］グリッドでタグのプロパティを設定することもできます。

例：

出力メンバーの「初期値」を 50 に設定します。



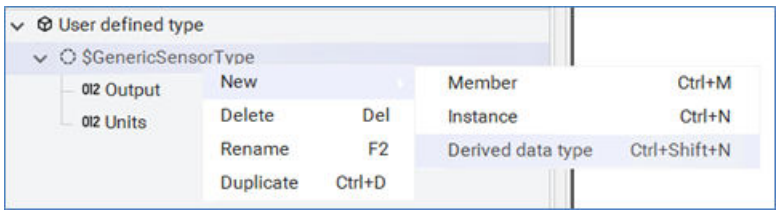
プロパティの詳細については、「プロパティ グリッドでのユーザー定義データ型の編集」を参照してください。

新しい派生データ型の作成

データ型を派生させて、そのすべてのメンバーを継承できます。

新しい派生データ型の作成

1. ［**ユーザー定義型**］ペインで目的のデータ型を右クリックして、コンテキスト メニューから［**新規**］を選択し、［**派生データ型**］を選択します。



   または

   ショートカット キー **Ctrl+Shift+N** を使用します。

   新しい派生データ型が作成されます。

   派生データ型のデフォルト名は <親データ型の名前>_00*n* です。*n* の初期値は 1 です。 たとえば、$Pump から派生データ型を作成した場合、最初の派生データ型のデフォルト名は "$Pump_001" です。派生データ型の名前には $ 記号の接頭辞が付けられます。この $ 記号を削除することはできません。派生データ型の名前には、英数字に加えて $、#、および _（アンダースコア）文字を使用できます。名前の最大長は 32 文字です。

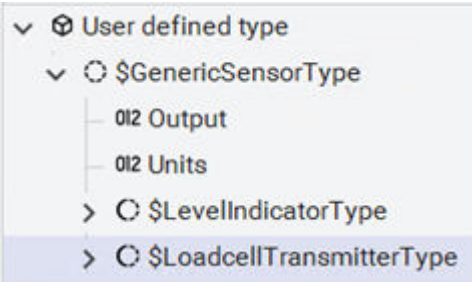
   ベース データ型のメンバーが派生データ型に追加されます。［名前］および［タイプ］のプロパティを除く親データ型のメンバーのプロパティを変更できます。新しいメンバーを派生データ型に追

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

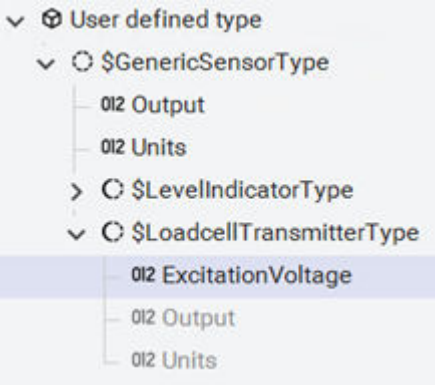
加できます。継承メンバーはグレーアウトされます。派生データ型のプロパティは、［**プロパティ**］ペインで変更できます。

例：

GenericSensorType から 2 つの派生データ型を作成し、LevelIndicatorType と LoadcellTransmitterType という名前を付けます。



次に、新しいメンバーを LoadcellTransmitterType データ型に追加し、整数型の ExcitationVoltage という名前を付けます。

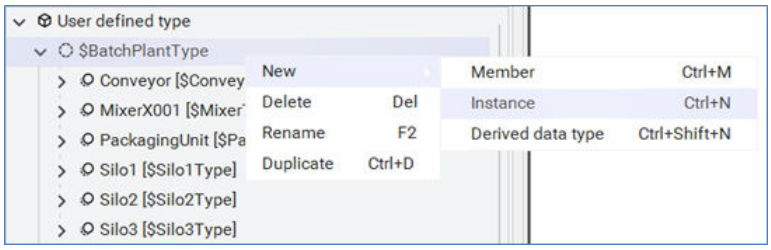


派生データ型は、親のすべてのデータ メンバーを継承し、グレーアウトとして表示されます。

新しいインスタンスの作成

インスタンスの作成

1. ［**ユーザー定義型**］ペインで目的のデータ型を右クリックして、コンテキスト メニューから［**新規**］を選択し、［**インスタンス**］を選択します。ツールバーの［**新規インスタンス**］アイコンをクリックすることもできます。



または

ツールバーで [New instance] を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

または

ショートカット キー **Ctrl+N** を使用します。

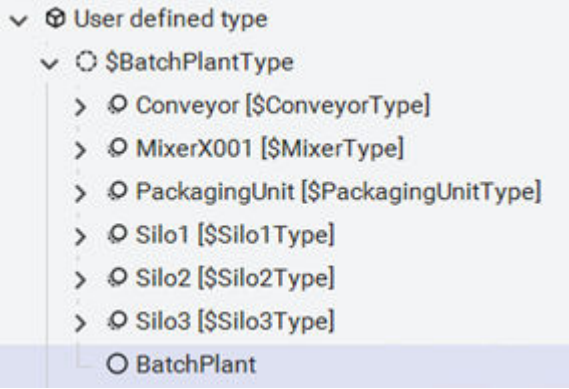
新しいインスタンスが作成されます。

2. UDT の名前を変更します。インスタンスのデフォルト名は <データ型名>_00*n* です。*n* の初期値は 1 です。 インスタンス名には $ 文字がプレフィックスとして付きません。インスタンスの名前には、英数字に加えて $、#、および _（アンダースコア）文字を使用できます。名前の最大長は 32 文字です。

ベース データ型のメンバーがインスタンスに追加されます。新しいメンバーをインスタンスに追加することはできません。継承メンバーはグレーで表示されます。インスタンスのプロパティは、［**プロパティ**］ペインで変更できます。［名前］および［タイプ］のプロパティを除くメンバーのプロパティを変更できます。1 つのデータ型または派生データ型の配下のすべてのメンバーは、同じデータ型または派生データ型の配下のインスタンスに属します。どのメンバーがどのインスタンスに属しているかは［**モデル – タグ名**］ペインで確認できます。すべてのインスタンスは、そのメンバと共に［**モデル – タグ名**］ペインの［**UDT**］に表示されます。

例：

$BatchPlantType からインスタンスを作成して BatchPlant という名前を付けます。



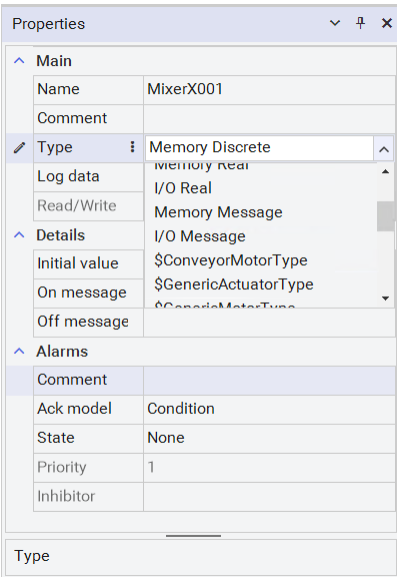
別のデータ型の下でのメンバーの入れ子

UDT のネストは、別のデータ型のメンバーを作成することによって行います。［**プロパティ**］ペインで 1 つのデータ型のメンバーを別のデータ型にポイントできます。入れ子を使用すると、タグを作成する時間と労力を削減できます。最大 6 つのレベルの UDT の入れ子がサポートされています。

別のデータ型の下でのメンバーの入れ子

1. 入れ子にするメンバーを選択します。
2. ［**プロパティ**］ペインの［**タイプ**］フィールドで、入れ子を作成するデータ型または派生データ型を選択します。［**型**］フィールドにデータ型の名前を入力してデータ型をフィルタすることもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 選択したデータ型または派生データ型の下でメンバーが入れ子になります。この名前は変更できます。たとえば、入れ子にされたメンバーの名前を "InletValve" に変更します。

注：この名前は、データ型のもう 1 つのインスタンスではありません。これは、データ型インスタンスのデータ型メンバーを参照するもう 1 つの方法です。

例：

"InletValve" は実際のインスタンスではありません。これはタグ名ディクショナリのルートには表示されず、クロス リファレンス ユーティリティのタグ階層のルートに表示されません。入れ子にされたメンバーをインスタンスにマップして、スクリプトとアニメーションで参照することができます。

UDT の最大の入れ子レベルは 6 です。以下の例を考えてみます。

Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6.Member1

この例の詳細を以下に示します。

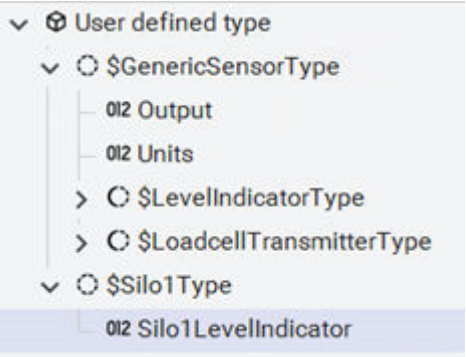
- "Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6" は UDT の 6 つのレベルです。
- "Member1" は、メモリ整数型などの基本 InTouch タグ タイプです。

注：循環参照はサポートされません。

例：

新しいデータ型を作成して Silo1Type という名前を付け、Silo1LevelIndicator というメンバーを追加します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



［**プロパティ**］グリッドから、プロパティ［**タイプ**］を別のデータ型に変更できます。LevelIndicatorType を選択したとします。

ネストされたメンバーが追加されます。

データ型をインポート

以前に作成した UDT のセットを含む .json ファイルをアプリケーションにインポートすることによって UDT のセットを効率的に構築できます。

UDT では JSON インポート中に次のモードがサポートされています。

- **ASK**：競合している場合に確認用のポップアップを表示します。この動作は（CSV ファイルからの）DBLoad/DBDump 操作で使用される ASK モードと似ています。これはデフォルト モードです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **SKIP**：競合するエントリを、変更せずにスキップします。DBLoad/DBDump 操作の IGNORE モードのように機能します。
- **OVERRIDE**： 競合している場合に既存のエントリを置換します。DBLoad/DBDump 操作の REPLACE モードのように動作します。
- **ABORT**：DBLoad/DBDump 操作の ABORT 操作と同様に、UDT インポート操作を停止します。

デフォルトでは、UDT のエクスポート時に JSON ファイルで ASK モードが構成されます。

注: 各アラーム グループに対してポップアップのプロンプトが表示されないようにするには、OVERRIDE モードを使用します。

JSON ファイルのモードを変更します。

1. インポートする目的の JSON ファイルを開きます（udtinfo-20241108.json や udtinfo-20241105-DWTP Tags Import.json など）。
2. 「モード」フィールドを以下のように更新します。

   JSON コンテンツ:

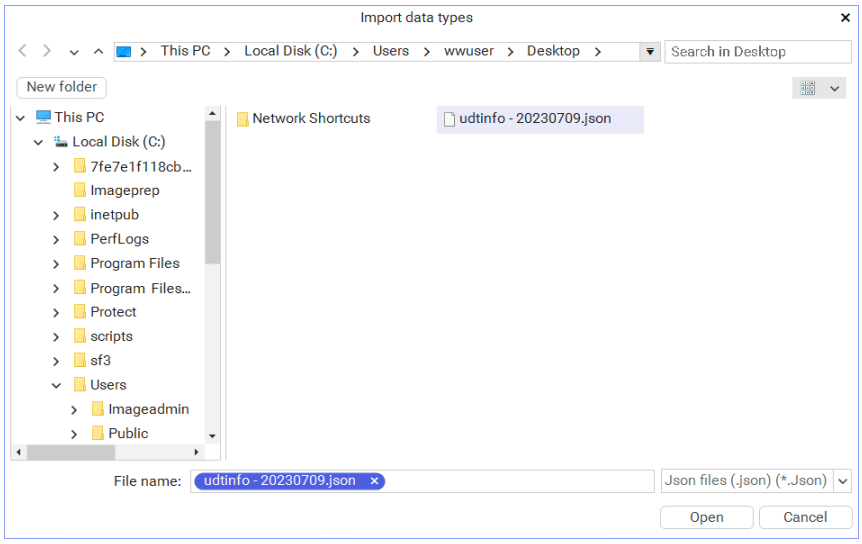
   {"バージョン":13,"モード":"OVERRIDE"
3. 更新した JSON ファイルを使用して、UDT のインポート操作を実行します。

データ型をインポート

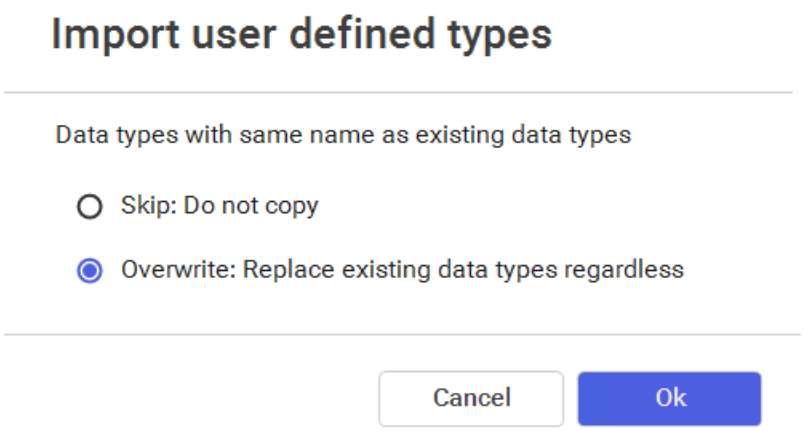
1. ［**ユーザー定義型**］ペインを開きます。
2. ［**ユーザー定義型**］を右選択してコンテキスト メニューから［**データ型をインポート**］を選択し、ツールバーの［**インポート**］アイコンを選択します。

3. データ型設定の詳細を含む .json ファイルにナビゲートし、目的のファイルを選択して［**開く**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**ユーザー定義型をインポート**］ダイアログ ボックスで以下の操作を行います。
   - 既存のデータ型と同じ名前のインポートされたデータ型をスキップするには、「**スキップ: コピーしない**」を選択します。
   - データ型名が既存のデータ型と同じである場合に、既存のデータ型を上書きして置き換えるには、「**上書き: 既存のデータ型に関係なく置換**」を選択します。



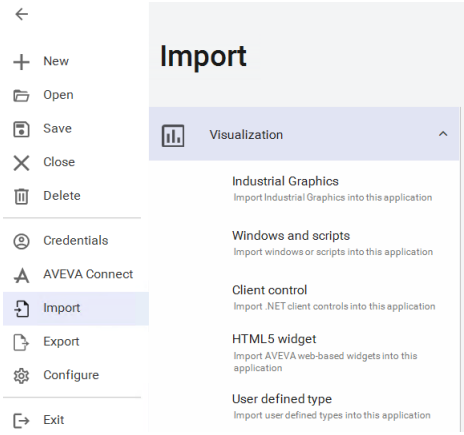
5. **[OK]** を選択します。

   前の手順で選択したオプションに従って、.json ファイルで使用可能なすべてのデータ型とその他のアイテムが表示されます。

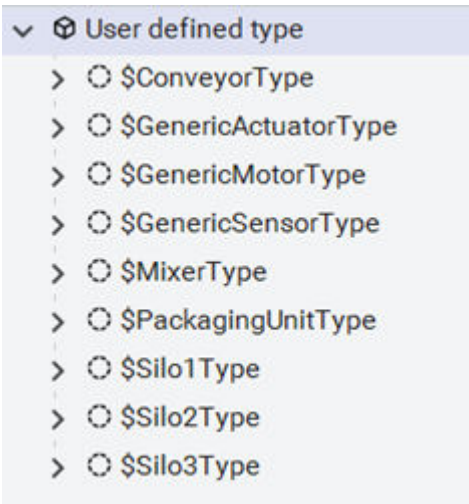
   注：上書きされたインスタンス値は、ファイルをインポートするときに保持されません。

［**ファイル**］メニューの［**インポート**］>［**表示**］>［**ユーザー定義型**］オプションを使用してデータ型をインポートすることもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



ファイルをインポートすると、UDT 階層が表示されます。

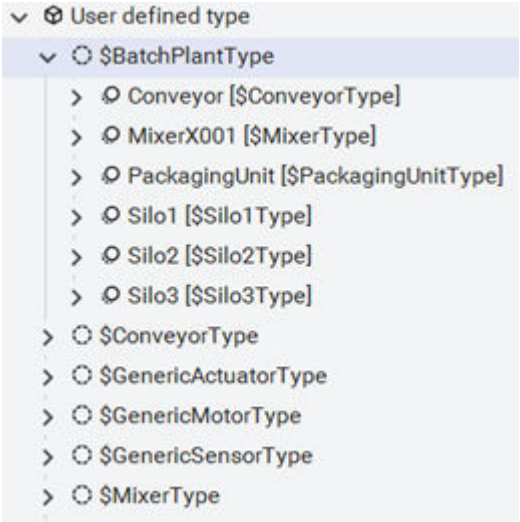


階層には他の UDT を追加できます。その場合、「BatchPlantType」という新しいデータ型を作成してメンバーを追加し、以下に示すように、各メンバーのプロパティ グリッドで型を変更します。

- Conveyor - 型 $ConveyorType
- MixerX001 - 型 $MixerType
- PackagingUnit - 型 $PackagingUnitType
- Silo1 - 型 $Silo1Type
- Silo2 - 型 $Silo2Type
- Silo3 - 型 $Silo3Type

結果の階層に新しいメンバーが反映されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



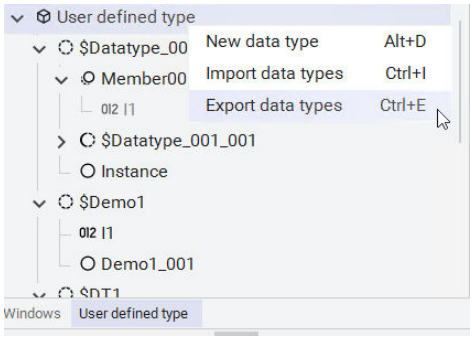
データ型のエクスポート

その他のアプリケーション、システムのその他のコンピュータ、または別のシステムのその他のコンピュータで使用するためにデータ型をエクスポートできます。

データ型のエクスポート

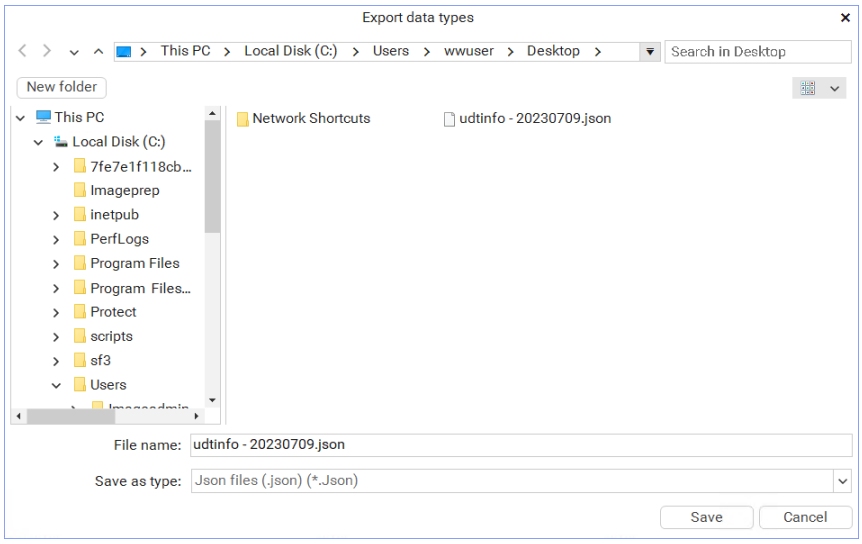
1. ［**ユーザー定義型**］ペインを開きます。

   ［**ユーザー定義型**］を右選択してコンテキスト メニューから［**データ型をエクスポート**］を選択するか、ツールバーの［**エクスポート**］アイコンを選択します。



   ［**データ型のエクスポート**］ウィンドウが開きます。

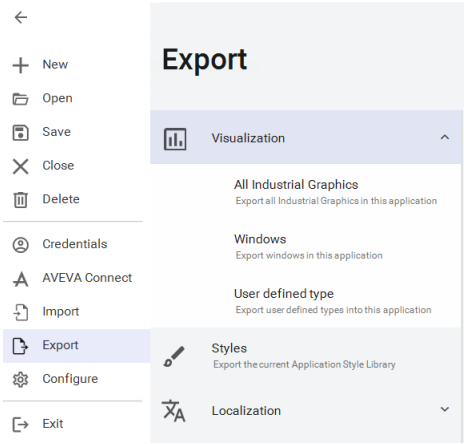
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. 必要に応じてファイル名を変更し、［**保存**］を選択します。

   選択した場所に .json ファイルが保存されます。エクスポートしたファイルは、外部の JSON エディタを使用して編集し、元のアプリケーションまたは別のアプリケーションにインポートできます。

［**ファイル**］メニューの［**エクスポート**］>［**表示**］>［**ユーザー定義型**］オプションを使用してデータをエクスポートすることもできます。



注：エクスポート中、すべての UDT が JSON ファイルに含まれます。

ユーザー定義データ型の管理

前のトピックで説明されている UDT を作成および更新する操作に加えて、基本的な管理操作を実行できます。

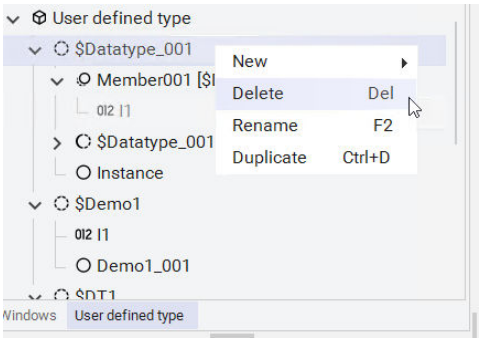
メンバーを削除する

- 削除するメンバーを右選択して、コンテキストメニューから[**削除**]を選択します。 ツールバーにある [**削除**] アイコンを選択して、メンバーを削除することもできます。複数のメンバーを選択するには **Ctrl** キーを使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データ型、派生データ型、またはインスタンスを削除する

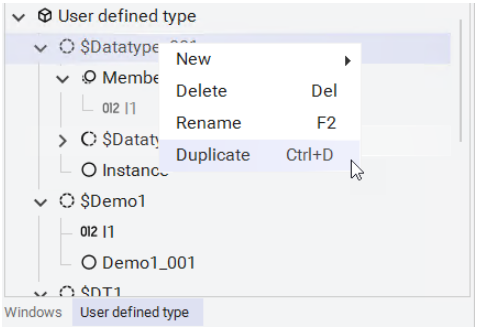
データ型、派生データ型、またはインスタンスを右選択してコンテキスト メニューから ［**削除**］を選択するか、ツールバーの［**削除**］アイコンを選択します。データ型およびそのすべてのメンバーが削除されます。



注：データ型を削除するには、配下のインスタンスと派生データ型を削除する必要があります。

データ型を複製する

- ［**ユーザー定義型**］ペインでベース データ型を右クリックして、コンテキスト メニューから［**複製**］を選択します。すべてのメンバーを含むベース データ型のコピーが作成されます。



注：複製できるのはデータ型と派生データ型だけです。インスタンスとメンバーを複製することはできません。

場所属性を追加する

1. ［**ユーザー定義型**］ペインを開きます。
2. データ型または派生データ型を強調表示し、ツールバーの **[場所の追加]** アイコンを選択します。

   緯度と経度の新しい属性がデータ型または派生データ型に追加されます。場所属性の使用の詳細については、このセクションの「マップアプリケーション」トピックを参照してください。

ユーザー定義データ型の一括編集

JSON ファイルを使用して複数の UDT を一括編集できます。エクスポートされた JSON ファイルを編集するには、一般的な JSON エディタを使用して UDT を追加、削除、または編集します。

UDT を一括で編集または作成する

1. 既存の複数の UDT を JSON ファイルにエクスポートします。UDT をエクスポートする方法の詳細については、「データ型のエクスポート」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. JSON ファイルをテキスト エディタまたは JSON エディタで開きます。
3. 既存の UDT をコピーしてファイルに貼り付けて UDT を編集または作成します。
4. 必要に応じて、その他のプロパティの名前の変更または編集を行います。
5. JSON ファイルを保存します。
6. JSON ファイルを WindowMakerWindowMaker の［**ユーザー定義型**］ペインにインポートします。インポートの詳細については、「データ型をインポート」を参照してください。

例：

```
{
  Version : 12
  Templates : [ 1 item
    0 : {
      Name : DT1
      Comments : value
      Members : [ 1 item
        0 : {
          Name : IODisc
          TagType : AtomicTag
          Source : Reference
          TagComment : value
          AlarmGroup : 1
          LogData : False
          LogEvents : False
          LocalTag : false
          TypeOf : Discrete
          IoConfig : { 2 props }
          AlarmConfiguration : { 3 props }
        }
      ]
      Overrides : [ 0 items
      ]
    }
  ]
  Derived Templates : [ 0 items
  ]
  Instances : [ 3 items
    0 : {
      Name : DT1_001
      Comments : value
      TypeOf : DT1
      Overrides : [ 3 items
        0 : {
          Name : IOInt
          ContextName : value
          DataType : Int
          Val : {
            DataType : Int
            DefaultValue : value
          }
```

ユーザー定義データ型の表示

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザー定義データ型は次の 2 つのいずれかを使用して表示できます。

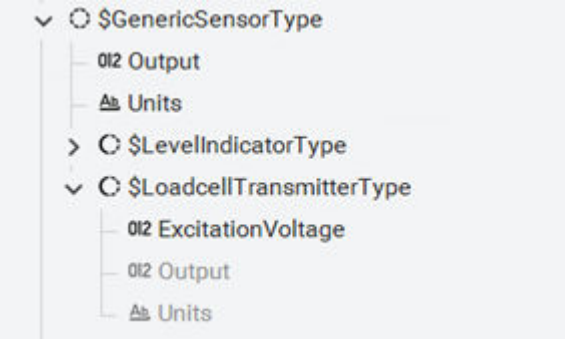
- [ユーザー定義型] ビューを通じて
- [モデル - タグ名] ビュー

[ユーザー定義型] ビュー

[ユーザー定義型] ペインには、ユーザー定義型、メンバーとメンバーのデータ型、派生データ型、およびインスタンスのリストが表示されます。

- データ型には "$" 文字の接頭辞が付きます。
- メンバー データ型には、「[$<データ型名>]」が追加されます。
- すべてのメンバーと派生データ型は、完全に展開してメンバーを表示できます。
- このビューでは、インスタンス名を展開することはできません。
- 各データ型に対して、すべての子アイテムは以下の示す順番で表示されます。グループ内では、アイテムはアルファベット順に並べ替えられます。
  a. 直接メンバー
  b. 継承メンバー
  c. 派生データ型
  d. インスタンス

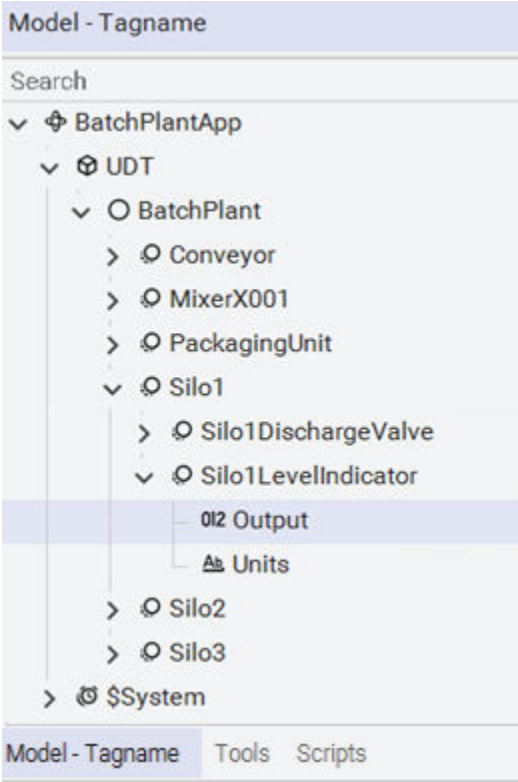
例：



[モデル - タグ名] ビュー

[モデル – タグ名] ビューは、UDT インスタンス ビューが追加された InTouch タグ ビューです。UDT インスタンスとそのメンバーを表示できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



プロパティ グリッドでのユーザー定義データ型の編集

注 : [プロパティ] グリッドで、次の値を編集する必要があります。タグ名辞書の UDT は表示または編集できません。

プロパティを更新する

UDT のプロパティは［**プロパティ**］グリッドで表示および変更できます。［**ユーザー定義型**］ペインまたは［**モデル - タグ名**］ペインで選択したアイテムのプロパティは、WindowMakerWindowMaker の右側の［**プロパティ**］ペインに表示されます。

- 各プロパティを強調表示してプロパティを選択および更新します。対応するプロパティの説明が［**プロパティ**］ペインの下部に表示されます。
- 行った変更は自動的に保存されます。
- プロパティのフィールドは、［**タイプ**］プロパティ フィールドで選択したオプションに基づいて変化します。
- ［**タイプ**］プロパティ フィールドには、UDT がその他のデータ型と共に表示されます。
- プロパティのリストが更新され、選択した［**タイプ**］に関連付けられたプロパティが反映されます。
- ［**名前**］プロパティと［**タイプ**］プロパティを除き、メンバーが定義されているすべてのタグ プロパティ値を編集できます。

メンバー データ型

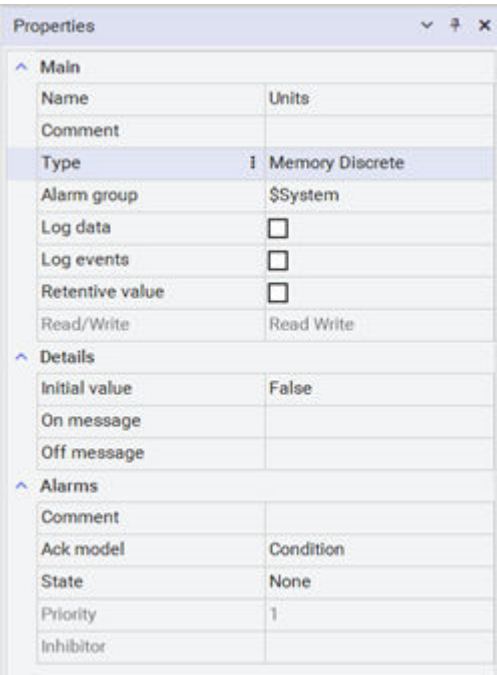
メンバーを別のデータ型に変更する場合、以下のプロパティだけが表示されます。この例では、$BatchPlantType > Silo1 メンバーを選択できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メンバーの編集

メンバーの型を基本 InTouch タグに変更すると、関連するすべてのタグ プロパティが編集可能になります。

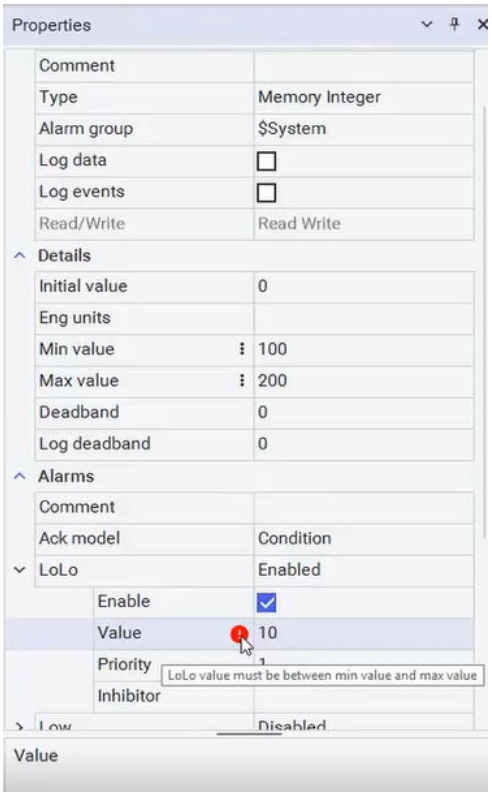
たとえば、$GenericSensorType > Units という名前の UDT を選択し、型をメモリ論理型に変更します。プロパティ グリッドが更新され、この理論型タグのすべての関連タグ プロパティが表示されます。



無効な値と必須フィールド

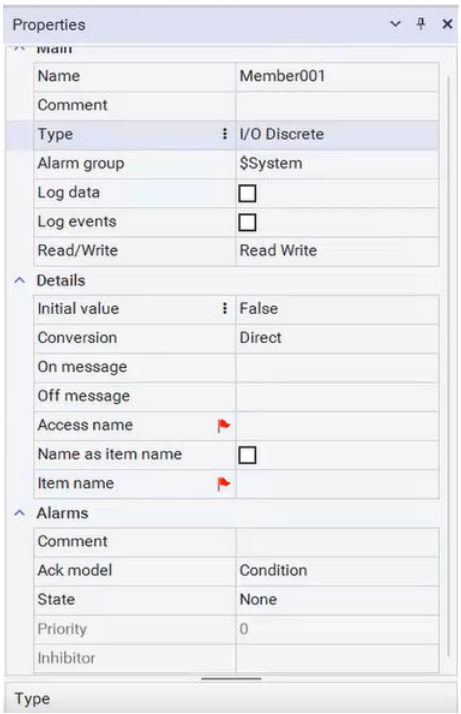
プロパティの無効な値は赤いアイコンで示されます。プロパティ値を更新し、その他のいずれかのユーザー定義型を選択すると、プロパティ名の前に赤いエラー アイコンが表示されます。エラー アイコンの上にカーソルを置くと、エラーの理由を示すツールヒントが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



値が必要なプロパティは赤いフラグで示されます。データ型レベルで値を入力する必要はありません。しかし、インスタンス レベルで値を入力または上書きすることが推奨されます。

例：



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値の上書きと変更の反映

［名前］プロパティと［タイプ］プロパティを除き、メンバーが定義されているデータ型レベルのすべてのタグ プロパティ値を編集できます。

例 1:

1. $GenericSensorType > Output を選択します。
2. 初期値を 50 に変更します。
3. 次に、$LevelIndicatorType > Output を選択します。
4. この初期値も 50 であることを確認します。
5. $GenericSensorType > Output を選択します。
6. 初期値を 60 に変更します。

   $LevelIndicatorType > Output に戻ります。この初期値も 60 に更新されていることを確認します。これは変更の反映の一例です。
7. 初期値を上書きすることもできます。初期値を 75 に設定します。

   初期値が太字になり、上書きされたことが示されます。

8. $GenericSensorType > Output を選択して初期値を 65 に変更します。

   このインスタンスでは値が上書きされているので、値は $LevelIndicatorType に反映されません。

例 2:

この例は、前の例 1 の延長です。

1. ［モデル - タグ名］ビューで、BatchPlant > Silo1 > Silo1LevelIndicator > Output を選択します。初期値は 75 です。

   これは $LevelIndicatorType データ型から継承されたものです。
2. 25 の上書き値を入力します。
3. BatchPlant > Silo2 > Silo2LevelIndicator > Output を選択して、初期値を 55 で上書きします。
4. BatchPlant > Silo3 > Silo3LevelIndicator > Output を選択します。初期値を 80 で上書きします。

変更の反映

データ型の変更は［ユーザー定義型］ビューに即時反映されます。

新規メンバーを追加

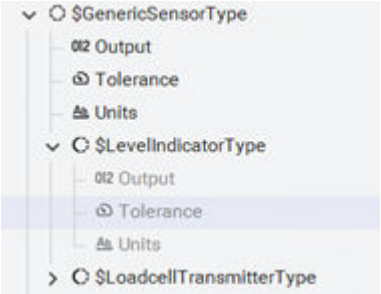
新規メンバーの追加は、派生データ型、メンバー データ型、およびインスタンスに反映されます。

例 :

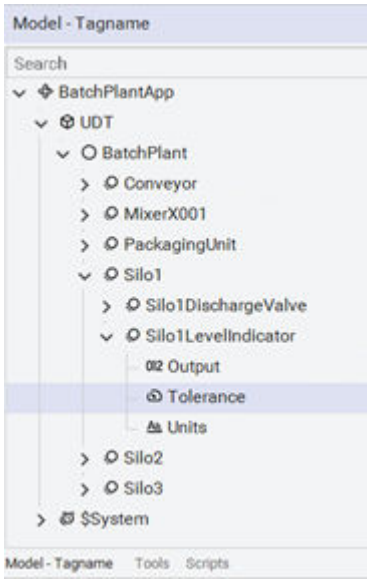
1. $GenericSensorType を選択します。
2. メモリ実数型の "Tolerance" という新しいメンバーを追加します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

この新しいメンバーは、その派生データ型 $LevelIndicatorType で使用可能になります。



この新しいメンバーは、インスタンス メンバー データ型でも使用可能です。



メンバー型の変更

既存のメンバーの変更は、派生データ型、メンバー データ型、およびインスタンスにも直ちに反映されます。

例：

前の例の延長:

1. $GenericSensorType > Tolerance に移動します。
2. プロパティ グリッドで、データ型をメモリ実数型からメモリ メッセージ型に変更します。

   $LevelIndicatorType > Tolerance もメモリ メッセージのデータ型に更新されます。

既存の機能の UDT サポート

注：例に示す UDT 名、メンバー名、およびプロパティは、説明目的でのみ使用されています。型とメンバーには実際のアプリケーションに適切な名前を付けてください。

UDT は以下の InTouch 機能でサポートされています。

- Intellisense

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

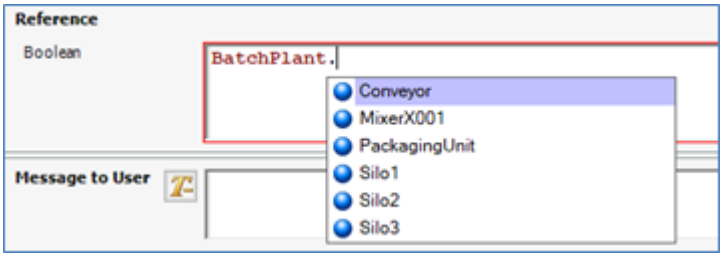
- アニメーション
- スクリプト
- 参照の変更
- UDT メンバーの参照
- インスタンスの UDT メンバーの産業用グラフィック エディタ キャンバスへのドラッグ アンド ドロップ
- アニメーションとスクリプト
- アラームおよびアラーム クライアント コントロール
- 履歴およびトレンド クライアント
- I/O タグとしての UDT メンバーの設定
- 所有オブジェクトとしての UDT インスタンスの使用
- Web Client
- マップアプリケーション
- 相互参照
- 未使用タグの削除
- スーパータグとの共存
- ライセンス
- ランタイム Tag Viewer

Intellisense

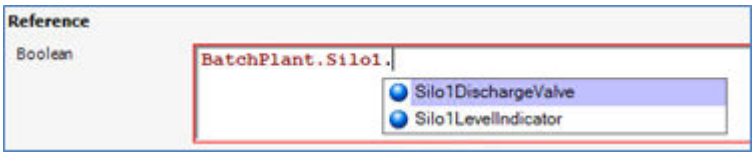
UDT インスタンスの Intellisense には、そのレベルだけのすべてのメンバーとメンバー データ構造が一覧表示されます。産業用グラフィック エディタは、スクリプト、アニメーション、およびカスタム プロパティで Intellisense をサポートします。ネイティブ InTouch はスクリプト作成でのみ Intellisense をサポートします。

- 産業用グラフィック: Intellisense は、スクリプト、アニメーション、およびカスタム プロパティでサポートされます。
- ネイティブ InTouch: Intellisense はネイティブ InTouch スクリプトでのみサポートされます。

例:



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



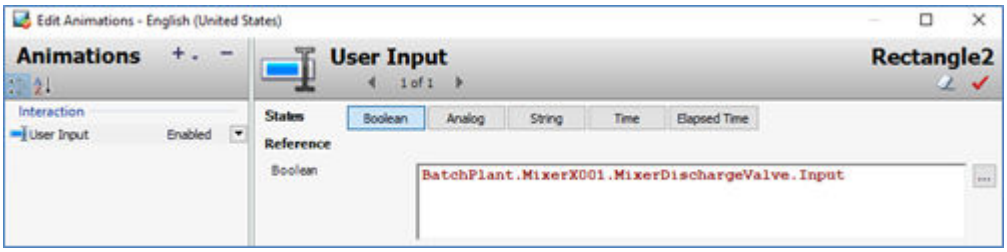
アニメーション

産業用グラフィック アニメーションとネイティブ InTouch アニメーションで UDT を設定できます。

注記: 値の表示アニメーションを含む UDT のタグの変更の動作はリモート タグと同じです。完全な参照を UDT タグ、またはドット フィールドを含む UDT タグで置き換える必要があります。

例:

産業用グラフィック アニメーションで UDT を設定します。



ネイティブ InTouch アニメーションで UDT を設定します。

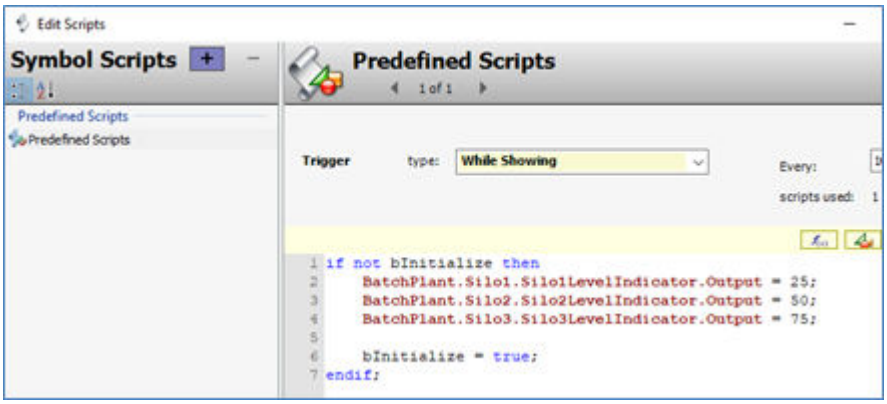
スクリプト

産業用グラフィック スクリプトとネイティブ InTouch スクリプトで UDT を設定できます。

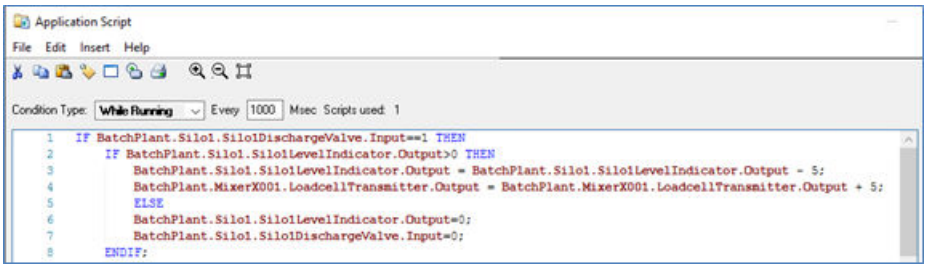
例:

産業用グラフィック スクリプトで UDT を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.



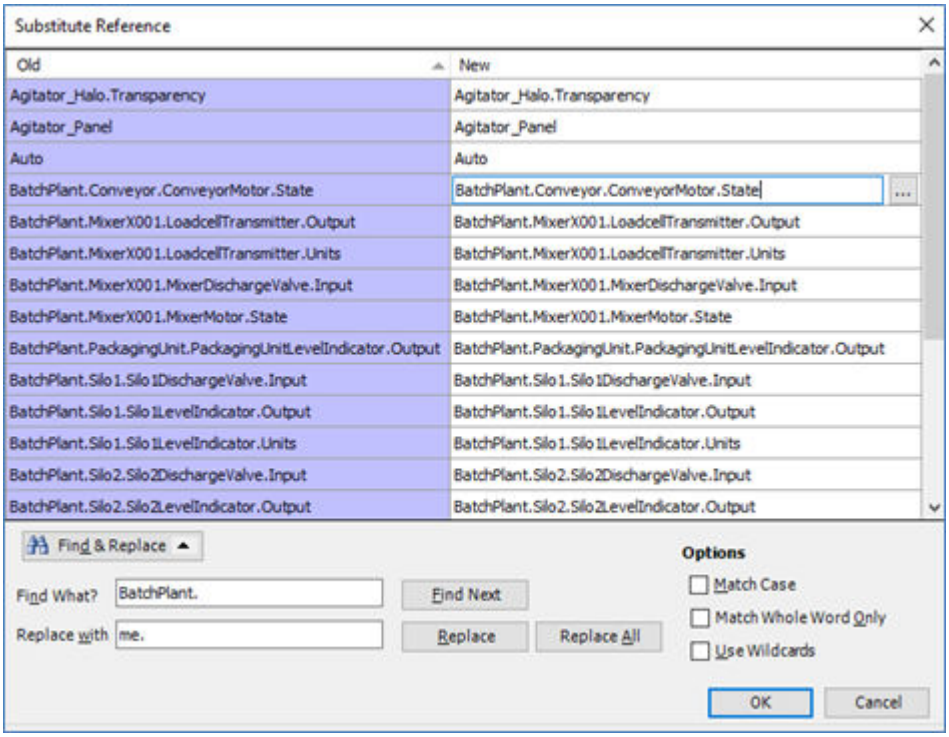
ネイティブ InTouch スクリプトで UDT を設定します。



参照の変更

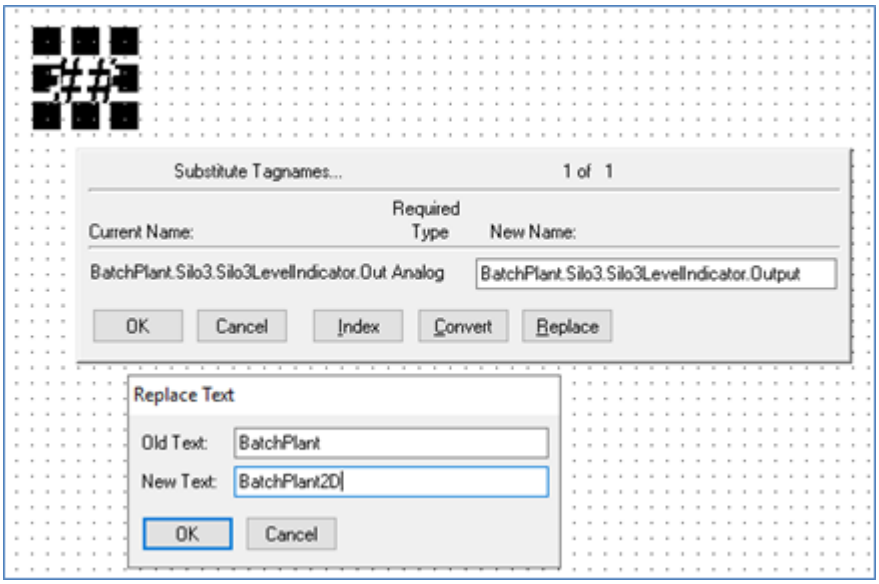
UDT インスタンス メンバーの参照は、産業用グラフィック エディタとネイティブ InTouch の両方で、他の InTouch タグと同じ方法で置換できます。

産業用グラフィックの参照の変更の例を以下に示します。



© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

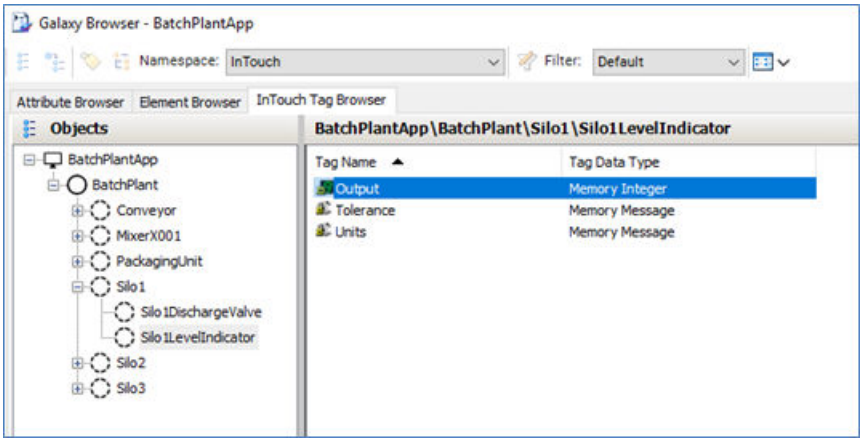
ネイティブ InTouch での参照の変更の例を以下に示します。



UDT メンバーの参照

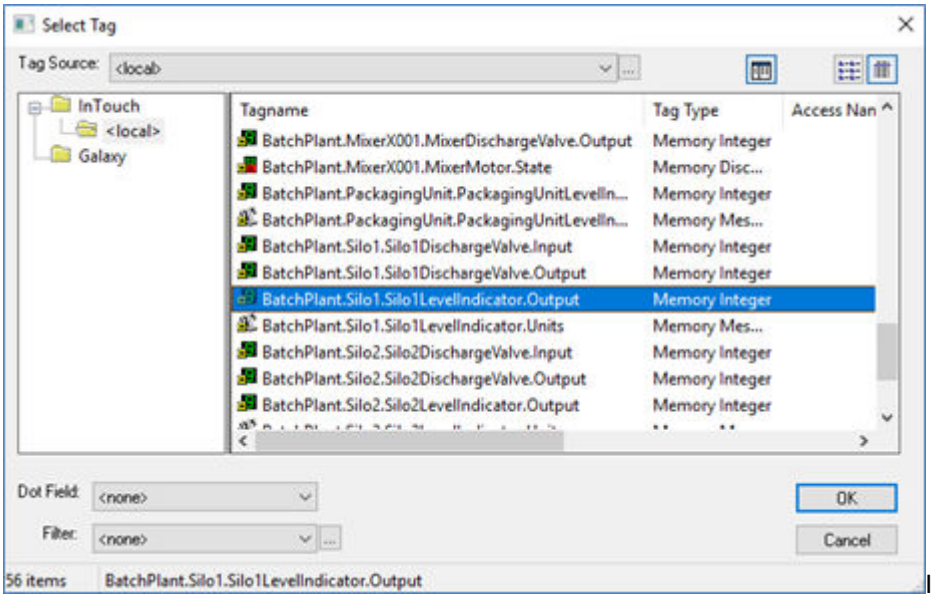
産業用グラフィック エディタでアニメーションを設定する場合、またはスクリプトを使用する場合、省略記号ボタンを使用して［**Galaxy ブラウザ**］を開き、UDT メンバーを参照します。ネイティブ InTouch のアニメーションまたはスクリプト オプションで、**タグ ブラウザ**を開いて UDT メンバーを参照します。

産業用グラフィック エディタでの参照の例を以下に示します。



ネイティブ InTouch での参照の例を以下に示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

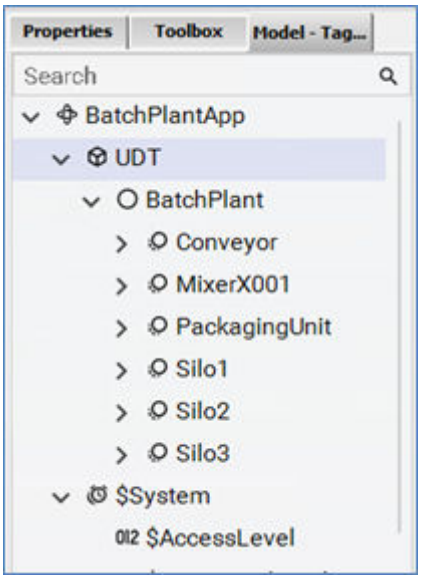


インスタンスの UDT メンバーの産業用グラフィック エディタ キャンバスへのドラッグ アンド ドロップ

産業用グラフィック エディタの［**モデル - タグ名**］ペインにも UDT インスタンスが表示されます。産業用グラフィック エディタで 1 つまたは複数の UDT メンバーをキャンバスにドラッグ アンド ドロップしてアニメーションを作成できます。

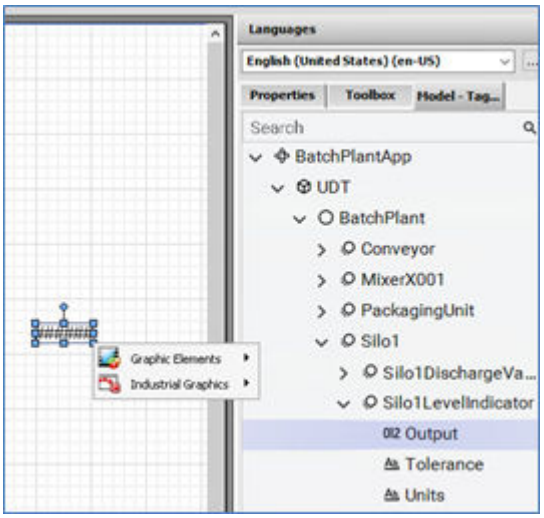
例:

［**モデル - タグ名**］でドラッグ アンド ドロップします。



産業用グラフィック エディタでドラッグ アンド ドロップます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



アニメーションとスクリプト

グラフィックに以下の設定が含まれている場合、UDT インスタンス メンバーは WindowViewerWindowViewer でサポートされます。

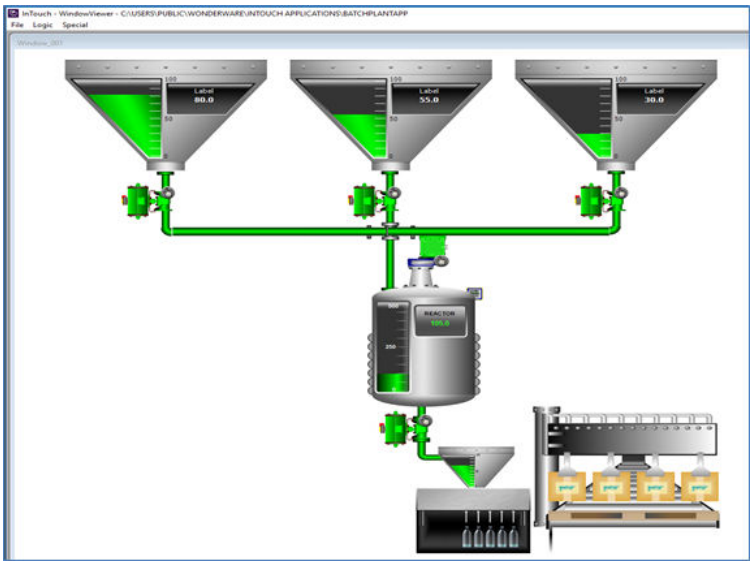
- 産業用グラフィックとネイティブ InTouch グラフィックのアニメーションおよびスクリプト

  または

- ネイティブ InTouch のスクリプト

例：

WindowViewerWindowViewer では次のように表示されます。



アラームおよびアラーム クライアント コントロール

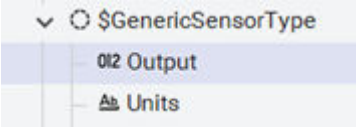
インスタンスの UDT メンバーは、その他の InTouch タグと同じ方法でアラームをサポートします。メンバーのアラーム プロパティは、［**プロパティ**］グリッドで更新できます。WindowViewerWindowViewer

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

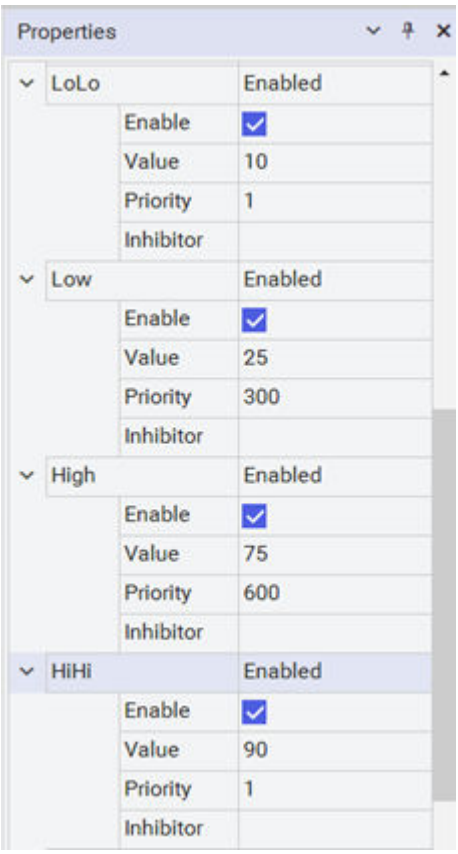
で、アラームの生成とアラーム クライアント コントロールでの表示を行っているインスタンスのメンバーを表示できます。

例 1：UDT メンバーのアラームの設定

1. ［ユーザー定義型］ビューで $GenericSensorType > Output メンバーを選択します。



2. ［プロパティ］グリッドで、次の図に示すように、このタグのアラーム プロパティを更新します。

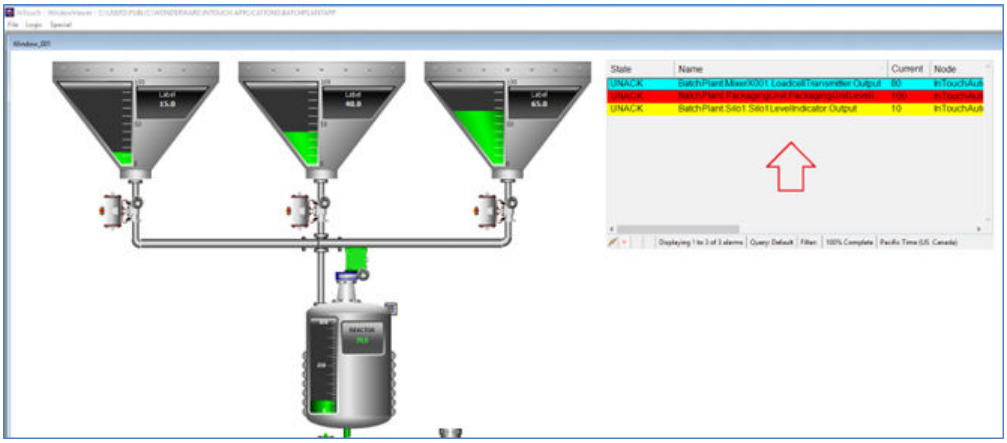


アラーム プロパティが UDT インスタント レベルまで反映されます。以下の UDT メンバーでは、前の例に示されているように、アラームを設定できます。

- BatchPlant.Silo1.Silo1LevelIndicator.Output
- BatchPlant.Silo2.Silo2LevelIndicator.Output
- BatchPlant.Silo3.Silo3LevelIndicator.Output

次の図は、WindowViewerWindowViewer における UDT のアラーム サポートの例を示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

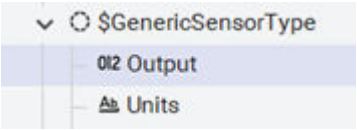


履歴およびトレンド クライアント

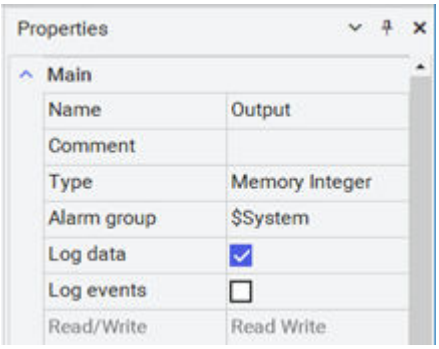
インスタンスの UDT メンバーは、その他の InTouch タグと同様に、データを LGH ファイルまたは Historian に記録できます。

ログ データの有効化の例

1. ［ユーザー定義型］ ビューで $GenericSensorType > Output メンバーを選択します。



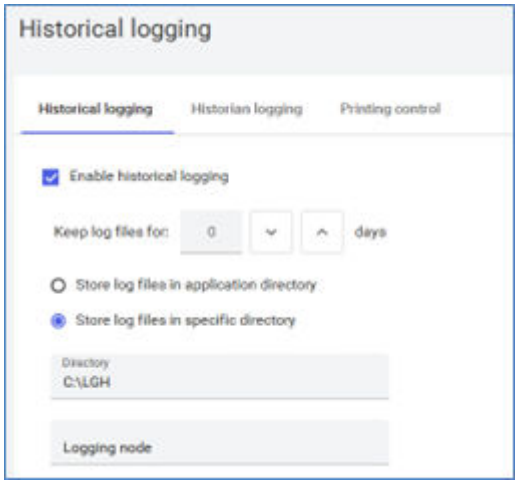
2. ［プロパティ］グリッドで、以下に示すように Log data プロパティが True（チェック済み）に更新されているかどうかを確認します。



LGH の履歴ログの有効化の例:

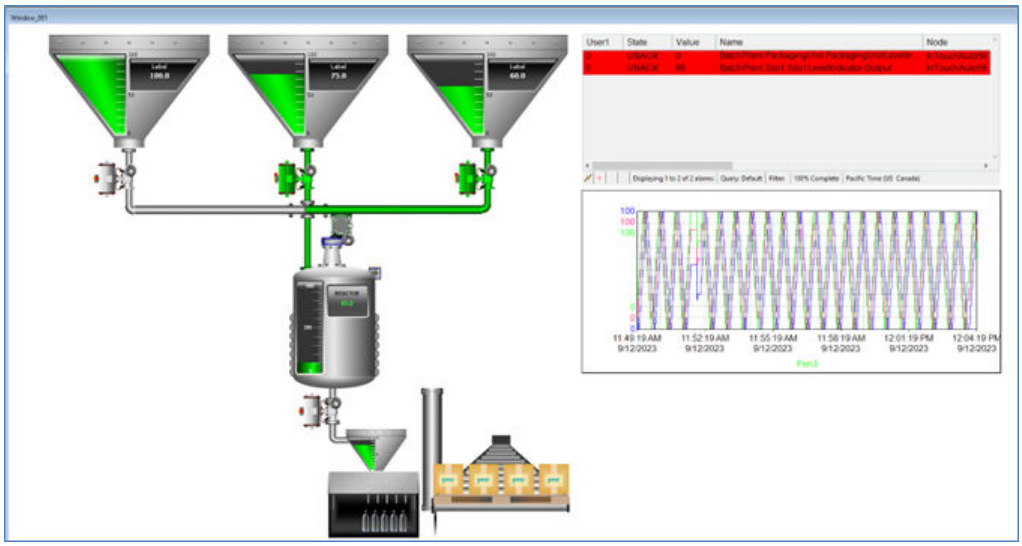
1. バックステージへ移動:［ファイル］>［設定］>［履歴ログ］

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.



2. 指定したディレクトリが存在しない場合、ファイル エクスプローラを使用して作成します。
3. WindowViewerWindowViewer を開き、ランタイムに移動し、LGH ファイルへのデータ履歴の記録を検証します。

   次の図は、LGH ファイルへの履歴データを記録する UDT メンバーを示し、トレンド クライアントはリアル タイムおよび履歴データをプロットしています。履歴データは、トレンドにバックフィルされます。



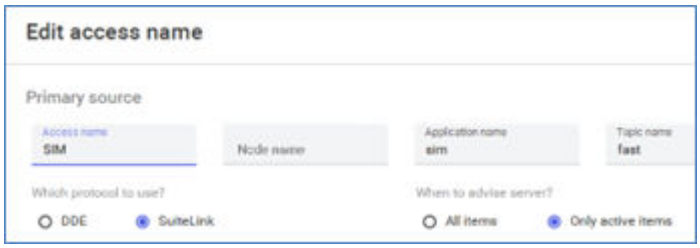
I/O タグとしての UDT メンバーの設定

その他の InTouch タグと同じ方法で UDT メンバーを I/O タグとして設定できます。[**プロパティ**] グリッドで、[**タイプ**]、[**アクセス名**]、および [**アイテム名**] フィールドを指定してメンバーを I/O タグとして設定します。

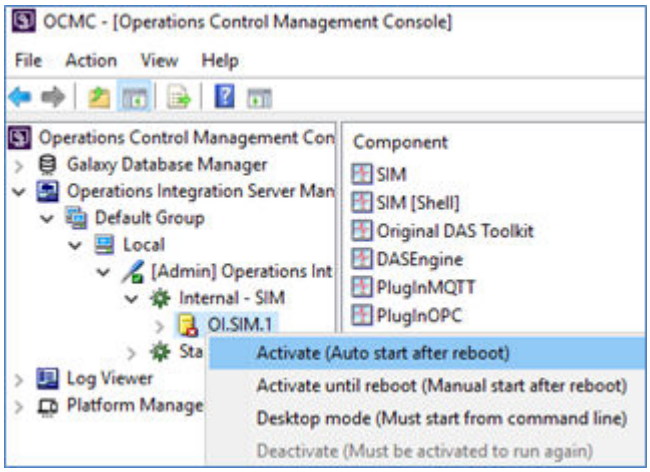
例: I/O タグとしての UDT メンバーの設定

1. ローカル Simulator サーバーのアクセス名 "SIM" を作成します。

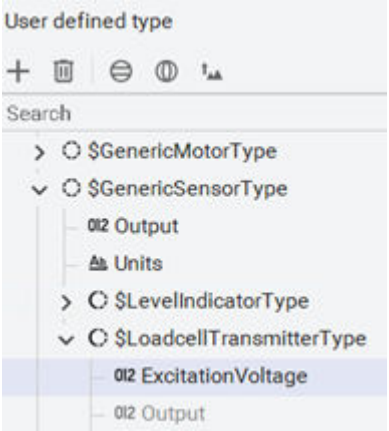
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. **Operations Control Management Console**（OCMC）から SIM サーバーをアクティブ化します。

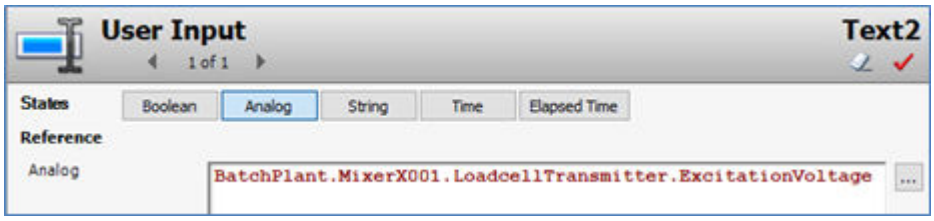


3. ExcitationVoltage 例メンバーを選択して設定します。

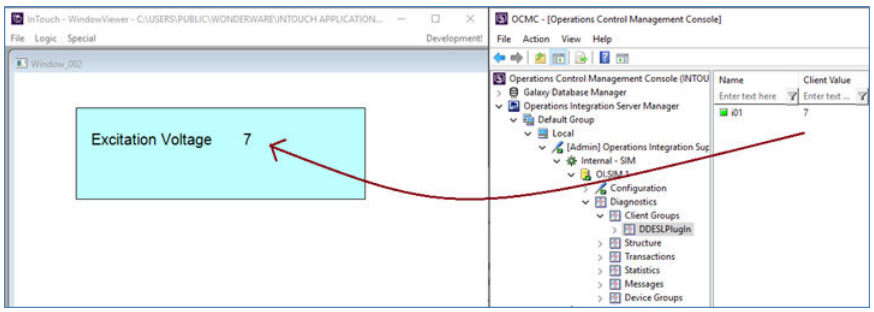


4. プロパティを設定ひてメンバーを I/O タグにします。インスタンス レベルでアクセス名およびアイテム名を上書きすることもできます。
   a. 型を I/O 整数型に設定します。
   b. SIM へのアクセス名を設定します（ドロップ ダウン リストから選択できます)。
   c. アイテム名を i01 に設定します。
5. グラフィックを作成し、以下の図に示す参照を含むアニメーションを追加します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. 作成したグラフィックを埋め込んでランタイムで表示します。値が I/O 参照にバインドされていることを確認します。



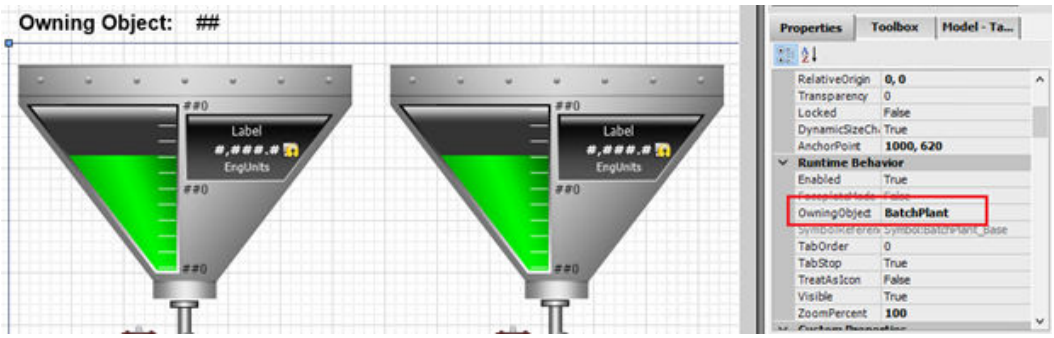
所有オブジェクトとしての UDT インスタンスの使用

埋め込まれたグラフィックの所有オブジェクトを設定して UDT インスタンスをポイントすることができます。ランタイム時、**me.** 相対参照は UDT インスタンスによって置き換えられます。ランタイム時に所有オブジェクトを別の UDT インスタンスに更新することもできます。

例: 所有オブジェクトとしての UDT インスタンスの使用

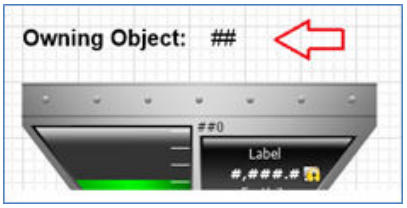
1. ［視覚化］フォルダで、ReactorDemo Symbols > MainDisplays に移動します。
2. BatchPlant_OwningObj グラフィックを編集します。
3. 埋め込まれたグラフィック BatchPlant_RelativeRef1 を選択します。

   プロパティ グリッドで、OwningObject 属性が UDT インスタンスである BatchPlant に設定されていることを確認します。



BatchPlantType から別の UDT を作成し、ランタイム時に入力アニメーションを使用して別の UDT insutansu に切り替えることができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



例: ランタイム時の所有オブジェクトから別の UDT インスタンスへの切り替え

1. BatchPlant_OwningObj グラフィックを別の枠ウィンドウに埋め込みます。
2. WindowViewerWindowViewer が起動されている場合は閉じます。
3. WindowViewerWindowViewer に切り替えます。
4. "me." 総体参照が BatchPlant にバインドされていることを確認します。

所有オブジェクトを別のインスタンスに変更できます。

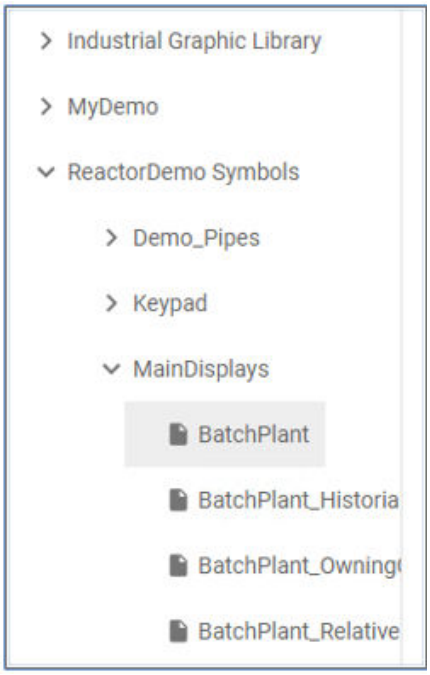
Web Client

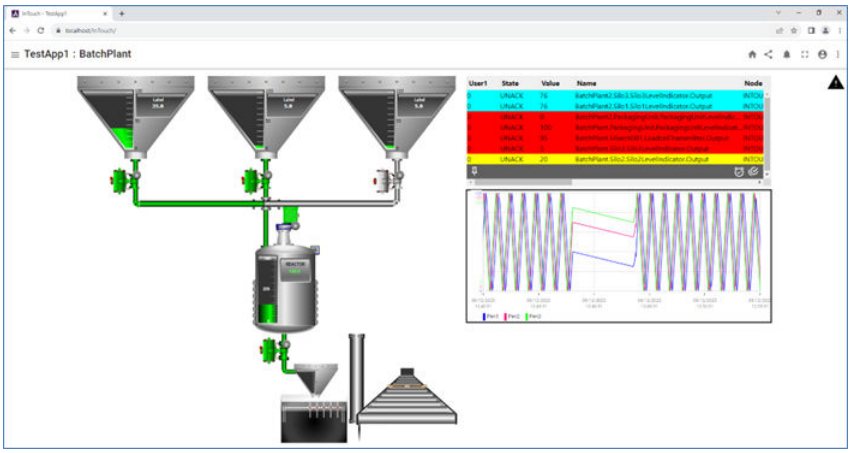
インスタンスの UDT メンバーは、Web Client でその他の InTouch タグと同じように機能します。

例：

WindowViewerWindowViewer でアプリケーションが実行している状態で、WindowMakerWindowMaker から Web Client を起動し、メニューから BatchPlant グラフィックを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



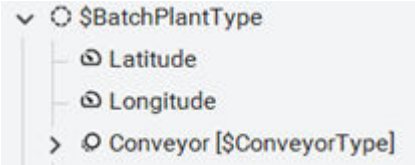


マップアプリケーション

インスタンスの UDT メンバーは、MapApp でその他の InTouch タグと同じように動作します。

場所メンバーの追加の例

1. この例の "BatchPlantType" を選択し、ツールバーの［**場所を追加**］を使用して新しいメンバー の経度および緯度を追加します。



2. プロパティ グリッドで以下の値を設定します。
   - 経度 = 33

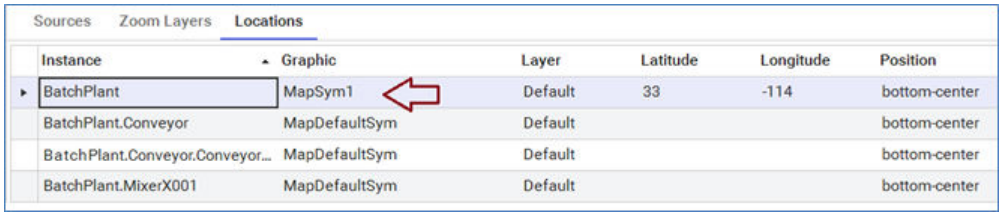
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 緯度 = -114

マップ設定の構成例

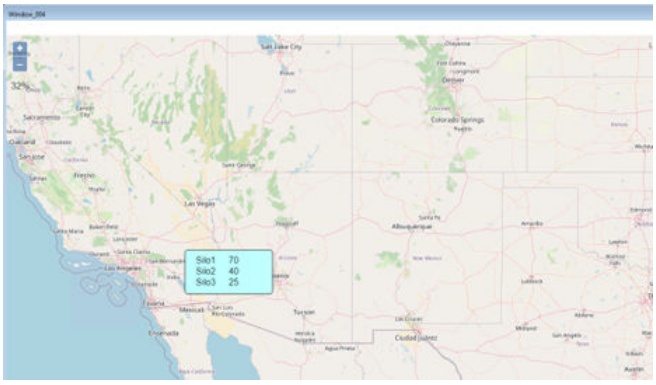
1. ［視覚化］フォルダで、ウィジェットに移動して［マップ アプリ］を選択します。
2. グラフィック "MapSym1" を "BatchPlant"に関連付けます。MapSym1 は、この UDT インスタンス BatchPlant に対して MapApp で表示するグラフィックです。

   MapApp 設定では、インスタンスおよびそのインスタンスのすべてのメンバー（メンバー データ型）がリストされます。



ランタイムでの **MapApp** の使用例

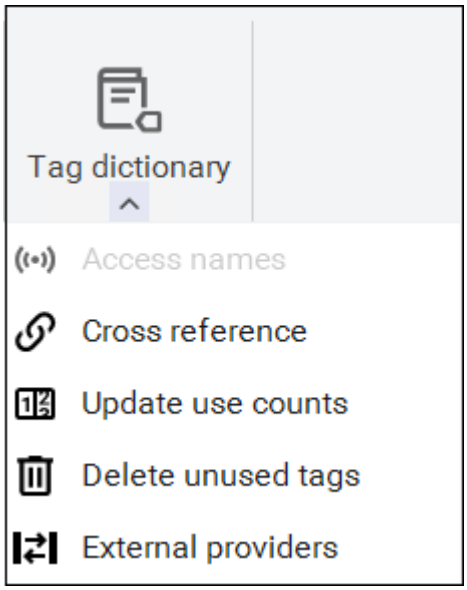
- グラフィック「MapAppMain」を別の枠ウィンドウに埋め込みます。WindowViewerWindowViewer を閉じ（既に実行している場合）、ランタイムに切り替えます。



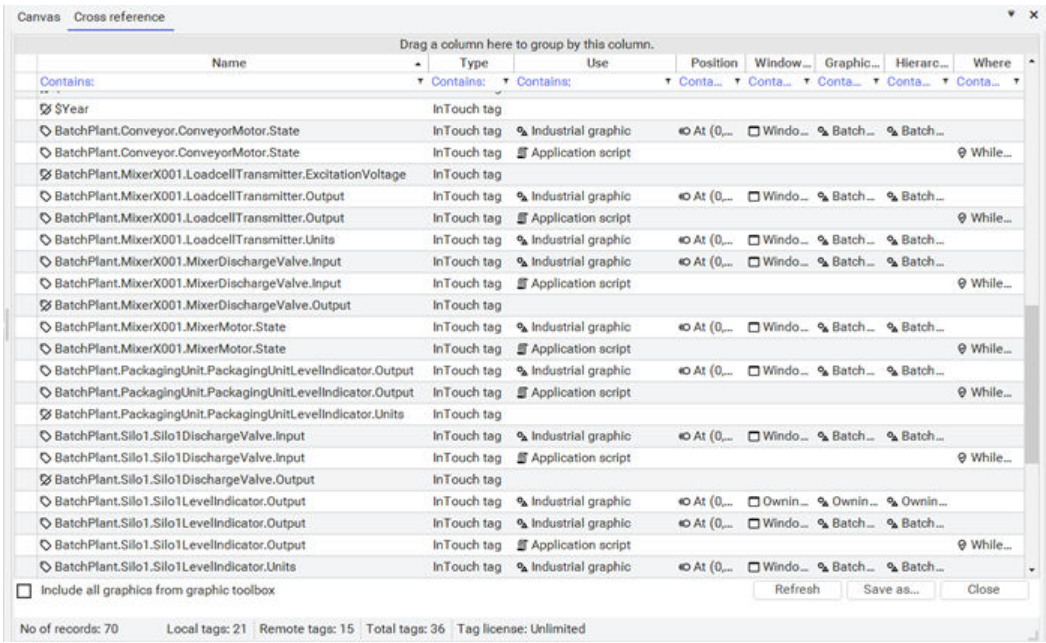
相互参照

クロス リファレンス ユーティリティを使用して、インスタンスの UDT メンバーが使用されている場所を見つけることができます。WindowMakerWindowMaker の［タグ ディクショナリ］を選択して［クロス リファレンス］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



相互参照



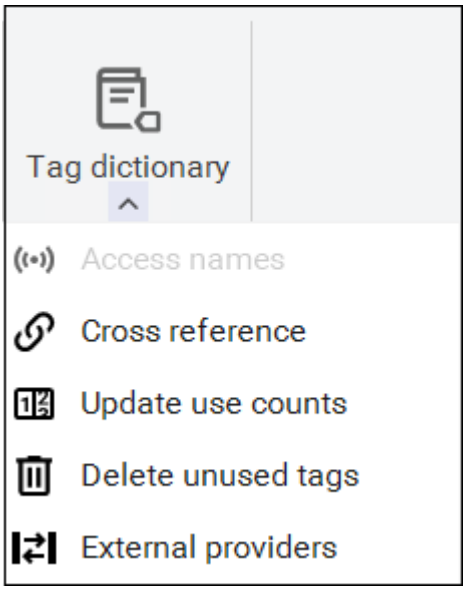
未使用タグの削除

インスタンスの未使用の UDT メンバーは、［未使用タグの削除］ページに表示され、そこから削除することができます。

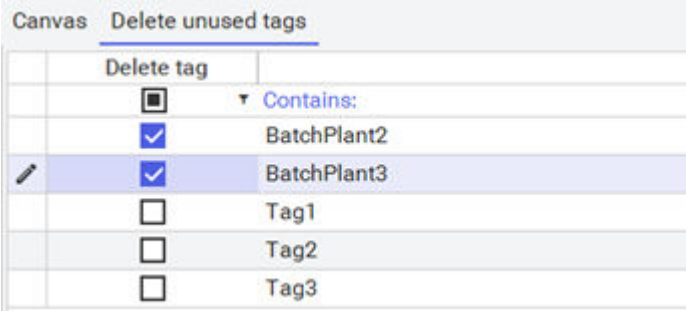
例:

1. $BatchPlantType の 2 つのインスタンスを作成し、それぞれに BatchPlant2 と BatchPlant3 という名前を付けます。
2. ［未使用タグの削除］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



未使用タグ BatchPlant2 と BatchPlant3 を削除できます。BatchPlant は使用中なので表示されません。



スーパータグとの共存

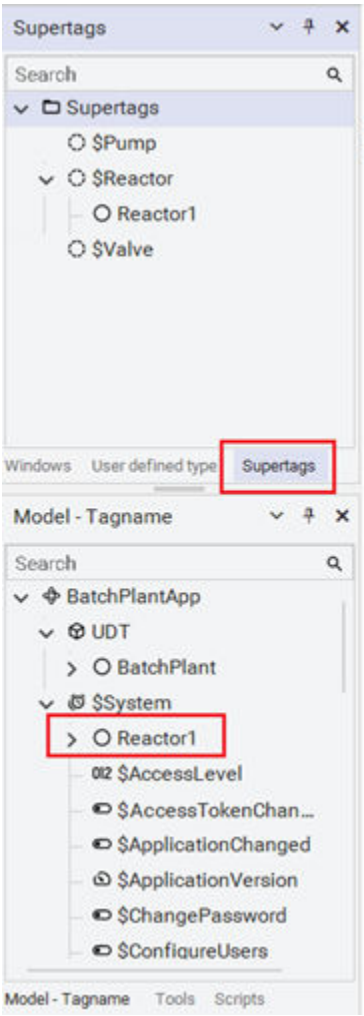
UDT はスーパータグと共存できます。スーパータグと UDT を同じ名前で作成できます。スーパータグインスタンスと UDT インスタンスは、スクリプト、アニメーション、アラーム、およびトレンドで連動することも、個別に機能することもできます。

共存の動作を以下に示します。

- デフォルトでは、スーパータグ ビューはオフになっています。
  - これは、新しいアプリケーションだけでなく、最新のリリースに移行されたアプリケーションにも該当します。
  - 必要な場合、スーパータグ ビューに戻すことができます。
- スーパータグでは引き続き "\" 構文が使用されます。
- スーパータグは、引き続き System Platform 2023 と同様に機能します。
- スーパータグ定義とユーザー定義型の間に名前の衝突はありません。
  - 同じ名前でスーパータグと UDT を作成できます（"Reactor" など）。
- スーパータグ インスタンスと UDT インスタンスは、スクリプト、アニメーション、アラーム、およびトレンドで連動することも、個別に機能することもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- スーパータグでは所有オブジェクトはサポートされません。したがって、スーパータグでは "me." はサポートされません。



Supertags definition and instance

Supertags instance

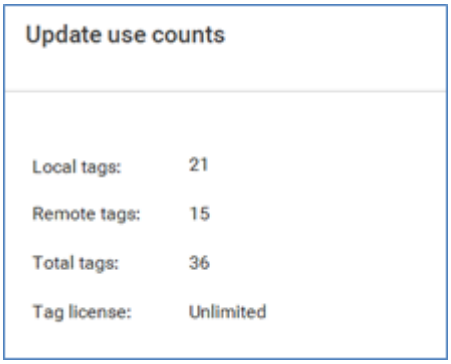
ライセンス

インスタンスの各 UDT メンバーは 1 つのローカル タグとしてカウントされます。UDT データ型または派生データ型はローカル タグの使用にカウントされません。

例:

次の例では、ローカル タグの合計数は 21 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

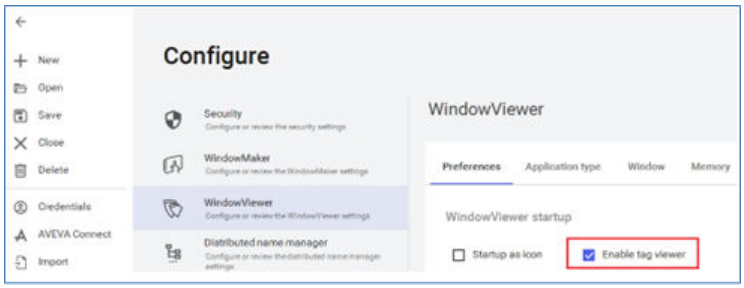


ランタイム Tag Viewer

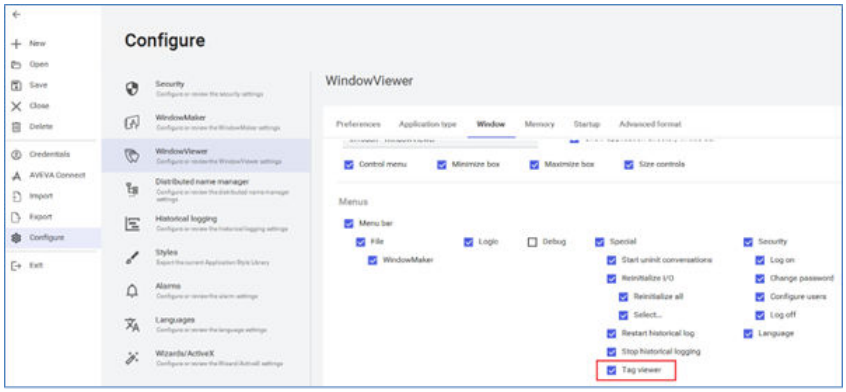
ランタイム Tag Viewer は UDT メンバーをサポートします。

Tag Viewer を開く前に、以下のことを確認します。

- ［ファイル］>［設定］>［**WindowViewer**］>［環境設定］で［**Tag Viewer** の有効化］チェック ボックスが選択されていること。

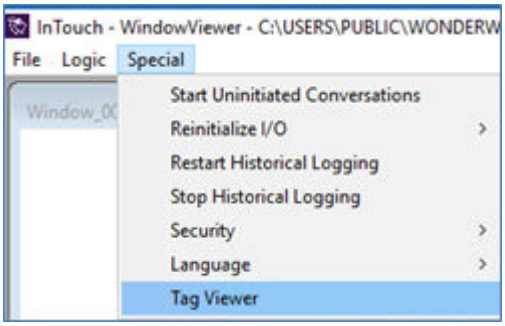


- ［ファイル］>［設定］>［**WindowViewer**］>［ウィンドウ］で［**Tag Viewer**］チェック ボックスが選択されていること。

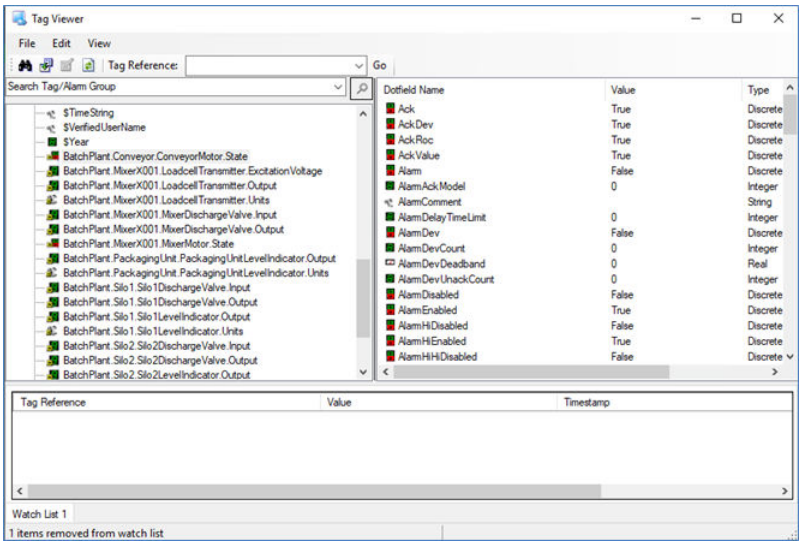


WindowViewerWindowViewer でアプリケーションを実行し、Tag Viewer を起動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



Tag Viewer に UDT インスタンス メンバーが表示されます。



UDT の制限事項

現在のリリースでは、以下の機能は UDT でサポートされていません。

- ActiveX コントロール プロパティでの UDT タグの使用。
- ウィザードでの UDT タグの使用（ボタン、クロック、枠など）。
- トレンドでの UDT タグの使用（履歴トレンド）。
- UDT タグでの SmartSymbol のインスタンス参照の上書き。
- SQL アクセス マネージャ バインド リストでの UDT タグの使用。
- レシピ マネージャでの UDT タグの使用。
- ネイティブ ユーザー入力アニメーションの最大および最小フィールドでの UDT タグの使用。
- InTouchProxy オブジェクトからのタグ ブラウザ。
- アラーム コメントでの UDT タグの使用。
- 言語の切り替えでの UDT タグの使用。
- タグの保持プロパティ。
- ディスクリート アラームやアナログ アラームなどのネイティブ アニメーションで使用される TBT の使用

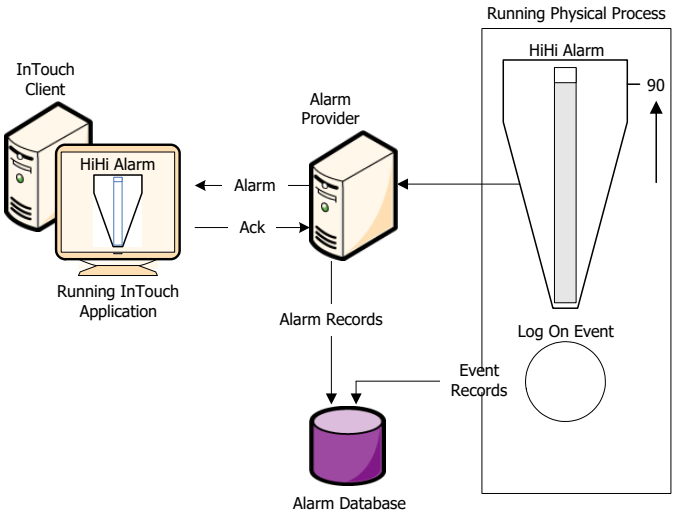
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 10 アラーム

アラームとイベントを生成してオペレータにプロセスの動作状況を通知する InTouch アプリケーションを作成できます。

- 問題を引き起こす可能性のあるプロセス状態に関する警告が、アラームによってランタイム オペレータに警告されます。通常は、プロセス値が定義済みのしきい値を超えた場合にトリガするようにアラームを設定します。オペレータは、アラームを確認する必要があります。
- イベントは、通常のシステム ステータス メッセージを表します。オペレータが InTouch アプリケーションにログオンするなど、システム条件が発生するときが典型的なイベントとなります。オペレータは、イベントを確認する必要はありません。

以下の図は、アプリケーションが実行されている間、InTouch HMI でアラームとイベントが処理される方法を示しています。アラームとイベントのデータは、アラーム データベースに保存されます。



イベント監視用のタグを設定できます。イベント メッセージは、タグの値が変更するたびにアラーム システムにログ記録されます。イベント メッセージには、値がどのように変更されたかに加えて、オペレータ、I/O、スクリプト、またはシステムのいずれによってその変更が開始されたのかという情報が含まれます。

アラームの設定

アラームを設定するには、簡単にアラーム条件を持つタグを設定します。

必要に応じて、以下の操作も実行できます。

- アラーム階層を定義します。
- アラームを無効にし、抑止します。
- アラーム コメントを設定します。
- その他のアラームとイベントのプロパティを設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム階層の定義

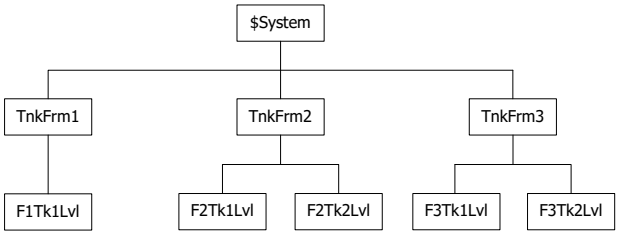
各 InTouch アラームはアラーム グループに属します。関連するアラームをグループに整理することによって、オペレータは簡単にアラームをフィルタしたり、表示したり、確認したりできるようになります。詳細については、「アーム グループ」を参照してください。

分散アラーム システムでは、アラーム グループ リストの基本としてアラーム グループが使用されます。分散アラーム システム グループ リストの作成の詳細については、「アラーム グループ リスト ファイルの作成」を参照してください。

アラーム グループの作成

アラーム グループの作成を開始する前に、アラーム グループを構成する方法と、アラーム グループ名を準備する必要があります。一貫したグループ命名規則を使用することによって、階層内でのグループの論理的な順序が制約されます。

以下の図で、階層内の同じレベルのグループに割り当てられた名前が類似していることに注意してください。



また、下位グループの名前に親の名前の一部を含めることによって、下位グループが親グループを参照していることにも注目してください。たとえば、第 3 レベルのアラーム グループの名前 F1Tk1Lvl にはグループ名の F1 プリフィックスを含めることによって、そのアラーム グループの親である TnkFrm1 を参照しています。階層内の異なるレベルのアラーム グループ間での親子関係を提示する命名規則を開発します。

注：アラーム グループは InTouch ライセンスではタグとして数えられないものの、データベースではタグとして数えられます。したがって、アラーム グループと実際のタグの合計数は、ユーザーの InTouch ライセンスで設定されている最大数を超えることはできません。

アラーム グループ名は、以下の要件を満たす必要があります。

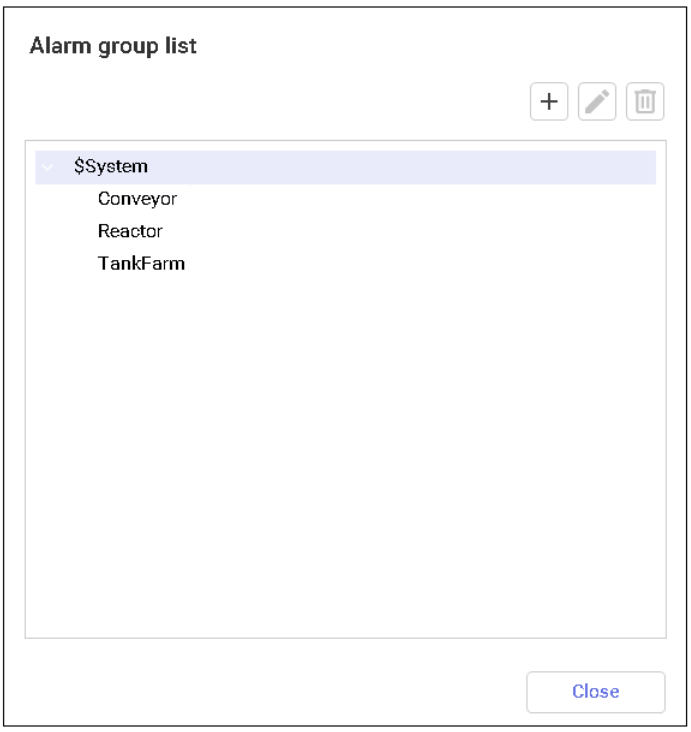
- 名前は 32 文字以内である必要があります。
- 名前は英数文字（A～Z、a～z または 0～9）で開始する必要があります。
- 名前には以下のキーボード文字（@、#、$、%、&、-、_、?、!、\）を含めることができ、名前の 2 番目の位置から使用を開始できます。
- 名前にハイフン（-）が含まれている場合、その名前は英字で開始する必要があります。
- 名前に空白文字を含めることはできません。
- 名前には少なくとも英字を 1 つ使用する必要があります。

アラーム グループを作成するには

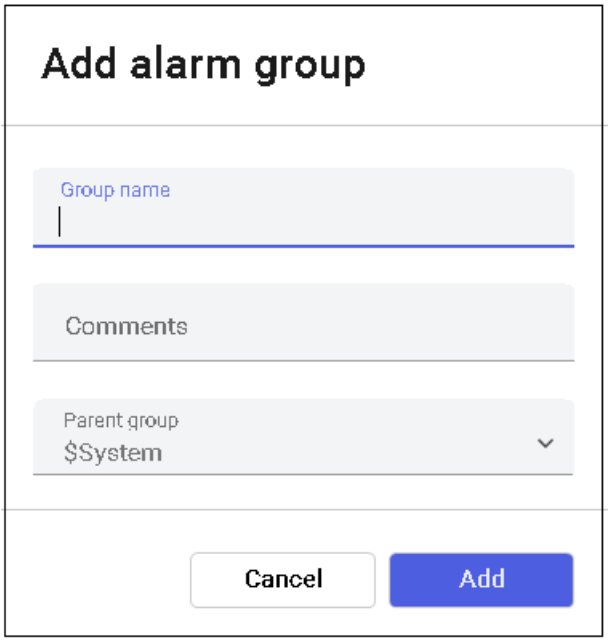
1. ［**ホーム**］メニューの［**アラーム**］グループで［**アラーム グループ**］をクリックします。

   ［**アラーム グループ リスト**］ウィンドウが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［+］アイコンをクリックして追加するか、Alt+A キーを押します。
   ［アラーム グループの追加］ダイアログ ボックスが表示されます。



3. ［グループ名］ボックスに、新しいアラーム グループの名前を入力します。
4. ［コメント］ボックスに、新しいアラーム グループのオプションのコメントを 49 文字以内で入力します。
5. アラーム グループを別の親グループに割り当てるには:

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. リストから**親グループ**名を選択します。InTouch アプリケーションに初めて定義されるアラーム グループである場合、このグループは親の $System グループに自動的に割り当てられます。

b. リストから新しい親グループを選択します。

6. ［**追加**］をクリックします。

   ［**アラーム グループ リスト**］ウィンドウが表示され、リストに追加された新しいアラーム グループが表示されます。

7. ［**閉じる**］をクリックします。

アラーム グループを変更する

アラーム グループは、以下の操作を実行するために変更できます。

- 名前を変更する
- 関連付けられたコメントを変更する
- 別のグループに再割り当てする

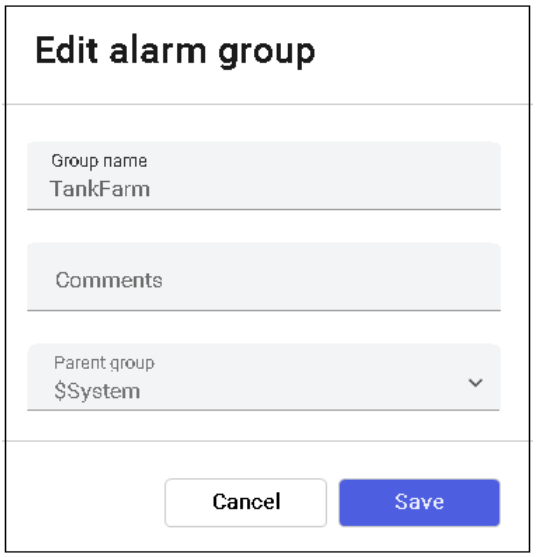
アラーム グループを変更する

1. ［**ホーム**］メニューの［**アラーム**］グループで［**アラーム グループ**］を選択します。

   ［**アラーム グループ リスト**］ウィンドウが表示されます。

2. 変更するアラーム グループを強調表示し、［編集］アイコンを選択するか、Alt+E を押します。

   ［**アラーム グループの編集**］ウィンドウが表示されます。



3. アラーム グループの名前やコメントを変更します。
4. アラーム グループを別の親グループに割り当てるには:

   a. ［**親グループ**］ドロップダウン リストから新しい親グループを選択します。

5. **保存**を選択します。
6. ［**OK**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム グループを削除する

アラーム グループを削除して、階層からそのグループを削除できます。削除されたアラーム グループに属すアラームやタグは、階層内で削除されたグループのすぐ上の親グループに自動的に再割り当てされます。また、削除された子グループも削除されたグループのすぐ上の親グループに再割り当てされます。

アラーム グループを削除する

1. ［ホーム］メニューの［アラーム］グループで［アラーム グループ］を選択します。

   ［アラーム グループ リスト］ウィンドウが表示されます。

2. アラーム グループを強調表示して **Delete** アイコンを選択するか、Alt+D を押します。メッセージが表示されたら、［はい］を選択します。
3. **[閉じる]**を選択します。

アラーム状況を使用したタグ変数の設定

アラームのタグ変数は、アラームのタイプと 1 つまたは複数のアラームしきい値を指定することによって設定できます。タグ変数の値が定義されたしきい値に達すると、アラームが発生します。タグ変数値のアラーム状況への移行情報はすべて、分散アラーム システムに報告されます。

論理値アラームの設定

論理値アラームは、論理型タグ変数に対応します。アラーム状況が論理型タグ変数の true （オン、はい、1）状態または false 状態（オフ、いいえ、0）に対応するかどうかを設定できます。

論理型タグ変数のアラーム状況を設定する

1. タグ名ディクショナリを開きます。
2. 既存の論理型タグ変数を選択するか、新しい論理型タグ変数を作成します。
3. ［タグ変数ディクショナリ］ダイアログ ボックスの一番上にある［アラーム］または［両方］のいずれかを選択して、論理値アラームの詳細設定ダイアログ ボックスを表示します。

4. ［確認モデル］領域で、タグ変数のアラーム確認モデルを選択します。
   - 確認の時間までにアラーム状況である、またはサブステートであるすべての移行を確認でカウントするには、［条件］を選択します。これがデフォルトの確認モデルです。
   - 確認をアラーム状況またはサブステートへの特定の移行に対してのみ行うには、［イベント指向］を選択します。最新のトランザクションを参照する場合のみ確認が受け入れられます。
   - 確認をアラーム状況やサブステートへの移行、または平常状態への復帰に対してのみ行うには、［拡張サマリ］を選択します。平常状態からの移行はすべて、新しい「平常への復帰」（RTN）グループの開始としてマーク付けされます。RTN グループ内でのすべての移行は、RTN グループ全体が確認されたと認識されるまで、個別に確認される必要があります。
5. ［アラーム コメント］ボックスに、最大 131 文字のアラーム コメントを入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注 :［アラーム コメント］ボックスには、二重引用符文字（"）を含めないでください。二重引用符を含むアラーム コメントをフェッチする場合、区切り文字として二重引用符を使用するプロセスが失敗します。

6. ［**アラーム状況**］領域で、アクティブなアラーム状況の論理型タグ変数値を**オン**または**オフ**にするかを選択します。
7. ［**優先度**］ボックスで 1 ～ 999 のアラーム優先度番号を割り当てます。デフォルトの優先度番号は、最高のアラーム優先度を示す 1 です。
8. オプションで、論理値アラームのアラーム抑止タグ変数を割り当てます。
   a. ［**アラーム抑止タグ変数**］ボックスで、ボタンを選択して、定義済みのタグ変数リストを持つ［**タグ変数を選択してください**］ダイアログ ボックスを表示します。
   b. テーブルを選択するには、リスト内でハイライトし、**[OK]**を選択します。抑止タグ変数として選択したタグ変数の名前が、［**アラーム抑止タグ変数**］ボックスに表示されます。

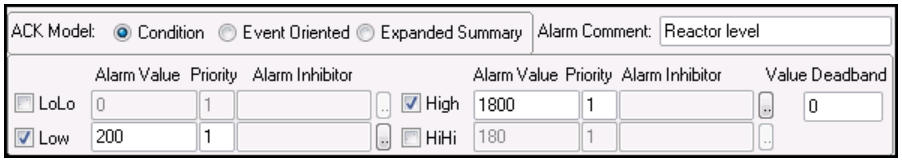
   アラームの抑止の詳細については、「アラームの抑止」を参照してください。
9. **保存**を選択します。
10. ［**閉じる**］選択して、［**タグ変数ディクショナリ**］ダイアログ ボックスを閉じます。

値型アラームの設定

値型アラームは、整数型または実数型のタグ変数に関連付けられています。タグ変数値が事前に設定された LoLo から HiHi までの範囲のしきい値から移行する場合に発生するアラームを設定できます。アラーム状況が任意のタグ変数の値やそのアラームの関連付けられた優先度に対応するかどうかを設定できます。

値型アラームの設定

1. タグ名ディクショナリを開きます。
2. 既存の実数型または整数型のタグ変数を選択するか、新しいタグ変数を作成します。
3. ［**タグ変数ディクショナリ**］ダイアログ ボックスの一番上にある［**アラーム**］または［**両方**］のいずれかを選択して、アラームの詳細設定ダイアログ ボックスを表示します。



4. ［**確認モデル**］領域で、タグ変数のアラーム確認モデルを選択します。
   - 確認の時間までにアラーム状況である、またはサブステートであるすべての移行を確認でカウントするには、［**条件**］を選択します。これがデフォルトの確認モデルです。
   - 特定のアラーム状況やサブステートへの移行に対してのみ確認を行うには、［**イベント指向**］を選択します。確認は最後のトランザクションを参照している場合にのみ受け入れられます。
   - 確認をアラーム状況やサブステートへの移行、または平常状態への復帰に対してのみ行うには、［**拡張サマリ**］を選択します。平常状態からの移行はすべて、新しい「平常への復帰」（RTN）グループの開始としてマーク付けされます。RTN グループ内でのすべての移行は、RTN グループ全体が確認されたと認識されるまで、個別に確認される必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［**アラーム コメント**］ボックスに、最大 131 文字のデフォルト コメントを入力します。このコメントは、タグ変数の .AlarmComment ドットフィールドに割り当てられます。

   注 :［アラーム コメント］ボックスには、二重引用符文字（"）を含めないでください。二重引用符を含むアラーム コメントをフェッチする場合、区切り文字として二重引用符を使用するプロセスが失敗します。

6. タグ変数の値が限界値を超えたかどうかを検出するために使用するアラームのタイプ（**LoLo**、**Lo**、**Hi**、**HiHi**）を選択します。
7. ［**アラーム値**］ボックスに、アラーム タイプに対する限界値を入力します。

   たとえば、**LoLo** アラームおよび **Lo** アラームの場合、タグ変数の値が［**アラーム値**］より小さい場合にアラーム状況となります。**Hi** アラームおよび **HiHi** アラームの場合、タグ変数の値が［**アラーム値**］を超えた場合にアラームが発生します。しきい値には、実数値を使用できます。

8. ［**デッドバンド**］ボックスに、アラーム状態から移行する前に、タグ変数値がアラーム値に対してどのくらいの範囲内でなければならないかを工学単位数で指定します。

   たとえば、アラーム状況から平常状態に戻るには、タグ変数値はアラームしきい値の範囲内に戻るだけではなく、指定したデッドバンド値以上の値に戻る必要があります。このデッドバンドは、タグ変数値がしきい値に近似であり、継続的にアラーム状態になったり、アラーム状態から離れたり変動している場合に、繰り返しの警告によってアラームが発生されることを避けるために使用されます。

9. オプションで、タグ変数のアラーム タイプ（LoLo、Lo、Hi、HiHi）に対してアラーム抑止タグ変数を割り当てます。

   a. ［**アラーム抑止タグ変数**］領域で、ボタンを選択して、定義済みのタグ変数リストを持つ［**タグ変数を選択してください**］ダイアログ ボックスを表示します。

   b. テーブルを選択するには、リスト内でハイライトし、**[OK]**をクリックします。抑止タグ変数として選択したタグ変数の名前が、［**アラーム抑止タグ変数**］ボックスに表示されます。

   抑止タグ変数の詳細については、「アラームの抑止」を参照してください。

10. **保存**を選択します。
11. ［**閉じる**］ を選択して、［**タグ変数ディクショナリ**］ダイアログ ボックスを終了します。

偏差アラームの設定

偏差アラームは、整数型または実数型のタグ変数に関連付けられます。現在のタグ変数値を目標値と比較することによって、アラームを設定できます。その後で、その差異の絶対値がタグ変数値の範囲のパーセンテージとして表される 1 つまたは複数のしきい値と比較されます。

たとえば、以下の値ではタグ変数の小偏差アラームと大偏差アラームの条件を設定しています。

最小値 = -1000

最大値 = 1000

小偏差 % = 10

大偏差 % = 15

基準値 = 500

これらの値を例として使用して、小偏差アラームと大偏差アラームの値は以下のように計算されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

1. タグ変数の値範囲の合計を計算します。

   1000 - (-1000) = 2000

2. タグ変数の値範囲の合計に小偏差と大偏差のパーセンテージを乗算します。

   2000 x 0.10 = 200 = 小偏差しきい値

   2000 x 0.15 = 300 = 大偏差しきい値

3. 基準値に対して、小偏差しきい値および大偏差しきい値を加算および減算します。

   500 - 200 = 300 = 小偏差下限値

   500 +200 = 700 = 小偏差上限値

   500 -300 = 200 = 大偏差下限値

   500 +300 = 800 = 大偏差上限値

偏差アラームを設定する

1. タグ名ディクショナリを開きます。
2. 既存の実数型または整数型のタグ変数を選択するか、新しいタグ変数を作成します。
3. ［**タグ変数ディクショナリ**］ダイアログ ボックスの一番上にある［**アラーム**］または［**両方**］のいずれかを選択して、アラームの詳細設定ダイアログ ボックスを表示します。

4. アナログ型タグ変数の値が指定した基準値に対して大偏差であるか小偏差であるかを判断するために使用する偏差（［**小偏差**］および［**大偏差**］）を選択します。
5. ［**% 偏差**］ボックスに、小偏差または大偏差のアラーム状況を発生させるための、アナログ型タグ変数の基準値からの逸脱パーセンテージを入力します。これはタグ変数の範囲に対するパーセンテージとして表されます。I/O 型タグ変数の場合、タグ変数の詳細設定ダイアログ ボックスに入力した［**工学値最小値**］と［**工学値最大値**］によって範囲が定義されます。メモリ型タグ変数の場合、範囲は最小値と最大値によって定義されます。
6. ［**基準値**］ボックスに、小偏差と大偏差のパーセンテージの基準となるタグ変数の参照値を入力します。
7. ［**偏差デッドバンド %**］ボックスに、タグ変数がアラーム状況から逸脱する前にタグ変数値が低下する必要のあるしきい値からの偏差パーセンテージを入力します。
8. **保存**を選択します。
9. ［**閉じる**］選択して、［**タグ変数ディクショナリ**］ダイアログ ボックスを閉じます。

変化率アラームの設定

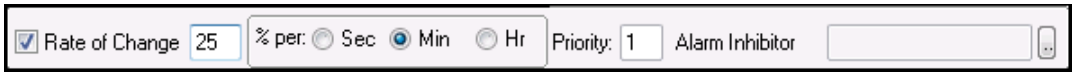
変更率アラームは、測定間隔内でアラームの絶対値が指定された制限を超えたときを検出します。値が変化するたびに、タグ変数は変化率アラームについてテストされます。変化率は、以前のタグ値、以前の変更が行われた時間、現在のタグ値、および現在の時間を使用して計算されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

特定の時間範囲内で計算された変化率は、タグに対して指定された変化率のパーセンテージ制限と比較されます。計算された変化率がパーセンテージ制限より大きい場合、アラーム状況がタグに設定されます。変化率アラームは、タグの変化率がアラーム制限を下回るまで有効なままになります。

変化率アラームを設定する

1. タグ名ディクショナリを開きます。
2. 既存の実数型または整数型のタグ変数を選択するか、新しいタグ変数を作成します。
3. ［**タグ変数ディクショナリ**］ダイアログ ボックスの一番上にある［**アラーム**］または［**両方**］のいずれかを選択して、アラームの詳細設定ダイアログ ボックスを表示します。以下の図は、変化率アラームに適用されるオプションのみを示しています。



4. ［**変化率**］ボックスをオンにします。
5. ［**%**］ボックスに、変化しきい値の最大許容パーセンテージを入力します。
6. 時間間隔の単位を、［**秒**］、［**分**］、または［**時**］の中から選択します。
7. ［**優先度**］ボックスに、アラーム優先度の値を 1 ～ 999 の範囲で入力します。
8. オプションで、変化率アラームのアラーム抑止タグ変数を割り当てます。
   a. ［**アラーム抑止タグ変数**］領域で、ボタンを選択して、定義済みのタグ変数リストを持つ［**タグ変数を選択してください**］ダイアログ ボックスを表示します。
   b. テーブルを選択するには、リスト内でハイライトし、**[OK]**を選択します。抑止タグ変数として選択したタグ変数の名前が、［**アラーム抑止タグ変数**］ボックスに表示されます。

   アラームの抑止の詳細については、「アラームの抑止」を参照してください。
9. **保存**を選択します。
10. ［**閉じる**］選択して、［**タグ変数ディクショナリ**］ダイアログ ボックスを閉じます。

0 = アラームを無効化

.AlarmEnabled ドットフィールドまたは AlarmDisabled ドットフィールドを使用して、タグ変数のすべてのアラームを一度に無効にしたり、有効にしたりできます。サブステートを持つアラームに対しては、各サブステートを個別に無効にできます。たとえば、アナログ値のアラームでは Hi を有効にし、HiHi を無効にすることができます。

ランタイム中、アラーム プロバイダでは、無効であるアラームやサブステートに対してアラームは生成されません。アラームが無効であるか有効であるかの変更は、ランタイムで行うことができます。

アラームが無効から有効の状態に移行するたびに、チェック ロジックにより、アラーム プロバイダによってそのアイテムをアラーム状況にする必要があるかどうかが判断されます。

アイテムがアラーム状況の間、アラームが無効となるかアクティブに抑止される場合、そのアイテムは強制的に別の（有効な）状態となります。どの状態となるべきかは、どの状態が使用可能であるかどうか、またそれらの状態が無効になっていないかどうかにより異なります。この処理は、アラーム タイプやしきい値に基づいて、アラーム プロバイダによって実行されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラームの抑止

オプションとして、アラームがアクティブ状態に移行することを防ぐ抑止アラーム タグ変数を各アラームまたはアラーム サブステートに割り当てることができます。

- 抑止タグ変数の値が True（0 以外または Null 以外）となり、その状態が維持されると、アラームは抑止されます。
- 同様に、抑止アラーム タグ変数が False（0 または Null）となり、その状態が維持されると、アラームは抑止されません。

抑止タグ変数は WindowMaker でのみ変更できます。また、抑止タグ変数の値は、ランタイムで変更できます。

抑止タグ変数は個別のアラーム サブステートに割り当てることができます。各サブステートは異なるタグ変数で抑止できます。また、一部のサブステートに抑止タグ変数を割り当てない状態で残すこともできます。

抑止されているアラーム（タグ変数が True）は、確認を待機していません。アラームがサブステートを持つ場合、引き続き使用可能であるサブステートでのみ確認を待機している状態となります。

アラームやサブステートは、個別に無効化したり、抑止すること（またはその両方）ができます。アラームは、それが有効になっており、アクティブに抑止されていない場合しかアクティブになりません。

アラームやサブステートに抑止タグ変数が割り当てられていない場合、常に False である抑止タグ変数を持つ場合と同じ結果になります。

移行によってアラームがアクティブに抑止されている状態から変化するたびに、チェック ロジックにより、InTouch でそのアイテムをアラーム状況にする必要があるかどうかが判断されます。

アイテムがアラーム状況の間、アラームがアクティブに抑止される場合、そのアイテムは強制的に別の(有効な）状態にされます。どの状態になるかは、どの状態が使用可能かどうか、またその使用可能な状態も無効になっていないかどうかによって異なります。この処理は、アラームのタイプやしきい値などに基づいて、InTouch によって処理されます。

確認を待機するアラーム（またはアラーム サブステート）がアクティブに抑止状態になった場合、アイテムを強制的に別の（有効な）状態に設定する必要があります。アイテムがアラーム状況であるかに関しては、InTouch によって適切な状態が決定される必要があります。

アラーム抑止タグ変数は、使用カウントとライセンス制限に含まれます。

アラーム抑止タグ変数の名前を取得するには、以下の読み取り専用のタグ変数ドットフィールドを使用します。

- AlarmDscInhibitor
- AlarmLoLoInhibitor
- AlarmLoInhibitor
- AlarmHiHiInhibitor
- AlarmHiInhibitor
- 1AlarmMajDevInhibitor
- AlarmMinDevInhibitor
- AlarmRocInhibitor

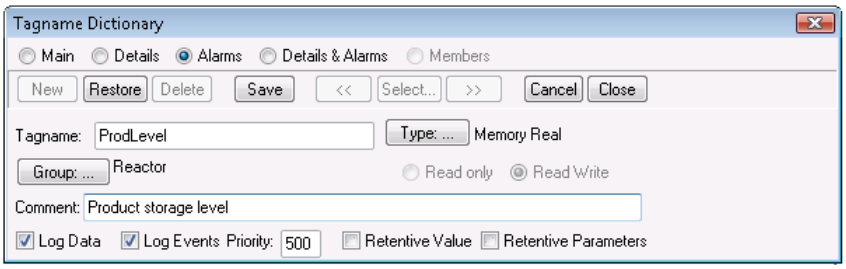
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

これらのフィールドはタグ変数の名前を返します。したがって、InTouch QuickScript で間接型タグ変数リファレンスでこの名前を使用して、アラーム抑止タグ変数の現在の値を特定したり、アラーム抑止タグ変数の値を変更したりできます。これにより、ランタイムでアラームのグループを強制的に有効にしたり、アクティブに抑止したりできます。

個々のタグのイベント プロパティの設定

タグを定義してイベントの監視を行う場合、タグの値が変化するたびにイベント メッセージがアラーム システムにログ記録されます。イベント メッセージでは、値がどのように変更されたかがログ記録されます。たとえば、オペレータ、I/O、QuickScript、またはシステムのいずれかによって変更が発生したかが記録されます。

1. タグ名ディクショナリを開きます。
2. イベントとしてログ記録されるデータに関連付ける既存のタグを選択するか、新しいタグを作成します。
3. ［**イベント ログ**］を選択します。［優先度］ボックスが使用可能になります。［優先度］に入力する値によって、タグのイベント優先度レベルが決定されます。



4. ［**優先度**］ボックスに、イベント優先度としての数字を 1 ～ 999 の範囲で割り当てます。
5. **保存**を選択します。
6. ［**閉じる**］選択して、［**タグ変数ディクショナリ**］ダイアログ ボックスを閉じます。

アラームとイベントのグローバル設定

アプリケーションによって生成されるすべてのアラームやイベントに適用される以下のグローバル設定を構成できます。

- 内部アラーム メモリ（バッファ）のサイズ。
- アラームが通常状態に戻ったことが確認を意味するかどうか。詳細については、「タグ値が平常に復帰するときの自動確認の使用」を参照してください。
- イベント ロギング。
- WindowViewer が再起動されるときに、アラームの有効化状態を保持するかどうか。
- アラームの確認コメントを一般のアラーム コメントとして使用するかどうか。詳細については、「アラーム コメントと確認コメントの使用」を参照してください。
- アラーム クライアント コントロール グリッドに LATCHED アラームを表示するかどうか。詳細については、「ラッチ済み状態の有効化」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム バッファ サイズを設定する

分散アラーム システムでの通信の大部分は、ノード間で送信されるアラーム クエリーとアラーム レコードにより構成されます。ノード内では、アラーム クエリーとアラーム レコードはアラーム バッファとも呼ばれる InTouch 内部アラーム メモリに保持され、ネットワーク トラフィックが最小限に抑えられています。アラーム バッファ サイズは、サマリまたは履歴アラーム クエリーに対してノードが保存できるアラームの最大数です。新しいレコード用の容量を作成するため、アラーム バッファは最も古いレコードを削除します。

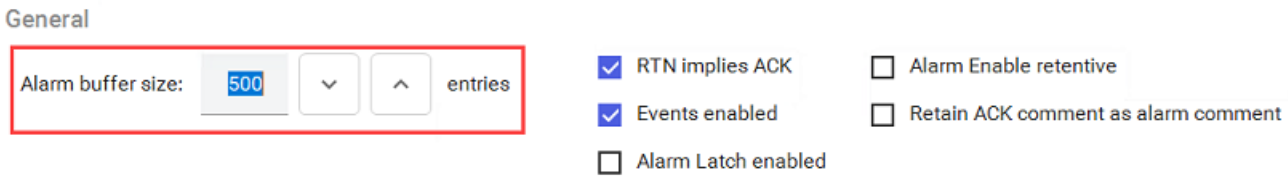
メモリに保存されているアラーム イベントのみがアプリケーション ウィンドウに表示できます。InTouch アプリケーションでアラーム ステータスが表示されない場合、バッファ サイズを 1 に設定して、ノード メモリを節約できます。

アラーム バッファに大きな値を割り当てると、ノードのパフォーマンスに影響が与えられる可能性があります。分散アラーム システムの場合、デフォルト値の 500 が推奨されます。

アラーム バッファ サイズを設定する

1. WindowMaker を開きます。
2. ［ファイル］メニューでで、［設定］をポイントし、［アラーム］を選択します。

   ［アラーム］設定画面が表示されます。



3. ［アラーム バッファ サイズ］ボックスに、サマリまたは履歴クエリー用のメモリ アラーム バッファに保存できるアラーム エントリの最大数を入力します。
4. 保存を選択します。

イベントを有効にする

イベントを有効にして、アプリケーション内にログ記録できます。イベントは、オペレータ操作、QuickScript、または I/O から発生するアプリケーション データの認識可能な変更を表します。

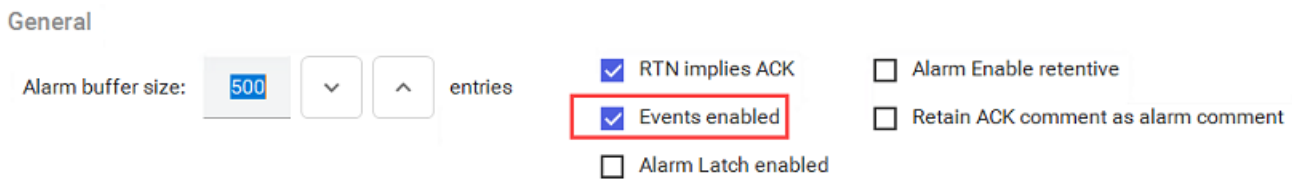
タグの［イベント ログ］プロパティは、タグに関連付けられたイベントが内部アラーム メモリに保存される前、またはアラーム データベースにログ記録される前に、タグ名ディクショナリから設定される必要があります。タグのイベント ロギングの指定の詳細については、「個々のタグのイベント プロパティの設定」を参照してください。

イベントを有効にする

1. WindowMaker を開きます。
2. ［ファイル］メニューでで、［設定］をポイントし、［アラーム］を選択します。

   ［アラーム］設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［**イベント有効**］チェック ボックスをオンにして、InTouch アプリケーションの実行中に発生するすべてのイベントをログに記録します。
4. **保存**を選択します。

アラームの有効化を保持する

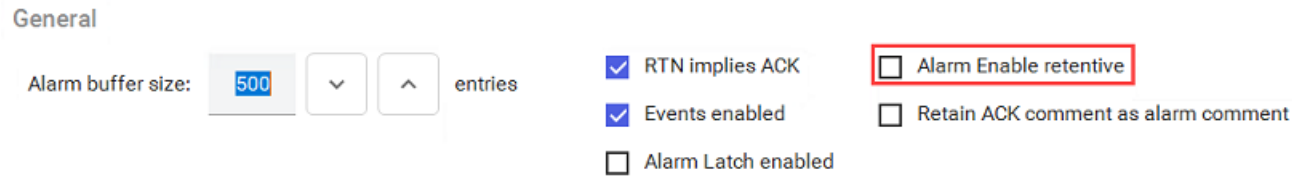
InTouch アプリケーションが停止され再起動されるとき、タグの **.AlarmEnabled** ドットフィールドに割り当てられている現在の値を保持するように選択できます。

.AlarmEnabled ドットフィールドに割り当てられている値によって、アラームとイベントのロギングがオンまたはオフに切り替わります。 **.AlarmEnabled** ドットフィールドは、タグまたはアラーム グループに割り当てることができます。**.AlarmEnabled** ドットフィールドがアラーム グループに割り当てられると、指定したアラーム グループ内でそのタグ変数に関連付けられているすべてのアラームがログ記録されるかどうか決定されます。

アラームの有効化を保持する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューでで、［**設定**］をポイントし、［**アラーム**］を選択します。

   ［**アラーム**］画面が表示されます。



3. ［**アラーム イベントの保持**］チェック ボックスをオンにして、InTouch アプリケーションの再起動時に **.AlarmEnabled** ドットフィールドの現在の状態を初期値として保持します。
4. **保存**を選択します。

ラッチ済み状態の有効化

LATCHED（ラッチ済み）状態を有効にして、アラーム クライアント コントロール グリッドで LATCHED アラームを表示できます。次の場合、アラームは LATCHED 状態になります。

- アラームが UNACK_RTN 状態から確認された場合。

  または

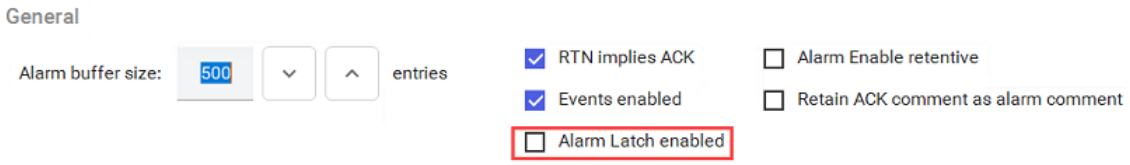
- アラームが ACK 状態から通常状態に戻った（RTN）場合。

ラッチ済み状態を有効にする：

1. WindowMaker を開きます。
2. ［**ファイル**］メニューでで、［**設定**］をポイントし、［**アラーム**］を選択します。

   ［**アラーム**］設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［アラーム ラッチ有効］チェック ボックスをオンにして、アラーム クライアント コントロール グリッドに LATCHED アラームを表示します。
4. 保存を選択します。

アラーム グループ リスト ファイルの作成

アラーム グループ リストを作成するには、分散名前マネージャを使用します。その後で、ローカル ノードおよびリモート ノードからの既存のアラーム グループをリストに追加します。

以下の表には、分散名前マネージャからアラーム グループを指定するための構文が示されています。

ノード	アラーム グループ構文
ローカル	\InTouch!Group_Name または .GroupName
リモート	\\Node_Name\InTouch!Group_Name または Node_Name.Group_Name

これらの例の場合、Node_Name は InTouch リモート ノードの名前です。Group_Name は、アラーム グループの名前です。アラーム グループ リストが定義されているローカル ノードでアラーム グループが定義されている場合は、前にピリオドを付けることによってアラーム グループ名を簡単に入力できます。たとえば、".Group_Name" と入力します。

アラーム グループ リストを作成する

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］を選択し、［分散名前マネージャ］を選択します。

   ［分散名前マネージャ］設定画面が表示されます。
3. ［分散アラーム］タブで［追加］（+）アイコンを選択します。

   ［分散アラームの追加］ダイアログが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. ［メンバー］ボックスに、クエリーに含める InTouch ノードとアラーム グループのリストを入力します。

   ノード名とアラーム グループ名は、標準グループ エントリ構文またはピリオドを使用したショートカット エントリを使用して入力できます。ショートカット エントリは、アラーム グループ リストを保存する際に標準グループ エントリに変換されます。

   注 : Node.Group および .Group の構文は、この設定ダイアログ ボックスでのみ使用できます。アラーム表示の設定やアラーム QuickScript 関数では有効ではありません。

5. ［追加］を選択して、このリストをアラーム グループ ファイルに追加します。

   メンバーの構文が自動的に変換されます。

6. 保存を選択します。
7. アラーム グループ リストの名前を Alarm Viewer コントロールのクエリーに追加します。これで、リストで指定したグループすべてのアラームが Alarm Viewer コントロールで表示されるようになります。

アラーム クエリー

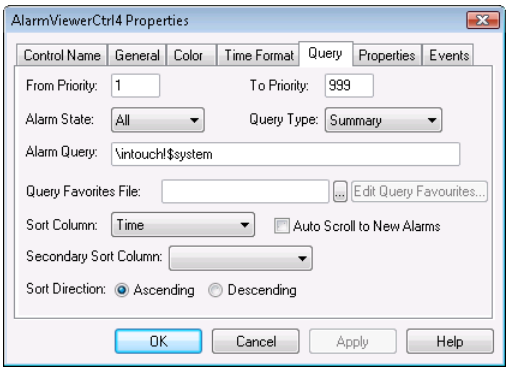
アラーム クエリーは以下のいずれかを取得します。

- InTouch 内部アラーム メモリまたはアラーム データベースからのアラームとイベント（履歴アラーム）。
- InTouch 内部アラーム メモリからの現在のアラーム（サマリ アラーム）

InTouch アラーム コントロールを設定する際、クエリー ソースを指定します。クエリー結果をフィルタするためのクエリー オプションを選択することもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以下の図は、Alarm Viewer ActiveX コントロールの［クエリー］タブを示しています。

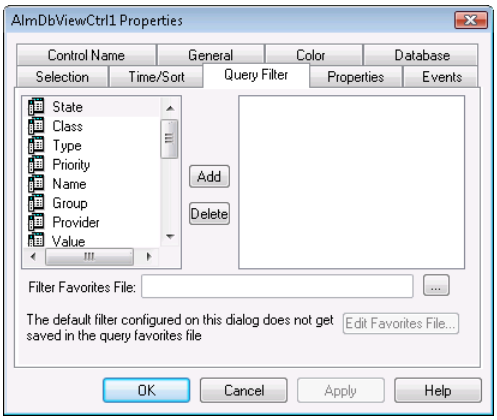


この例では、以下の条件によって選択されたアラーム データを表示するアラーム表示を作成します。

- アラーム優先度（1～999）
- アラーム状態（すべて、確認、または未確認）
- クエリー タイプ（サマリまたは履歴）
- アラーム グループ（ローカルまたはリモート データ ソース）

クエリーは「クエリー設定」ファイルと呼ばれる .xml ファイルに保存できます。ランタイム時、このファイルに保存された選択条件を使用して別のクエリーを実行することによって、アラーム表示を新しいアラーム データで更新できます。

その他の InTouch アラーム ActiveX コントロールはより広範囲のクエリー条件を提供しています。以下の図は、Alarm DB Viewer コントロールの［クエリー フィルタ］タブを示しています。



クエリーは、ダイアログ ボックスの左ペインに表示されているリストからのアラーム属性またはイベント属性を選択することによって作成します。その後、値を選択した属性に割り当てます。最後に、ブール型演算子を使用して属性を組み合わせて、クエリー フィルタ条件を設定します。

アラーム メモリからアラーム レコードおよびイベント レコードを選択するためのクエリー関数またはドットフィールドを含む QuickScript を記述できます。次の Alarm Viewer コントロール ステートメントでは、ApplyQuery() メソッドを使用して、アラーム メモリに対してクエリーを実行しています。

```
#AlarmViewerCtrl1.ApplyQuery ("\InTouch!$System",500,600,"All", "Historical");
```

このステートメントは、500 ～ 600 の間の優先度で「\InTouch!$System」で指定されているすべての履歴アラームを取得します。選択したアラーム レコードは Alarm Viewer コントロール画面に表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム クエリーの例

ローカル ノードでは、この構文の後にアラーム クエリーを使用します。

\Provider!AlarmGroup

以下に例を示します。

\InTouch!$System

リモート ノードには次のクエリー構文を使用します。

\\ノード名\Provider!AlarmGroup

たとえば、MyNode1 いうノードでは次のようになります。

\\MyNode1\InTouch!$System

Galaxy からのアラームのクエリーを実行するには、**「"Galaxy" の代わりに "Galaxy_<Galaxy name>" を使用して登録する」**チェックボックスをオンにして、次の構文を使用します。この構文では、特定のコンピュータ上の特定の領域にあるオブジェクトの特定のアラーム名からアラームを取得します。アラーム名は、属性名またはアラームの初期の名前の場合があります。Galaxy 名は、アラーム ウィンドウの［プロバイダー］カラムに表示されます。

\\NodeName\Galaxy_GalaxyName!AreaName!ObjectName.AlarmName

次の構文では、特定の領域からすべてのアラームを取得します。

\Galaxy_GalaxyName!AreaName

次の構文では、2 つの領域からアラームを取得します。

\Galaxy_GalaxyName!Area1 \Galaxy_GalaxyName!Area2

次の構文では、指定したコンピュータ ノードの Platform から指定した領域内のすべてのアラームを取得します（デフォルト）。

\\NodeName\Galaxy_GalaxyName!AreaName

単一のワイルドカードを使用して、指定した領域内の複数のアラーム名に一致させることもできます。次の構文では、領域「AreaName」にあり「Tank」という文字で始まるすべてのオブジェクトからすべてのアラームを取得します。

\Galaxy_GalaxyName!AreaName!Tank*

次の構文では、「AreaName」領域にあるすべてのオブジェクトから「Hi」という名前が付いているすべてのアラームを取得します。

\Galaxy_GalaxyName!AreaName!*.Hi

注：InTouch の代わりに Galaxy からアラームをクエリしている場合、グループ名に使用できる文字数は 128 です。

詳細な InTouch クエリー情報の取得

以下の表には、各 InTouch クエリー ソースのクエリーに関する詳細情報のリファレンスが一覧表示されています。

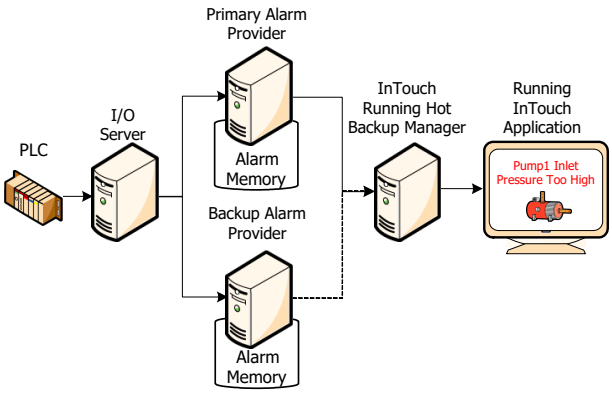
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クエリ ソース	参照先
Alarm DB Logger Manager	ログに記録するアラームを設定する
Alarm Printer ユーティリティ	印刷するアラームを設定する
Alarm Viewer コントロール	表示するアラームを設定する
Alarm Tree Viewer コントロール	表示するプロバイダとグループの設定
Alarm Pareto コントロール	分析するアラームの設定
分散アラーム表示オブジェクト	表示するアラームを設定する
Hot Backup ManagerHot Backup Manager	アラーム レコード マッピング ファイルを作成する

アラーム冗長性による工場セキュリティの強化

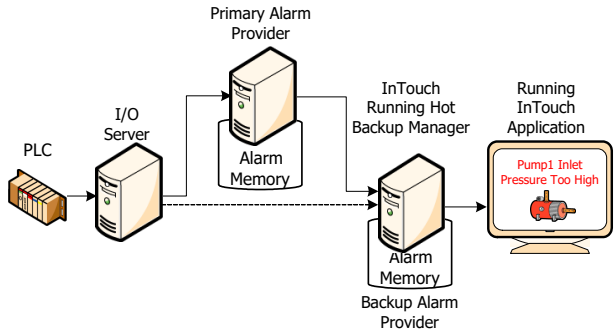
InTouch 分散アラーム システムは通知を発行し、ネットワーク内のリモート ノードで実行中のアプリケーションからアラーム確認を受け取ります。アラーム プロバイダ アプリケーションでは、メモリ内にアラーム データが保存されます。アラーム コンシューマ アプリケーションはその他のノードでクライアントとして実行して、アラームプロバイダからのアラームをリモートでクエリーしたり、表示したり、確認したりします。

重複するアラーム プロバイダを作成するには、Alarm Hot Backup Manager を使用します。以下の図は、Hot Backup Manager がセカンダリの現在のアラーム リポジトリをバックアップ プロバイダとして使用する方法を示しています。



次の図に示すように、Hot Backup Manager とバックアップ プロバイダを同じノード上で実行することもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



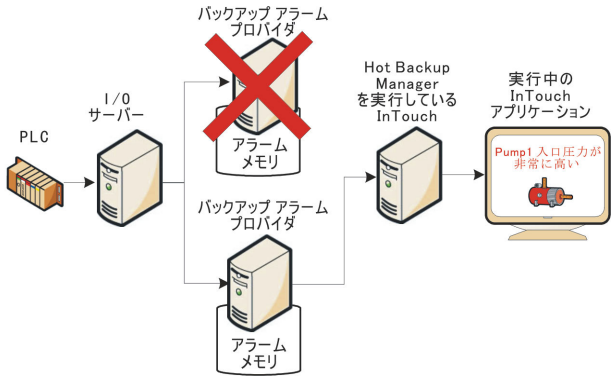
Hot Backup の理解

Hot Backup には、プライマリとバックアップ（Hot Backup ペア）という 2 つのアラーム プロバイダをポイントする 1 つの名前（Hot Backup ペア名）が提供されています。InTouch HMI アラームのコンシューマは、この 1 つの Hot Backup ペア名を参照して、プライマリまたはバックアップ アラーム プロバイダのいずれかからアラームを取得できます。次のアラーム クライアントは、Hot Backup ペアのクエリーをサポートします。

- InTouch HMI Alarm DB Logger Manager
- InTouch HMI Alarm Printer
- InTouch HMI Alarm Viewer コントロール
- アラーム クライアント コントロール

両方のプロバイダ ノードが正常に動作している場合、アラーム コンシューマはプライマリ プロバイダからアラーム データを受け取ります。ただし、プライマリ プロバイダが失敗した場合、アラーム コンシューマは代わりにバックアップからアラーム データを受け取ります。プライマリ プロバイダが使用できる場合でもバックアップ プロバイダを使用するように Hot Backup Manager を設定することもできます。このバックアップ オプションを選択しない場合、プライマリ プロバイダが使用可能な場合、プライマリ プロバイダに自動的に切り替わります。詳細については、「Hot Backup ペアを作成する」を参照してください。

以下の図は、プライマリ アラーム プロバイダが失敗した後でも引き続きアラーム コンシューマがアラームを受け取る様子を示しています。アラーム コンシューマは引き続き Hot Backup ペアを参照しますが、バックアップ プロバイダがアラーム データを提供します。



Hot Backup のアラーム プロバイダの指定

以下のアラーム プロバイダのどれでも指定することができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- グラフィック エディタでは、「InTouch:」と入力しなくても

  InTouch をアラーム プロバイダとして指定する場合、プライマリ ノードとバックアップ ノードの両方に対してプロバイダが InTouch である必要があります。ただし、バックアップ ノードのアラーム グループは、プライマリ ノードとは異なるものにできます。

- Galaxy<1/>Galaxy

  Galaxy または Galaxy_<Galaxy 名> アラーム プロバイダを指定するには、IDE で AppEngine 冗長性を設定する必要があります。

- Galaxy_<Galaxy 名>

  Galaxy 名に対する有効な文字は、英数字と特殊記号 $、#、および _ です。

  <1>Galaxy_<Galaxy</1><2> </2><3>名</3><4>></4><5> </5><6>を指定する場合、</6>プライマリ ノードとバックアップ ノードの両方に対して <7>Galaxy </7><8>名が同じである必要があります。</8>

  Galaxy または Galaxy_<Galaxy 名> をアラーム プロバイダとして指定する場合、バックアップ ノードのプロバイダは Galaxy または Galaxy_<Galaxy 名> にできます。

  バックアップ ノードのアラーム グループは、プライマリ ノードと同じである必要があります。

注：システムは、2 つの異なる Galaxy から生成されたアラームの Hot Backup ペアをサポートしません。

Hot Backup Manager について

Hot Backup Manager では、プライマリ プロバイダとバックアップ プロバイダの間でアラーム確認が同期化されます。アラームがプライマリ プロバイダで確認される場合、同じアラームがバックアップ プロバイダでも同時に確認されます。

Hot Backup Manager：

- 設定ユーティリティを提供して、バックアップ ペアを作成します。
- 設定ユーティリティを提供して、Hot Backup ペアのプライマリ プロバイダとバックアップ プロバイダの間でアラーム レコードをマップします。
- InTouch アラーム バックアップ ペアの間でアラーム確認を同期化します。
- 分散アラーム システムが開始および停止するときに、Hot Backup システムに属するすべてのノード間で通信が確立されます。

TSE セッションでの Alarm Hot Backup Manager の使用

［ツール］ > ［アプリケーション］を使用することで、異なる WindowMaker の Terminal Services Edition (TSE) セッションで、1 つの Alarm Hot Backup Manager のみを起動できます。

［スタート］〉［プログラム］ショートカットを使用することで、各 TSE セッション内で 1 つの Alarm Hot Backup Manager を起動できます。最後に保存された設定は、以前の設定を上書きします。

Hot Backup ペアのアラームのクエリーは、ランタイム時に TSE セッション内でサポートされています。

Hot Backup ペアを設定する

Alarm Hot Backup Manager では、プロバイダ アプリケーションを実行している 2 つのホスト ノードからバックアップ ペアが作成されます。Hot Backup Manager は、WindowMaker から起動できます。Hot Backup ペアを設定するには：

- Hot Backup ペアを作成します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アラーム レコードのキー フィールドを設定します。
- アラーム レコードのキー フィールドをマップします。
- アラーム レコード マップを Hot Backup Manager にインポートします。

Hot Backup ペアを作成する

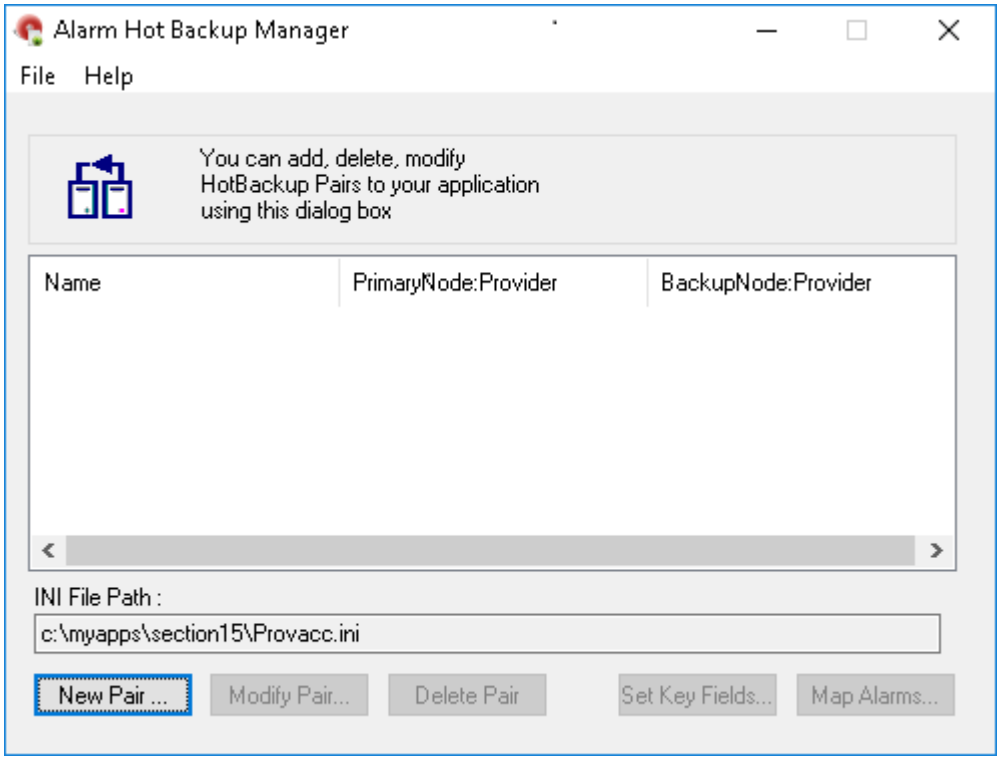
Hot Backup ペアを設定するには:

- Hot Backup ペアに名前を割り当てます。
- プライマリ アラーム プロバイダを識別します。
- バックアップ アラーム プロバイダを識別します。

設定情報を含む Provacc.ini ファイルも指定できます。

Hot Backup ペアを設定する

1. Alarm Hot Backup Manager を開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm Hot Backup Manager**］をダブルクリックします。

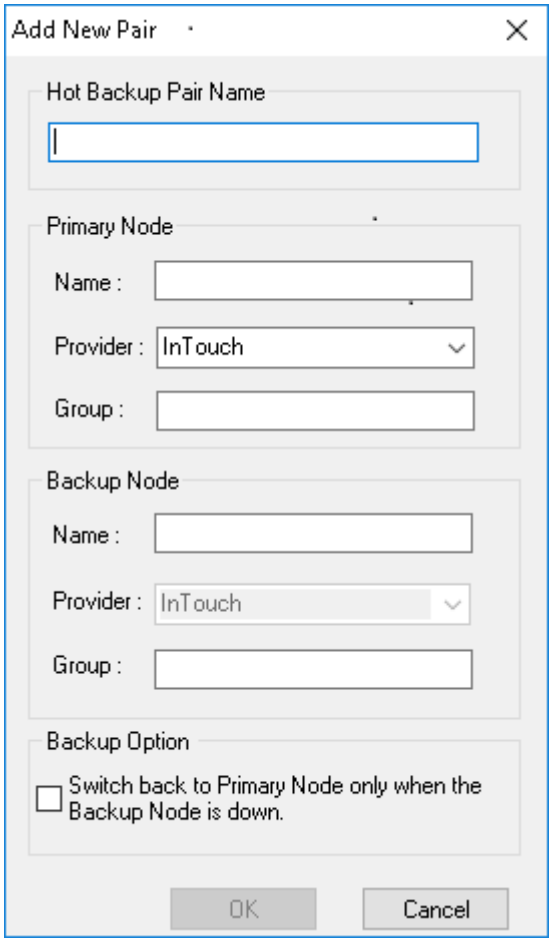


2. ［ファイル］メニューで、［開く］を選択します。Provacc.ini ファイルを強調表示し、［**OK**］をクリックします。

   デフォルトでは、Alarm Hot Backup Manager は最後に開いた InTouch アプリケーション フォルダにある Provacc.ini ファイルを確認します。InTouch アプリケーション フォルダにある Provacc.ini ファイルを使用する必要があります。それ以外の場合は、別に指定したフォルダに Provacc.ini ファイルのコピーを作成し、それを Hot Backup Manager で使用するために選択します。

3. 新規ペアを選択します。［新規ペアの追加］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**Hot Backup ペアの名前**］ボックスに、新しいバックアップ ペアの一意の名前を入力します。

   ペア名には、32 文字以下の英数字を使用できます。ペア名には、ドル記号（$）、ポンド記号（#）、およびアンダースコア（_）文字を使用できます。

5. ［**メイン ノード**］領域で、プライマリ ノードを設定します。以下の手順を実行します。

   a. ［**名前**］ボックスに、プライマリ プロバイダ アプリケーションを実行中のコンピュータのノード名を入力します。ノード名は、Hot Backup Manager に固有である必要があります。存在しないノード名を入力した場合、またはノード名が別の Hot Backup ペアで使用されている場合、エラーメッセージが表示されます。

   b. ［**プロバイダ**］ボックスで、ドロップダウン リストからアラーム プロバイダを選択します。デフォルトは InTouch です。

   c. ［**グループ**］ボックスに、プライマリ プロバイダからのアラームをクエリーするアラーム グループの名前を入力します。

6. ［**バックアップ ノード**］領域で、バックアップ ノードを設定します。以下の手順を実行します。

   a. ［**名前**］ボックスに、バックアップ プロバイダ アプリケーションを実行中のコンピュータのノード名を入力します。これは Hot Backup Manager が実行されているのと同じノードでもかまいません。

   b. Galaxy または Galaxy_<Galaxy 名> のアラーム プロバイダを指定した場合、［**プロバイダ**］ボックスで、バックアップ プロバイダを選択します。

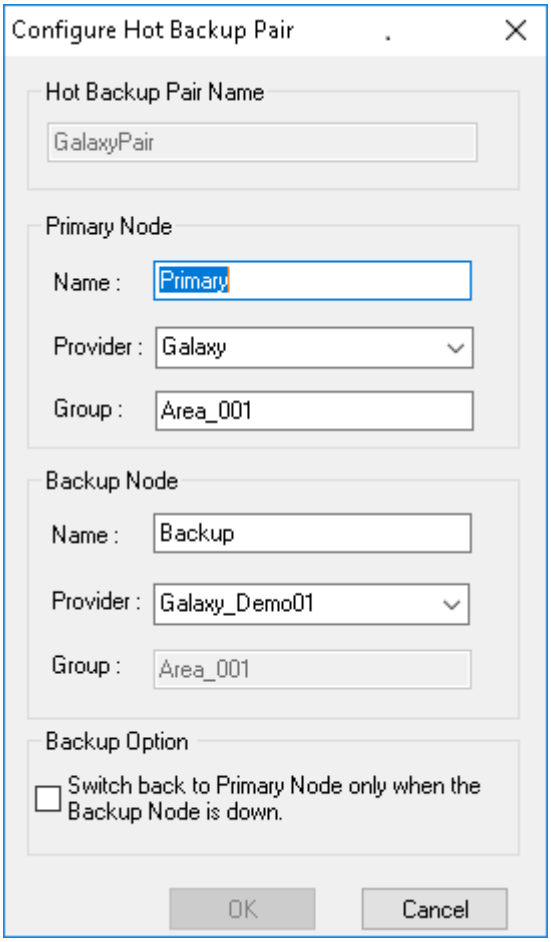
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Galaxy または Galaxy_<Galaxy 名> をプライマリ ノード アラーム プロバイダとして指定した場合、バックアップ ノードのプロバイダは Galaxy または Galaxy_<Galaxy 名> である必要があります。

InTouch をプライマリ ノード アラーム プロバイダとして指定した場合、バックアップ ノードのプロバイダは InTouch である必要があります。

c. ［**グループ**］ボックスに、バックアップ プロバイダからのアラームをクエリーするアラーム グループの名前を入力します。

<1>Galaxy </1><2>または</2><3> Galaxy_<Galaxy</3><4> 名</4><5>> </5><6>をアラーム プロバイダとして指定した場合、バックアップ ノード グループはプライマリ ノード グループと同じである必要があり、それは編集できません。</6>



7. プライマリ ノードが使用可能なときでもバックアップ ノードを継続して使用するには、［**バックアップ ノードが使用できない場合にのみプライマリ ノードに切り替える**］を選択します。（このオプションは英語以外のオペレーティング システムではテストされていません。）

   デフォルトでは、このオプションは選択されていません。

8. ［**OK**］を選択します。
9. ［**ファイル**］メニューで、［**保存**］を選択します。
10. WindowMaker を再起動します。

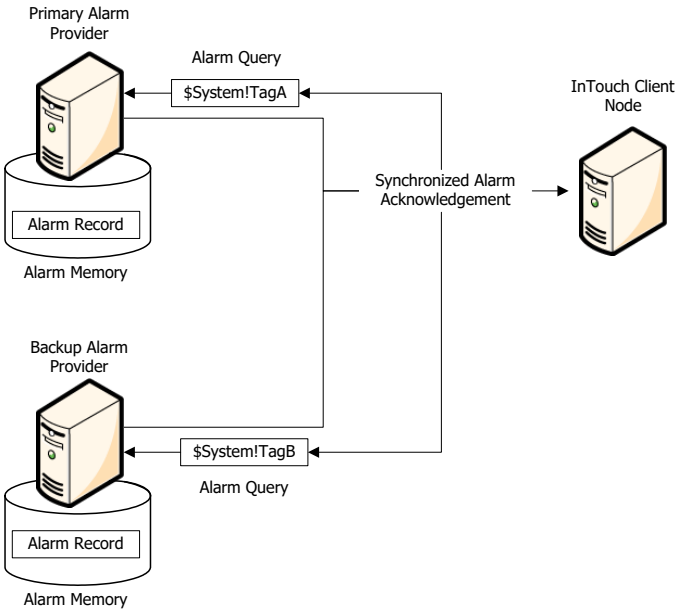
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Hot Backup ペアのアラーム キー フィールドの設定

プライマリ プロバイダとバックアップ プロバイダの間でアラーム確認を同期化するには、タグ変数アラーム レコード フィールドの組み合わせを識別する必要があります。フィールドのこの組み合わせによって、各プロバイダの現在のアラーム リポジトリに保存されるペアのアラーム レコードに固有のマッピング キーが生成されます。

InTouch アラーム プロバイダに対してのみ、アラーム キー フィールドを設定、およびアラームをマップできます。

以下の図は、標準クエリー内のアラーム レコード フィールドに基づく同期化されたアラーム確認要求を示しています。



マッピング キーは、デザイン時およびランタイムのアラーム レコードの組み合わせで構成できます。デザイン時アラーム レコードは、タグ変数ディクショナリから定義される場合、タグ変数のアラーム プロパティに基づいています。

たとえば、アラーム名フィールドはプライマリおよびバックアップのノード アプリケーションで定義されるタグ変数の名前を使用するため、デザイン時で認識されています。QuickScript またはオペレータ アクションによって、アプリケーションの実行中にレコードとして保存されるアラーム プロパティが定義または変更されます。

マップ キーは、デザイン時およびランタイムのアラーム レコード フィールドの組み合わせから作成できます。マップ キーは、各プロバイダの現在のアラーム リポジトリから 1 つのレコードのみを選択する必要があります。キー フィールドでは、固有のクエリーが作成される必要があります。

Hot Backup Manager を使用して、アラーム レコード フィールドからマッピング キー リストを作成します。

注：Galaxy または Galaxy_<Galaxy 名> アラーム プロバイダを指定する場合、［**キー フィールドの設定**］と［**アラームのマップ**］オプションの両方が無効です。

Hot Backup ペアのアラーム フィールド マッピング リストを作成する

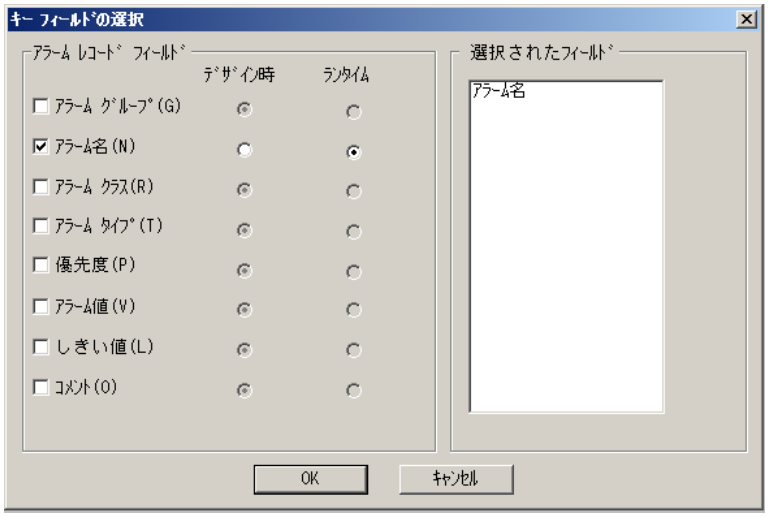
1. Alarm Hot Backup Manager を開きます。以下の手順を実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. ［ツール］ビューで［アプリケーション］を展開します。

b. ［**Alarm Hot Backup Manager**］をダブルクリックします。

2. リストから Hot Backup ペアを選択します。
3. キーフィールドを設定 を選択します。［キー フィールドの選択］ダイアログ ボックスが表示されます。



4. ［アラーム レコード フィールド］領域で、マッピング キー リストに含めるアラーム レコード フィールドを選択します。

   選択したアラーム レコード フィールドは、［選択されたフィールド］リスト ボックスに表示されます。

5. 選択したアラーム レコード フィールド用に［デザイン時］または［ランタイム］を選択します。
6. ［**OK**］を選択します。
7. WindowMaker を再起動します。

アラーム レコード マッピング ファイルを作成する

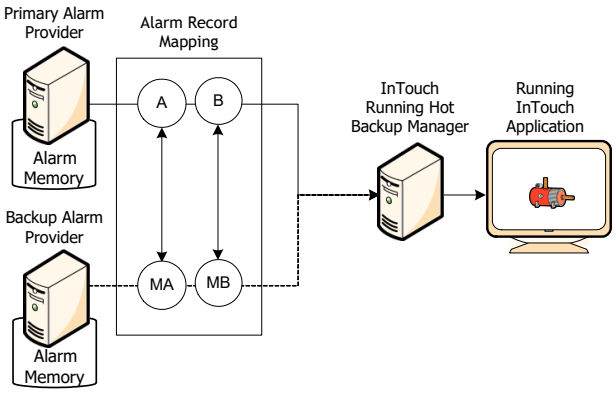
プライマリとバックアップのアラーム プロバイダが異なるアプリケーションを実行しているときはいつでも、Hot Backup ペアのアラーム レコードをマップする必要があります。アラーム レコード マッピングでは、プライマリ プロバイダとバックアップ プロバイダの異なるアラーム レコード間の対応が確立されます。たとえば、割り当てられた InTouch タグ変数名に基づいてアラーム レコードをマップできます。名前は異なる可能性がありますが、アラーム レコードは、2 つのプロバイダ アラーム リポジトリの間で論理的に一貫性を保っています。

注：プライマリ プロバイダとバックアップ プロバイダの両方が同じアプリケーションを実行している場合は、アラーム レコード マッピング ファイルの作成は必要ありません。マッピング ファイルが提供されていない場合、分散アラーム システムでは、プライマリ プロバイダとバックアップ プロバイダが同じアラーム レコードを使用して同じアプリケーションを実行していると想定されます。

マッピングによって、アラーム確認は異なるアプリケーションを実行中のプロバイダの間で同期化できます。分散アラーム システムがプロバイダのアラームを確認する際、別のプロバイダでどのアラームを確認するべきなのかも把握しています。

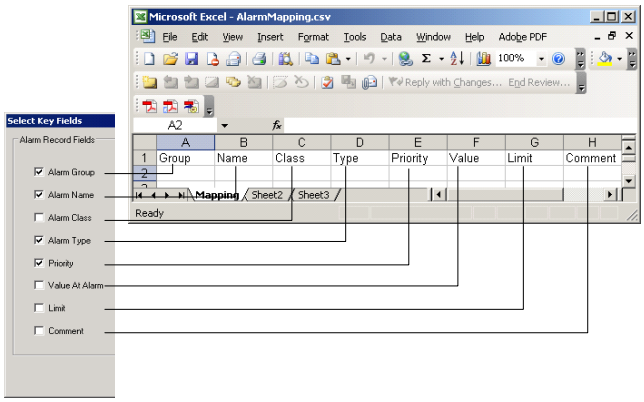
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以下の図は、Hot Backup ペアの 2 つのプロバイダ間のアラーム レコード マッピングを示しています。この例では、プライマリ プロバイダの A アラーム レコードと B アラーム レコードがバックアップ プロバイダの対応するアラーム レコードにマップされます（MA と MB）。



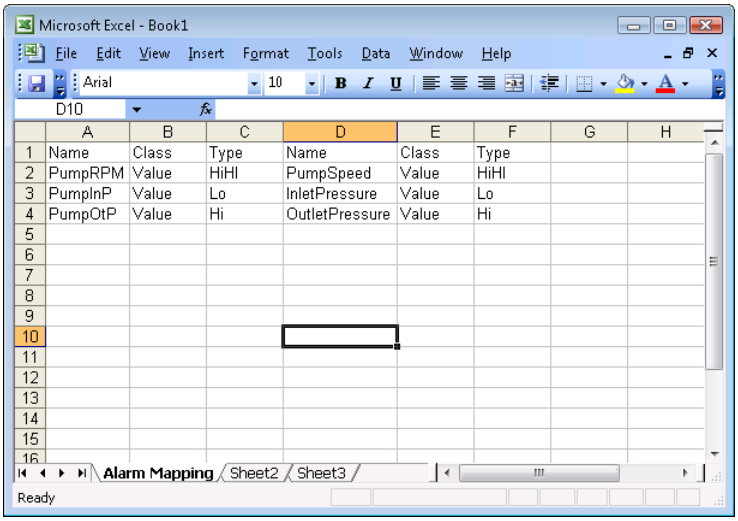
Hot Backup Manager は、Microsoft Excel またはメモ帳のようなテキスト エディタで作成するカンマ区切り値（CSV）ファイルからアラーム レコード マップをインポートします。マッピング ファイルには、プライマリ プロバイダとバックアップ プロバイダの対応するアラーム レコードを関連付けるアラーム レコード フィールドの順序付きリストが含まれます。

タグ変数アラーム レコード フィールドをマッピング ファイルのヘッダとして指定する必要があります。ファイル内のヘッダの順序は、[**キー フィールドの選択**] ダイアログ ボックスに表示されるアラーム レコード フィールドと一致する必要があります。以下の図は、[**キー フィールドの選択**] ダイアログ ボックスのアラーム レコード フィールドの順序に一致する Excel ファイルのカラム ヘッダを示しています。



マッピング キーを生成するために使用されるアラーム フィールド レコードの選択されたヘッダのみを含むマッピング ファイルを作成できます。以下の図は、アラーム名、クラス、およびタイプのヘッダのみを含む Excel ファイルを示しています。ヘッダを追加するとき、ヘッダの順序は、[**キー フィールドの選択**] ダイアログ ボックスのアラーム レコード フィールドの順序と常に一致する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



	A	B	C	D	E	F
1	Name	Class	Type	Name	Class	Type
2	PumpRPM	Value	HiHI	PumpSpeed	Value	HiHI
3	PumpInP	Value	Lo	InletPressure	Value	Lo
4	PumpOtP	Value	Hi	OutletPressure	Value	Hi

カラムの左セットでプライマリ プロバイダのアラーム フィールド レコードを指定します。同じように、カラムの右セットでバックアップ プロバイダの同じレコードを指定します。

マッピング ファイル カラム ヘッダ	アラーム フィールド レコードに割り当てられる値
グループ	タグ変数が割り当てられたアラーム グループの名前。アラーム グループ名には空白を含めることはできません。
名前	アラーム レコードがマップされるタグ変数の名前。タグ変数名に空白を含めることはできません。
クラス	タグ変数に割り当てられるアラームのクラス。 可能なクラスの値は以下のとおりです。 • VALUE - 値アラーム • DEV - 偏差アラーム • ROC - 変更率アラーム • DSC - 論理値アラーム
タイプ	アラーム クラスに関連付けられるアラーム条件のタイプ • LOLO、LO、HI、HIHI - 値アラーム • MinDev と MajDev - 偏差アラーム • ROC - 変更率アラーム • DSC - 論理値アラーム
優先度	アラーム条件に割り当てられる優先度。優先度は 1 ～ 999 の数である必要があります。
値	以下の注記を参照してください。
しきい値	以下の注記を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マッピング ファイル カラム ヘッダ	アラーム フィールド レコードに割り当てられる値
コメント	以下の注記を参照してください。

アラーム値、しきい値、およびコメントのカラム：

- 特定のノードの特定のレコードに対する「クラス」または「タイプ」の値が不明である場合、「アラーム値」と「しきい値」のカラム値には、Null 以外の値であればどれでも使用できます。
- 特定のノードの特定のレコードに対する「クラス」の値が Value、Dev、または ROC である場合、「アラーム値」と「しきい値」のカラム値は 1234567890.-+eE の文字のみを受け入れることができます。
- 特定のノードの特定のレコードに対する「タイプ」の値が LOLO、LO、HI、HIHI、MinDev、MajDev、または ROC である場合、「アラーム値」と「しきい値」のカラムには 1234567890.-+eE の文字のみを受け入れることができます。
- 特定のノードの特定のレコードに対する「クラス」または「タイプ」のうちいずれかの値が DSC である場合、「アラーム値」と「しきい値」のカラム値は Null 以外の値であればどれでも使用できます。
- 「コメント」カラムの値には、制限はありません。
- マッピング ファイルのすべてのレコードは、固有のものである必要があります。Hot Backup Manager では、インポート プロセスの間に重複レコードが省略されます。インポート プロセスが完了した後で、詳細を表示できます。

グループ、アラーム名、および優先度など、アラーム レコードのフィールド値を組み合わせて、アラーム レコードを固有に識別する「合成マッピング キー」を生成できます。

InTouch アラーム プロバイダは、アラームを生成したタグ変数の名前と、［アラーム名］フィールドを同じにします。したがって、Hot Backup ペアが指定されている場合、アラーム グループ名とタグ変数名の組み合わせを使用して、マッピング キーを生成できます。

以下に例を示します。

プロバイダ ノード	バックアップ ノード
$System!TagA	$System!TagB

プロバイダが固有フィールドとしてアラーム名フィールドとコメント フィールドを一緒に持つ場合、マッピング キーはアラーム名とコメントの組み合わせにすることができます。

プロバイダ ノード	バックアップ ノード
tagA!CommentA	tagB!CommentB

3 番目のプロバイダのその他のフィールドの組み合わせに対しても同じです。

アラーム レコード マッピング ファイルのインポート

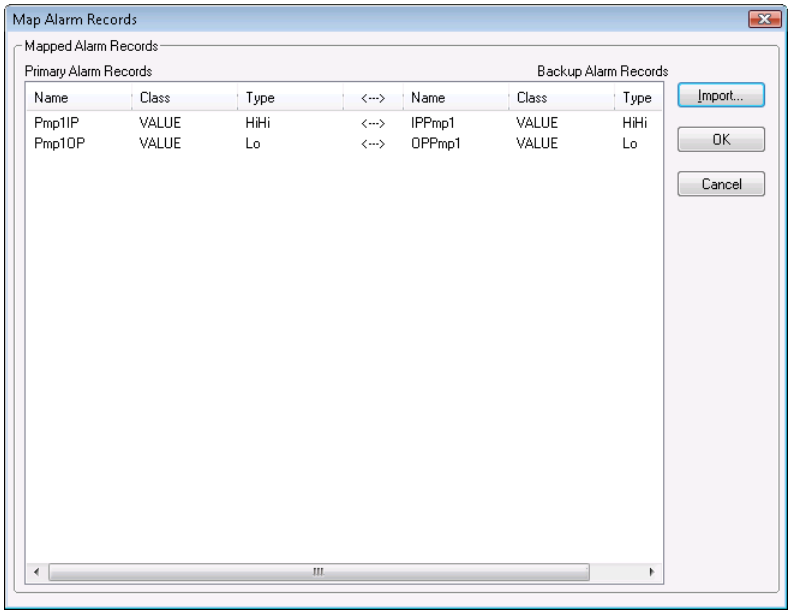
アラーム レコード マッピング ファイルの内容を Hot Backup Manager にインポートできます。

マッピング ファイルをインポートするとき、［アラーム クラス］フィールドと［アラーム タイプ］フィールドの間の検証は行われません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム レコードをマッピング ファイルからインポートする

1. Alarm Hot Backup Manager を開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm Hot Backup Manager**］をダブルクリックします。
2. リストから Hot Backup ペアを選択します。
3. **Map Alarms** を選択します。［アラーム レコードのマップ］ダイアログ ボックスが表示されます。



4. **Import** を選択します。［開く］ダイアログ ボックスが表示されます。マッピング ファイルを選択し、［開く］を選択します。

   Hot Backup Manager がファイルからのレコードのインポートを開始します。
5. すべてのマッピング レコードがインポートされたら、［**OK**］を選択します。
6. ［ファイル］メニューで、［保存］を選択します。
7. WindowMaker を再起動します。

アラーム マッピング ファイル インポート問題のトラブルシューティング

以下の状況では、ファイルのインポートが妨げられます。

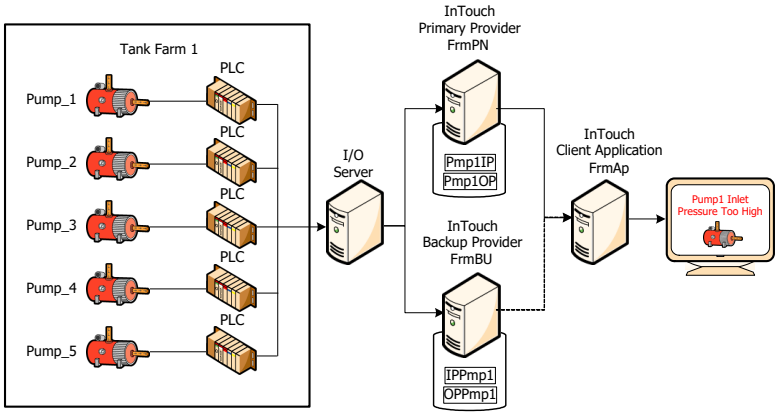
- 必要なカラム数に、インポート ファイルのすべてのレコードの値が入力されている必要があります。レコードに対して、値が多すぎても少なすぎてもいけません。
- インポート ファイルの見出しは、［キー フィールドの選択］ダイアログ ボックスと同じ見出しでなければならず、同じ順序でなければなりません。

インポートされるレコードに間違ったエントリがあると、その特定のレコード番号を省略するか、インポート プロセス自体を中止するよう求めるメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Hot Backup ペアの例

このセクションでは、アラーム バックアップ ペアを設定する一般的な作業シナリオについて説明します。以下の図は、InTouch タンク ファーム アプリケーションの Hot Backup ペアの設定を示しています。この例では、InTouch アプリケーションがアラーム状況としてポンプ圧力を監視しています。



3 台のコンピュータはすべて InTouch HMI を実行しています。Hot Backup ペアには、プライマリ プロバイダとして FrmPN、バックアップ プロバイダとして FrmBU が含まれています。これらの 2 つのノードは、InTouch 分散アラーム システム内でアラーム プロバイダとして機能します。

Hot Backup Manager は FrmAp ノードで実行します。InTouch クライアント アプリケーションは FrmAp で実行され、Hot Backup ペアの 2 つのプロバイダからアラームを取り込みます。

FrmPN を実行している InTouch アプリケーションは、ポンプ入口と出口の圧力用に 2 つのサマリ アラームを生成します。2 つのアラームは、TnkFrm1 アラーム グループに属します。FrmBU で実行している InTouch アプリケーションは、ポンプ圧力用に 2 つの論理的に等しいアラームを生成します。

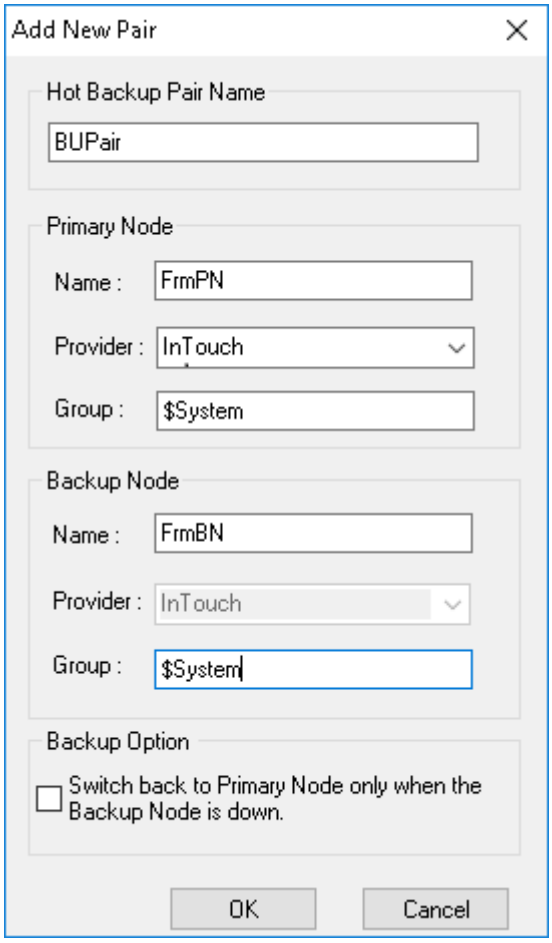
冗長な Hot Backup ペアを設定するときは、以下の操作を行います。

- Hot Backup ペアを作成します。
- アラーム レコードのキー フィールドを設定します。
- アラーム レコード マッピング ファイルを作成します。
- アラーム レコード マッピング ファイルをインポートします。

Hot Backup ペアを作成します。

1. WindowMaker で InTouch アプリケーションを開きます。この例では、アプリケーションは FrmAp ノードで実行します。
2. Alarm Hot Backup Manager を開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm Hot Backup Manager**］をダブルクリックします。
3. 新規ペアを選択します。［新規ペアの追加］ダイアログ ボックスが表示されます。
4. 次の図に示すように、［新規ペアの追加］ダイアログ ボックスのオプションを入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. ［**OK**］を選択して、［**Alarm Hot Backup Manager**］ダイアログ ボックスに戻ります。
6. ［**Alarm Hot Backup Manager**］ダイアログ ボックスが InTouch で表示されます。

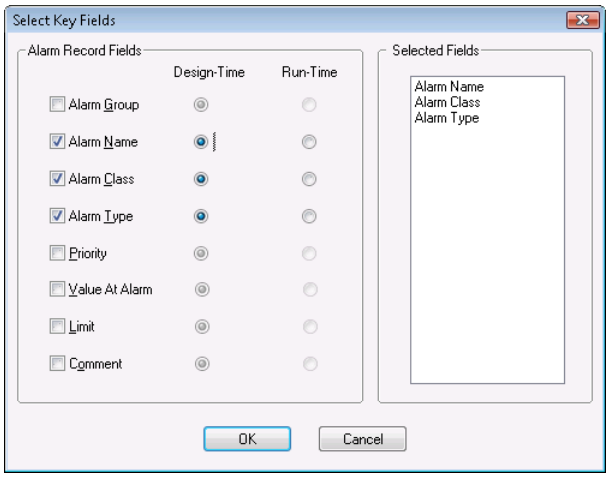
Hot Backup ペアを作成する最初の手順が完了しました。次に、以下の<1>手順に従って、各プロバイダのアラーム リポジトリに保存されているペアのアラーム レコードに固有のマッピング キーを生成します。</1>

アラーム レコードのキー フィールドをマップします。

1. リストで Hot Backup ペアを選択します。
2. **キーフィールドを設定** を選択します。［**キー フィールドの選択**］ダイアログ ボックスが表示されます。

   次の図に示すように、［**新規ペアの追加**］ダイアログ ボックスのオプションを入力します。プライマリ プロバイダとバックアップ プロバイダの間のアラームは論理的には一貫していますが、異なる名前が割り当てられており、異なるアラーム グループに属します。各プロバイダのアラーム リポジトリに保存されているレコードへの固有のマッピング キーは、［**デザイン時**］オプションとして、［**アラーム グループ**］、［**アラーム名**］、［**アラーム クラス**］、および［**アラーム タイプ**］を選択することによって生成できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［**OK**］を選択します。メッセージが表示されたら、［**はい**］を選択します。

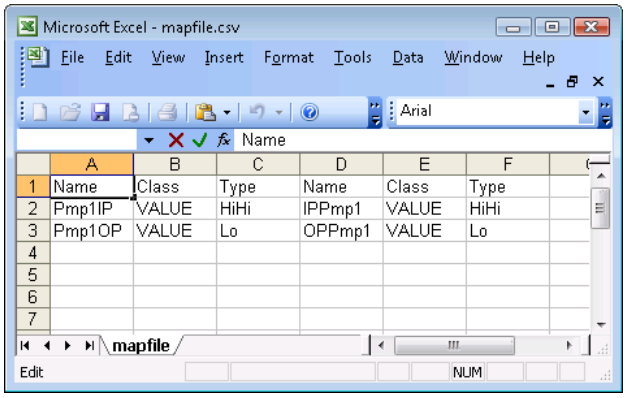
アラーム レコード マッピング キーを作成する 2 番目の手順が完了しました。

ここでは、3 つのノードすべてが InTouch アプリケーションを実行しています。2 つのプロバイダ ノードは同等のアラームを生成しますが、異なるタグ変数を使用しています。プライマリ プロバイダは、ポンプの入口と出口の圧力が高すぎる場合、2 つのサマリ アラームを生成します。バックアップ プロバイダは、同じポンプ圧力アラーム状況に対して 2 つの論理的に同一であるアラームを生成します。次に、以下の手順に従って<1>、各プロバイダのアラーム リポジトリに保存されている等しいレコードを関連付けるマッピング ファイルを作成します。</1>

アラーム マッピング ファイルを作成する

1. Excel またはメモ帳のようなテキスト エディタで .csv ファイルを作成します。
2. ファイル ヘッダの名前を［**キー フィールドの選択**］ダイアログ ボックスのアラーム レコード フィールド オプションと同じ順序で入力します。

   この例では、ファイル ヘッダをアラーム グループ、アラーム名、アラーム クラス、およびアラーム状況のタイプ別に順序付ける必要があります。
3. ファイルの各行で 2 つのプロバイダ間のアラームをマップします。
4. 以下の Excel ファイルの例では、ヘッダとアラーム状況を Hot Backup ペアの 2 つのプロバイダに対して指定する方法を示しています。マッピング ファイルを、クライアント ノードで実行している Hot Backup Manager にアクセス可能な場所に保存します。



アラーム レコード マッピング ファイルを作成する 3 番目の手順が完了しました。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

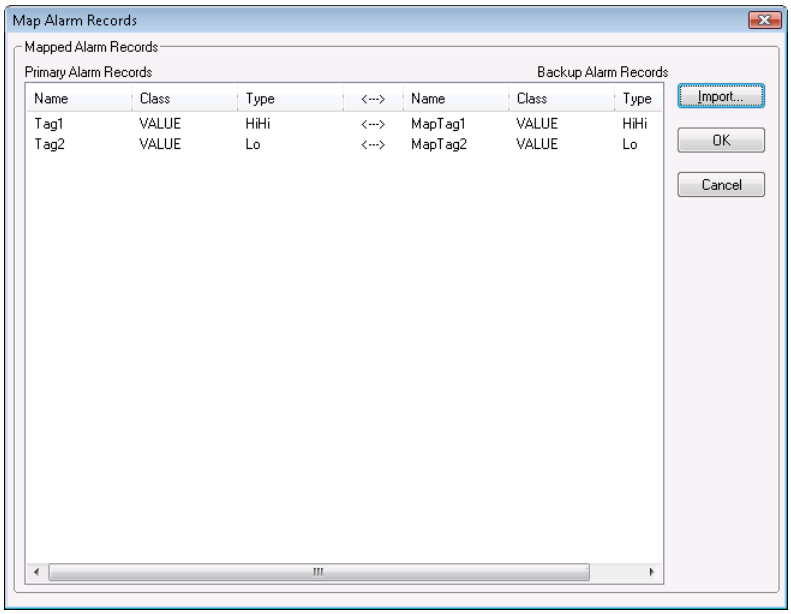
最後の手順で、アラーム マッピング ファイルの内容を Hot Backup Manager にインポートします。この例では、クライアント アプリケーションは 2 つのアラーム プロバイダ間でどのポンプ圧力アラーム レコードを確認するべきかを把握しています。

アラーム レコード マッピング .csv ファイルをインポートする

1. **Alarm Hot Backup Manager** を開きます。

   以前に作成したアラーム バックアップ ペアを一覧表示する必要があります。

2. **Map Alarms** を選択します。［**アラーム レコードのマップ**］ダイアログ ボックスが表示されます。
3. **Import** を選択します。［**ファイルを開く**］ダイアログ ボックスが表示されます。
4. マッピング ファイルを選択し、［**開く**］を選択します。［**アラーム レコードのマップ**］ダイアログ ボックスが開き、ファイルからのアラーム マッピング レコードが一覧表示されます。



5. ［**OK**］を選択します。Hot Backup Manager がファイルからのレコードのインポートを開始します。
6. すべてのマッピング レコードがインポートされたら、［**OK**］を選択します。

これで Hot Backup アプリケーションを実行できます。

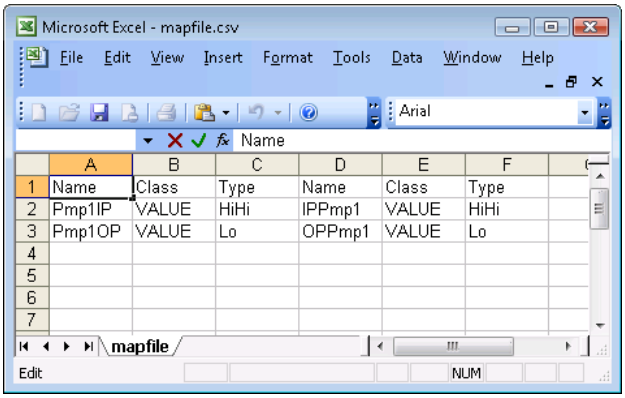
確認同期化の例

例：

- アラーム名、アラーム クラス、およびアラーム タイプを、デザイン時キー フィールドとして選択します。
- ランタイム フィールドとして、アラーム グループが選択されています。

以下のアラーム レコード マップ ファイルは、Microsoft Excel を使用して作成されました。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



Pmp1IP アラームは IPPmp1 アラームにマップされます。両方とも、VALUE のクラスおよび HiHi のタイプです。

Pmp1OP アラームは OPPmp1 アラームにマップされます。両方とも、VALUE のクラスおよび Lo のタイプです。

- Pmp1IP の HiHi アラームがプライマリ アラーム ノードで確認されるとき、セカンダリ プロバイダ ノードの IPPmp1 の HiHi アラームにも確認が表示されます。ただし、両方のノードのアラーム グループ名が同じままである必要があります。
- Pmp1OP の Lo アラームがプライマリ アラーム ノードで確認されるとき、セカンダリ プロバイダ ノードの OPPmp1 の Lo アラームにも確認が表示されます。ただし、両方のノードのアラーム グループ名が同じままである必要があります。

確認の同期化は、デザイン時とランタイムのマッピングが一致した場合にのみ発生します。

マッピング用に、デザイン時とランタイムのアラーム レコード フィールドの任意の組み合わせを選択できます。ただし、マッピングの結果が複数のリファレンスにならないことを確認してください。

たとえば、クラスと優先度など、2 つのアラーム レコード フィールドが選択されると、1 つ以上のアラームが条件に一致する可能性が高くなります。このような場合、Hot Backup の同期化が機能する保証はありません。確認を反映させるプロセスにおいて、条件に一致するランダムなアラームまでが確認される一方で、他の一致するアラームが未確認のままになることがあります。

Hot Backup ペアについての注記

- Hot Backup は、InTouch 7.11 以降のクライアントでのみ使用できます。
- 分散アラーム表示オブジェクトで Hot Backup ペアをクエリーして、その後で再びプライマリ プロバイダを別にクエリーすると、分散アラーム表示オブジェクトに重複したレコードが表示されます。
- プライマリ プロバイダのレコードが確認され、（確認が行われたときにダウン状態であった）セカンダリ プロバイダを後で起動した場合、確認されたレコードのタイムスタンプはプライマリ プロバイダのレコードと同一です。
- Hot Backup ペアをクエリーしているアラーム コンシューマでは、個別のノード名がプロバイダとして表示されます。
- マッピング用に、デザイン時とランタイムのアラーム レコード フィールドの任意の組み合わせを選択できます。
- ［値］と［しきい値］の各キー フィールドをマッピングするとき、これらの値は小数点第 4 位までに四捨五入され、マップされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- デザイン時とランタイムのマッピングの特定の組み合わせを持たないアラーム レコードでは、デフォルトのランタイム マッピングが使用されます。
- Hot Backup ユーティリティでは、完全に同じノード名で複数の Hot Backup ペアを作成できます。この機能は、Galaxy と Galaxy_<name> のプロバイダ タイプでのみ使用できます。

  例えば、ユーティリティにこの 2 つのペアを加えることができるようになった：

例 1

ペア名: Pair1

プライマリノード:Node1

プライマリプロバイダー:Galaxy

プライマリグループ:A01/

バックアップ ノード:Node2

バックアッププロバイダー:Galaxy

バックアップ グループ:A01

例 2

ペア名: Pair2

プライマリノード:Node1

プライマリプロバイダー:Galaxy

プライマリグループ:A02/

バックアップ ノード:Node2

バックアッププロバイダー:Galaxy

バックアップ グループ:A02

ユーティリティのエラー レポート機能が拡張され、この方法でペアを追加しようとする際に潜在的な問題に関するユーザー フィードバックが提供されるようになりました。

注： InTouch 以外のペアを使用する場合、以前のペアと同じノード名とグループ名を持つペアを入力することはできません。同じノード名を持つ 2 番目のペアを追加しようとすると、エラーが発生します。以下に例を示します。

ペア 1: ノード 1: n1 ノード 2: n2 グループ: a1

ペア 2: ノード 1: n1 ノード 2: n2 グループ: a1

アラーム監査記録の作成

InTouch アラーム プロバイダがオペレーティング システムまたは ArchestrA 認証のいずれかを使用するよう設定されている場合にアラームが発生すると、アラーム表示の［ユーザー フル ネーム］カラムには、ユーザーがログオンしていると想定して、ユーザーのフル ネームが含まれます。

たとえば、ユーザー ID が JohnS、フル ネームが John Smith というユーザーが PLANT_FLOOR ドメインに登録されている場合、［ユーザー フル ネーム］カラムには John Smith が含まれます。アラームが後で確認され、確認を実行するノードがオペレーティング システムまたは ArchestrA セキュリティを使用するよう設定されている場合、［ユーザー フル ネーム］カラムは更新され、確認ユーザーのフル ネームが表

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

示されます。それ以外の場合、アラーム表示にはコンピュータ名と $Operator タグの内容が連結されて表示されます。

InTouch セキュリティには、オペレータのフル ネームとアラーム確認を含めることができます。これは、アラーム検出に関するレコードでも可能です。ほとんどの組織では、ログイン ID はユーザーのフル ネームではなく、略語や役割分類です。

プロバイダやコンシューマ InTouch ノードに対してオペレーティング システム認証を設定する場合:

- アラームが生成され、確認が実行されると、アラーム表示でフル ネームが表示されます。
- アラームが生成され、確認が実行されると、アラーム プリンタでフル ネームが印刷されます。
- Alarm DB Logger では、各アラーム レコードの［Operator］フィールドと［AckOperator］フィールドにドメイン名、ログオン ユーザー ID、およびフル ユーザー名が記録されます。これにより、組織内に同じフル ネームの従業員が 2 人存在する場合でも、個別に識別できます。
- ［ユーザー］フィールドには、DomainName\UserName の形式でユーザー アカウントが表示されます。

認証を要求するアラームを確認する

一部の業種では、特定のタイプのアラームを確認する際に、明示的な「署名」またはユーザー認証を要求します。InTouch は、このタイプの認証操作を行うために、アラーム クライアント コントロールまたはスクリプト機能を利用します。アラームを確認するのに特定のユーザー署名を要求するかどうかを決定する主要な要素は、アラームの優先度です。

アラームは、ユーザーが確認するのに認証を要求するように設定することができます。認証を要求するようにアラームを設定するとは、ランタイム時に、アラームを確認するためユーザーが自分の資格情報を入力する必要があることを意味しています。システムがスマート カード リーダーを使用するように設定されている場合には、ユーザーはスマート カードの資格情報を入力することができます。

このセクションでは、認証を要求する WindowViewer で、アラームを確認する方法について説明します。

アラームの確認に認証を要求するようアラーム クライアントを設定する方法の詳細については、アラーム クライアント コントロール文書を参照してください。

SignedAlarmAck() 関数を使用して、アラームを確認するための認証の動作を設定する方法の詳細については、『*産業用グラフィック エディタ ユーザー 文書*』の「グラフィック スクリプトの追加と維持」の章を参照してください。

認証を要求する WindowViewer のアラームの確認について

認証を要求するよう設定されたアラームの確認には、次の事柄が必要です。

- Galaxy に対してセキュリティが有効にされている必要があります。
- 配置された InTouchViewApp に対してセキュリティが有効にされている必要があります。
- InTouchViewApp は、少なくとも 1 つ、アラームとイベントを表示し、アラームの確認のために署名（認証）を要求するように設定され、アラームの優先度範囲の最小値と最大値が設定されているアラーム コントロールを含んでいる必要があります。
- ランタイム オペレータは Galaxy 内で設定された適切なユーザー名、パスワード、および権限を持っている必要があります。オペレーターは、WindowViewer を実行するのに十分な権限を持っている必要があります。

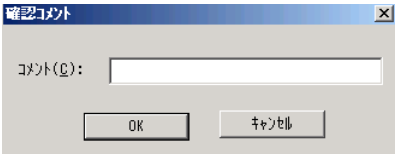
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

現在誰が InTouch アプリケーションのランタイム ユーザーとしてログインしているかには関わりなく、アラームの確認のために署名を要求するよう設定されたアラームでは、常にユーザー認証が必要とされます。この操作はログオンしているユーザーのセッションには影響を与えません。

アラーム確認のランタイム時の基本的な動作

オペレータが 1 つか複数のアラームを選択して、それらを確認しようと試みます。

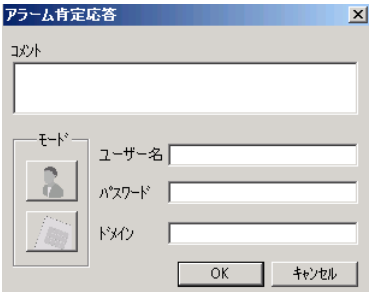
- アラーム クライアントは、アラームが確認署名を要求するように設定されているかどうかをチェックします。設定されていた場合には、選択されたアラームのうち、設定された優先度範囲の中に入っているものがあるかどうかをチェックします。入っていたものがある場合には、選択されたアラームすべてを確認するための署名を要求します。
- 署名が要求されていない場合には、ポップアップの確認ダイアログが表示されます。そのように設定されていた場合には、確認コメント用のテキスト ボックスが表示されます。



- オペレータのアカウント情報が有効であれば、アラーム クライアントは、選択されたアラームすべてを確認しようと試みます。
- アラーム クライアント グリッドでは、証明を行ったユーザーは、選択されたアラームすべてを確認した人物として識別されます。
- アラーム クライアントは、それが署名された確認であることを示すため、確認コメントの先頭に「署名された確認」というプレフィックスを付けます。
- 選択されたアラームのうち、その優先度が要求されている範囲に入っているものがなかった場合には、アラーム クライアントは、それが標準の確認であることを示すため、確認コメントの先頭に「標準の確認」というプレフィックスを付けます。

認証を要求するアラームを確認する

1. WindowViewer で、1 つまたは複数の未確認のアラームを選択します。右選択して、実行する確認操作を選択します。いずれかのアラームが署名を要求するように設定されていた場合には、[**確認アラーム**] ダイアログ ボックスが表示されます。



   スマート カードが有効でなった場合には、スマート カードまたはパスワード認証の一方を選択するための [**モード**] ボタンは無効になります。

2. コメントを入力するように設定されていた場合には、[**コメント**] ボックスにコメントを入力します。
3. ネットワーク ユーザー アカウントを使用して認証する場合には、ユーザー アカウントのオプションが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以下の手順を実行します。

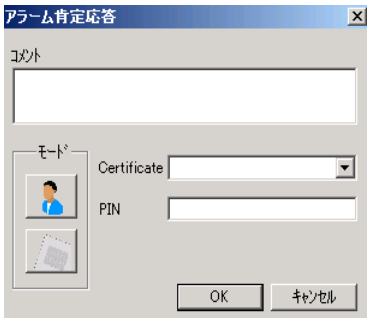
a. ［**ユーザー名**］ボックスに、ユーザー名を入力します。デフォルトでは現在ログオンしているユーザーの名前が表示されます。現在ログオンしているユーザーがいない場合、空白のボックスが表示されます。

b. ［**パスワード**］ボックスに、ユーザー名に関連付けられているパスワードを入力します。

c. ［**ドメイン**］ボックスに、ドメイン名を入力します。セキュリティ モードが ArchestrA Galaxy になっている場合には、表示されるドメイン名は ArchestrA になります。変更することはできません。

d. ［**OK**］を選択します。

4. スマート カードを使用して認証する場合には、スマート カードの［**確認アラーム**］ダイアログ ボックスが表示されます。



スマート カードを使用して認証するには、以下の操作を行います。

a. ［**証明書**］リストで、自分のスマート カード証明書を選択します。証明書のリストは、ドメイン名／ユーザー名の形式で表示されます。証明書のリストが表示されるには、その時点でコンピュータに接続されているリーダーにスマート カードが挿入されている必要があります。デフォルトでは現在ログオンしているユーザーの証明書が表示されます。［**セキュリティ書き込み**］ダイアログ ボックスが開いているときにカードの抜き差しを行った場合、証明書のリストは自動的に更新されます。

b. ［**PIN**］ボックスに、スマート カードの PIN を入力します。

c. ［**OK**］を選択します。

代わりに名前、パスワード、およびドメイン名で認証する場合には、［モード］ボタンを選択します。手順 3 に移動します。

分散アラーム表示オブジェクトの操作

InTouch HMI のこのバージョンには、分散アラーム表示オブジェクトが含まれており、InTouch バージョン 7 以前で開発されたアプリケーションをサポートします。分散アラーム表示オブジェクトは従来のオブジェクトなので、InTouch HMI のバージョン 7 より新しいバージョンでアラーム表示を行う場合は、この代わりに Alarm Viewer コントロールを使用する必要があります。

InTouch HMI のこのバージョンには、分散アラーム表示オブジェクトが含まれており、InTouch バージョン 7 以前で開発されたアプリケーションをサポートします。分散アラーム表示オブジェクトは従来のオブジェクトなので、InTouch HMI のバージョン 7 より新しいバージョンでアラーム表示を行う場合は、この代わりに Alarm Viewer コントロールを使用する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

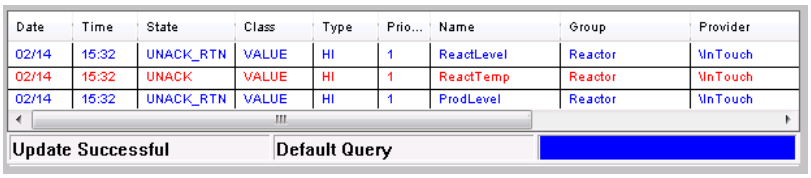
分散アラーム表示オブジェクトの操作

InTouch HMI のこのバージョンには、分散アラーム表示オブジェクトが含まれており、InTouch バージョン 7 以前で開発されたアプリケーションをサポートします。分散アラーム表示オブジェクトは従来のオブジェクトなので、InTouch HMI のバージョン 7 より新しいバージョンでアラーム表示を行う場合は、この代わりに Alarm Viewer コントロールを使用する必要があります。

InTouch HMI のこのバージョンには、分散アラーム表示オブジェクトが含まれており、InTouch バージョン 7 以前で開発されたアプリケーションをサポートします。分散アラーム表示オブジェクトは従来のオブジェクトなので、InTouch HMI のバージョン 7 より新しいバージョンでアラーム表示を行う場合は、この代わりに Alarm Viewer コントロールを使用する必要があります。

分散アラーム表示オブジェクトについて

分散アラーム表示オブジェクトには、ローカル アラームとリモート アラームの両方を表示する単一表示の機能が提供されています。



この表示オブジェク機能には、組み込みスクロール バー、サイズ変更可能なカラム、複数のアラーム選択、ステータス バー、ショートカット メニュー、およびアラーム優先度と状況に基づく色が含まれています。

分散アラーム表示オブジェクトには、アラーム表示の外観（表示される情報を含む）、さまざまなアラーム条件に使用される色、表示するアラーム グループとアラーム優先度レベルを設定できるプロパティが含まれています。

表示オブジェクトの詳細については、「ランタイムでの分散アラーム表示オブジェクトの使用」を参照してください。

分散アラーム表示オブジェクトのガイドライン

分散アラーム表示オブジェクトを使用するときは、以下のガイドラインを確認してください。

- 関連付けられている QuickScript 関数が変更する表示を識別できるように、各表示には識別子が必要です。この識別子は、［**アラーム設定**］ダイアログ ボックスの［**表示名**］ボックスに入力しますが、各表示に対して固有である必要があります。
- ウィンドウ コントロール オブジェクトやグラフィック オブジェクトなど、他の InTouch オブジェクトと表示が重ならないようにする必要があります。WindowMaker で分散アラーム表示オブジェクトを選択し、表示の「ハンドル」を確認することによって、簡単に確認できます。ハンドルは画面の他のオブジェクトと接触できません。
- 表示は慎重に使用する必要があります。1 つの画面に多数の表示を配置すると、システムのパフォーマンスが低下する可能性があります。可能な限り、ウィンドウの表示数を制限し、追加の表示に対しては新しいウィンドウを呼び出します。

デザイン時の分散アラーム表示オブジェクトの設定

以下を設定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ステータス バー、スクロール バーなどの一般機能
- カラムとソート順
- アラーム レコードを取得するために使用するアラーム クエリー
- アラーム レコードが表示される時間形式
- 表示されるアラーム レコードのフォントと色
- カラムのサイズ変更、アラームの選択、ショートカット メニューへのアクセスなどランタイム ユーザーが表示で実行できること

分散アラーム表示オブジェクトを作成する

ウィザードの場合と同じように、分散アラーム表示オブジェクトを作成します。

分散アラーム表示オブジェクトを作成する

1. ［**描画**］メニューの［**挿入**］グループで［**ウィザード**］を選択します。

   ［**ウィザードの選択**］ダイアログ ボックスが表示されます。
2. ウィザードのリストで、［**アラームの表示**］を選択します。
3. **Dist. Alarm Display** ウィザードをダブルクリックします。ダイアログ ボックスが閉じて、カーソルが「貼り付け」モードの状態のウィンドウが再表示されます。
4. 分散アラーム表示オブジェクトを貼り付ける位置をウィンドウ内で選択します。ウィザードのサイズを変更するには、選択ハンドルをドラッグします。

これで、表示を設定する準備が整いました。

グリッドの表示プロパティの設定

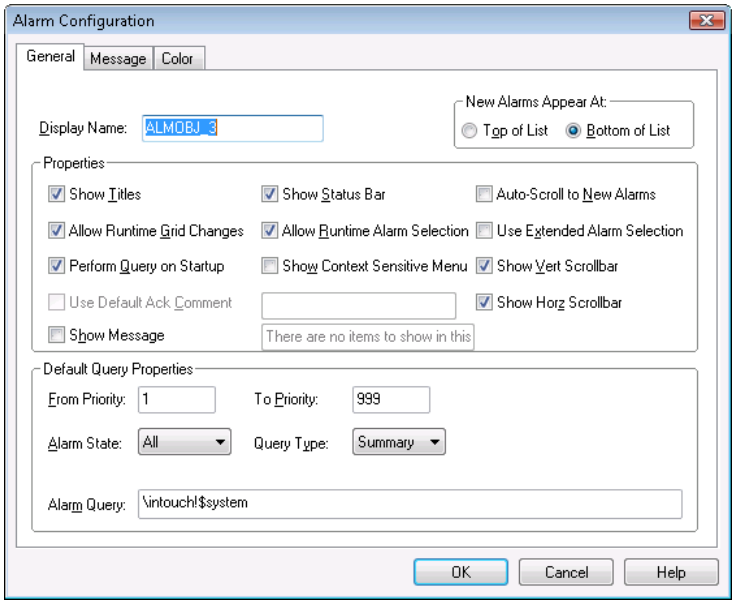
表示グリッドの場合は、以下の内容を設定できます。

- タイトル バー、ステータス バー、スクロール バー
- 新しいアラームがグリッドに表示される場所、自動スクロールの有無
- 表示するアラーム レコードがない場合に表示されるデフォルト メッセージ

グリッドの外観を設定する

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［表示名］ボックスに、アラーム表示の名前を入力します。この名前は、使用される各アラーム表示に対して固有である必要があります。この名前はアラーム確認やクエリーなど、このオブジェクトを参照するタスクの実行に対して、システムを通じて使用されます。
3. ［新規アラームの表示箇所］領域で、オブジェクト内で新しいアラームを表示する場所を設定します。
   - 最新のアラームをリストの一番上に表示するには、［リストの一番上］を選択します。
   - 最新のアラームをリストの一番下に表示するには、［リストの一番下］を選択します。
4. ［プロパティ］領域で、タイトル バー、ステータス バー、およびスクロール バーを設定します。以下のいずれかを実行します。
   - アラーム メッセージ タイトル バーを表示するには、［タイトルの表示］チェック ボックスをオンにします。
   - ステータス バーを表示するには、［ステータス バーの表示］チェック クスをオンにします。
   - 垂直スクロール バーを表示するには、［垂直スクロールバーの表示］チェック ボックスをオンにします。
   - 水平スクロール バーを表示するには、［水平スクロールバーの表示］チェック ボックスをオンにします。
5. 表示するアラーム レコードがない場合、デフォルト メッセージを表示するには、［メッセージの表示］チェックボックスをオンにします。ボックスにメッセージを入力します。
6. 選択を自動的に新規アラームに移動させるには、［新規アラームに自動スクロール］チェック ボックスをオンにします。新しいアラームは、表示オブジェクト内に現在表示されていないアラームです。
7. ［OK］を選択します。

ランタイムでユーザーがアクセスできる機能の管理

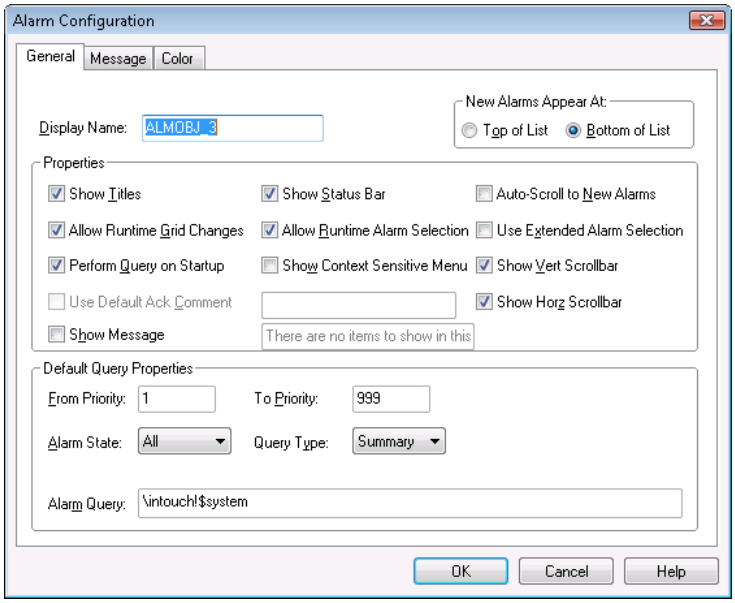
ランタイム ユーザーが、カラム設定を変更し、アラームを選択し、ショートカット メニューを開けるようにできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注：スクリプト関数を使用して、ショートカット メニューに表示されるコマンドを実行できます。

ランタイム機能を設定する

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。



2. ランタイムで利用できるオプションを設定します。以下のいずれかを実行します。
   - ユーザーがカラム設定を変更できるようにするには、［**実行中のグリッド変更可能**］チェック ボックスをオンにします。
   - ショートカット メニューを有効にするには、［**コンテキスト メニューを表示**］チェック ボックスをオンにします。
   - ユーザーがアラームを選択できるようにするには、［**実行中にアラームの選択可能**］チェック ボックスをオンにします。
   - Ctrl キーまたは Shift キーを押さえたままマウスで選択することによって、ユーザーが複数のアラームを選択できるようにするには、［**拡張アラーム選択の使用**］チェック ボックスをオンにします。デフォルトでは、アラームを簡単に選択して選択を切り替えます。
3. ［**OK**］を選択します。

表示するアラームを設定する

分散アラーム表示オブジェクトを設定し、以下の内容に基づいてアラームを表示できます。

- 優先度
- 確認済み、未確認などの状況
- サマリまたは履歴などのタイプ

  アラーム クエリーを設定するときは、テキストのみを使用します。タグ変数は使用できません。クエリーの例は以下のとおりです。

  アラーム グループへのフル パス：

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

\\ノード\InTouch!Group

ローカル アラーム グループへのフル パス

\InTouch!Group

別のグループ リスト：

GroupList

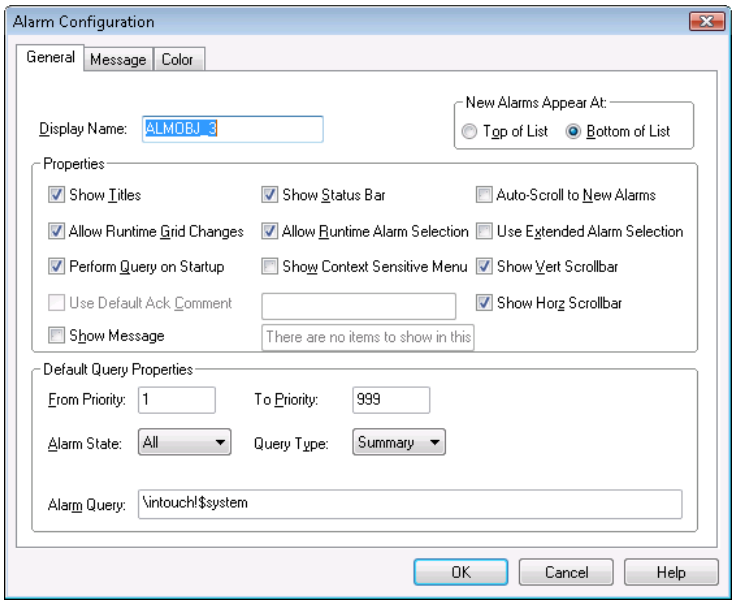
複数のクエリーを実行するには、各クエリーをスペースで区切ります。以下に例を示します。

\InTouch!Group GroupList

デフォルトのクエリー プロパティは、［**起動時にクエリー実行**］チェック ボックスをオンにした場合、または almDefQuery() 関数が実行される場合のみ使用できます。

表示するアラームを設定する

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。



2. ［**起動時にクエリー実行**］チェック ボックスをオンにします。
3. ［**クエリー属性のデフォルト**］領域で、オブジェクトのデフォルト クエリーを設定します。
   - ［**優先度始点**］ボックスに、デフォルトの最小アラーム優先度を入力します。
   - ［**優先度始点**］ボックスに、デフォルトの最大アラーム優先度を入力します。
   - ［**アラーム状況**］リストで、クエリーするデフォルトのアラーム状況を選択します（全て、未確認、確認）。
   - ［**クエリー タイプ**］リストで、［サマリ］または［履歴］のいずれかのアラーム タイプを選択します。
   - ［**アラーム クエリー**］ボックスに、最初のアラーム クエリーを入力します。
4. ［**OK**］を選択します。

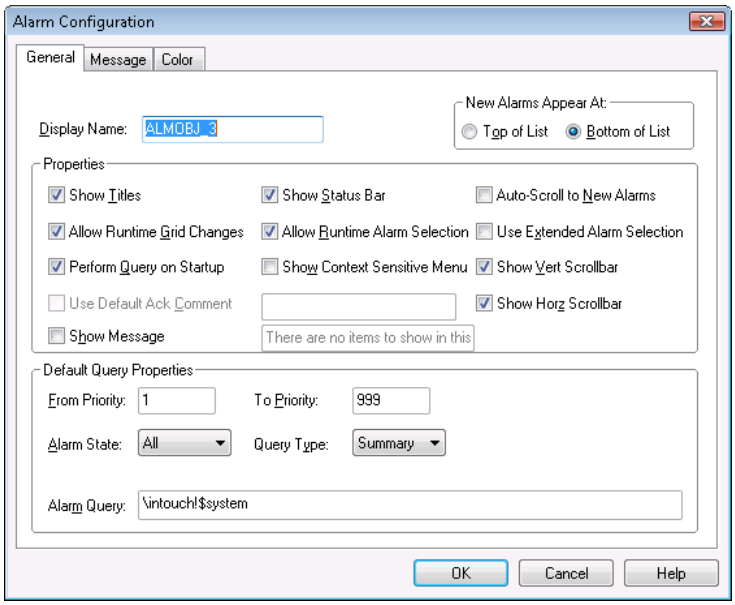
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

デフォルトのアラーム コメントの設定

オペレータがアラームを確認するときに使用されるデフォルトのコメントを設定できます。デフォルトのコメントを設定しない場合、オペレータがアラームを確認するときに、オペレータがコメントを入力できるダイアログ ボックスが表示されます。このダイアログ ボックスはテキストを入力することも、空白のまま残すこともできます。

デフォルトのコメントを設定する

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。



2. ［**コンテキスト メニューを表示**］チェック ボックスをオンにします。
3. ［**デフォルトの確認コメントを使用**］チェック ボックスをオンにし、ボックスにコメント テキストを入力します。
4. ［**OK**］を選択します。

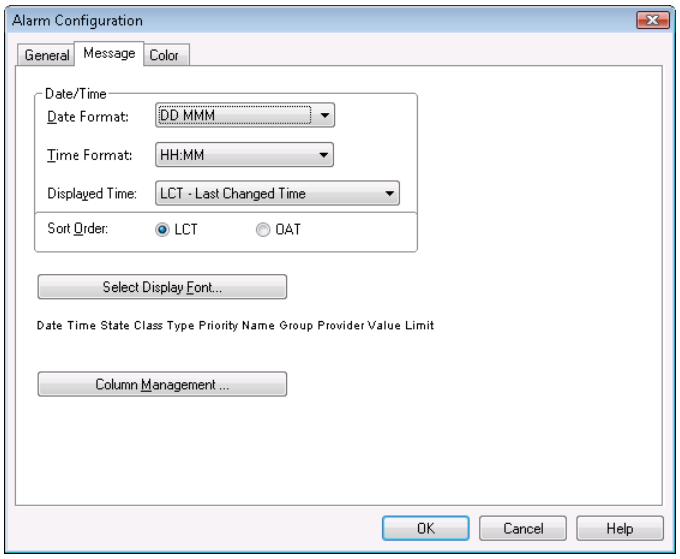
アラーム レコードの時間形式の設定

オリジナルのアラーム時刻です。アラームが開始した日付／時間スタンプです。タグが I/O 型タグであり、そのサーバーがタイムスタンプを渡すことができる場合は、I/O サーバーからのタイムスタンプです。

アラーム表示時間形式を設定する

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。
2. **メッセージ** タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［日付］リストで、日付の形式を選択します。使用できる形式は以下のとおりです。

選択	内容	選択	内容
DD MMM	28 2	**MM/DD**	02/28
DD MMM YYYY	28 2 2007	**MM/DD/YY**	02/28/07
DD/MM	28/07	**MMM DD**	2 28
DD/MM/YY	28/02/07	**MMM DD YYYY**	2 28 2007
YY/MM/DD	07/02/28	**YYYY/MM/DD**	2007/02/28

4. ［時間］リストで、時間の形式を選択します。このリストの値は、時間の形式を指定するためのテンプレートとして使用します。たとえば、時間を 10:24:30 AM と指定するには、HH:MM:SS AP を使用します。テンプレートの文字は以下のとおりです。

文字	説明
AP	AM／PM 形式を選択します。たとえば、午後 3 時は 3:00 PM と表示されます。 AP が指定されていない場合、時間はデフォルトで 24 時間制で表示されます。たとえば、午後 3 時は 15:00 と表示されます。
HH	アラーム／イベントが発生した時間を示します。
MM	アラーム／イベントが発生した分を表示します。
SS	アラーム／イベントが発生した秒を表示します。
SSS	アラーム／イベントが発生したミリ秒を表示します。

5. ［ソート順］領域で、アラームをオブジェクトに保存する順序を設定します。
   - ［**OAT**］を選択して、アラーム発生の日付／時間スタンプである元のアラーム時間を使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **LCT** を選択して、最後のアラーム変更時間を使用します。これはアラームのインスタンスに対して発生した最新の状態変化（アラームの開始、サブステートの変化、平常への復帰、または確認）の日付／時間スタンプです。

6. ［**OK**］を選択します。

アラーム レコードのフォントの設定

コントロール内のレコードと見出しのフォントを設定します。

フォントのプロパティを設定する

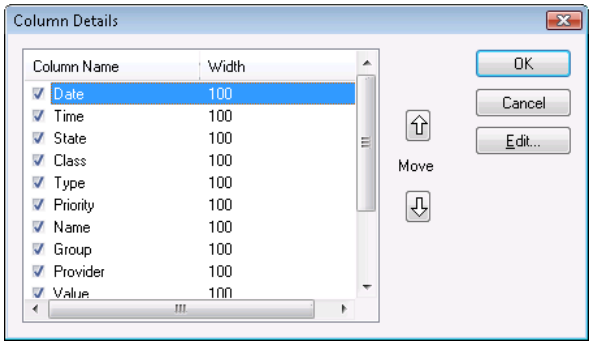
1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。
2. **メッセージ** タブを選択します。
3. ［**表示フォントの選択**］を選択します。Windows 標準の［**フォント**］ダイアログ ボックスが表示されます。
4. フォントを設定して、［**OK**］を選択します。

アラーム レコードのカラムの設定

表示するカラムを選択し、カラム順を指定し、カラムの名前と幅を設定できます。

表示カラムの詳細を設定する

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。
2. [**メッセージ**] タブを選択します。
3. **カラムの管理**を選択します。［**カラムの詳細**］ダイアログ ボックスが表示されます。



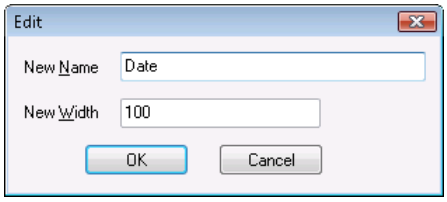
4. 分散アラーム表示オブジェクトのカラムを表示するには、［**カラム名**］リストのチェック ボックスをオンにします。少なくとも 1 つのカラムを選択する必要があります。以下の表は、カラムについて説明しています。

カラム	説明
日付	［**メッセージ**］タブで、選択した形式の日付を表示します。
Time	［**メッセージ**］タブで、選択した形式の時間を表示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カラム	説明
状態	アラームの状態を表示します。
クラス	アラームのカテゴリを表示します。
Type	アラーム タイプを表示します。
優先度	アラーム優先度を表示します。
名前	アラーム／タグ変数を表示します。
グループ	アラームのグループ名を表示します。
プロバイダ	アラーム プロバイダ名を表示します。
値	アラームが発生した際のタグ変数値を表示します。
しきい値	タグ変数のアラームしきい値を表示します。
Operator	アラーム状況に関連付けられているログオンしたオペレータの ID を表示します。
コメント	タグ変数のコメントを表示します。このコメントは、データベースでタグ変数のアラームが定義されたときに［**アラーム コメント**］ボックスに入力されたものです。

5. カラムの順序を移動するには、カラム名を選択してから［移動］矢印ボタンを使用してください。［**カラムの詳細**］ダイアログ ボックスの一番上にあるカラム名が、アラーム表示で一番左に表示されます。
6. カラム名と幅を編集するには、カラム名を選択してから［**編集**］を選択します。選択したカラムの［**編集**］ダイアログ ボックスが表示されます。



   a. デフォルトのカラム名以外にカラム名を表示するには、［**新しい名前**］ボックスに新しい名前を入力します。
   b. ［**新しい幅**］ボックスに、デフォルトのカラム幅を入力します。カラム幅は、1 ～ 999 ピクセルの範囲で設定できます。デフォルトのカラム幅は 100 ピクセルです。
   c. ［**編集**］ダイアログ ボックスの［**OK**］を選択します。
7. ［**カラムの詳細**］ダイアログ ボックスの［**OK**］を選択します。
8. ［**適用**］を選択します。

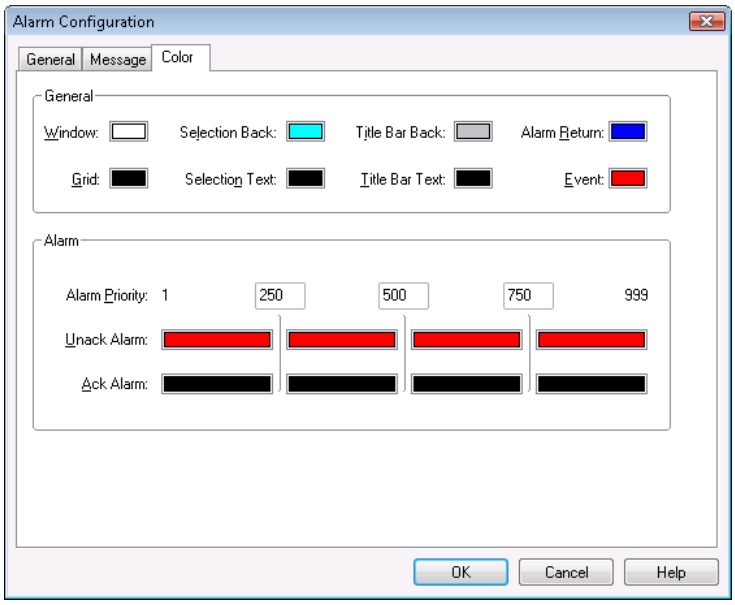
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム レコードの色の設定

背景色や選択したレコードの色など分散アラーム表示オブジェクトのさまざまな部分の色を設定できます。さまざまな範囲のアラーム レコードのさまざまな色も設定できます。

アラーム表示の色を設定する

1. 分散アラーム表示オブジェクトを右選択し、[**プロパティ**] を選択します。[**アラーム設定**] ダイアログ ボックスが表示されます。
2. **色**タブを選択します。



3. [**表示項目**] 領域で、それぞれの色のボックスを選択し、InTouch カラー パレットを開きます。次のそれぞれに使用する色を選択します。

オプション	説明
ウィンドウ	表示背景色を設定します。
グリッド	グリッドの色を設定します。
背景の選択	反転表示されたテキスト背景色を設定します。
テキストの選択	反転表示されたテキストの色を設定します。
タイトル バーの背景	タイトル バーの背景色を設定します ([**タイトルの表示**] オプションをオンにした場合にのみ表示されます)。
タイトル バーのテキスト	タイトル バー テキストの色を設定します([**タイトルの表示**] オプションをオンにした場合にのみ表示されます)。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オプション	説明
アラーム復帰	平常状態に復帰したアラーム（確認されずに平常に戻ったアラーム）の色を設定します。
イベント	イベント アラームの色を設定します。

4. ［**アラーム優先度**］ボックスに、アラーム表示の境界値を設定します。
5. ［**未確認アラーム**］と［**確認アラーム**］の色ボックスを選択して、InTouch パレットを開きます。使用するパレットの色を選択します。
6. ［**OK**］を選択します。

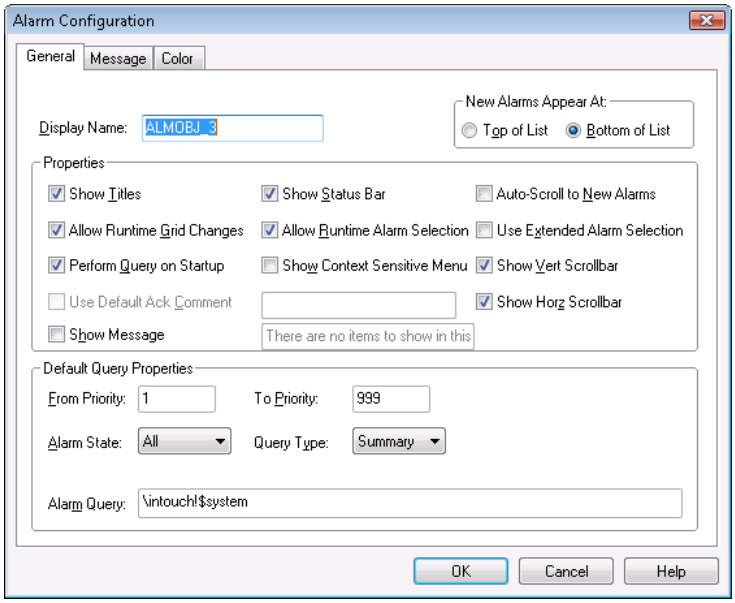
表示タイプの設定

分散アラーム表示オブジェクトには、アクティブなアラームのサマリまたは履歴アラームのリストを表示できます。

表示タイプはランタイムで動的に変更することもできます。これを行うには、たとえば、almQuery() スクリプト関数を実行します。almQuery() スクリプト関数では、指定した分散アラーム表示オブジェクト (例: "AlmObj_1") を指定された表示タイプ (例: "Summary") に設定するために使用できるパラメータを使用します。詳細については、「almQuery() 関数」を参照してください。

クエリーのデフォルトを設定する

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。



2. ［**クエリー タイプ**］リストで、ランタイム デフォルトに使用するアラーム表示のタイプを選択します。
3. ［**OK**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ローカル アラームを監視するための分散表示の使用

分散アラーム表示オブジェクトを使用すると、ローカル アラームとリモート アラームの両方の表示と確認を行うことができます。

ローカル アラームだけを監視するように表示を設定するには

1. 分散アラーム表示オブジェクトを右選択し、［**プロパティ**］を選択します。［**アラーム設定**］ダイアログ ボックスが表示されます。
2. ［**アラーム クエリー**］ボックスに、`\InTouch!$System` を入力します。

   任意の有効なアラーム グループを $System の代わりに使用できます。\InTouch!$System のみを含むアラーム グループ リストを定義し、直接リファレンスの代わりにこのグループ リストを使用することもできます。
3. 表示タイプの他のパラメータを設定し、アプリケーションが必要とするフィルタを設定します。

ランタイムでの分散アラーム表示オブジェクトの使用

分散アラーム表示オブジェクトでは、グリッドを使用して、アラーム メッセージを表示します。このグリッドを使用すると、カラム ハンドルを選択し、必要な幅にドラッグするだけで簡単にカラムの幅のサイズを変更できます。この機能はランタイム中でのみ使用できます。

グリッドのカラム変更は保存されないため、グリッド カラム変更を行い、アラーム表示が含まれているウィンドウを閉じると、再びそのウィンドウを開いたときにはグリッド カラムはデフォルトのカラム幅に戻ります。WindowMaker でデフォルトのカラム幅を調整できます。

グリッドを使用すると、リスト ボックス内の 1 つまたは複数のアラームを選択できます。選択したアラームは、almAckSelect() QuickScript 関数を使用して確認できます。分散アラーム表示オブジェクトを設定するときは、選択動作を定義して、切り替え選択（アイテムごと）または複数選択（Ctrl キーまたは Shift キーを押したまま、マウス クリックを使用して複数のアラームを選択）のいずれかを許可できます。ランタイム中の選択はオフに設定できます。

表示される各アラーム メッセージの色は、アラームの優先度や確認の状態に応じて、最大 8 色の中から選択できます。

設定時にオンにした場合は、スクロール バーを利用できます。

アラーム表示オブジェクトには、コントロールが設定される方法に基づいて、アラーム レコードを検索するコントロールがあります。

サイズ変更可能な表示カラム

分散アラーム表示オブジェクトはグリッドを使用して、アラーム メッセージを保持します。ランタイムで、カラムを選択し、ドラッグするだけで簡単にカラムの幅のサイズを動的に変更できます。また、縦のグリッド線をダブルクリックして、カラムのサイズを自動的に変更できます。この機能はランタイム中でのみ使用できます。カラムの幅のサイズ変更は、設定時にオンにする必要があります。

アラーム表示が含まれるウィンドウを閉じるとき、グリッド カラムの変更は保存されません。

複数の選択

アラーム表示が設定される方法に基づいて、リスト ボックスの 1 つまたは複数のアラームを選択できます。

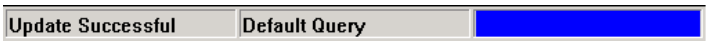
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム メッセージの色

アラーム メッセージの色は、アラームの優先度や確認の状態に応じて、最大 8 色の中から選択できます。

ステータス バー

アラーム オブジェクトが設定される方法に基づいて、ステータス バーには、ステータス メッセージ、現在のアラーム クエリー、および進行状況バーが表示されます。



ステータス バーの左部分には、コントロールの現在のステータスが表示されます。

これらのインジケータでは、表示クエリーの現在の状況の概要と分散アラーム表示オブジェクトで使用できる非表示機能についての詳細が提供されます。「停止」に設定されている場合、ステータス バーの右ペインが赤となります。「非表示」に設定されている場合、ステータス バーの左ペインが赤となります。

「非表示」に設定されている場合、「非表示中」という単語が左ペインに表示されます。

機能	説明
ステータス メッセージ	ステータス バーの左端にあるステータス メッセージには、現在のクエリー ステータスの詳細な説明が表示されます。
アラーム クエリー	アラーム クエリーでは、現在のアラーム クエリーが視覚的に表示されます。
進行状況バー	ステータス バーの右端にある進行状況バーには、現在のクエリーの進行状況が視覚的に表示されます。

ステータス メッセージは以下のとおりです。

ステータス メッセージ	状況／インジケータ	進行状況バー
なし	クエリーなし	なし
更新中	クエリー中	青／緑
更新が完了しました	クエリー終了	赤
非表示	クエリー名	青一色
停止	クエリー名	赤

ショートカット メニュー

アラーム オブジェクトが設定される方法に基づいて、オブジェクトを右クリックし、共通コマンドのショートカット メニューを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

このコマンドを選択します。	目的
選択アラームを確認	選択したアラームを確認します。
その他を確認	表示内のすべてのアラームを確認するか、表示されているアラーム、選択したグループ、選択したタグ名、および優先度のみを確認します。
選択項目を非表示	選択したアラームを非表示にします。
その他を非表示	表示内のすべてのアラームを非表示にするか、表示されているアラーム、選択したグループ、選択したタグ名、および選択した優先度のみを非表示にします。
クエリー設定	［**アラーム クエリー**］ダイアログ ボックスを開き、使用するクエリーを選択できます。
統計	［**アラームの統計**］ダイアログ ボックスを開きます。
非表示	［**アラーム非表示**］ダイアログ ボックスを開きます。
停止	現在の表示を停止します。
選択項目を閉じる	選択したアラームを解除します。
その他を閉じる	表示内のすべてのアラームを解除するか、表示されているアラーム、選択したグループ、選択したタグ名、および選択した優先度のみを解除します。

アラーム クエリーの選択と設定

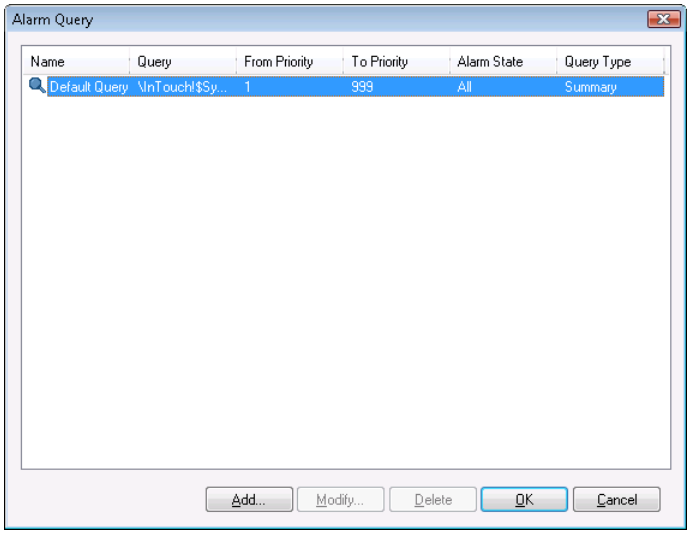
ショートカット メニューの［**クエリー設定**］コマンドをクリックし、以前定義されたアラーム クエリーのリストからアラーム クエリーを素早く選択します。また、新しい名前付きクエリーを作成したり、既存のクエリーを編集したり、既存のクエリーを削除したりすることもできます。

注 : 分散アラーム表示に複数行アラーム クエリーが表示される場合、改行部分が文字化けします。ただし、機能には影響ありません。

表示用のアラーム クエリーを選択する

1. ランタイムで、分散アラーム表示を右クリックし、［**クエリー設定**］をクリックします。［**アラーム クエリー**］ダイアログ ボックスが表示されます。

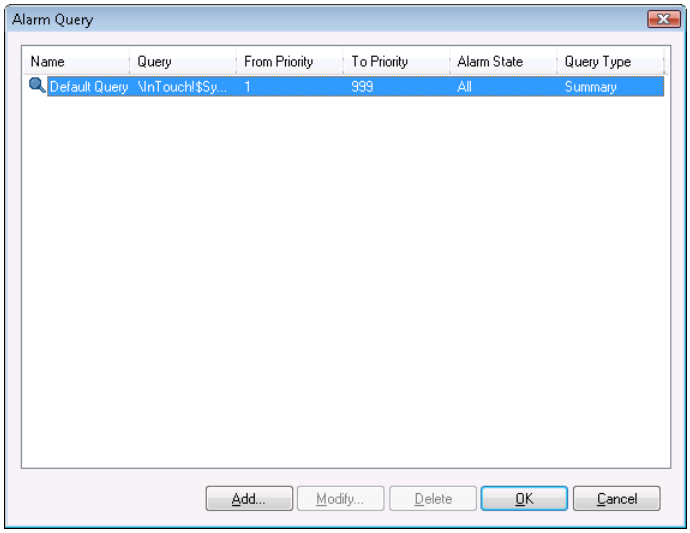
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. 現在定義されているクエリーのリストで、使用する名前付きクエリーを選択します。
3. ［**OK**］を選択します。分散アラーム表示オブジェクトには、選択したクエリーに関するアラーム情報が表示されます。

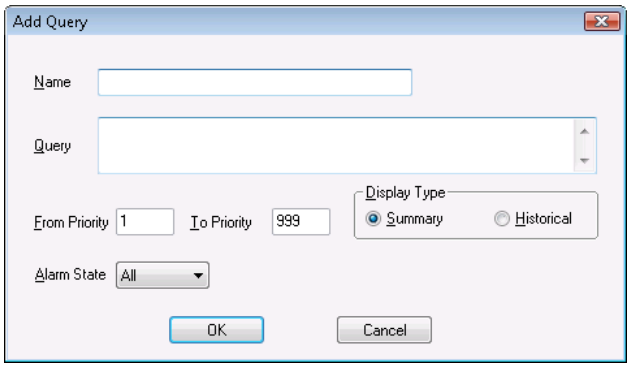
新しい名前付きクエリーを追加するには

1. ランタイムで、分散アラーム表示を右選択し、［**クエリー設定**］を選択します。［**アラーム クエリー**］ダイアログ ボックスが表示されます。



2. **追加**を選択します。［**クエリーの追加**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. クエリーを設定します。以下の手順を実行します。
   a. ［**名前**］ボックスに、クエリーの名前を入力します。
   b. ［**クエリー**］ボックスに、実行する InTouch アラーム クエリーのセットを入力します。1 つまたは複数のアラーム プロバイダとグループを指定できます。
   c. ［**優先度始点**］ボックスに、最低アラーム優先度の値（1 ～ 999）を入力します。［**優先度終点**］ボックスに、最高アラーム優先度の値（1 ～ 999）を入力します。
   d. ［**アラーム状況**］リストで、アラーム クエリーに使用するアラーム状況を選択します。
   e. ［**表示タイプ**］領域で、表示するアラームのタイプを選択します。詳細については、「サマリ アラームと履歴アラーム」を参照してください。
4. ［**OK**］を選択して、［**クエリーの追加**］ダイアログ ボックスを閉じます。
5. ［**アラーム クエリー**］ダイアログ ボックスで、［**OK**］を選択します。

既存の名前付きクエリーを変更するには

1. ランタイムで、分散アラーム表示を右選択し、［**クエリー設定**］を選択します。［**アラーム クエリー**］ダイアログ ボックスが表示されます。
2. 現在定義されているクエリーのリストで、変更する名前付きクエリーを選択します。
3. [**Modify**] を選択します。［**クエリーの修正**］ダイアログ ボックスが表示されます。
4. 必要な修正を行ってから［**OK**］を選択して、［**クエリーの修正**］ダイアログ ボックスを閉じます。
5. ［**アラーム クエリー**］ダイアログ ボックスの［**OK**］を選択します。

注 : 変更されているアラーム クエリーを使用しているその他の分散アラーム表示オブジェクトには、変更は自動的に適用されません。

既存の名前付きクエリーを削除するには

1. ランタイムで、分散アラーム表示を右選択し、［**クエリー設定**］を選択します。［**アラーム クエリー**］ダイアログ ボックスが表示されます。
2. 現在定義されているクエリーのリストで、変更する名前付きクエリーを選択します。
3. **削除**を選択します。メッセージが表示されたら、［**はい**］を選択します。
4. ［**アラーム クエリー**］ダイアログ ボックスの［**OK**］を選択します。

注 : 削除されているアラーム クエリーを使用しているその他の分散アラーム表示オブジェクトには、削除は自動的に適用されません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

関数とドットフィールドを使用した分散アラーム表示オブジェクトの制御

関数とドットフィールドを使用して、ランタイムで分散アラーム表示オブジェクトを制御できます。

返される関数のエラー番号のリストについては、「エラーの説明」を参照してください。

プロパティの取得または設定

プロパティには、GetProperty*X*() 関数を使用してアクセスできます。ここで、X はデータ タイプ（論理型を表す D、整数型を表す I、メッセージ型を表す M）です。以下に例を示します。

```
GetPropertyM(ControlName.Property, MsgTag)
```

GetPropertyX() 関数の詳細については、「組み込み関数」を参照してください。

GetPropertyX() 関数が含まれるスクリプトを実行するとき、プロパティ値は MsgTag に保存されます。複数の行が選択されている場合、MsgTag に割り当てられるプロパティは、複数選択の最初の行にあるタグ変数値です。

アラームを確認する

分散アラーム表示オブジェクトは、クエリーできる任意のアラームを確認できます（サマリ表示のみ）。分散アラーム表示オブジェクトには、アラーム確認関数が含まれます。これらの関数は、ローカル アラームとアラーム グループを確認するために使用される **.Ack** ドットフィールドを補足します。これらの関数を使用すると、すべてのアラーム、表示されるアラーム、および選択したアラームを確認できます。

グループ メンバシップ、優先度、アプリケーション名、およびタグ変数名のような特性によって、アラームを確認することもできます。

- almAckAll() 関数
- 関数とドットフィールドを使用した分散アラーム表示オブジェクトの制御
- almAckGroup() 関数
- almAckPriority() 関数
- almAckRecent() 関数
- almAckTag() 関数
- almAckSelect() 関数
- almAckSelectedGroup() 関数
- almAckSelectedPriority() 関数
- almAckSelectedTag() 関数

almAckAll() 関数

現在のクエリー内のすべてのアラームを確認します。サマリ モードで分散アラーム表示オブジェクトに現在表示されていないアラームも含まれます。

カテゴリ

アラーム

構文

```
[Result=]almAckAll(ObjectName,Comment);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

Comment
アラーム確認コメントです。

例

```
MessageTag = "Acknowledge All by " + $Operator;
almAckAll("AlmObj_1",MessageTag);
```

参照項目

Ack()、almAckGroup()、almAckTag()、almAckDisplay()、almAckRecent()、almAckSelect()、almAckSelectedGroup()、almAckSelectedPriority()、almAckSelectedTag()

almAckDisplay() 関数

サマリ モードで分散アラーム表示オブジェクトに現在表示されているアラームのみを確認します。

カテゴリ

アラーム

構文

```
[Result=]almAckDisplay(ObjectName,Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

Comment
アラーム確認コメントです。

例

```
almAckDisplay("AlmObj_1","Display Acknowledgement");
```

参照項目

Ack()、almAckAll()、almAckGroup()、almAckTag()、almAckRecent()、almAckSelect()、almAckSelectedGroup()、almAckSelectedPriority()、almAckSelectedTag()

almAckGroup() 関数

指定したプロバイダとグループ名に一致する、指定されている分散アラームオブジェクトに表示されるすべてのアラームを確認します。

カテゴリ

アラーム

構文

```
[Result=]almAckGroup( “ObjectName”, ApplicationName, GroupName, Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ApplicationName
たとえば、\\node1\Intouch などアプリケーション名です。

GroupName
たとえば、$System など InTouch アラーム グループの名前です。

Comment
アラーム確認コメントです。

例

```
MessageTag = “Acknowledge group, Turbines, by ” + $Operator;
almAckGroup(“AlmObj_1”, “\Intouch”, “Turbine”, MessageTag);
```

almAckPriority() 関数

アラームのアプリケーション名、アラーム グループ、および優先度範囲に一致する最後のクエリーの結果として、指定されている分散アラームオブジェクトに表示されるすべてのアラームを確認します。

カテゴリ

アラーム

構文

```
[Result=]almAckPriority(ObjectName, ApplicationName, GroupName, FromPri, ToPri, Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
たとえば、\\node1\Intouch などアプリケーション名です。

GroupName
たとえば、$System などグループの名前です。

FromPri
アラーム優先度範囲の開始番号です。たとえば、100 です。

ToPri
アラーム優先度範囲の終了番号です。たとえば、900 です。

Comment
アラーム確認コメントです。

例

```
almAckPriority(“AlmObj_1”, “\\node1\Intouch”, “Turbines”, 10, 100, “Range 10 to 100
acknowledged”);
```

almAckRecent() 関数

最後に発生したアラームを確認します。

構文

```
[Result=]almAckRecent(ObjectName, Comment)
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Comment
アラーム確認コメントです。

例

```
almAckRecent("AlmObj_1",$DateString);
```

almAckTag() 関数

最後のクエリーの結果として、指定されている分散アラームオブジェクトに表示されるすべてのアラームを確認します。アラームは、アプリケーション名、グループ名、タグ変数名、およびクエリーで指定された優先度範囲に一致する必要があります。

カテゴリ

アラーム

構文

```
[Result=]almAckTag("ObjectName", ApplicationName, GroupName, TagName, FromPri, ToPri,
Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
アプリケーションの名前です。たとえば、\\node1\Intouch です。

GroupName
アラーム グループの名前です。たとえば、$System です。

TagName
値がアラーム状況にあるタグ変数の名前です。

FromPri
アラーム優先度範囲の開始番号です。たとえば、100 です。

ToPri
アラーム優先度範囲の終了番号です。たとえば、900 です。

Comment
アラーム確認コメントです。

例

```
almAckTag("AlmObj_1", "\\node1\Intouch", "Turbines", "Valve1", 10, 100, "Acknowledged for
Valve1");
```

参照項目

Ack()、almAckAll()、almAckGroup()、almAckDisplay()、almAckRecent()、almAckSelect()、almAckSelectedGroup()、almAckSelectedPriority()、almAckSelectedTag()

almAckSelect() 関数

サマリ モードで分散アラーム表示オブジェクトで選択したアラームのみを確認します。

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
[Result=]almAckSelect(ObjectName,Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

Comment
アラーム確認コメントです。

例

この例では、昼間勤務または夜間勤務の間に発生したアラームのみを確認します。

```
IF ($Hour >= 0 and $Hour < 8) THEN
   AckTag = "Night Shift";
ELSE
   AckTag = "Day Shift";
ENDIF;
almAckSelect ("AlmObj_1",AckTag);
```

参照項目

Ack()、almAckAll()、almAckGroup()、almAckTag()、almAckDisplay()、almAckRecent()、almAckSelectedGroup()、almAckSelectedPriority()、almAckSelectedTag()

almAckSelectedGroup() 関数

指定された分散アラーム表示オブジェクト内で選択される 1 つまたは複数のアラームの同じグループ名を持つ同じプロバイダ名とグループ名のすべてのアラームを確認します。

カテゴリ

アラーム

構文

```
[Result=]almAckSelectedGroup(ObjectName,Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

Comment
アラーム確認コメントです。

例

```
MessageTag = "Acknowledge selected groups by " + $Operator;
almAckSelectedGroup ("AlmObj_1", MessageTag);
```

参照項目

Ack(), almAckAll()、almAckGroup()、almAckTag()、almAckDisplay()、almAckRecent()、almAckSelect()、almAckSelectedPriority()、almAckSelectedTag()

almAckSelectedPriority() 関数

指定された分散アラーム表示オブジェクト内で選択される 1 つまたは複数のアラームの同じ優先度値を持つ同じプロバイダ名とグループ名のすべてのアラームを確認します。優先度は、選択したアラームレコードの最低優先度および最大優先度から計算されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

アラーム

構文

```
[Result=]almAckSelectedPriority(ObjectName, Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

Comment
アラーム確認コメントです。

例

```
MessageTag = "Acknowledge selected priorities by " + $Operator;
almAckSelectedPriority ("AlmObj_1", MessageTag);
```

参照項目

Ack()、almAckAll()、almAckGroup()、almAckTag()、almAckDisplay()、almAckRecent()、almAckSelect()、almAckSelectedGroup()、almAckSelectedTag()

almAckSelectedTag() 関数

指定された分散アラーム表示オブジェクト内の 1 つまたは複数の選択したアラームと同じ優先度を持つ同じプロバイダ名およびグループ名からの同じタグ変数を持つすべてのアラームを確認します。この関数は、InTouch がアラーム プロバイダの場合にのみ機能します。

カテゴリ

アラーム

構文

```
[Result=]almAckSelectedTag(ObjectName,Comment);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

Comment
アラーム確認コメントです。

例

```
MessageTag = "Acknowledge selected tagnames by " + $Operator;
almAckSelectedTag ("AlmObj_1", MessageTag);
```

参照項目

Ack()、almAckAll()、almAckGroup()、almAckTag()、almAckDisplay()、almAckRecent()、almAckSelect()、almAckSelectedGroup()、almAckSelectedPriority()

アラームの選択

分散アラーム表示オブジェクトからアラームを選択するスクリプトを作成できます。すべてのアラームを選択することも、選択したアラームのみを選択することも、現在のアラームのカウントを取得することもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- almSelectAll() 関数
- almUnselectAll() 関数
- almSelectionCount() 関数
- almSelectGroup() 関数
- almSelectItem() 関数
- almSelectPriority() 関数
- almSelectTag() 関数

データ ソース、アラーム優先度、および InTouch タグ変数に基づいて、特定のアラームを選択することもできます。

almSelectAll() 関数

指定された分散アラーム表示オブジェクト内のすべてのアラームの選択を切り替えます。

カテゴリ

アラーム

構文

```
[Result=]almSelectAll(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

例

```
If $AccessLevel > 8000 THEN
   almSelectAll("AlmObj_1");
   almAckSelect("AlmObj_1", "Ack Selected by a Manager");
ENDIF;
```

参照項目

almSelectItem()、almSelectGroup()、almSelectPriority()、almSelectTag()、almUnSelectAll()

almUnselectAll() 関数

指定された分散アラーム表示オブジェクト内のすべての選択済みアラームを選択解除します。

カテゴリ

アラーム

構文

```
[Result=]almUnselectAll(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

例

```
If $AccessLevel == 9999 THEN
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
   almAckSelect("AlmObj_1", "Comment");{This alarm can be acknowledged by only
   Administrator}
   ELSE
   almUnselectAll("AlmObj_1");
ENDIF;
```

参照項目

almSelectAll()、almSelectItem()、almSelectGroup()、almSelectPriority()、almSelectTag()

almSelectionCount() 関数

分散アラーム表示オブジェクトでオペレータによって選択されたアラーム数を返します。

カテゴリ

アラーム

構文

```
[Result=]almSelectionCount(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

例

AlarmCount タグ変数には、分散アラーム表示オブジェクトからオペレータによって選択されたアラーム数が割り当てられます。

```
AlarmCount = almSelectionCount("AlmObj_1");
```

almSelectGroup() 関数

表示の最後のクエリーの結果として、結果アラームに同じアラーム グループ名が含まれる、指定された分散アラーム表示オブジェクトによって含まれるすべてのアラームの選択を切り替えます。

カテゴリ

アラーム

構文

```
[Result=]almSelectGroup(ObjectName, ApplicationName,GroupName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
アプリケーションの名前です。たとえば、\\node1\Intouch です。

GroupName
グループの名前です。たとえば、$System です。

例

```
almSelectGroup("AlmObj_1","\InTouch","Turbine");
```

参照項目

almSelectAll()、almSelectItem()、almSelectPriority()、almSelectTag()、almUnSelectAll()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

almSelectItem() 関数

アラーム表示オブジェクト内で最後に選択または選択解除されたアイテムの選択を切り替えます。

構文

```
[Result=]almSelectItem(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

例

```
almSelectItem("AlmObj_1");
```

参照項目

almSelectAll()、almSelectGroup()、almSelectPriority()、almSelectTag()、almUnSelectAll()

almSelectPriority() 関数

表示の最後のクエリーの結果として、結果アラームが指定した優先度範囲内である、指定された分散アラーム表示オブジェクトのすべてのアラームの選択を切り替えます。

カテゴリ

アラーム

構文

```
[Result=]almSelectPriority( "objectName", ApplicationName, GroupName, FromPri, ToPri );
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
アプリケーションの名前です。たとえば、\\node1\Intouch です。

GroupName
グループの名前です。たとえば、$System です。

FromPri
アラームの開始優先度です。たとえば、100 または整数型タグ変数です。

ToPri
アラームの終点優先度です。たとえば、900 または整数型タグ変数です。

例

```
almSelectPriority("AlmObj_1","\\node1\Intouch", "Turbines",10,100);
```

参照項目

almSelectAll()、almSelectItem()、almSelectGroup()、almSelectTag()、almUnSelectAll()

almSelectTag() 関数

表示の最後のクエリーおよび指定されたタグ変数の結果として、指定された分散アラーム表示オブジェクトのすべてのアラームの選択を切り替えます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

アラーム

構文

```
[Result=]almSelectTag (ObjectName, ApplicationName, GroupName, TagName, FromPri, ToPri);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
アプリケーションの名前です。たとえば、\\node1\Intouch です。

GroupName
グループの名前です。たとえば、$System です。

TagName
アラーム タグ変数の名前です。

FromPri
アラームの開始優先度です。たとえば、100 または整数型タグ変数です。

ToPri
アラームの終点優先度です。たとえば、900 または整数型タグ変数です。

例

```
almSelectTag("AlmObj_1","\\node1\Intouch","Turbines","Valve1",10,100);
```

参照項目

almSelectAll()、almSelectItem()、almSelectGroup()、almSelectPriority()、almUnSelectAll()

選択したアラーム レコードに関する情報の取得

選択したアラームに関する情報を返すスクリプトを作成できます。スクリプトで以下のドットフィールドおよび関数を使用します。

- .AlarmTime ドットフィールド
- .AlarmDate ドットフィールド
- .AlarmName ドットフィールド
- .AlarmValue ドットフィールド
- .AlarmClass ドットフィールド
- .AlarmType ドットフィールド
- .AlarmState ドットフィールド
- .AlarmLimit ドットフィールド
- .AlarmPri ドットフィールド
- .AlarmGroupSel ドットフィールド
- .AlarmAccess ドットフィールド
- .AlarmProv ドットフィールド

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- .AlarmOprName ドットフィールド
- .AlarmOprNode ドットフィールド
- .AlarmComment ドットフィールド

.AlarmTime ドットフィールド

アラームが発生した時間を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトで選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmTime",TagName);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データタイプ

文字列（読み取り専用）

例

この例では、AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almTime** はメモリ メッセージ型タグ変数です。

```
GetPropertyM("AlmObj_1.AlarmTime",almTime);
```

押しボタン QuickScript で使用した場合、このステートメントはアラームの発生時刻を **almTime** タグ変数に返します。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmType、.AlarmValue

.AlarmDate ドットフィールド

選択したアラームに関連付けられている日付を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmDate",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

押しボタン QuickScript で使用した場合、このステートメントは日付を **almDate** タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmDate",almDate);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almDate** は選択したアラームに関連付けられているタグ変数の日付を取得するメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmName ドットフィールド

選択したアラームに関連付けられているタグ変数の名前を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmDate",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

押しボタン QuickScript で使用した場合、このステートメントはアラームの名前を **almName** に返します。

```
GetPropertyM("AlmObj_1.AlarmName",almName);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almName** は選択したアラームに関連付けられているタグ変数名を受け取るメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmValue ドットフィールド

選択したアラームに関連付けられているタグ変数のアラーム値を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmValue,TagName);
```

パラメータ

オブジェクト名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

備考

この関数は、メッセージ型タグ変数を使用して、数値を取得します。これは、GetProperty 関数が実数をサポートしていないためです。StringToReal() 関数を使用して、結果を実数型タグ変数に割り当てることができます。

例

押しボタン QuickScript で使用した場合、このステートメントは値を **almValue** に返します。

```
GetPropertyM("AlmObj_1.AlarmValue",explicitValue);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almValue** は選択したアラームに関連付けられるタグ変数のアラーム値を含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmClass ドットフィールド

選択したアラームに関連付けられるタグ変数のアラーム クラスを返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmClass",Tagname);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データタイプ

文字列（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のステートメントは、選択したアラームに関連付けられているアラーム クラスを返します。

```
GetPropertyM("AlmObj_1.AlarmClass",almClass);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almClass** は選択したアラームに関連付けられるタグ変数のアラーム クラスを含むメモリ メッセージ型タグ変数です。

押しボタン QuickScript で使用した場合、このステートメントはアラームのアラーム クラスを **almClass** タグ変数に返します。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmType ドットフィールド

選択したアラームに関連付けられているタグ変数のアラーム タイプを返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmType",TagName);
```

パラメータ

オブジェクト名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

押しボタン QuickScript で使用される場合、このステートメントはオペレータがアラームを確認するときに、**almType** タグ変数に対して選択されたアラームのタイプを返します。

```
GetPropertyM("AlmObj_1.AlarmType",almType);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almType** は選択したアラームに関連付けられるタグ変数のアラーム タイプを含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmValue

.AlarmState ドットフィールド

選択したアラームの状況を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmState",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

押しボタン QuickScript で使用した場合、このステートメントは選択したアラームの状態を almState タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmState",almState);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almState** は選択したアラームに関連付けられるタグ変数のアラーム状況を含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmTime、.AlarmType、.AlarmValue

.AlarmLimit ドットフィールド

選択したアラームに関連付けられているタグ変数のしきい値を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM ("ObjectName.AlarmLimit",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

この関数は、メッセージ型タグ変数を使用して、数値を取得します。これは、GetProperty 関数が実数をサポートしていないためです。StringToReal() 関数を使用して、結果を実数型タグ変数に割り当てることができます。

例

押しボタン QuickScript で使用した場合、このステートメントは選択したアラームのしきい値を almLimit タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmLimit",explicitLimit);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almLimit** は選択したアラームに関連付けられるタグ変数のアラームしきい値を含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmPri ドットフィールド

選択したアラームに関連付けられているタグ変数の優先度（1 ～ 999）を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmType",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

押しボタン QuickScript で使用した場合、このステートメントはアラーム優先度を almPrilvl タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmPri",almPri;
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almPri** は選択したアラームに関連付けられるタグ変数の優先度レベルを含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.AlarmGroupSel ドットフィールド

選択したアラームに関連付けられているタグ変数のアラーム グループを返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmState",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

押しボタン QuickScript で使用した場合、このステートメントはアラーム グループの名前を almGroup タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmGroupSel",almGroup);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almGroup** は選択したアラームに関連付けられるタグ変数のアラーム グループを含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmGroup、.AlarmName

.AlarmAccess ドットフィールド

選択したアラームに関連付けられているタグ変数のアクセス名を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
GetPropertyM("Objectname.AlarmAccess",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

押しボタン QuickScript で使用した場合、アラームに関連付けられるタグ変数のアクセス名を almAccess タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmAccess",explicitAccess);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almAccess** は選択したアラームに関連付けられるタグ変数のアクセス名を含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType

.AlarmProv ドットフィールド

選択したアラームに関連付けられているタグ変数のアラーム プロバイダを返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmType",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

押しボタン QuickScript で使用した場合、このステートメントはプロバイダ名を almProv タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmProv", almProv);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almProv** は選択したアラームに関連付けられるタグ変数のプロバイダ名を含むメモリ メッセージ型タグ変数です。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmOprName ドットフィールド

選択したアラームを確認したログオン ユーザーの名前を返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmOprName",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

```
GetPropertyM("AlmObj_1.AlarmOprName",almOprName);
```

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almOprName** はタグ変数に関連付けられるアラームに応答するユーザーの名前を含むメモリ メッセージ型タグ変数です。

押しボタン QuickScript で使用した場合、このステートメントはユーザーの名前を almOprName タグ変数に返します。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmOprNode ドットフィールド

選択したアラームに関連付けられているタグ変数に対するオペレータ ノードを返します。アラームは、サマリ モードで分散アラーム表示オブジェクトから選択する必要があります。

ターミナル サービス環境でアラームが確認されると、オペレータ ノードは、対応するオペレータがターミナル サービス セッションを確立しているクライアント マシンの名前になります。ノード名を取得できない場合、ノードの IP アドレスが代わりに使用されます。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmOprNode",TagName);
```

パラメータ

ｵﾌﾞｼﾞｪｸﾄ名
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データ型

文字列（読み取り専用）

例

```
GetPropertyM("AlmObj_1.AlarmOprNode",almOprNode);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

AlmObj_1 は分散アラーム表示オブジェクトの名前で、**almOprNode** は選択したアラームに関連付けられるタグ変数のオペレータ ノードの名前を含むメモリ メッセージ型タグ変数です。

押しボタン QuickScript で使用した場合、このステートメントはユーザーのノードを almOprNode タグ変数に返します。

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmComment ドットフィールド

タグ変数ではなく、アラームを記述する読み取り／書き込み文字列であるアラーム コメントを返します。デフォルトでは、新しいアプリケーションのコメントは空です。

ただし、古い InTouch アプリケーションが InTouch バージョン 7.11 以降に変換される場合、後方互換性のために、タグ変数コメントが .AlarmComment ドットフィールドにコピーされます。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmComment",TagName);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

データタイプ

文字列（読み取り専用）

例

以下の例は、AlmObj_1 分散アラーム表示オブジェクトで選択したタグ変数のアラーム コメントが返され、almComment タグ変数に配置されます。

```
GetPropertyM("AlmObj_1.AlarmComment", almComment);
```

参照項目

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

アラーム クエリーの設定

以下のクエリー関数を使用して、アラーム メモリからレコードを取得します。

- almDefQuery() 関数
- almQuery() 関数
- almSetQueryByName() 関数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

almDefQuery() 関数

デフォルト プロパティを使用してクエリーを実行し、指定された分散アラーム表示オブジェクトを更新します。

カテゴリ

アラーム

構文

```
[Result=]almDefQuery(ObjectName);
```

引数

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

備考

WindowMaker で分散アラーム表示オブジェクトを開発する間に、デフォルトのクエリー プロパティが指定されます。

例

```
almDefQuery("AlmObj_1");
```

参照項目

almQuery()、almSetQueryByName()

almQuery() 関数

指定された分散アラーム表示オブジェクトを更新するクエリーを実行して、指定したパラメータを使用します。

カテゴリ

アラーム

構文

```
[Result=]almQuery(ObjectName,AlarmList,FromPri,ToPri,State,Type);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

AlarmList
アラーム クエリー／名前マネージャのエイリアスを設定し、たとえば「\intouch!$System」やメッセージ型タグ変数に対するクエリーを実行します。

FromPri
表示するアラームの開始優先度です。たとえば、100 または整数型タグ変数です。

ToPri
表示するアラームの終了優先度です。たとえば、900 または整数型タグ変数です。

State
表示するアラームのタイプを指定します。たとえば、「UnAck」またはメッセージ型タグ変数です。有効な状況は、All、UnAck、または Ack です。

タイプ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

更新された表示に示されるアラーム レコードのタイプ：
"Hist" ＝履歴アラーム
"Summ" ＝サマリ アラーム

例

このステートメントは、優先度が 500 ～ 600 の MyAlarmListGroup で指定されたすべての履歴アラームを取得します。アラームは AlmObj_1 アラーム表示に表示されます。

```
almQuery("AlmObj_1","MyAlarmListGroup",500,600,"All","Hist");
```

この例では、MyAlarmListGroup は、名前マネージャ セットアップから設定されたアラーム リストです。

参照項目

almDefQuery()、almSetQueryByName()

almSetQueryByName() 関数

ユーザー定義クエリー設定ファイルからのパラメータを使用して、分散アラーム表示オブジェクトの指定されたインスタンスに対して新しいアラーム クエリーを開始します。

カテゴリ

アラーム

構文

```
[Result=]almSetQueryByName(ObjectName, QueryName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

QueryName
クエリー設定を使用して作成されたクエリーの名前です。

備考

これは分散アラーム表示オブジェクトの特定インスタンスに対するクエリーです。画面上に、独自のクエリーを持つ表示がいくつか存在する可能性があります。

例

この例は、「Turbine Queries」という名前のクエリーからのパラメータを使用して、新しいクエリーを開始します。

```
almSetQueryByName("AlmObj_1","Turbine Queries");
```

参照項目

almQuery()、almDefQuery()

現在のクエリー プロパティの確認

以下のドットフィールドを使用して、アラーム メモリ クエリーのステータスを返します。

- .AlarmGroup ドットフィールド
- .QueryType ドットフィールド
- .QueryState ドットフィールド
- .Successful ドットフィールド

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- .PriFrom ドットフィールド
- .PriTo ドットフィールド

.AlarmGroup ドットフィールド

分散アラーム表示オブジェクトを形成するために使用される現在のクエリーが含まれます。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmGroup",TagName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意のメッセージ型タグ変数です。

備考

この読み取り専用ドットフィールドには、指定された分散アラーム表示オブジェクトで使用される現在のアラーム クエリーが含まれます。このクエリーは、アラーム グループのリストまたは直接のアラーム プロバイダ リファレンスの場合があります。

データタイプ

文字列（読み取り専用）

例

このステートメントは、AlmObj_1 分散アラーム オブジェクトによって使用される現在のアラーム クエリーを **CurrentQuery** タグ変数に返します。

```
GetPropertyM("AlmObj_1.AlarmGroup",CurrentQuery);
```

参照項目

GetPropertyM(), .AlarmGroupSel, .AlarmName

.QueryType ドットフィールド

現在のアラーム クエリーのタイプを表示します。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyI( "ObjectName.QueryType",TagName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意の整数型タグ変数です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

この読み取り専用ドットフィールドには、指定された分散アラーム表示オブジェクトで使用される現在のクエリー タイプが含まれます。

データタイプ

整数型（読み取り専用）

有効値

1 = 履歴

2 = サマリ

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトの現在のクエリー タイプを AlmQueryType タグ変数に返します。

```
GetPropertyI("AlmObj_1.QueryType",AlmQueryType);
```

参照項目

GetPropertyI()、.QueryState

.QueryState ドットフィールド

現在のアラーム状況クエリー フィルタを表示します。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyI( "ObjectName.QueryState",TagName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意の整数型タグ変数です。

備考

この読み取り専用ドットフィールドには、指定された分散アラーム表示オブジェクトによって使用される現在のクエリー フィルタが含まれます。

データタイプ

整数型（読み取り専用）

有効値

0 = 全て

1 = 未確認

2 = 確認

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトの現在のクエリー フィルタを AlmQueryState タグ変数に返します。

```
GetPropertyI("AlmObj_1.QueryState", AlmQueryState);
```

参照項目

GetPropertyI()、.QueryType

.Successful ドットフィールド

現在のクエリーが成功したかどうかを示します。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyD( "ObjectName.Successful",TagName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
関数が処理されるとき、プロパティ値を保持する論理型タグ変数

備考

この読み取り専用ドットフィールドには、指定された分散アラーム表示オブジェクトの最後のクエリーの状況が含まれます。

データタイプ

論理型（読み取り専用）

有効値

0 = クエリーでエラー発生

1 = クエリー成功

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトの最後のクエリーの状況を AlmFlag タグ変数に返します。

```
GetPropertyD("AlmObj_1.Successful",AlmFlag);
```

参照項目

GetPropertyD()

.PriFrom ドットフィールド

現在のクエリーで使用されるアラーム優先度範囲の最低値を返します。

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

```
[ErrorNumber=]GetPropertyI("ObjectName.PriFrom", Tagname);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
任意の整数型タグ変数です。

データタイプ

整数型（読み取り専用）

例

以下のステートメントは、分散アラーム表示オブジェクト AlmObj_1 で使用されるクエリーの最小優先度値を、**MinPri** 整数型タグ変数に返します。

```
GetPropertyI("AlmObj_1.PriFrom",MinPri);
```

参照項目

GetPropertyI()、.PriTo、.AlarmPri

.PriTo ドットフィールド

現在のクエリーで使用されるアラーム優先度範囲の最大値が含まれます。

使用法

```
[ErrorNumber=]GetPropertyI("ObjectName.PriTo", Tagname);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
関数が処理されるときにプロパティ値を保持する整数型タグ変数です。

データタイプ

整数型（読み取り専用）

例

以下のステートメントは、分散アラーム表示オブジェクト AlmObj_1 によって使用されるクエリーの最大優先度値を、MaxPri 整数型タグ変数に返します。

```
GetPropertyI("AlmObj_1.PriTo",MaxPri);
```

参照項目

GetPropertyI()、.PriFrom、.AlarmPri

分散アラーム表示オブジェクトの更新の確認

以下のドットフィールドを使用して、分散アラーム表示オブジェクトにすべての現在アラームが含まれているか、または保留中の更新があるかどうかを確認します。

- .ListChanged ドットフィールド
- .PendingUpdates ドットフィールド

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.ListChanged ドットフィールド

分散アラーム表示オブジェクト用の新規アラームまたは更新されたアラームがあるかどうかを示します。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyD("ObjectName.ListChanged",TagName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
関数が処理されるとき、プロパティ値を保持する論理型タグ変数

備考

この読み取り専用ドットフィールドには、分散アラーム表示オブジェクトで更新される必要のある変更があるかどうかについての状況が含まれます。このプロパティは、プロパティを読み取る際に自動的にリセットされます。

データ タイプ

論理型（読み取り専用）

有効値

0 = 表示オブジェクトに対する新しいアラームまたは更新がありません

1 = 表示オブジェクトに対する新しい更新があります

例

```
以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトの新規アラームまたは更新の状況を
AlmDispStat タグ変数に返します。
GetPropertyD("AlmObj_1.ListChanged",AlmDispStat);
```

参照項目

GetPropertyD()

.PendingUpdates ドットフィールド

分散アラーム表示オブジェクトに対する保留中の更新の数を示します。表示が停止し、新しいアラームレコードが作成されたときには、一般的に保留中の更新があります。これらの更新は表示されませんが、保留中の更新のカウントは増加します。

カテゴリ

アラーム

使用法

```
[ErrorMessage=]GetPropertyI( "ObjectName.PendingUpdates", TagName);
```

引数

ObjectName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
関数が処理されるときにプロパティ値を保持する整数型タグ変数です。

備考

この読み取り専用ドットフィールドには、指定された分散アラーム表示オブジェクトの保留中の更新数が含まれます。ゼロより大きい任意の値は、分散アラーム表示オブジェクトに新規アラーム データがあることを示しています。この値は、オブジェクトがリフレッシュされるたびにリセットされます。

データタイプ

整数型（読み取り専用）

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトの保留中の更新がある場合、整数 1 以上を AlarmPendingUpdates タグ変数に返します。

```
GetPropertyI( "AlmObj_1.PendingUpdates",AlarmPendingUpdates);
```

参照項目

GetPropertyI()

アラームを非表示にする

分散アラーム表示オブジェクトは、例外基準に一致するアラーム コンシューマで 1 つまたは複数のアラームを非表示にできます。アラームが例外基準に一致する場合、表示のインスタンスに表示されません。

QuickScript 関数を使用して、アラームを非表示にできます。

- almSuppressAll() 関数
- almUnsuppressAll() 関数
- almSuppressDisplay() 関数
- almSuppressGroup() 関数
- almSuppressPriority() 関数
- almSuppressTag() 関数
- almSuppressSelected() 関数
- almSuppressSelectedGroup() 関数
- almSuppressSelectedPriority() 関数
- almSuppressSelectedTag() 関数
- almSuppressRetain() 関数
- .SuppressRetain ドットフィールド

almSuppressAll() 関数

サマリ モードで分散アラーム表示オブジェクトに現在表示されていないインスタンスも含め、現在のクエリーのアラームの現在および今後のすべてのインスタンスを非表示にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
[Result=] almSuppressAll(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

備考

この関数は almAckAll() と同様に機能します。画面を確認して、スクロールで上下に移動してアラームをすべて確認した場合に表示されるすべてのアラームを識別することによって、非表示にするアラームを識別します。

例

```
almSuppressAll("AlmObj_1");
```

参照項目

almSuppressGroup()、almSuppressTag()、almSuppressDisplay()、almSuppressPriority()、almSuppressRetain()、almSuppressSelected()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almSuppressSelectedTag()、almUnSuppressAll()

almUnsuppressAll() 関数

非表示のアラームをすべてクリアします。

構文

```
[Result=] almUnSuppressAll(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

例

```
almUnSuppressAll("AlmObj_1");
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressTag()、almSuppressDisplay()、almSuppressPriority()、almSuppressRetain()、almSuppressSelected()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almSuppressSelectedTag()

almSuppressDisplay() 関数

サマリ モードで、分散アラーム表示オブジェクトで表示される現在および今後のアラームを非表示にします。

構文

```
[Result=]almSuppressDisplay(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

この関数は対応する almAckDisplay() 関数と同様に機能します。現在表示されているすべてのアラームを識別して、非表示にするアラームを識別します。

例

```
almSuppressDisplay("AlmObj_1");
```

almSuppressGroup() 関数

指定したプロバイダとグループ名を持つ任意のアラームの現在および今後の発生を非表示にします。

構文

```
[Result=]almSuppressGroup(ObjectName, ApplicationName,GroupName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
アプリケーションの名前です。たとえば、\\node1\InTouch です。

GroupName
アラーム グループの名前です。たとえば、$System です。

例

```
almSuppressGroup( "AlmObj_1","\InTouch","Turbines");
```

参照項目

almSuppressAll()、almSuppressTag()、almSuppressDisplay()、almSuppressPriority()、almSuppressRetain()、almSuppressSelected()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almSuppressSelectedTag()、almUnSuppressAll()

almSuppressPriority() 関数

同じプロバイダ名とグループ名を持つ指定された優先度範囲の任意のアラームの現在および今後の発生を非表示にします。

構文

```
[Result=]almSuppressPriority(ObjectName, ApplicationName, GroupName, FromPri, ToPri);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
アプリケーションの名前です。たとえば、\\node1\InTouch です。

GroupName
グループの名前です。たとえば、$System です。

FromPri
アラームの開始優先度です。たとえば、100 または整数型タグ変数です。

ToPri
アラームの終点優先度です。たとえば、900 または整数型タグ変数です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

```
almSuppressPriority("AlmObj_1","\\node1\Intouch", "Turbines",10,100);
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressTag()、almSuppressDisplay()、almSuppressRetain()、almSuppressSelected()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almSuppressSelectedTag()、almUnSuppressAll()

almSuppressTag() 関数

同じプロバイダ名、グループ名、および優先度範囲を持つ指定したタグ変数に属する任意のアラームの現在および今後の発生を非表示にします。

カテゴリ

アラーム

構文

```
[Result=]almSuppressTag(ObjectName, ApplicationName, GroupName, TagName, FromPri, ToPri,
AlarmClass, AlarmType);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

ApplicationName
アプリケーションの名前です。たとえば、\\node1\InTouch です。

GroupName
グループの名前です。たとえば、$System です。

TagName
アラーム タグ変数の名前です。

FromPri
アラームの開始優先度です。たとえば、100 または整数型タグ変数です。

ToPri
アラームの終点優先度です。たとえば、900 または整数型タグ変数です。

AlarmClass
アラームのクラスです。たとえば、"Value" です。

AlarmType
アラームのタイプです。たとえば、"HiHi" です。

例

```
almSuppressTag("AlmObj_1","\\node1\Intouch", "Turbines","Valve1",10,100,"Value","LoLo");
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressDisplay()、almSuppressPriority()、almSuppressRetain()、almSuppressSelected()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almSuppressSelectedTag()、almUnSuppressAll()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

almSuppressSelected() 関数

サマリ モードで、分散アラーム表示オブジェクトで選択されるアラームの現在および今後の発生を非表示にします。

カテゴリ

アラーム

構文

```
[Result=]almSuppressSelected(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

備考

この関数は、**almAckSelect()** 関数と同様に機能します。表示オブジェクトで選択したアラームによってアラームを識別します。

例

```
almSuppressSelected("AlmObj_1");
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressTag()、almSuppressDisplay()、almSuppressPriority()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almSuppressSelectedTag()、almUnSuppressAll()

almSuppressSelectedGroup() 関数

指定された分散アラーム表示オブジェクト内の同じプロバイダ名を持つ 1 つまたは複数の選択したアラームの同じグループに属すアラームの現在および今後の発生を非表示にします。

カテゴリ

アラーム

構文

```
[Result=]almSuppressSelectedGroup(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

備考

この関数は almAckSelectedGroup() 関数と同様に機能します。選択されたアラームを識別し、それらのアラームが属すグループを識別し、これらのグループからの今後の発生を非表示にします。

例

```
almSuppressSelectedGroup("AlmObj_1");
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressTag()、almSuppressDisplay()、almSuppressSelected()、almSuppressSelectedPriority()、almSuppressSelectedTag()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

almSuppressSelectedPriority() 関数

指定された分散アラーム表示オブジェクト内で、同じプロバイダ名およびグループ タグ変数を持つ 1 つまたは複数の選択されたアラームの同じ優先度に属すアラームの現在および今後の発生を非表示にします。

カテゴリ

アラーム

構文

```
[Result=]almSuppressSelectedPriority(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

備考

優先度は、選択したアラーム レコードの最低優先度および最大優先度から計算されます。

この関数は almAckSelectedPriority() 関数と同様に機能します。表示で選択されたアラームを識別し、これらのアラームの対応する優先度を識別し、同じ優先度を持つアラームの今後の発生を非表示にします。

例

```
almSuppressSelectedPriority("AlmObj_1");
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressTagName()、almSuppressDisplay()、almSuppressSelected()、almSuppressSelectedGroup()、almSuppressSelectedTag()、almAckSelectedPriority()

almSuppressSelectedTag() 関数

同じプロバイダ名、グループ名、および優先度範囲を持つ 1 つまたは複数の選択されたアラームの同じタグ変数に属す任意のアラームの現在および今後の発生を非表示にします。

カテゴリ

アラーム

構文

```
[Result=]almSuppressSelectedTag(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

例

```
almSuppressSelectedTag("AlmObj_1");
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressTag()、almSuppressDisplay()、almSuppressSelectedAlarm()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almAckSelectedTag()、almUnSuppressAll()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

almSuppressRetain() 関数

後続のクエリーで発生したすべてのアラームを非表示にします。

カテゴリ

アラーム

構文

```
[Result=]almSuppressRetain(ObjectName,SuppressionRetainFlag);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

SuppressionRetainFlag
任意の論理型またはアナログ型のタグ変数、0 またはゼロ以外の値です。後続のクエリーに対しても非表示情報が保持される場合は TRUE、それ以外の場合は FALSE です。

備考

アラーム クエリーが変更されたときフラグが 0 である場合、非表示フィルタは削除されます。

例

```
almSuppressRetain("AlmObj_1", 0);
```

参照項目

almSuppressAll()、almSuppressGroup()、almSuppressTag()、almSuppressDisplay()、almSuppressPriority()、almSuppressSelected()、almSuppressSelectedGroup()、almSuppressSelectedPriority()、almSuppressSelectedTag()、almUnSuppressAll()

.SuppressRetain ドットフィールド

分散アラーム表示オブジェクトの非表示状態を保持する機能のステータスを読み取りまたは書き込みします。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyD( "ObjectName.SuppressRetain",TagName);
[ErrorNumber=]SetPropertyD( "ObjectName.SuppressRetain",TagName);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

Tagname
スクリプトが処理されるとき、プロパティ値を保持する論理型タグ変数です。

データタイプ

論理型（読み取り／書き込み）

有効値

0 = オフを保持します

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

1 = オンを保持します

例

以下のステートメントは、SupRtn 論理型タグ変数から、「AlmObj_1」の非表示保持のステータスを設定します。

```
SetPropertyD("AlmObj_1.SuppressRetain", SupRtn);
```

参照項目

GetPropertyD()、SetPropertyD()

アラーム表示のスクロール

以下の関数とドットフィールドを使用して、分散アラーム表示オブジェクト内のアラーム リストを垂直または水平にスクロールします。表示を停止することもできます。

- almMoveWindow() 関数
- .Freeze ドットフィールド
- .PrevPage ドットフィールド
- .NextPage ドットフィールド

almMoveWindow() 関数

分散アラーム表示オブジェクトのアラーム リストを垂直または水平にスクロールします。

カテゴリ

アラーム

構文

```
[Result=]almMoveWindow(ObjectName,Option,Repeat);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

Option
実行するスクロール操作のタイプ：

タイプ	説明
LineDn	次の行へ移動
LineUp	前の行へ移動
PageDn	次のページへ移動
PageUp	前のページへ移動
Top	リストの一番上へ移動
Bottom	リストの一番下へ移動
PageRt	右ページへ移動
PageLf	左ページへ移動

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タイプ	説明
Right	リストの最後へ移動（右側）
Left	リストの先頭へ移動（左側）

Repeat
操作が繰り返される回数です。

例

```
almMoveWindow("AlmObj_1", "Bottom", 0);
almMoveWindow("AlmObj_1", "LineDn", 3);
almMoveWindow("AlmObj_1", "PageUp", 0);
```

.Freeze ドットフィールド

.Freeze ドットフィールドは、停止ステータスを読み込むか、分散アラーム表示オブジェクトを停止または停止解除します。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyD("ObjectName.Freeze", TagName);
[ErrorNumber=]SetPropertyD("ObjectName.Freeze", TagName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
関数が処理されるとき、プロパティ値を保持する論理型タグ変数

備考

分散アラーム表示オブジェクトの停止ステータスを含む、または変更する読み取り／書き込みドットフィールドです。アラーム表示オブジェクトが停止されたとき、表示されるアラームは更新することも、新しいアラームを追加することもできません。停止によって、アラームが点滅するかどうかに影響は与えられません。

データタイプ

論理型（読み取り／書き込み）

有効値

0 = 停止がオフ

1 = 停止がオン

例

以下のステートメントは、AlmFreeze 論理型タグ変数からの値によって "AlmObj_1" の Freeze プロパティを設定します。

```
SetPropertyD("AlmObj_1.Freeze",AlmFreeze);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

GetPropertyD()、SetPropertyD()

.PrevPage ドットフィールド

分散アラーム表示オブジェクトを 1 ページ（アラームが全体に表示された 1 画面）上にスクロールします。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]SetPropertyD("ObjectName.PrevPage",0);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

備考

このプロパティが設定されると、分散アラーム表示オブジェクトで前のページが表示されます。リストの先頭に達した場合を除いて、前のページが表示された後で、変数が自動的に 1 に設定されます。リストの先頭に達した場合、値は 0 のままです。

データタイプ

論理型（読み取り／書き込み）

参照項目

GetPropertyD()、SetPropertyD()、.NextPage、.PageNum、.TotalPages

.NextPage ドットフィールド

分散アラーム表示オブジェクトを 1 ページ（アラームが全体に表示された 1 画面）下にスクロールします。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]SetPropertyD("ObjectName.NextPage",0);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

備考

このプロパティが設定されると、分散アラーム表示オブジェクトで次のページが表示されます。リストの末尾に達した場合を除いて、次のページが表示された後で、変数が自動的に 1 に設定されます。リストの先頭に達した場合、値は 0 のままです。

データタイプ

論理型（読み取り／書き込み）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

GetPropertyD()、SetPropertyD()、.PrevPage、.PageNum、.TotalPages

アラーム統計とカウントの表示

以下の関数とドットフィールドを使用して、現在の分散アラーム表示オブジェクトに関する統計情報を表示します。

- almShowStats() 関数
- .PageNum ドットフィールド
- .TotalPages ドットフィールド
- .NumAlarms ドットフィールド
- .ProvidersReq ドットフィールド
- .ProvidersRet ドットフィールド

almShowStats() 関数

指定した分散アラーム表示オブジェクトの［**アラームの統計**］ダイアログ ボックスを表示します。

カテゴリ

アラーム

構文

```
[Result=]almShowStats(ObjectName);
```

引数

ObjectName
アラーム オブジェクトの名前です。たとえば、AlmObj_1 です。

例

```
almShowStats("AlmObj_1");
```

.PageNum ドットフィールド

アラーム オブジェクトで表示されている現在のページ番号が含まれます。

カテゴリ

アラーム

構文

```
[ErrorNumber=]GetPropertyI("ObjectName.PageNum",TagName);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
分散アラーム表示オブジェクトから現在表示されているページ番号を保持する整数型タグ変数

備考

この読み取り専用ドットフィールドは、指定された分散アラーム表示オブジェクトで現在表示されているページ番号を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データタイプ

整数型（読み取り専用）

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトから現在表示されているページ番号を **AlarmPage** 整数型タグ変数に返します。

```
GetPropertyI("AlmObj_1.PageNum",AlarmPage);
```

参照項目

GetPropertyI()、.NextPage、.PrevPage、.TotalPages

.TotalPages ドットフィールド

分散アラーム表示オブジェクトの合計ページ数が含まれます。

カテゴリ

アラーム

構文

```
[ErrorNum=]GetPropertyI("ObjectName.TotalPages", TagName);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
指定された分散アラーム表示オブジェクトに含まれるアラーム ページの合計数を取得する整数型タグ変数

備考

このドットフィールドは、指定された分散アラーム表示オブジェクトに含まれるアラーム ページの合計数を返します。1 ページは、指定時間に画面のオブジェクトに表示されるすべてのアラームと等しくなります。

データタイプ

整数型（読み取り専用）

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトに含まれるページの合計数を Alm**TotalPages** 整数型タグ変数に返します。

```
GetPropertyI( "AlmObj_1.TotalPages",AlmTotalPages);
```

参照項目

GetPropertyI()、NextPage、PrevPage、PageNum

.NumAlarms ドットフィールド

分散アラーム表示オブジェクト内のアラーム数が含まれます。

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
[ErrorNum=]GetPropertyI("ObjectName.NumAlarms", Tagname);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
指定された分散アラーム表示オブジェクトに登録されるアラームの現在数を保持する整数型タグ変数です。これには、表示されるアラームだけでなく、登録されているすべてのアラームが含まれます。

備考

この読み取り専用ドットフィールドは、分散アラーム表示オブジェクト内のアラームの合計数を返します。

データタイプ

整数型（読み取り専用）

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトによって使用されるアラームの現在数を **AlarmCount** 整数型タグ変数に返します。

```
GetPropertyI("AlmObj_1.NumAlarms",AlarmCount);
```

参照項目

GetPropertyI()

.ProvidersReq ドットフィールド

指定された分散アラーム表示オブジェクトによって使用される現在のクエリーで必要とされるアラーム プロバイダ数が含まれます。

カテゴリ

アラーム

構文

```
[ErrorNumber=]GetPropertyI( "ObjectName.ProvidersReq",TagName);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
指定された分散アラーム表示オブジェクトに登録されるアラーム プロバイダの現在数を保持する整数型タグ変数です。これには、表示されるアラームだけでなく、登録されているすべてのアラームが含まれます。

データタイプ

整数型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のステートメントは、分散アラーム表示オブジェクト "AlmObj_1" によって使用される現在のクエリーで必要とされるアラーム プロバイダ数を返します。この値は、TotalProv 整数型タグ変数に書き込まれます。

```
GetPropertyI("AlmObj_1.ProvidersReq",TotalProv);
```

参照項目

GetPropertyI()、.ProvidersRet

.ProvidersRet ドットフィールド

指定された分散アラーム表示オブジェクトによって使用される現在のクエリーによって返されるアラーム プロバイダ数が含まれます。

カテゴリ

アラーム

使用法

```
[ErrorNumber=]GetPropertyI ("ObjectName.ProvidersRet",TagName);
```

パラメータ

ObjectName
分散アラーム表示オブジェクトの名前です。たとえば、AlmObj_1 です。

TagName
指定された分散アラーム表示オブジェクトにアラームを正常に返したアラーム プロバイダ数を保持する整数型タグ変数

備考

この読み取り専用ドットフィールドには、指定された分散アラーム表示オブジェクトによって使用される現在のクエリーによって返されるアラーム プロバイダ数が含まれます。

データタイプ

整数型（読み取り専用）

例

以下のステートメントは、AlmObj_1 分散アラーム表示オブジェクトにアラームを正常に返したアラーム プロバイダ数を返します。この値は、**RetProv** 整数型タグ変数に書き込まれます。

```
GetPropertyI("AlmObj_1.ProvidersRet",RetProv);
```

参照項目

GetPropertyI()、.ProvidersReq

エラーの説明

以下の表は、エラー番号について説明しています。この表に存在しない番号が返された場合、そのエラーは不明なエラーです。

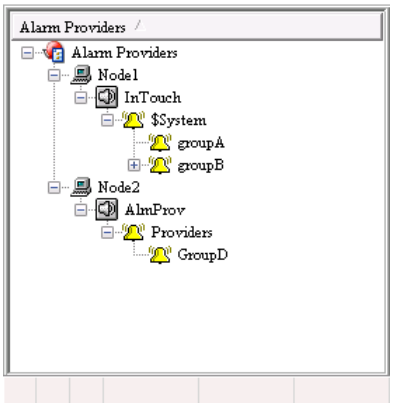
エラー番号	説明
0	成功

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

エラー番号	説明
-1	一般エラー
-2	使用できるメモリが不十分です
-3	プロパティは読み取り専用です
-4	指定したアイテムはすでに存在します
-5	オブジェクト名が不明です
-6	プロパティ名が不明です

アラーム階層の表示

Alarm Tree Viewer ActiveX コントロールには、アラーム クエリーによって選択されたアラーム プロバイダのアラーム グループ階層が表示されます。Alarm Tree Viewer コントロールに表示されるアイテムには、アラーム プロバイダ、ノード、およびグループが含まれます。



Alarm Tree Viewer コントロールを使用することによって、Alarm Viewer コントロールの操作性が向上します。Alarm Tree Viewer コントロールでオペレータがアラーム プロバイダを選択する際に Alarm Viewer コントロールが新しいアラーム プロバイダをクエリーするようにスクリプトを作成できます。

Alarm Tree Viewer コントロールをどのように表示するか、またどのデータを表示するかを設定できます。詳細については、「Alarm Viewer コントロールの設定」を参照してください。

Alarm Tree Viewer コントロールの設定が終了したら、表示されているデータを以下のように変更できます。

- 名前によるデータのソート
- ツリーの更新
- 別のクエリーの実行

ActiveX コントロールの詳細については、「ActiveX コントロール」を参照してください。

Alarm Viewer コントロールの設定

Alarm Tree Viewer コントロールに対して、以下の内容を設定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 色を含む一般的なコントロール外観
- テキスト フォント
- 自動リフレッシュ
- ランタイムにユーザーがアクセスできる機能
- 表示するプロバイダとグループの選択
- カスタム保存クエリー
- アラーム グループのソート順

これらのオプションは、WindowMaker で Alarm Tree Viewer コントロールの実行中に設定できます。

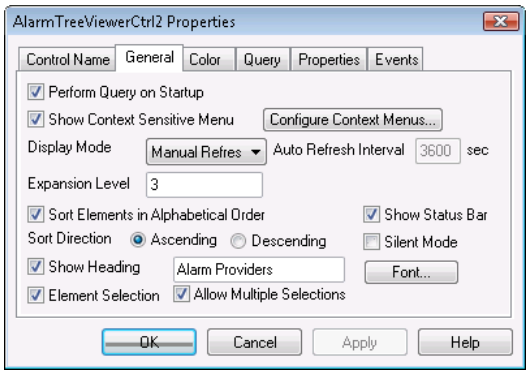
外観と色の設定

Alarm Tree Viewer コントロールの表示外観を設定する場合、以下の操作を行えます。

- ステータス バーを含めます。
- カラム ヘッダーを含めます。
- 表示要素の色を設定します。

外観を設定する

1. Alarm Tree Viewer コントロールを右選択して、［プロパティ］を選択します。 ［**AlarmTreeViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. **全般** タブを選択します。

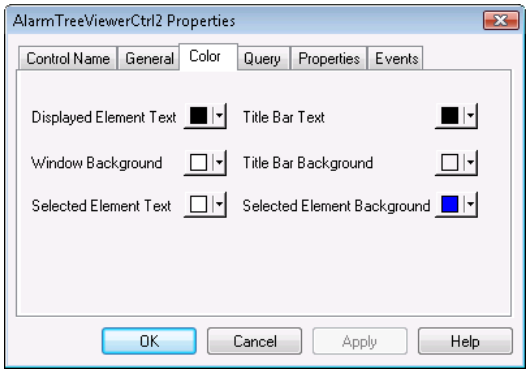


3. ランタイム ユーザーに対する Alarm Tree Viewer コントロールの表示方法を設定します。以下のいずれかを実行します。
   - デフォルトのクエリー プロパティを使用してツリーを自動的に更新するには、［**起動時にクエリー実行**］チェック ボックスをオンにします。それ以外の場合、ユーザーは［リフレッシュ］コマンドを実行して、ツリーを更新する必要があります。
   - ショートカット メニューを有効にするには、［**コンテキスト メニューを表示**］チェック ボックスをオンにします。メニューに表示するコマンドを設定するには、［コンテキスト メニューの設定］を選択します。詳細については、「ランタイムでユーザーがアクセスできる機能の管理」を参照してください。
   - ［**表示モード**］リストで、ツリーを更新する方法を選択します。自動更新の場合は、［**自動更新間隔**］ボックスに更新間隔を入力します。範囲は 5 ～ 32767 秒です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**拡張レベル**］ボックスに、ツリーの拡張レベル数を入力します。この設定により、コントロールを手動で更新するときに、アラーム ツリーが開くアラーム グループの分岐レベルが決定されます。値 1 はプロバイダのみ表示し、値 2 はプロバイダの直接アラーム グループを表示します。
- ツリー要素をアルファベット順にソートするには、［**アルファベット順に要素をソート**］チェック ボックスをオンにします。また、［**昇順**］または［**降順**］を選択して、ソート順を設定します。
- 階層の上に見出しを表示するには、［**見出しの表示**］チェック ボックスをオンにします。ボックスには、見出しバーのテキストを入力します。
- Alarm Tree Viewer コントロールの一番下にステータス バーを表示するには、［**ステータス バーの表示**］チェック ボックスをオンにします。
- ツリーのフォント プロパティを設定するには、［フォント］を選択します。Windows 標準の［**フォント**］ダイアログ ボックスが表示されます。
- ユーザーがツリーで要素を選択できるようにするには、［**要素の選択**］チェック ボックスをオンにします。
- Ctrl キーおよび Shift キーを使用して、1 つまたは複数の要素を選択できるようにするには、［**複数選択可**］チェック ボックスをオンにします。
- Alarm Tree Viewer コントロールでランタイムのエラー メッセージが表示されないようにするには、［サイレント モード］チェック ボックスをオンにします。エラー メッセージは、常にロガーに送信されます。

4. **適用** を選択します。
5. **色**タブを選択します。



1. パレットのボタンを選択して、Alarm Tree Viewer コントロールの表示要素に色を割り当てます。

   タイトル バーのテキスト、ウィンドウの背景、選択された要素テキスト、および選択された要素背景の色を設定できます。

2. **適用** を選択します。

フォントの設定

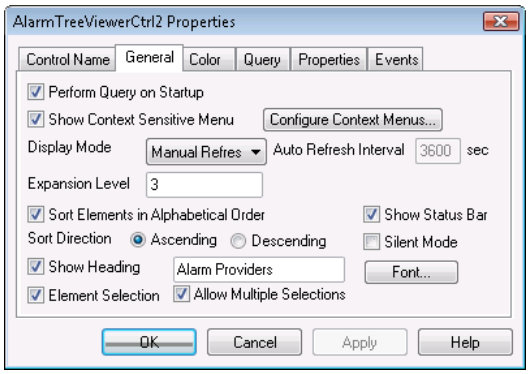
Alarm Tree Viewer コントロールのテキスト外観を設定できます。

フォントを設定する

1. Alarm Tree Viewer コントロールを右選択して、［**プロパティ**］を選択します。［**AlarmTreeViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. **全般** タブを選択します。



3. **フォント**を選択します。Windows 標準の［**フォント**］ダイアログ ボックスが表示されます。フォントを設定して、［**OK**］を選択します。
4. **[OK]** を選択します。

自動更新に設定する

Alarm Tree Viewer コントロールをランタイム時に自動更新するように設定できます。それ以外の場合、オペレータは手動で Alarm Tree Viewer コントロールを更新する必要があります。

自動更新に設定する

1. Alarm Tree Viewer コントロールを右選択して、［**プロパティ**］を選択します。

   ［**AlarmTreeViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. ［**全般**］タブを選択します。
3. ［**表示モード**］リストで、ツリーを更新する方法（［**手動更新**］または［**自動更新**］）を選択します。自動更新の場合は、［**自動更新間隔**］ボックスにツリーの更新間隔を入力します。範囲は 5 ～ 32767 秒です。
4. **適用** を選択します。

Alarm Tree View の表示更新のチューニング

Alarm Tree View の表示を更新するとき、ツリーに一覧表示されるアラーム プロバイダがクエリーに含まれたアラーム プロバイダに一致しないことがあります。これは、リモート アラーム プロバイダにアラーム グループの非常に大きな階層がある場合、またはアラーム プロバイダとのネットワーク接続が遅い場合に発生することがあります。

Alarm Tree View の表示を適切に更新するには、InTouch.ini ファイルに入力してチューニングできる 3 つの設定があります。これらの設定は、クエリーに対する完全な応答を待機するまでの時間、およびその間にアラーム クエリーを再試行する頻度を指定します。

デフォルトでは、これらの設定は InTouch.ini ファイルに含まれていないので、手動で入力してチューニングする必要があります。設定を明示的に InTouch.ini ファイルに入力しない場合、デフォルト値が使用されます。

AlarmTreeFastRetryMax

この設定は、クエリーが送信された直後に、そのツリー データ取得の最後の高速試行（毎秒 1 回）が実行されるまでの時間を決定します。値は秒単位です。

使用できる値：1 ～ 32767

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

デフォルト値：10

例：

```
[InTouch]
AlarmTreeFastRetryMax=5
```

AlarmTreeSlowRetryInterval

高速再試行が完了した後、この設定は、ツリー データ取得の追加の再試行を実行する頻度（秒）を決定します。

使用できる値：1 ～ 32767

デフォルト値：5

例：

```
[InTouch]
AlarmTreeSlowRetryInterval=10
```

AlarmTreeTotalRetryMax

この設定は、両方の種類の再試行を実行する最大期間（秒）を指定します。

使用できる値：1 ～ 32767

デフォルト値：30

例：

```
[InTouch]
AlarmTreeTotalRetryMax=60
```

デフォルト設定の再試行動作

たとえば、デフォルトの再試行設定およびデフォルトの最大再試行期間（30 秒）を使用する場合は、以下の処理が行われます。

- 高速再試行間隔の間、10 秒間、1 秒ごとに取得の再試行が行われます。
- 低速再試行間隔の間、5 秒間、1 秒ごとに取得の再試行が行われます。

表示更新が完了したと見なされる場合

表示は、以下の時点まで引き続き更新されます。

- アラーム クエリーで指定されたすべてのプロバイダーへの接続が完了した時点
- すべてのプロバイダの階層ツリーが表示された時点

表示の更新回数は、クエリーが送信されるたびに変更された値またはデフォルト値にリセットされます。

ランタイム時における機能へのアクセスの管理

Alarm Tree Viewer コントロールには、オペレータがランタイム中にコントロールを右クリックして開くことができるショートカット メニューが含まれています。メニューに表示するコマンドは設定できます。

ランタイムのショートカット メニューを設定する

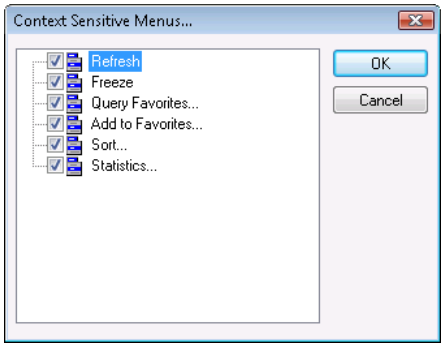
1. Alarm Tree Viewer コントロールを右選択して、［**プロパティ**］を選択します。

   ［**AlarmTreeViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［全般］タブを選択します。
3. ショートカット メニューを有効にするには、［**コンテキスト メニューを表示**］チェック ボックスをオンにします。
4. ［**コンテキスト メニューの設定**］を選択します。

   ［**コンテキスト メニュー**］ダイアログ ボックスが表示されます。



5. ショートカット メニューに表示する各コマンドのチェック ボックスをオンにします。ショートカット コマンドは少なくとも 1 つ選択する必要があります。

コマンド	説明
最新の情報への更新	Alarm Tree Viewer コントロールに表示されるデータを更新します。
停止	ツリーの停止/停止解除のモード切り替えを可能にします。
クエリー設定	使用できるリストからクエリー設定を選択するための、［**アラーム クエリー**］ダイアログ ボックスを表示します。
設定ファイルに追加	［**クエリーの追加**］ダイアログ ボックスで、新しいクエリーを追加できます。
ソート	［**ソート**］ダイアログ ボックスを表示して、Alarm Tree Viewer コントロールのデータを昇順または降順にソートします。
統計	Alarm Tree Viewer コントロールに現在表示されている取得されたアラーム プロバイダのパーセンテージを表示する［**アラームの統計**］ダイアログ ボックスを表示します。

6. ［**OK**］を選択し、［**コンテキスト メニュー**］ダイアログ ボックスを閉じます。
7. **適用** を選択します。

表示するプロバイダとグループの設定

Alarm Tree Viewer コントロールに属するアラーム プロバイダとグループのアラーム クエリーを設定します。アラーム クエリーは、スペースで区切られたアラーム プロバイダのリストです。アラーム プロバイダの有効な構文は以下のとおりです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム プロバイダへのフル パス:

\\Node\ProviderName

ローカル アラーム プロバイダへのパス:

\ProviderName

複数のクエリーの場合、各クエリーをスペースで区切ります。以下に例を示します。

\inTouch \\Node17\InTouch \MyNode\InTouch

デフォルトのアラーム クエリーは \InTouch です。アラーム クエリーに対してタグ変数を使用することはできません。

複数のアラーム グループをクエリーして、その後で 1 つまたは複数のグループの配置を解除した場合、Alarm Tree Viewer コントロールでは自動的に更新は行われず、これらのグループは表示から削除されません。また、アラーム プロバイダの登録を解除するには、アラーム プロバイダを停止して再起動する必要があります。

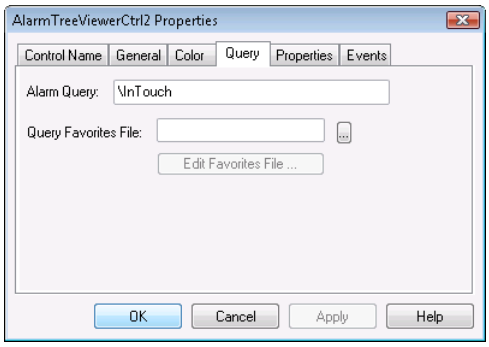
アラーム クエリーで \Galaxy を指定して Galaxy をクエリーする場合、Galaxy に配置されたすべての InTouch アラーム プロバイダが表示されます。以下に例を示します。

\\Node\Galaxy!Area[名前]

同じ名前のグループが含まれる複数のアラーム プロバイダを持つノードから情報をクエリーする場合、ツリーの最後のアラーム プロバイダに対するレコードが表示されます。

アラーム クエリーを設定する

1. Alarm Tree Viewer コントロールを右選択して、[**プロパティ**] を選択します。[**AlarmTreeViewerCtrl のプロパティ**] ダイアログ ボックスが表示されます。
2. **[XML]**タブを選択します。



3. [**アラーム クエリー**] ボックス、最初のアラーム クエリーへのパスを入力します。
4. **適用** を選択します。

クエリー設定を使用したカスタム保存クエリーの作成

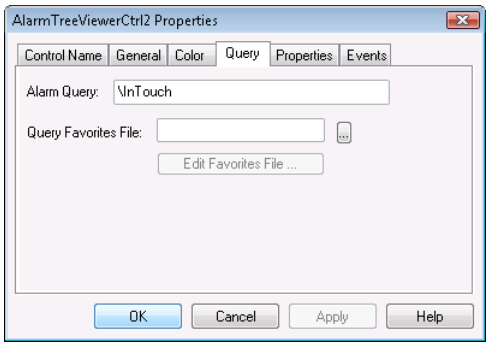
オペレータがショートカット メニューから選択できるクエリー設定のリストを設定できます。

クエリー ファイルは任意のフォルダに保存できます。InTouch アプリケーション フォルダに保存する必要はありません。アラーム クエリー ファイルは、.xml ファイルです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クエリー設定ファイルを設定します。

1. Alarm Tree Viewer コントロールを右選択して、[**プロパティ**] を選択します。[**AlarmTreeViewerCtrl のプロパティ**] ダイアログ ボックスが表示されます。
2. **[XML]**タブを選択します。



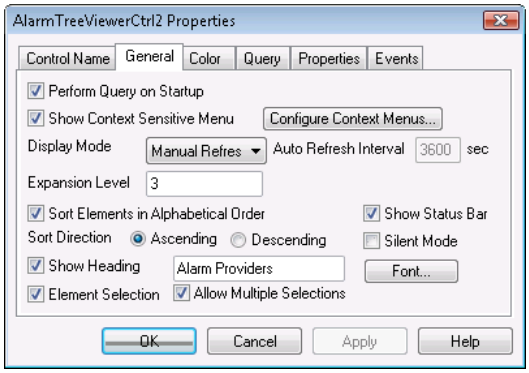
3. クエリー設定ファイルを設定します。
   a. [**クエリー設定ファイル**] ボックスに、ネットワーク パスとファイル名を入力するか、参照ボタンを選択してファイルを参照します。
   b. **クエリー設定**ファイルを編集するには、[**設定ファイルの編集**] ボタンを選択します。[**アラーム クエリー**] ウィンドウが開き、設定ファイルのフィルタを追加、変更、または削除できます。完了したら、[**OK**] を選択して変更を保存し、ウィンドウを閉じます。
4. **[OK]** を選択します。

アラーム グループのソート順の設定

Alarm Tree Viewer コントロールでは、ノードとアラーム グループをアルファベットの昇順または降順のいずれかで表示できます。

アラーム グループのソート順を設定する

1. Alarm Tree Viewer コントロールを右選択して、[プロパティ] を選択します。[**AlarmTreeViewerCtrl のプロパティ**] ダイアログ ボックスが表示されます。
2. [**全般**] タブを選択します。



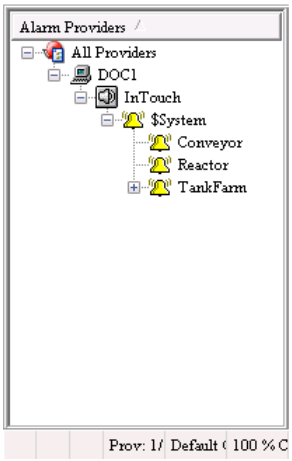
3. アラーム グループをアルファベット順で一覧表示するには、[**アルファベット順に要素をソート**] チェック ボックスをオンにします。
4. [**昇順**] または [**降順**] を選択して、ソート順を指定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. **[OK]** を選択します。

ランタイムでの Alarm Tree Viewer コントロールの使用

Alarm Tree Viewer コントロールを使用して、アラーム プロバイダとアラーム グループの階層をナビゲートできます。



Alarm Tree Viewer コントロールには、複数のノードとアラーム プロバイダを表示できます。

- ノードはコンピュータ アイコンで示されます。
- アラーム プロバイダはスピーカー アイコンで示されます。
- アラーム グループはベル アイコンで示されます。

1 つまたは複数のアラーム グループを選択した状態で、Alarm Tree Viewer コントロールや Alarm DB View コントロールで使用できるアラームのクエリーを生成できます。複数のアラーム グループを選択するには、Shift キーを押したままグループを選択します。すべてのグループの選択を解除するには、空白領域を選択します。

コントロールが設定されている方法に基づいて、ランタイムでのショートカット メニューに以下の 1 つまたは複数のコマンドが表示されます。

- **リフレッシュ** – アラームの手動による更新を施行します。
- **停止** – アラームの更新を停止します。
- **クエリー設定** – ［**アラーム クエリー**］ダイアログ ボックスが開き、以前定義されたアラーム クエリーのリストからアラーム クエリーを選択できます。
- **設定ファイルに追加** – 選択したグループ（存在する場合）に基づいてクエリー文字列が入力された状態の［**クエリーの追加**］ダイアログ ボックスが開きます。
- ［**ソート**］ – アラーム グループをアルファベットの昇順または降順でソートするオプションを持つ［**ソート**］ダイアログ ボックスが開きます。
- ［**統計**］ – 取得済みのアラーム プロバイダのパーセンテージを示す［**アラームの統計**］ダイアログ ボックスが開きます。

Alarm Tree Viewer コントロールのステータス バー情報の理解

Alarm Tree Viewer コントロールのステータス バーはウィンドウの一番下に表示され、以下の情報が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 現在のクエリーの名前
- 現在のクエリーの完了状況を示すパーセンテージ

クエリー設定の使用

Alarm Tree Viewer コントロールのショートカット メニューの［クエリー設定］コマンドを使用すると、定義済みのアラーム クエリーのリストから素早くアラーム クエリーを選択して実行できます。また、新しい名前付きクエリーを作成したり、既存のクエリーを編集したり、既存のクエリーを削除したりすることもできます。

アラーム クエリーを選択して実行する

1. ランタイムで Alarm Tree Viewer コントロールを右クリックします。
2. **クエリー設定**を選択します。［アラーム クエリー］ダイアログ ボックスが表示されます。
3. 現在定義されているクエリーのリストから表示する名前付きクエリーを選択します。
4. **[OK]** を選択します。Alarm Tree Viewer コントロールに、選択したクエリーのアラーム グループ情報が表示されます。

Alarm Tree Viewer コントロールの ActiveX プロパティの使用

スクリプトを使用して Alarm Tree Viewer コントロールのプロパティの値を直接設定するか、InTouch タグまたは I/O 参照に割り当てることができます。プロパティの設定の詳細については、「ActiveX コントロールのスクリプト」を参照してください。

以下の表には、Alarm Tree Viewer コントロールのプロパティが一覧表示されています。色の値の設定の詳細については、「ActiveX コントロールの色の設定」を参照してください。

プロパティ名	説明
AddtoFavoritesMenu(お気に入りに追加メニュー)	［設定ファイルに追加］ショートカット メニュー コマンドを有効または無効にします。
AlarmQuery(アラーム クエリ)	最初のアラーム クエリーを表示し、クエリーを変更できるようにします。有効な構文は、\\<ノード>\<プロバイダ> または \<プロバイダ> です。
ElementSelection (エレメント選択)	ランタイム中にオペレータがツリーの要素を選択することができるかどうかを制御します。
ExpansionLevel(拡張レベル)	コントロールを手動で更新するときに、アラーム ツリーが開く分岐レベルを設定します。値 1 はプロバイダのみ表示し、値 2 はプロバイダの直接アラーム グループを表示します。
フォント	コントロールに表示されるレコードと見出しのフォントを取得または設定します。
FreezeMenu(フリーズメニュー)	［停止］メニュー コマンドを有効または無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ名	説明
HeaderText(ヘッダーテキスト)	Alarm Tree Viewer コントロールの見出しに表示されるテキストを取得または設定します。
MultiSelection(複数を選択)	Alarm Tree Viewer コントロールで複数の要素を選択できるようにします。
QueryFavoritesFile(クエリ お気に入りファイル)	クエリー設定ファイルの名前を取得または設定します。
QueryFavoritesMenu(クエリ お気に入りメニュー)	［クエリー設定］メニュー コマンドを有効または無効にします。
QueryStartup(クエリスタートアップ)	選択した場合、デフォルトのクエリー プロパティを使用して、Alarm Tree Viewer コントロールを自動的に更新します。選択しない場合、Alarm Tree Viewer コントロールを更新するには再クエリーする必要があります。
RefreshInterval(更新間隔)	コントロールの自動更新間隔を秒単位で取得します。
RefreshMenu(更新メニュー)	ショートカット メニューに［リフレッシュ］コマンドを表示するかどうかを決定する値を取得または設定します。
SelTextBackColor(テキスト背景色の選択)	選択した要素の背景色を取得または設定します。
SelTextColor(テキスト色の選択)	選択した要素のテキスト色を取得または設定します。
ShowContextMenu	ショートカット メニューを有効または無効にします。
ShowHeading(見出しの表示)	Alarm Tree Viewer コントロールのタイトル バーを表示または非表示にします。
ShowStatusBar(ステータスバーの表示)	ステータス バーが表示されるかどうか決定する値を取得するか、設定します。
SilentMode(サイレントモード)	コントロールがサイレント モードであるかどうかを決定する値を取得または設定します。
SortElements(並べ替えエレメント)	Alarm Tree Viewer コントロールのソートを有効または無効にします。
SortMenu(並べ替えメニュー)	［ソート］メニュー コマンドを有効または無効にします。
SortOrder(ソート順)	ソート順を取得または設定します。可能な値は「昇順」および「降順」です。それぞれ、0 および 1 で示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ名	説明
StatsMenu(統計メニュー)	［統計］メニュー コマンドを有効または無効にします。
TextColor(テキスト色)	Alarm Tree Viewer コントロールのテキスト色を取得または設定します。
ﾀｲﾄﾙ背景色	タイトル バーの背景色を取得または設定します。ShowHeading プロパティが設定されている場合のみ使用できます。
ﾀｲﾄﾙ前景色	タイトル バーの前景色を取得または設定します。ShowHeading プロパティが設定されている場合のみ使用できます。
ｳｨﾝﾄﾞｳの色	Alarm Tree Viewer コントロールのウィンドウの背景色を取得または設定します。

Alarm Tree Viewer コントロールの ActiveX メソッドの使用

Alarm Tree Viewer コントロールのメソッドをスクリプトで使用すると、以下の操作を行えます。

- コントロールに関する情報を取得します。
- アラーム階層内の特定のエントリに関する情報を取得します。
- コントロールを停止します。
- クエリー文字列を作成します。
- クエリーを実行します。

メソッドの呼び出しの詳細については、「ActiveX コントロールのスクリプト」を参照してください。

Alarm Tree Viewer コントロールに関する情報を取得する

以下のメソッドを使用して、Alarm Tree Viewer コントロールに関する情報を取得できます。

- AboutBox() メソッド, AboutBox() メソッド, AboutBox() メソッド
- GetElementCount() メソッド, GetElementCount() メソッド

AboutBox() メソッド

Alarm Tree Viewer の［バージョン情報］ダイアログ ボックスを表示します。

GetElementCount() メソッド

ツリーの要素の合計数を取得します。

構文

```
Object.GetElementCount()
```

例

コントロールの名前は AlarmTreeViewerCtrl1 であり、nTag1 は整数型または実数型のタグ変数です。

```
nTag1 = #AlarmTreeViewerCtrl1.GetElementCount();
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Alarm Tree Viewer の特定のエントリに関する情報を取得する

一組のメソッドを使用して、Alarm Tree Viewer コントロールのウィンドウに表示されている要素に関する情報を取得できます。

- CheckElementMembership() メソッド
- GetElementCount() メソッド, GetElementCount() メソッド
- GetElementName() メソッド
- GetElementPath() メソッド
- GetSelectedElementCount() メソッド
- GetSelectedElementName() メソッド
- GetSelectedElementPath() メソッド
- GetSubElementCount() メソッド
- GetSubElementName() メソッド
- GetSubElementPath() メソッド

CheckElementMembership() メソッド

下位のツリー要素が上位のツリー要素の一部であるかどうかを確認します。

構文

```
Object.CheckElementMembership(PathName, DescendantElementName, AncestorElementName)
```

パラメータ

PathName
パスの名前です。たとえば、\InTouch または \\NodeName です。

DescendantElementName
下位要素の名前です。たとえば、GroupA です。

AncestorElementName
上位要素の名前です。たとえば、GroupB です。

GetElementCount() メソッド

ツリーの要素の合計数を取得します。

構文

```
Object.GetElementCount()
```

例

コントロールの名前は AlarmTreeViewerCtrl1 であり、nTag1 は整数型または実数型のタグ変数です。

```
nTag1 = #AlarmTreeViewerCtrl1.GetElementCount();
```

GetElementName() メソッド

インデックスに対応する要素名を取得します。

構文

```
Object.GetElementName(ElementIndex)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

ElementIndex
要素のインデックスです。

例

コントロール名は AlarmTreeViewerCtrl1 であり、StrTag はメッセージ型タグ変数です。

```
StrTag = #AlarmTreeViewerCtrl1.GetElementName(3);
```

GetElementPath() メソッド

インデックスに対応する要素パスを、指定された拡張レベルまで取得します。

構文

```
Object.GetElementPath(ElementIndex, ExpansionLevel)
```

パラメータ

ElementIndex
要素のインデックスです。

ExpansionLevel
拡張レベルです。

例

コントロール名は AlarmTreeViewerCtrl1 であり、StrTag はメッセージ型タグです。インデックス 17 の要素の 4 レベルまでのパスを返します。

```
StrTag = #AlarmTreeViewerCtrl1.GetElementPath(17, 4);
```

GetSelectedElementCount() メソッド

ツリー内で選択した要素の数を取得します。

構文

```
Object.GetSelectedElementCount()
```

例

コントロールの名前は AlarmTreeViewerCtrl1 であり、nTag1 は整数型または実数型のタグ変数です。

```
nTag1 = #AlarmTreeViewerCtrl1.GetSelectedElementCount();
```

GetSelectedElementName() メソッド

Alarm Tree Viewr コントロールで選択した要素の名前を取得します。

構文

```
Object.GetSelectedElementName()
```

例

コントロール名は AlarmTreeViewerCtrl1 であり、StrTag はメッセージ型タグです。

```
StrTag = #AlarmTreeViewerCtrl1.GetSelectedElementName();
```

GetSelectedElementPath() メソッド

選択した要素のパスを、指定した拡張レベルまで取得します。

構文

```
Object.GetSelectedElementPath(ExpansionLevel)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

ExpansionLevel
拡張レベルです。

例

コントロール名は AlarmTreeViewerCtrl1 であり、StrTag はメッセージ型タグ変数です。

```
StrTag = #AlarmTreeViewerCtrl1.GetSelectedElementPath(3);
```

GetSubElementCount() メソッド

指定した要素からのサブ要素の合計数を取得します。

構文

```
Object.GetSubElementCount(Path, ElementName)
```

パラメータ

Path
パスの名前です。次に例を示します。
\\NodeName\InTouch
Path パラメータが空白である場合、Alarm Tree Viewer コントロールは指定した要素名に一致するツリーの最初の要素を検索します。

ElementName
要素の名前です。たとえば、Group1 です。

例

コントロールの名前は AlarmTreeViewerCtrl1 であり、nTag1 は整数型または実数型のタグ変数です。

```
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount("", "Group1" );
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount( "\\NodeName", "Group1" );
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount( "\InTouch", "Group1" );
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount( "\\NodeName\InTouch", "Group1" );
```

GetSubElementName() メソッド

指定した要素の場合、対応するインデックスのサブ要素の名前を取得します。

構文

```
Object.GetSubElementName(Path, ElementName, ElementIndex)
```

パラメータ

Path
パスの名前です。次に例を示します。
\\NodeName\InTouch
Path パラメータが空白である場合、Alarm Tree Viewer コントロールは指定した要素名に一致するツリーの最初の要素を検索します。

ElementName
要素の名前です。たとえば、Group1 です。

ElementIndex
要素のインデックスです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

コントロール名は AlarmTreeViewerCtrl1 であり、StrTag はメッセージ型タグ変数です。

```
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("", "Group1", 1);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("\\NodeName", "Group1", 1);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("\InTouch", "Group1", 1);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("\\NodeName\InTouch", "Group1", 1);
```

GetSubElementPath() メソッド

要素名のインデックスからのサブ要素のパスを、指定した拡張レベルまで取得します。

構文

```
Object.GetSubElementPath(Path, ElementName, ElementIndex, ExpansionLevel)
```

パラメータ

Path
パスの名前です。次に例を示します。
\\NodeName\InTouch
Path パラメータが空白である場合、Alarm Tree Viewer コントロールは指定した要素名に一致するツリーの最初の要素を検索します。

ElementName
要素の名前です。たとえば、Group1 です。

ElementIndex
要素のインデックスです。

ExpansionLevel
拡張レベルです。

例

コントロール名は AlarmTreeViewerCtrl1 であり、StrTag はメッセージ型タグ変数です。

```
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("", "Group1", 1, 3);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("\\NodeName", "Group1", 1, 3);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("\InTouch", "Group1", 1, 3);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("\\NodeName\InTouch", "Group1", 1, 3);
```

ツリーの停止

Freeze() メソッドを使用すると、Alarm Tree Viewer コントロール ツリーを今後発生する変更内容で更新しないようにできます。

Freeze() メソッド

Alarm Tree Viewer コントロール ツリーを停止します。

構文

```
Object.Freeze(Frozen)
```

パラメータ

Frozen
ツリーが更新できるかどうかを制御します。

1 = ツリーを停止します。

0 = ツリーの停止を解除します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

Tag1 はメモリ論理型タグ変数として定義されており、コントロールの名前は AlarmTreeViewerCtrl1 です。

```
Tag1 = 1;
#AlarmTreeViewerCtrl1.Freeze(Tag1);
```

選択によるクエリー文字列の作成

GetAlarmQueryFromSelection() メソッドを使用して、Alarm Tree Viewer コントロールで選択した要素からのクエリー文字列を取得できます。クエリー文字列は、Alarm Viewer コントロールで動的に使用できます。

GetAlarmQueryFromSelection() メソッド

Alarm Tree Viewer コントロールで選択した要素のクエリー文字列を返します。

構文

```
Object.GetAlarmQueryFromSelection()
```

例

コントロール名は AlarmTreeViewerCtrl1 であり、StrTag はメッセージ型タグ変数です。次に例を示します。StrTag は \\NodeName\InTouch\GroupA に設定されます。

```
StrTag = #AlarmTreeViewerCtrl1.GetAlarmQueryFromSelection();
```

クエリーを実行する

メソッドを使用して Alarm Tree Viewer コントロールでクエリーを実行できます。クエリー設定ファイルに保存されている既存のクエリーを取得するかアラーム プロバイダの新しい収集を指定する文字列を設定するメソッドを使用して、Alarm Tree Viewer に対してクエリーを実行できます。

- SetQueryByName() メソッド SetQueryByName()めそっど, SetQueryByName() メソッド SetQueryByName()めそっど
- SetQueryByString() メソッド

SetQueryByName() メソッド SetQueryByName()めそっど

渡されるクエリー名によって指定されている通りに現在のクエリーを設定します。クエリーはクエリー設定ファイルに存在する必要があります。

構文

```
オブジェクト。SetQueryByName(QueryName)
```

パラメータ

クエリ名
クエリー設定を使用して作成されたクエリーの名前です。たとえば、Turbine Queries です。

例

コントロールの名前は AlarmTreeViewerCtrl1 です。

```
#AlarmTreeViewerCtrl1.SetQueryByName("Turbine Queries");
```

SetQueryByString() メソッド

アラーム プロバイダの新しい収集を指定する新しい文字列として現在のクエリーを設定します。

構文

```
Object.SetQueryByString(NewQuery)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

NewQuery
アラーム クエリーが含まれる文字列です。以下に例を示します。
\\RepositoryNode\InTouch

例

コントロールの名前は AlarmTreeViewerCtrl1 です。

```
#AlarmTreeViewerCtrl1.SetQueryByString("\\RepositoryNode\InTouch");
```

メソッドとプロパティを使用するときのエラー処理

Alarm Tree Viewer コントロールのすべてのエラー メッセージは、Logger に送信されます。Alarm Tree Viewer コントロールをサイレント モードで実行するように設定すると、ランタイム エラーは表示されません。

Alarm Tree Viewer コントロール ActiveX イベントを使用したスクリプトのトリガ

マウス クリックやダブルクリックなどの Alarm Tree Viewer コントロール イベントに、QuickScript を割り当てることができます。イベントが発生すると、QuickScript が実行されます。

Alarm Tree Viewer コントロールは、以下のイベントをサポートしています。

- クリック
- ダブルクリック
- 終了
- 起動

Click イベントは ClicknElementID というパラメータを 1 つ持ち、これはランタイム時にクリックされたツリー内の要素を識別します。

DoubleClick イベントは DoubleClicknElementID というパラメータを 1 つ持ち、これはランタイム時にダブルクリックされたツリー内の要素を識別します。

Click イベントおよび DoubleClick イベントの場合、-1 の ElementID が［すべてのプロバイダ］ノードに戻されます。

注 : ユーザー インターフェイス メソッドが StartUp イベントから呼び出される場合、コントロールはまだ表示されていないため、Alarm Tree Viewer コントロールでは無視されます。これらのメソッドには、AboutBox()、CheckElementMembership()、Freeze()、GetAlarmQueryFromSelection()、GetElementCount()、GetElementName()、GetElementPath()、GetSelectedElementCount()、GetSelectedElementName()、GetSelectedElementPath()、GetSubElementCount()、GetSubElementName()、GetSubElementPath()、および Refresh() があります。

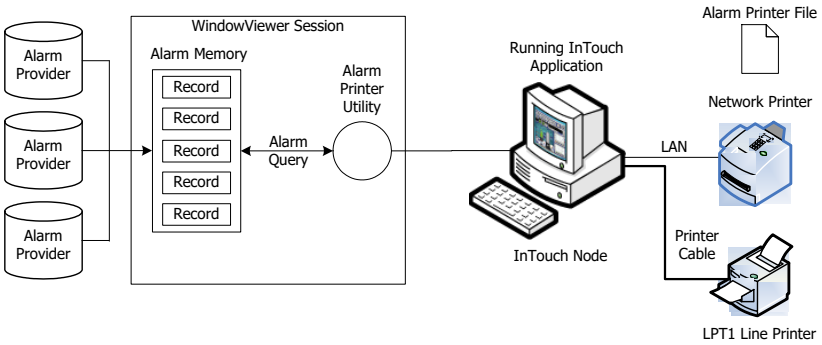
ActiveX イベントのスクリプトの詳細については、「ActiveX コントロールのスクリプト」を参照してください。

アラームの印刷

複数ノードからのアラームを印刷するには、InTouch Alarm Printer を使用します。専用のライン プリンタまたはネットワーク プリンタを使用して、イベントごとにアラーム メモリに保存されるアラーム レコ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ードを印刷できます。また、Alarm Printer を使用して、アラーム レコードをファイルに保存することもできます。



分散アラーム システムは、特定のイベントが発生するごとにライン プリンタでそれらのイベントを印刷するように設定できます。通常、致命的なエラーが発生した場合にすぐにアラームを印刷して、情報を記録します。一般的に、シリアル ポートまたはパラレル ポート経由で InTouch アプリケーションを実行中のコンピュータに直接接続されているドット マトリックス プリンタを使用します。Windows のネットワーク プリンタとレーザー プリンタは、実際に 1 枚のページを印刷する前に全ページがメモリに保持されるため、破壊的なイベントのデータ収集には適切ではありません。

アラーム印刷とログの設定

Alarm Printer のインスタンスは複数実行できます。Alarm Printer の各インスタンスは別のプリンタに印刷するように設定され、個別のアラーム クエリーで設定される必要があります。

Alarm Printer の個別のインスタンスが、特定の優先度範囲でアラームを印刷するように設定することもできます。たとえば、1 つの Alarm Printer インスタンスで優先度の高いアラームだけを印刷し、別のインスタンスで優先度の低いアラームだけを印刷するように設定できます。同様に、Alarm Printer の 1 つのインスタンスを使用して、工場の 1 つの領域からのアラームを印刷し、別のインスタンスでは別の領域からのアラームを印刷するように設定できます。

Alarm Printer 設定を拡張子 .alc を持つファイルに保存できます。必要な数の設定ファイルを作成できます。Alarm Printer では、実行中の Alarm Printer の各インスタンスに対して個別の設定ファイルが使用されます。

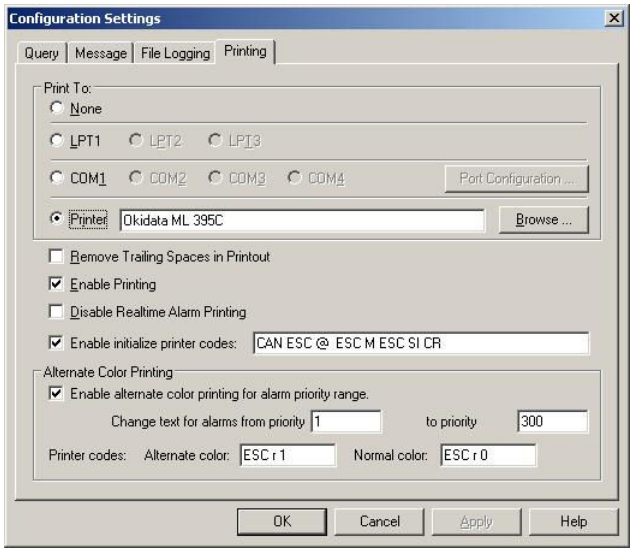
プリンタを設定する

Alarm Printer で使用するプリンタを指定する必要があります。また、アラーム発生時にアラームを印刷するかどうかを選択する必要があります。アラームを印刷する場合は、専用プリンタを使用します。ローカルに接続されたプリンタまたはネットワークに接続されたプリンタを使用できます。任意の Windows プリンタを Alarm Printer の出力先として使用できます。

プリンタを設定する

1. Alarm Printer ユーティリティを開きます。以下の手順を実行します。
   a. WindowMaker の［**ツール**］ビューで、［**アプリケーション**］を展開します。
   b. ［**Alarm Printer**］をダブルクリックします。
2. メニュー バーで、［**設定**］を選択します。［**プロパティの設定**］ダイアログ ボックスが表示されます。
3. ［**印刷**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［出力先］領域で、アラーム プリンタへの接続を選択します。
   - プリンタを使用しない場合は、［なし］を選択します。
   - InTouch アプリケーションを実行中のコンピュータにパラレル ポート経由で接続されたプリンタを使用する、［**LPT1-3**］を選択します。
   - InTouch アプリケーションを実行中のコンピュータにシリアル ポート経由で接続されたプリンタを使用する、［**COM1-4**］を選択します。［**ポート設定**］を選択して、［**COM のプロパティ**］ダイアログ ボックスを表示し、選択した COM ポートに割り当てられたデフォルト値を変更します。
   - InTouch アプリケーションを実行中のコンピュータにネットワーク経由で接続されたプリンタを使用する、［**プリンタ**］を選択します。ボックスに、プリンタの名前を入力するか、［**参照**］を選択して、使用可能なプリンタを選択します。

   注：必要なプリンタが表示されない場合は、Windows のプリンタの追加ウィザードを使用してプリンタを追加します。

5. プリンタで空白行または空白ページが印刷されないようにする、［**後に続くスペースを削除**］チェック ボックスをオンにします。
6. アラームを印刷する、［**印刷可能**］チェック ボックスをオンにします。
7. Alarm Printer でアラームの発生時にアラームが印刷されないようにする、［**リアルタイムのアラーム印刷を無効にする**］チェック ボックスをオンにします。
8. または、特定の設定でプリンタを初期化し、指定した優先度範囲に合わせて印刷特性を変更するよう印刷コマンド シーケンスを設定します。詳細については、「アラーム印刷コマンドの設定」を参照してください。
9. **[OK]** を選択します。

アラーム印刷コマンドの設定

プリンタを最初に開いたとき（クエリーが開始および停止するたび）にプリンタ コマンド初期化シーケンスを送信するよう Alarm Printer を設定できます。プリンタ コマンド初期化シーケンスを使用してプリンタを設定できます（文字ピッチや左右の印刷マージンなど）。

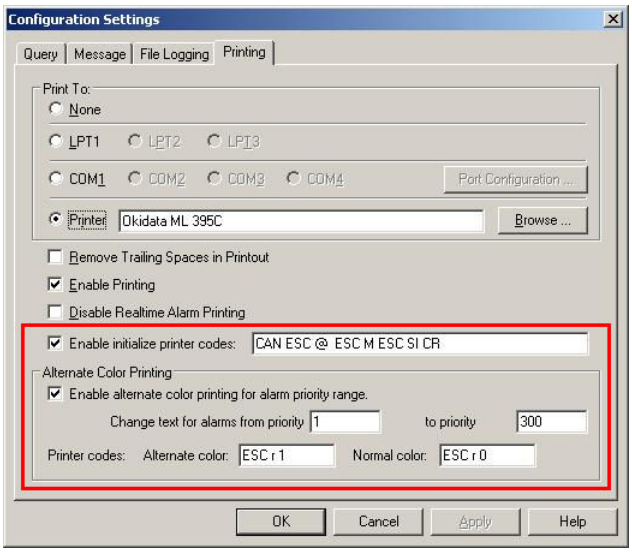
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

また、印刷色でアラームがアラーム優先度の指定範囲内にあるかどうかを示すよう、代替の印刷色および標準の印刷色を設定できます。代替および標準の印刷コマンドでは、必ずしも色を指定する必要はありません。これらのコマンドを使用して、ダブルストライク、強調印刷、下線、斜体、またはその他の特殊印刷特性を有効/無効にし、特定のアラーム優先度をその他の優先度と視覚的に異なるように表示できます。

注：白黒印刷の場合、ダブルストライク印刷の方が強調印刷よりも高速であることがあります。

印刷コマンド設定

[プロパティの設定] ダイアログ ボックスの **[印刷]** タブには、プリンタ コマンド シーケンスを指定する設定があります。



次の表は、これらの設定について説明しています。

設定	説明
プリンタ コード゛の初期化を有効にする:	クエリーが開始および停止するたびに、該当するボックスに入力したコマンド シーケンスがプリンタに送信されます。
アラーム優先度範囲の代替色印刷を有効にする	このオプションを設定すると、指定したアラーム優先度範囲で使用する代替色および標準色のコマンドが渡されます。
アラーム優先度始点のテキストを変更する	少ない数値（優先度の高いアラーム）から大きい数値（優先度の低いアラーム）までのアラームの優先度範囲を入力します。
代替色	アラーム優先度が指定範囲内にあるとき、毎回 1 行のアラーム テキストをプリンタに送信するプリンタ コマンド シーケンスがプリンタに送信されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

設定	説明
標準色	アラーム優先度が指定範囲内にあるとき、毎回 1 行のアラーム テキストをプリンタに送信するプリンタ コマンド シーケンスはプリンタに送信されません。

プリンタ コマンド シーケンスの例

Okidata Microline 395C カラー ドット マトリックス プリンタのプリンタ コマンド シーケンスの例を以下に示します。このコマンド シーケンスの詳細については、プリンタに付属するドキュメントを参照してください。

プリンタ初期化コマンド：

```
CAN ESC @ ESC M ESC SI CR
```

コマンドの要素の説明を以下に示します。

- CAN – プリンタの内部バッファに残っているデータをキャンセルします。
- ESC @ – プリンタをメニュー デフォルトに設定します。
- ESC M – プリンタを 12 CPI（Elite）に設定します。
- ESC SI – プリンタを圧縮に設定して 20 CPI を取得します。
- CR – プリンタ ヘッドを左マージンに移動します。

印刷色を赤に設定する代替色コマンド：

```
Esc r 1
```

印刷色を黒に設定する標準色コマンド：

```
Esc r 0
```

プリンタ コマンド エスケープ シーケンス構文

各プリンタ コマンド エスケープ シーケンスは、1 つ以上のバイトで構成されます。これらのバイトの詳細については、プリンタのドキュメントを参照してください。プリンタが ESC/P or ESC/P2 構文を解析できる場合、バイトの詳細を参照できます。これらの構文に関する情報は、インターネットで公開されています。

一般的に、プリンタ コマンド エスケープ シーケンス テキストの長さに関する情報は提供されていません。パーサは、テキスト トークンのストリームを解釈し、連続するバイト値のシーケンスに変換します。その後、バイト値は、適切なとき（プリンタの初期化時やアラームのプリンタ設定を変更および復元する必要があるときなど）にプリンタにそのまま送信されます。

プリンタ コマンド エスケープ シーケンスをテキストとして指定する場合、各バイト文字はスペース文字で区切る必要があります。スペース文字は、トークンが開始および終了する場所をパーサに示します。たとえば、プリンタのドラフト モードを ESCx0 でアクティブにすることができる場合、以下のコマンド シーケンスを入力します。

```
Esc x 0
```

スペース文字をプリンタに印刷する場合は、そのバイトを表す略記（SP）を使用します。

注：パーサでは大文字と小文字が区別されるので、すべての略記は大文字で入力する必要があります。たとえば、esc や Esc ではなく、ESC を使用します。

以下の種類のアイテムをコマンド シーケンスに含めることができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 制御文字の略記名
- 数値、文字、または印刷可能なその他の文字
- ASCII 文字の 10 進数
- ASCII 文字の 16 進数

16 進数または 10 進数の制御文字および印刷可能文字のリストについては、公開されている ASCII テーブルを参照してください。

制御文字の名前の入力

制御文字はキーボードから容易に入力できないので、制御文字の名前を入力できます。制御文字名のリストについては、公開されている ASCII テーブルの Abbr カラムを参照してください。一般的に ASCII テーブルに一覧表示されていない唯一の文字は特殊文字です。特殊文字は、SP として入力する必要があります。

注：パーサは、一般的な C 言語エスケープ文字（\r、\n、\t など）を処理しません。

印刷可能文字の入力

ボックスに入力して印刷可能文字を入力します。唯一の例外はスペース文字です（16 進数値は 0x20、10 進数値は 32）。スペース文字は SP として入力する必要があります。文字の値は、ASCII テーブルで表現されている番号として解釈されます。この番号はプリンタに送信される値です。

印刷可能文字のリストについては、公開されている ASCII テーブルを参照してください。

文字の 10 進数値の入力

ボックスに入力して文字の 10 進数値を入力します。入力できる最大の 10 進数値は 255 です（16 進数値は 0xFF）。大きすぎる 10 進数値（497 など）を入力しても警告は表示されず、255 が返されます。ネイティブ数値はサポートされません。

文字の 16 進数値の入力

ボックスに入力して文字の 16 進数値を入力します。有効な値の範囲は 0x00 ～ 0xFF です。

印刷するアラームを設定する

Alarm Printer はアラーム メモリをクエリーし、印刷またはログ ファイルに保存するレコードを選択します。クエリーは、以下の内容に基づいて内部アラーム メモリからレコードを選択します。

- アラーム優先度
- 現在のアラーム状況（未確認／確認済み）
- アラーム グループ メンバーシップ

各アラームには、アラームの重要度を表す優先度番号が割り当てられています。アラーム優先度範囲は 1～999 です。最も重要度の高いアラームには優先度 1 が割り当てられています。最も重要度の低いアラームには優先度 999 が割り当てられています。

ネットワークまたはプリンタの接続に失敗すると、Alarm Printer では、すべてのアラームの再印刷は行われず、接続が失敗する前に印刷されていないアラームのみが印刷されます。

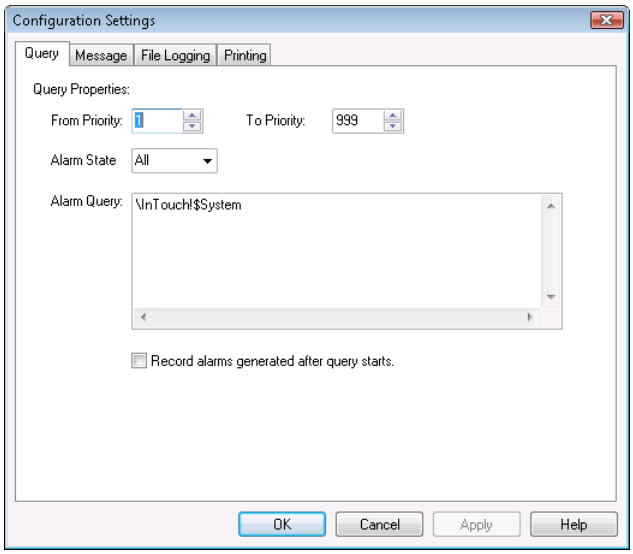
印刷するアラームを設定する

1. Alarm Printer ユーティリティを開きます。以下の手順を実行します。
   a. WindowMaker の［**ツール**］ビューで、［**アプリケーション**］を展開します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

b. ［**Alarm Printer**］をダブルクリックします。

2. メニュー バーで、［**設定**］を選択します。［**プロパティの設定**］ダイアログ ボックスが表示されます。
3. **[XML]**タブを選択します。



4. ［**優先度始点**］ボックスに、優先度が最も高いアラーム値（1～999）を入力します。
5. ［**優先度終点**］ボックスに、優先度が最も低いアラーム値（1 ～ 999）を入力します。
6. ［**アラーム状況**］リストで、アラーム状況を選択します。

選択	表示する内容
すべて	すべてのアラーム
確認	確認されたアラームのみ
未確認	未確認のアラームのみ

7. ［**アラーム クエリー**］ボックスに、1 つまたは複数のアラーム クエリーを入力します。1 つまたは複数のアラーム プロバイダとグループを指定できます。複数のクエリーを区切るにはスペースを使用ます。
8. クエリーが開始した後で発生したアラームのみを含めるには、［**クエリー開始後アラームを記録する**］チェック ボックスをオンにします。アラーム メモリに存在し、Akarm Printer がクエリーを開始する前にトリガされたアラーム レコードは無視されます。
9. **[OK]** を選択します。

印刷とファイル出力の形式の設定

選択した各オプションは、印刷出力で個別のフィールドとして表示されます。指定されたフィールド最大長を超えるデータを含むフィールドは、指定された制限に切り詰められます。

アラーム印刷とファイル出力の形式を設定する

1. Alarm Printer ユーティリティを開きます。以下の手順を実行します。

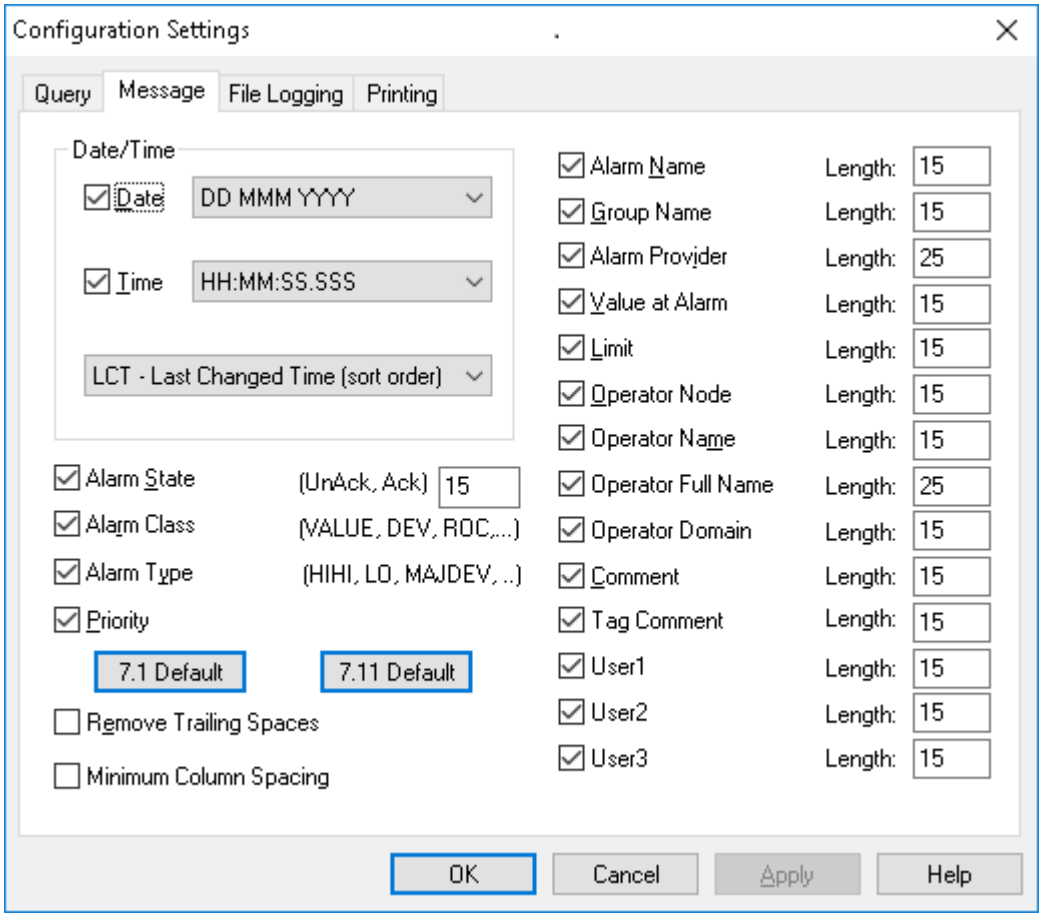
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. WindowMaker の［ツール］ビューで、［アプリケーション］を展開します。

b. ［**Alarm Printer**］をダブルクリックします。

2. メニュー バーで、［設定］を選択します。［プロパティの設定］ダイアログ ボックスが表示されます。

3. [メッセージ] タブを選択します。



4. ［日付／時間］領域で、［日付］チェック ボックスをオンにし、リストから日付形式を選択します。一覧表示される日付形式には、以下のコンポーネントが含まれています。

オプション	説明
DD	日付を表す 2 桁の数（01 ～ 31）
MM	月を表す 2 桁の数（01 ～ 12）
YY	年を表す下 2 桁の数
YYYY	年を表す 4 桁の数
MMM	月を表す 3 文字の略語

5. ［時間］チェック ボックスをオンにし、リストから時間形式を選択します。一覧表示された形式には、以下のコンポーネントが含まれます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オプション	説明
AP	AM/PM 時間形式を選択します。たとえば、午後 3 時は 15:00 と表示されます。AP が指定されていない場合、時間はデフォルトで 24 時間制で表示されます。たとえば、午後 3 時は 15:00 と表示されます。
HH	アラーム/イベントが発生した時間を表す 2 桁の数（00 ～ 23 または 01 ～ 12）
MM	アラーム/イベントが発生した分を表す 2 桁の数（00 ～ 59）
SS	アラーム/イベントが発生した秒を表す 2 桁の数（00 ～ 59）
SSS	アラーム/イベントが発生したミリ秒を表す 3 桁の数

6. アラームの開始時刻に従った、アラーム レコードにアラームが表示される順序を選択します。

オプション	説明
OAT	（元のアラーム時刻）アラームが開始した日付/時刻スタンプです。
LCT	（最終変更時刻）アラームのインスタンスに対して発生した最新の状態変化（アラームの開始、サブステートの変化、平常への復帰、または確認）の日付/時刻スタンプです。
LCT（確認時には OAT）	（最終変更時刻、ただし ACK 時には元のアラーム時刻）- アラームが未確認の間は LCT、確認された後は OAT が使用されます。

7. 印刷するアラーム情報を選択します。

オプション	説明
アラーム状態	アラーム状態を印刷します（未確認、確認など）。
アラーム クラス	アラーム クラスを印刷します（VALUE、DEV、ROC など）。
アラーム タイプ	アラーム タイプを印刷します（HIHI、LO、MAJDEV など）。
優先度	アラーム優先度（1 ～ 999）を印刷します。
7.1 デフォルト（ボタン）	InTouch バージョン 7.1 用の Alarm Printer ユーティリティのデフォルト設定を選択します。
7.11 デフォルト（ボタン）	InTouch バージョン 7.11 用の Alarm Printer ユーティリティのデフォルト設定を選択します。
後に続く空白を削除	実際のフィールド値がそのフィールドに設定されている値より短い場合、印刷フィールドから末尾の余分なスペースを削除します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オプション	説明
最小カラム スペース	より多くのフィールドが印刷されたページに適合するように、カラム間のスペースを削減します。
アラーム名	アラーム名（タグ）を印刷します。［長さ］ボックスに、アラーム名に使用できる文字数を入力します （半角 64 文字まで）。
グループ名	アラームのグループ名を印刷します。［長さ］ボックスに、アラーム グループ名に使用できる文字数を入力します （半角 64 文字まで）。
アラーム プロバイダ	アラーム プロバイダの名前を印刷します。［長さ］ボックスに、アラーム プロバイダ名に使用できる文字数を入力します （半角 64 文字まで）。
アラーム値	タグの値を印刷します。［長さ］ボックスに、アラーム値に使用できる文字数を入力します（半角 32 文字まで）。
しきい値	タグのアラーム制限を印刷します。［長さ］ボックスに、アラームしきい値に使用できる文字数を入力します （半角 32 文字まで）。目的の精度レベルを提供するために十分な文字数である必要があります。
オペレータ ノード	アラーム状況に関連付けられたオペレータ ノードを印刷します。［字数］フィールドに、ユーザー ノードに使用できる文字数を入力します（半角 64 文字まで）。 ターミナル サービス環境で、ユーザー ノードはそれぞれのユーザーがターミナル サービス セッションを確立したクライアント コンピュータの名前です。ノード名を取得できない場合、ノードの IP アドレスが代わりに使用されます。
オペレータ名	アラーム状況に関連付けられたオペレータ ノードを印刷します。［長さ］ボックスに、オペレータ名に使用できる文字数を入力します（半角 16 文字まで）。
オペレータ フルネーム	ログオンしたオペレータのフル ネームを印刷します。
オペレータ ドメイン	アラーム状況に関連付けられているログオンしたオペレータ ドメインを印刷します。
コメント	タグに関連付けられたアラーム コメントを印刷します。［長さ］ボックスに、コメントに使用できる文字数を入力します （半角 131 文字まで）。
タグ コメント	タグのコメントを印刷します。
ユーザー 1	アラームに対応するユーザー定義番号 1 の数値を印刷します。
ユーザー 2	アラームに対応するユーザー定義番号 2 の数値を印刷します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オプション	説明
ユーザー 3	アラームに関連付けられたユーザー定義文字列プロパティの文字列値を印刷します。最大文字数は 131 です。

8. **適用** を選択します。

アラームのログ ファイルの設定

アラーム レコードは、ファイルにログ記録できます。デフォルトでは、Alarm Printer では以下の命名規則に基づいてログ ファイルに名前が割り当てられます。

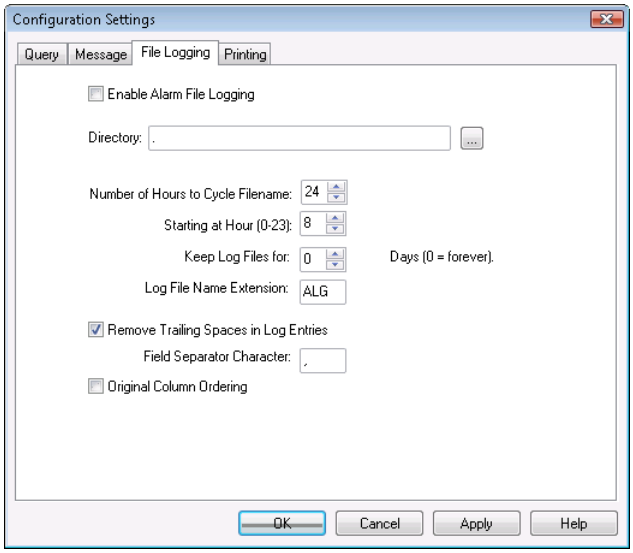
YYMMDDHH.ALG

表記法	説明
YY	ログ ファイルが作成された年を表す下 2 桁
MM	ログ ファイルが作成された月を表す 2 桁の数（01 ～ 12）
DD	ログ ファイルが作成された日を表す 2 桁の数（01 ～ 31）
HH	ログ ファイルが作成された時間を表す 2 桁の数（00 ～ 23）

アラーム レコード ログを設定する

1. Alarm Printer ユーティリティを開きます。以下の手順を実行します。
   a. WindowMaker の［**ツール**］ビューで、［**アプリケーション**］を展開します。
   b.［**Alarm Printer**］をダブルクリックします。
2. メニュー バーで、［**設定**］を選択します。［**プロパティの設定**］ダイアログ ボックスが表示されます。
3.［**ファイル ログ**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. アラーム レコードをログ ファイルに保存する、[**アラーム ファイル ログを有効にする**] チェック ボックスをオンにします。
5. [**ディレクトリ**] ボックスに、パスを入力するか、アラーム ログ ファイルを保存するフォルダの場所を参照します。
6. [**アラーム ログ ファイル名作成サイクル**] ボックスで、アラーム レコードを個別のログ ファイルに保存するために適切な時間数を入力します。

   有効なエントリは 1 ～ 24 です。InTouch アプリケーションを継続的に実行する場合は、長さの等しい日次ログ ファイルを作成するファイル ログ間隔を選択します。たとえば、[**アラーム ログ ファイル名作成サイクル**] を 6 に設定すると、長さの等しい 4 つの日次ログ ファイルが作成されます。
7. [**記録開始時間**] ボックスに、毎日ファイルへのアラーム レコードのログ記録を開始する時間を入力します。

   <1>有効なエントリは</1><2> 0 ～ 23 </2><3>です。</3>

   たとえば、ある石油精製所では 1 日 3 交代で稼動しています。最初の交代は、6:00 に開始します。管理者は、交代ごとにアラーム レコードが記録されることを求めています。この場合、[**アラーム ログ ファイル名作成サイクル**] オプションに 8 を入力します。[**記録開始時間（0-23）**] オプションに 6 を入力します。Alarm Printer では、06:00 ～ 14:00、14:00 ～ 22:00、22:00 ～ 06:00 からのログ ファイルが 3 つ作成されます。
8. [**ログ ファイル保存期間**] ボックスに、ログ ファイルを保持する日数を入力します。

   Alarm Printer では、ログ ファイルは現在の日付から指定した日数だけ保持されます。保存期間を過ぎたログ ファイルは削除されます。ログ ファイルを無制限に保存するには、0 を入力します。
9. [**ログ ファイル名拡張子**] ボックスで、デフォルトの ALG ファイル名拡張子を受け入れるか、別の 3 文字の拡張子をログ ファイルに割り当てます。

   .csv 拡張子を使用すると、ログ ファイルを Excel またはメモ帳に直接インポートできます。
10. ログ ファイル内のエントリの末尾のスペースを削除する、[**後に続くスペースを削除**] チェック ボックスをオンにします。

    ログ ファイル内の各レコードの最後にフィールド区切り文字を指定することもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

11. アラーム表示と同じ順序を保持してファイル レコードをログ記録する、［**オリジナル カラム順序**］チェック ボックスをオンにします。
12. **適用** を選択します。

設定ファイルの保存とロード

WindowMaker から Alarm Printer を起動すると、［**Alarm Printer**］ダイアログ ボックスにデフォルト設定が表示されます。Alarm Printer の設定は、アラーム設定ファイル（.alc）に保存されます。

プリンタ設定は、アラームを印刷する前にロードできるファイルに保存できます。

特定のアラーム設定ファイルは、ランタイムでコマンドラインから開いた場合、または *.alc ファイル名をダブルクリックして開いた場合に表示できます。

新しい Alarm Printer 設定ファイルを作成する

1. Alarm Printer ユーティリティを開きます。以下の手順を実行します。
   a. WindowMaker の［**ツール**］ビューで、［**アプリケーション**］を展開します。
   b. ［**Alarm Printer**］をダブルクリックします。
2. ［**ファイル**］メニューで、［**新規**］を選択し、Alarm Printer をデフォルト値で表示します。
3. メニュー バーで、［**設定**］を選択します。［**プロパティの設定**］ダイアログ ボックスが表示されます。
4. アラームを設定します。
5. ［**ファイル**］メニューで、［**保存**］を選択します。

既存の Alarm Printer 設定ファイルを編集する

1. ［**ファイル**］メニューで、［**開く**］を選択します。
2. 編集する Alarm Printer 設定ファイルを選択します。
3. ファイルを編集します。
4. ［ファイル］メニューで、［**保存**］を選択します。既存ファイルを変更しないで新しいファイルに変更を保存するには、［**名前を付けて保存**］を選択します。

Alarm Printer でのアラームの印刷の概要

各クエリーは、現在開いている Alarm Printer 設定ファイル（.alc）に指定されているすべてのアラームをログ記録します。ファイルが指定されていない場合、Alarm Printer の設定中に現在選択されている設定が使用されます。

Alarm Printer では、複数のクエリーを実行できます。各クエリーは異なるパラメータを使用し、Alarm Printer の個別のインスタンスと関連付けられます。Alarm Printer の 2 つのインスタンスが同じクエリーを実行している場合、エントリは重複されます。

Alarm Printer の実行中は、クエリーを手動で開始または停止できます。印刷を有効にしていることを確認してください。

アラーム クエリーを開始する

- ［**クエリー**］メニューで、［**開始／停止**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム クエリーを停止する

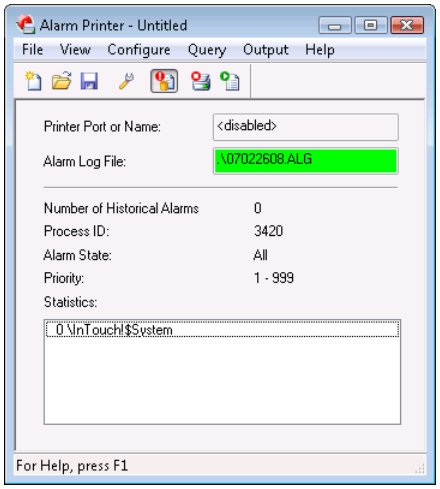
- ［クエリー］メニューで、［開始／停止］を選択します。

アラームのファイルへのログ

Alarm Printer を使用すると、アラーム レコードをファイルにログ記録できます。現段階で、［プロパティの設定］ダイアログ ボックスですでにロギングが設定されている状態となっています。

アラームのファイルへのログ

1. Alarm Printer ユーティリティを開きます。
   a. WindowMaker の［ツール］ビューで、［アプリケーション］を展開します。
   b. ［**Alarm Printer**］をダブルクリックします。
2. 必要に応じて、クエリーを設定し、ロギング用にアラーム レコードを収集します。
3. ［ロギングの切り替え］ボタンを選択します。
4. アラーム クエリーを実行します。



クエリーで収集されるアラーム レコードは、設定済みのログ ファイルに書き込まれます。

特定の設定での Alarm Printer の起動

システムの起動時に次のコマンドをバッチ ファイルで実行することによって、Alarm Printer を自動的に起動し、特定のファイルを開くことができます。

```
ALMPRT.EXE MYQUERY.ALC
```

ここで、`MYQUERY.ALC` は、開かれる Alarm Printer 設定ファイルの名前です。.exe ファイル名拡張子はオプションです。

バッチ ファイルが InTouch HMI がインストールされているフォルダに切り替わることを確認します。

システムの予期しないシャットダウンや再起動によるクエリー データの消失を防ぐために、次のコマンドをバッチ ファイルから実行し、Alarm Printer を自動的に起動して、特定のクエリーを自動的に実行することができます。

```
ALMPRT.EXE -q MYQUERY.ALC
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

コマンドで -q を使用すると、システムの起動時にクエリーが自動的に実行されます

スクリプトを使用した Alarm Printer の制御

スクリプトで Alarm Printer 関数を使用して、アラーム印刷を制御できます。

Alarm Printer 関数は、関数が正常に完了したかどうかを示す整数のエラー コードを返します。以下の表は、エラー コードを示しています。

エラー コード	エラー メッセージ
0	成功
1	インスタンスが見つからないか、実行中ではありません
2	インターフェイスが初期化されていません
3	仮想メモリにアクセスできませんでした
4	無効なエラー コードです
5	すでに実行中のインスタンスが多すぎます
6	結果文字列が長くなりすぎます
7	無効なインスタンス インデックスが関数に渡されました
8	Alarm Printer アプリケーションにメッセージをポストできませんでした
9	Alarm Printer アプリケーションからの応答を待機できませんでした
20	「優先度終点」は「優先度始点」に等しいか、大きい値である必要があります。
21	無効な優先度値です
22	無効なアラーム状況です
23	クエリーが実行しているためコマンドを実行できませんでした
24	クエリー文字列が有効ではありません
25	無効なクエリー処理状況です
26	無効な印刷状況セレクタです
27	Alarm Printer ウィンドウが受け取るコマンドが認識されません
28	クエリーが開始できませんでした

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Alarm Printer インスタンスまたはクエリーの停止および開始

Alarm Printer インスタンスとアラーム クエリーを開始したり、停止したりするには、以下の関数を使用します。

- APUStartInstance() 関数
- APUStartQuery() 関数
- APUStopInstance() 関数
- APUStopQuery() 関数

APUStartInstance() 関数

設定ファイルに指定された値で、Alarm Printer のインスタンスを最小化状態で開始します。

カテゴリ

表示

構文

```
[Result=] APUStartInstance(sFilePath, iTagInstance);
```

引数

sFilePath
設定ファイルのフル パス（入力文字列）

iTagInstance
整数型タグ変数関数は、正常に実行されると、インスタンス番号を返します。

備考

Alarm Printer の最大 16 個のインスタンスを開始できます。この関数は、インスタンス番号（0 ～ 15）を iTagInstance パラメータに書き込みます。新しい Alarm Pronter インスタンスが開始するとき、インスタンス番号は増加します。

このインスタンス番号は、Alarm Printer インスタンスを識別するために、他の Alarm Printer 関数によって使用されます。

この関数は 0 を返します。エラーの場合は、エラー コードを返します。

Alarm Printer プログラムは、アラームの処理をアラーム メモリから自動的に開始しません。**APUStartQuery()** 関数を使用して、インスタンスに対するアラーム メモリのデータの処理を開始します。

例

```
Status = APUStartInstance("c:\MyAlarmCfg\Area1Alarms.alc", Inst);
```

参照項目

APUStartQuery()、APUStopInstance()、APUStopQuery()

APUStartQuery() 関数

アラーム メモリから処理されるレコードの日付および時間しきい値を設定し、クエリーを開始します。

カテゴリ

表示

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
[Result=] APUStartQuery(iInstance, iYear, iMonth, iDay, iHour, iMinute);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iYear
年を表す数

iMonth
月を表す数

iDay
日を表す数

iHour
時間を表す数

iMinute
秒を表す数

備考

クエリーがすでに実行している際、別のクエリーを開始しようとするとエラーが発生します。

すべての日付と時間の値を 0 に設定すると、すべてのアラームが印刷されます。これは、0 が 1900 年 1 月 1 日 の 00:00 時と解釈されるためです。指定された時間と日付はローカル時間です。年を表す値が -1 の場合、日付はコマンドが処理される現在の時間に設定されます。

整数型のエラー コードを返します。

例

```
Status = APUStartQuery(Inst,2007,4,16,22,12);
```

参照項目

APUStartInstance()、APUStopInstance()、APUStopQuery()

APUStopInstance() 関数

Alarm Printer の指定インスタンスを停止します。印刷されるレコードのさらなる追加は停止し、現在実行している印刷クエリーは停止し、プログラムのインスタンスは閉じられます。

カテゴリ

表示

構文

```
[Result=] APUStopInstance(iInstance);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

備考

整数型のエラー コードを返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

```
Status = APUStopInstance(5);
```

参照項目

APUStartInstance()、APUStartQuery()、APUStopQuery()

APUStopQuery() 関数

指定のインスタンスにクエリーの実行を停止するよう要求します。アプリケーションの実行は継続されますが、クエリーは処理されません。**APUStartQuery()** を呼び出すと、インスタンスがクエリーを開始できます。

カテゴリ

表示

構文

```
[Result=] APUStopQuery(iInstance);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

備考

整数型のエラー コードを返します。

例

```
Status = APUStopQuery(5);
```

参照項目

APUStartInstance()、APUStartQuery()、APUStopInstance()

アラーム クエリー情報のクエリー

Alarm Printer 設定ファイル内で設定されたタグ変数優先度と値に基づいてクエリー パラメータを取得するには、以下の関数を使用します。

- APUGetAlarmGroupText() 関数
- APUGetQueryFromPriority() 関数
- APUGetQueryToPriority() 関数
- APUGetConfigurationFilePath() 関数
- APUGetPrinterJobCount() 関数
- APUGetQueryAlarmState() 関数
- APUGetQueryProcessingState() 関数

APUGetAlarmGroupText() 関数

アラーム クエリー アラーム グループ テキストを取得します。

カテゴリ

表示

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
[Result=] APUGetAlarmGroupText(iInstance, sTagGroup);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

sTagGroup
テキスト - アラーム グループ

備考

初期アラーム グループ テキストは、指定したインスタンスを持つ Alarm Printer が使用している .alc 設定ファイルから読み取られます。アラーム グループ テキストは、sTagGroup パラメータの InTouch メッセージ型タグに渡されます。

整数型のエラー コードを返します。

例

TagGroup メッセージ型タグには、関数が実行された後で、以下の値が含まれる可能性があります。
\intouch!$system

```
Status = APUGetAlarmGroupText(Inst,TagGroup);
```

参照項目

APUGetConfigurationFilePath()、APUGetPrinterJobCount()、APUGetQueryAlarmState()、APUGetQueryFromPriority()、APUGetQueryProcessingState()、APUGetQueryToPriority()

APUGetQueryFromPriority() 関数

クエリーの「優先度始点」の値を取得します。

カテゴリ

表示

構文

```
[Result=] APUGetQueryFromPriority(iInstance, iTagPriority);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iTagPriority
「優先度始点」の値を受け取る整数型タグ変数

備考

初期優先度は、指定インスタンスを持つ Alarm Printer が使用している .alc 設定ファイルから読み取られます。「優先度始点」の値は、iTagPriority パラメータの InTouch 整数型タグ変数に渡されます。

整数型のエラー コードを返します。

例

この例では、FromPri は、「優先度始点」の値を含む整数型タグ変数です。たとえば、1 です。

```
Status = APUGetQueryFromPriority(Inst, FromPri);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

APUGetAlarmGroupText()、APUGetConfigurationFilePath()、APUGetPrinterJobCount()、APUGetQueryAlarmState()、APUGetQueryProcessingState()、APUGetQueryToPriority()

APUGetQueryToPriority() 関数

クエリーの「優先度終点」を取得します。

カテゴリ

表示

構文

```
[Result=] APUGetQueryToPriority(iInstance,iPriority);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iPriority
「優先度終点」の値を受け取る整数型タグ変数

備考

APUGetQueryToPriority() 関数を含むスクリプトと同時に、別のクエリーを実行することはできません。「優先度終点」の値は、関数の iPriority パラメータに書き込まれます。これは整数型タグ変数です。

整数型のエラー コードを返します。

例

整数型タグ変数 ToPri は、「優先度終点」の値を受け取ります。たとえば、999 です。

```
Status = APUGetQueryToPriority(Inst,ToPri);
```

参照項目

APUGetAlarmGroupText()、APUGetConfigurationFilePath()、APUGetPrinterJobCount()、APUGetQueryAlarmState()、APUGetQueryFromPriority()、APUGetQueryProcessingState()

APUGetConfigurationFilePath() 関数

クエリーに使用される .alc 設定ファイルのフル ファイル パスを返します。

カテゴリ

表示

構文

```
[Result=] APUGetConfigurationFilePath(iInstance, sTagFilePath);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

sTagFilePath
Alarm Printer インスタンスが使用している設定ファイルのファイル パスの名前を取得するメッセージ型タグ変数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

ファイル パス テキストは、この関数の sTagFilePath パラメータとして指定できるメッセージ型タグ変数に返されます。

整数型のエラー コードを返します。

例

CfgFilePath メッセージ型タグ変数は、Inst 整数型タグ変数に含まれる Alarm Printer インスタンスと関連付けられている設定ファイルのファイル パスを受け取ります。たとえば、c:\MyAlarmCfg\Area1Alarms.alc などです。

```
Status = APUGetConfigurationFilePath(Inst, CfgFilePath);
```

参照項目

APUGetAlarmGroupText()、APUGetPrinterJobCount()、APUGetQueryAlarmState()、APUGetQueryFromPriority()、APUGetQueryProcessingState()、APUGetQueryToPriority()

APUGetPrinterJobCount() 関数

このインスタンスによって使用されるプリンタの最新の Windows プリンタ状況ジョブ カウントを返します。

カテゴリ

表示

構文

```
[Result=] APUGetPrinterJobCount(iInstance, iTagCount);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iTagCount
カウント値を受け取る整数型タグ変数

備考

この関数は、iTagCount パラメータでカウント値を返します。これは整数型タグ変数です。

クエリーが実行していない限り、結果は現在の内容ではありません。アラームが印刷されていない限り、結果は現在の内容ではありません。ジョブ カウントは通常、プリンタが最初に開かれたときに更新され、またアラーム行が印刷されるたびに更新されます。

返されるジョブ カウント値は Windows プリンタに対してのみ有効であり、パラレル ポートまたはシリアル ポートに関連付けられているプリンタに対してはあまり意味を持ちません。

整数型のエラー コードを返します。

例

PJCount は、指定インスタンスからのカウント値を受け取る整数型タグ変数です。

```
Status = APUGetPrinterJobCount(Inst, PJCount);
```

参照項目

APUGetAlarmGroupText()、APUGetConfigurationFilePath()、APUGetQueryAlarmState()、APUGetQueryFromPriority()、APUGetQueryProcessingState()、APUGetQueryToPriority()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

APUGetQueryAlarmState() 関数

クエリーのアラーム状態を返します。

カテゴリ

表示

構文

```
[Result=] APUGetQueryAlarmState(iInstance, iTagState);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iTagState

指定インスタンスに関連付けられたクエリーのアラーム状態を受け取る整数型タグ変数。値の意味は以下のとおりです。

0 = 全て

1 = 確認

2 = 未確認

備考

初期アラーム状態は .alc ファイルから読み取られます。この関数は、整数型タグである関数の iTagState パラメータでアラーム状態を返します。

整数型のエラー コードを返します。

例

AlmState は、0、1、または 2 を含む整数型タグ変数です。

```
Status = APUGetQueryAlarmState(Inst,AlmState);
```

参照項目

APUGetAlarmGroupText()、APUGetConfigurationFilePath()、APUGetPrinterJobCount()、APUGetQueryFromPriority()、APUGetQueryProcessingState()、APUGetQueryToPriority()

APUGetQueryProcessingState() 関数

アラーム クエリー処理の状況を返します。

カテゴリ

表示

構文

```
[Result=] APUGetQueryProcessingState(iInstance, iTagState);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iTagState

関数から処理状況を受け取る整数型タグ変数。値の意味は以下のとおりです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

0 = 停止

1 = 開始

備考

この関数は、整数型タグ変数である iTagState パラメータに整数値を返します。

整数型のエラー コードを返します。

例

ProcState は整数型タグ変数であり、クエリーが実行していない場合は 0、クエリーが実行している場合は 1 に設定されます。

```
Status = APUGetQueryProcessingState(Inst, ProcState);
```

参照項目

APUGetAlarmGroupText()、APUGetConfigurationFilePath()、APUGetPrinterJobCount()、APUGetQueryAlarmState()、APUGetQueryFromPriority()、APUGetQueryToPriority()

インスタンス情報のクエリー

Alarm Printer インスタンスの現在の状況を返すには、以下の関数を使用します。

- APUFindAlarmGroupInstance() 関数
- APUFindFileInstance() 関数
- APUFindPrinterInstance() 関数
- APUGetInstanceCount() 関数
- APUIsInstanceUsed() 関数

APUFindAlarmGroupInstance() 関数

指定アラーム グループ文字列を使用して、Alarm Printer の最初のインスタンスを返します。

カテゴリ

表示

構文

```
[Result=] APUFindAlarmGroupInstance(sGroup, iInstance);
```

引数

sGroup
インスタンス間で検出されるアラーム グループの名前です。

iInstance
指定グループ名を使用する検出されたインスタンスの値を受け取る整数型タグ変数

備考

この関数は、iInstance パラメータとして整数型タグ変数にインスタンス番号を返します。初期アラーム グループ文字列は alc ファイルから読み取られます。インスタンスが見つからない場合、関数はエラー コード（使用可能なインスタンスなし）として 1 を返し、整数型タグ変数パラメータに 0 を書き込みます。

整数型のエラー コードを返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

FoundInstance は、クエリーとして $System を使用する検出された最初のインスタンスの番号を受け取る整数型タグ変数です。

```
Status = APUFindAlarmGroupInstance("$System", FoundInstance);
```

参照項目

APUFindFileInstance()、APUFindPrinterInstance()、APUGetInstanceCount()、APUIsInstanceUsed()

APUFindFileInstance() 関数

指定の .alc 設定ファイルを使用して、Alarm Printer の最初のインスタンスを見つけます。

カテゴリ

表示

構文

```
[Result=] APUFindFileInstance(sFilePath, iInstance);
```

引数

sFilePath
検出されるインスタンスを持つ .alc 設定ファイルのパス

iInstance
インスタンスの番号を受け取る整数型タグ変数

備考

この関数は、iInstance パラメータとして整数型タグ変数にインスタンス番号を返します。Alarm Printer の必要なインスタンスを初期に取得するには、この関数を使用します。ファイル パス文字列一致では、大文字と小文字は区別されません。

インスタンスが見つからない場合、関数はエラー コード（使用可能なインスタンスなし）として 1 を返し、整数型タグ変数パラメータに 0 を書き込みます。

整数型のエラー コードを返します。

例

InstFound は、設定ファイル c:\MyAlarmCfg\Area1Alarms.alc を使用する最初のインスタンスの番号を受け取る整数型タグ変数です。

```
Status = APUFindFileInstance("c:\MyAlarmCfg\Area1Alarms.alc", InstFound);
```

参照項目

APUFindAlarmGroupInstance()、APUFindPrinterInstance()、APUGetInstanceCount()、APUIsInstanceUsed()

APUFindPrinterInstance() 関数

指定のプリンタ名またはポートを使用して、Alarm Printer の最初のインスタンスを検出します。

カテゴリ

表示

構文

```
[Result=] APUFindPrinterInstance(sPrinter, iInstance);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

sPrinter
検出されるインスタンスを持つプリンタの名前です。

iInstance
インスタンスの番号を受け取る整数型タグ変数

備考

この関数は、iInstance パラメータとして整数型タグ変数にインスタンス番号を返します。Alarm Printer の必要なインスタンスを初期に取得するには、この関数を使用します。プリンタ名は保存され、.alc ファイルから読み取られます。プリンタ名文字列一致では、大文字と小文字は区別されません。インスタンスが見つからない場合、関数はエラー コード（使用可能なインスタンスなし）として 1 を返し、整数型タグ変数パラメータに 0 を書き込みます。

整数型のエラー コードを返します。

例

FoundInst は関連付けられた .alc ファイルでプリンタ名として LPT1 を使用する最初のインスタンスの番号を受け取る整数型タグ変数です。

```
Status = APUFindPrinterInstance("WINDOWST1", FoundInst);
```

参照項目

APUFindAlarmGroupInstance()、APUFindFileInstance()、APUGetInstanceCount()、APUIsInstanceUsed()

APUGetInstanceCount() 関数

Alarm Printer の実行中のインスタンスの数を返します。最大 16 個のインスタンスです。

カテゴリ

表示

構文

```
[Result=] APUGetInstanceCount(iCount);
```

引数

iCount
インスタンスの数を受け取る整数型タグ変数

備考

この関数は、パラメータとして整数型タグ変数にインスタンス数を返します。同時に実行中の最初の 16 個を超えるインスタンスはどれも動的に制御できません。また、ステータスを取得することもできません。

整数型のエラー コードを返します。

例

iCount は整数型のタグ変数です。ランタイム値が 7 は、実行している Alarm Printer インスタンスが現在 7 個あるという意味です。

```
Status = APUGetInstanceCount( iCount );
```

参照項目

APUFindAlarmGroupInstance()、APUFindFileInstance()、APUFindPrinterInstance()、APUIsInstanceUsed()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

APUIsInstanceUsed() 関数

インスタンスが現在使用中であることを示す論理値を返します。

カテゴリ

表示

構文

```
[Result=] APUIsInstanceUsed(iInstance);
```

引数

iInstance
Alarm Printer のインスタンスの数（0 ～ 15）

備考

結果は 0 または 1 です。

0 = インスタンスは使用されていません。

1 = インスタンスは使用されています。

例

InUse は論理型タグ変数です。インスタンス 5 が使用中の場合は true（1）に設定され、インスタンス 5 が使用中でない場合は false（0）に設定されます。

```
InUse = APUIsInstanceUsed(5);
```

参照項目

APUFindAlarmGroupInstance()、APUFindFileInstance()、APUFindPrinterInstance()、APUGetInstanceCount()

プリンタ情報のクエリー

Alarm Printer のステータスを返すには、以下の関数を使用します。

- APUGetPrinterName() 関数
- APUGetPrinterStatus() 関数

APUGetPrinterName() 関数

このインスタンスによって使用されるプリンタの Windows プリンタ名およびポート名を返します。

カテゴリ

表示

構文

```
[Result=] APUGetPrinterName(iInstance, sTagPrinter);
```

引数

iInstance
Alarm Printer のインスタンス番号（0 ～ 15）

sTagPrinter
指定インスタンスに関連付けられた設定のプリンタ名またはポート名を受け取るメッセージ型タグ変数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

この関数は、インスタンスに対してプリンタが設定されていない場合、値 NONE を返します。プリンタ名は保存され、.alc ファイルから読み取られます。この関数は、sTagPrinter パラメータとして、プリンタ名またはポート名をメッセージ型タグ変数に返します。

整数型のエラー コードを返します。

例

PrtName は、Alarm Printer のインスタンス 3 に関連付けられた設定のプリンタ名またはポート名を受け取るメッセージ型タグ変数です。

```
Status = APUGetPrinterName(3,PrtName);
```

参照項目

APUGetPrinterStatus()

APUGetPrinterStatus() 関数

このインスタンスによって使用される Windows プリンタの最新のステータスを返します。

カテゴリ

表示

構文

```
[Result=] APUGetPrinterStatus(iInstance, iSelector, iTagStatus);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iSelector
以下を指定する整数値:

0 = Alarm Printer エラーのステータスを取得します。

1 = Alarm Printer 用紙切れのステータスを取得します。

2 = Alarm Printer オフラインのステータスを取得します。

3 = Alarm Printer オーバーフローのステータスを取得します。

iTagStatus
指定のインスタンス番号および iSelector パラメータの選択タイプに関連付けられたプリンタのステータスを受け取る整数型または実数型のタグ変数

備考

この関数は、iTagStatus パラメータとして整数型または実数型のタグ変数にプリンタのステータスを返します。クエリーが実行していて、アラームが印刷されていない場合、結果は現在の内容ではありません。ステータスは通常、プリンタが最初に開くとき、およびアラーム行が印刷されるときに更新されます。

このステータス情報は、Microsoft または Windows のドライバ標準に基づいて、プリンタに照会されています。すべてのプリンタ製造業者が以下の標準に従っているわけではありません。したがって、すべてのプリンタでステータス情報が返されるわけではありません。

整数型のエラー コードを返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

PrtStat は、インスタンス 5 に関連付けられたプリンタから「プリンタ オフライン」ステータスを受け取る整数型タグ変数です。

```
Status = APUGetPrinterStatus(5, 2, PrtStat);
```

参照項目

APUGetPrinterName()

アラーム クエリー情報の設定

Alarm Printer クエリーで使用されるパラメータを設定するには、以下の関数を使用します。

- APUSetAlarmGroupText() 関数
- APUSetQueryAlarmState() 関数
- APUSetQueryFromPriority() 関数
- APUSetQueryToPriority() 関数
- APUSetTimeoutValues() 関数

APUSetAlarmGroupText() 関数

アラーム クエリー アラーム グループ テキストを設定します。

カテゴリ

View

構文

```
[Result=] APUSetAlarmGroupText(iInstance, sGroup);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

sGroup
アラーム グループ テキスト

備考

クエリーが実行中の場合、この関数は正常に実行されません。

整数型のエラー コードを返します。

例

この例では、Alarm Printer インスタンス 1 のクエリーを \InTouch!GroupA に設定します。

```
Status = APUSetAlarmGroupText(1, "\InTouch!GroupA");
```

参照項目

APUSetQueryAlarmState()、APUSetQueryFromPriority()、APUSetQueryToPriority()、APUSetTimeoutValues()

APUSetQueryAlarmState() 関数

クエリーのアラーム状況を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

表示

構文

```
[Result=] APUSetQueryAlarmState(iInstance, iState);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iState
以下の値を持つ整数：

0 = 全て

1 = 確認

2 = 未確認

備考

APUSetQueryAlarmState() 関数を使用してスクリプトを実行している間、同時にクエリーを実行することはできません。

整数型のエラー コードを返します。

例

この例では、Alarm Printer インスタンス 3 のクエリーを設定して、確認アラームのみをクエリーします。

```
Status = APUSetQueryAlarmState(3, 1);
```

参照項目

APUSetAlarmGroupText()、APUSetQueryFromPriority()、APUSetQueryToPriority()、APUSetTimeoutValues()

APUSetQueryFromPriority() 関数

アラーム クエリーの下限または優先度始点を設定します。

カテゴリ

表示

構文

```
[Result=] APUSetQueryFromPriority(iInstance, iPriority);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iPriority
優先度始点の整数値（1 ～ 999）

備考

クエリーが実行中の場合、この関数は正常に実行されません。

整数型のエラー コードを返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例では、Inst 整数型タグ変数のインスタンス値に関連付けられているクエリーの「優先度始点」の値を FromPri 整数型タグ変数の値に設定します。

```
Status = APUSetQueryFromPriority(Inst, FromPri);
```

参照項目

APUSetAlarmGroupText()、APUSetQueryAlarmState()、APUSetQueryToPriority()、APUSetTimeoutValues()

APUSetQueryToPriority() 関数

クエリーの「優先度終点」を設定します。

カテゴリ

表示

構文

```
[Result=] APUSetQueryToPriority(iInstance, iPriority);
```

引数

iInstance
Alarm Printer のインスタンス（0 ～ 15）

iPriority
「優先度終点」の整数値は 1 ～ 999 の範囲です。指定インスタンスの同じクエリーの「優先度始点」以上に設定する必要があります。

備考

有効なアラーム優先度範囲を設定するには、「優先度終点」は「優先度始点」以上である必要があります。**APUSetQueryToPriority()** 関数を含むスクリプトは、クエリーと同時に実行できません。

整数型のエラー コードを返します。

例

この例では、インスタンス 0 から 240 に関連付けられたクエリーの「優先度終点」の値を設定します。

```
Status = APUSetQueryToPriority(0,240);
```

参照項目

APUSetAlarmGroupText()、APUSetQueryAlarmState()、APUSetQueryFromPriority()、APUSetTimeoutValues()

APUSetTimeoutValues() 関数

APUSetTimeoutValues() 関数は、タイムアウト間隔（秒単位）を設定します。タイムアウト間隔は、プログラム実行中に、メモリ アクセスによって発生するエラー数、または有効応答の取得失敗が確認される回数を制御します。

デフォルトのメモリ アクセス タイムアウトは 2 秒です。デフォルトの短い応答待機時間は 10 秒です。デフォルトの長い応答待機時間は 20 秒です。

カテゴリ

表示

構文

```
[Result=] APUSetTimeoutValues(iMemory, iShort, iLong);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

iMemory
整数 - アクセス タイムアウト

iShort
短い応答待機時間（整数）

iLong
長い応答待機時間（整数）

例

```
Status = APUSetTimeoutValues(iMemory,iShort,iLong);
```

参照項目

APUSetAlarmGroupText()、APUSetQueryAlarmState()、APUSetQueryFromPriority()、APUSetQueryToPriority()

Alarm Printer エラーの処理

Alarm Printer エラー コードをエラー メッセージに変換するには、APUTranslateErrorCode() 関数を使用します。

APUTranslateErrorCode() 関数

APU 関数の 1 つによって返されるエラー コードを、エラー コードを簡単に説明する英語の文字列に変換します。

カテゴリ

表示

構文

```
[Result=] APUTranslateErrorCode(iErrorCode, sTagMessage);
```

引数

iErrorCode
通常は、他のほとんどの APU 関数によって返される整数型のエラー コードです。

sTagMessage
エラー メッセージを受け取るメッセージ型タグ

備考

この関数は、sTagMessage パラメータとして、エラー メッセージをメッセージ型タグに返します。不明なエラー コードが渡された場合、この関数は失敗する可能性があります。

整数型のエラー コードを返します。

例

この例では、現在実行している Alarm Printer のインスタンス 15 が存在しない場合、メッセージ型タグ errmsg を「インスタンスがありません。」に設定します。

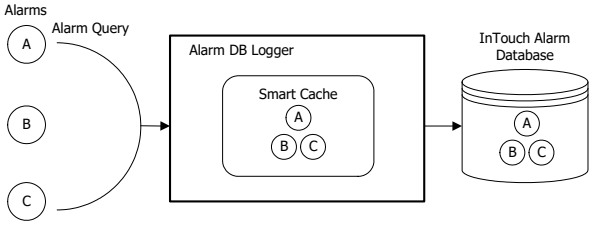
```
Status = APUTranslateErrorCode(APUSetAlarmGroupText(15,"$system"), ErrMsg);
```

アラーム データベースへのアラームの記録

InTouch 分散アラーム システムには、アラームやイベントをアラーム データベースにログする Alarm DB Logger ユーティリティが含まれています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Alarm DB Logger はアラーム コンシューマです。1 つまたは複数のクエリーを使用して Alarm DB Logger を設定し、InTouch アラーム プロバイダからのアラームを選択します。クエリーで選択したアラームは、スマート キャッシュと呼ばれるメモリ キャッシュに一時的に格納されます。Alarm DB Logger は、一定の周期でスマート キャッシュの内容をアラームおよびイベント レコードとしてアラーム データベースに書き込みます。



Alarm DB Logger は自動再接続が可能です。データベースへの接続が失われた場合は、Logger は一定の周期でデータベースへの接続をチェックし、アラーム データベースとの接続が再確立されると、ログへの記録が再開されます。

サービスとして実行されているか標準のアプリケーションとして実行されているかに関係なく、Alarm DB Logger はすべてのエラーを Logger に報告します。

Alarm DB Logger Manager 用の SQL Server アカウント

Alarm DB Logger Manager は SQL Server データベースに以下のアカウントを作成して使用します。

アカウント	パスワード
wwAdmin	wwAdmin
wwPower	wwpower
wwUser	wwUser

Alarm DB Logger Manager の使用

Alarm DB Logger Manager は、ログ処理を開始または停止する場合に使用します。Alarm DB Logger Manager は、サービスまたは標準のアプリケーションとして起動できます。

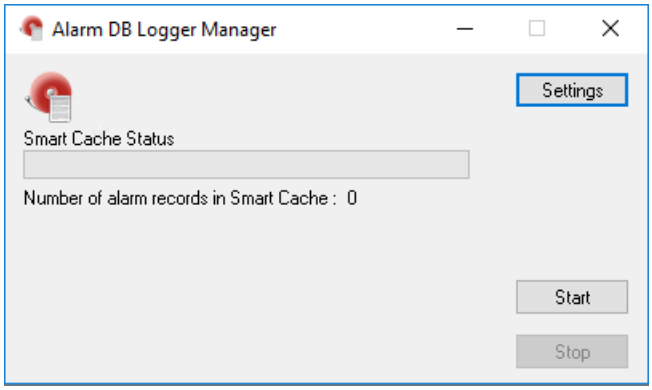
アラームのログは Alarm DB Logger 設定ウィザードを使用して設定します。このウィザードは Alarm DB Logger Manager から起動します。

Alarm DB Logger Manager には、スマート キャッシュに格納されたアラーム レコードの割合が表示されます。アラームがキャッシュされるのは、SQL Server との接続がダウンした場合や、Alarm DB Logger がアラーム データベースにログできる速度よりも速くアラームが発生した場合です。

Alarm DB Logger Manager を開く

1. ［ツール］ビューで［アプリケーション］を展開します。
2. ［**Alarm DB Logger Manager**］をダブルクリックします。Alarm DB Logger Manager が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

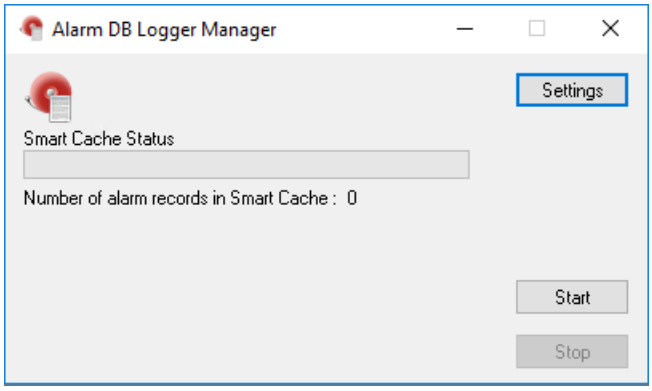


アラーム データベース ログの開始と停止

Alarm DB Logger はレコードをアラーム データベースに書き込みます。Alarm DB Logger Manager の設定ウィザードで設定した実行モードに応じて、サービスまたは標準のアプリケーションとして起動および動作します。

アラームのログを開始または停止する

1. ［ツール］ビューで［アプリケーション］を展開します。
2. ［**Alarm DB Logger Manager**］をダブルクリックします。Alarm DB Logger Manager が表示されます。



［スマート キャッシュ ステータス］に、メモリ内キャッシュに格納されているアラーム レコードの割合が表示されます。

3. ［開始］を選択して、アラームのロギングを開始します。
4. ［中止］を選択して、アラームのロギングを終了します。

アラーム データベース ロギングの設定

InTouch のアラームやイベントのデータをアラーム データベースにログするには、Alarm DB Logger Manager から以下の操作を実行します。

- アラーム データベースとの接続の設定
- アラーム データベースにログするアラームの選択
- レコードをアラーム データベースにログする周期の設定
- Alarm DB Logger の実行方法の選択

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

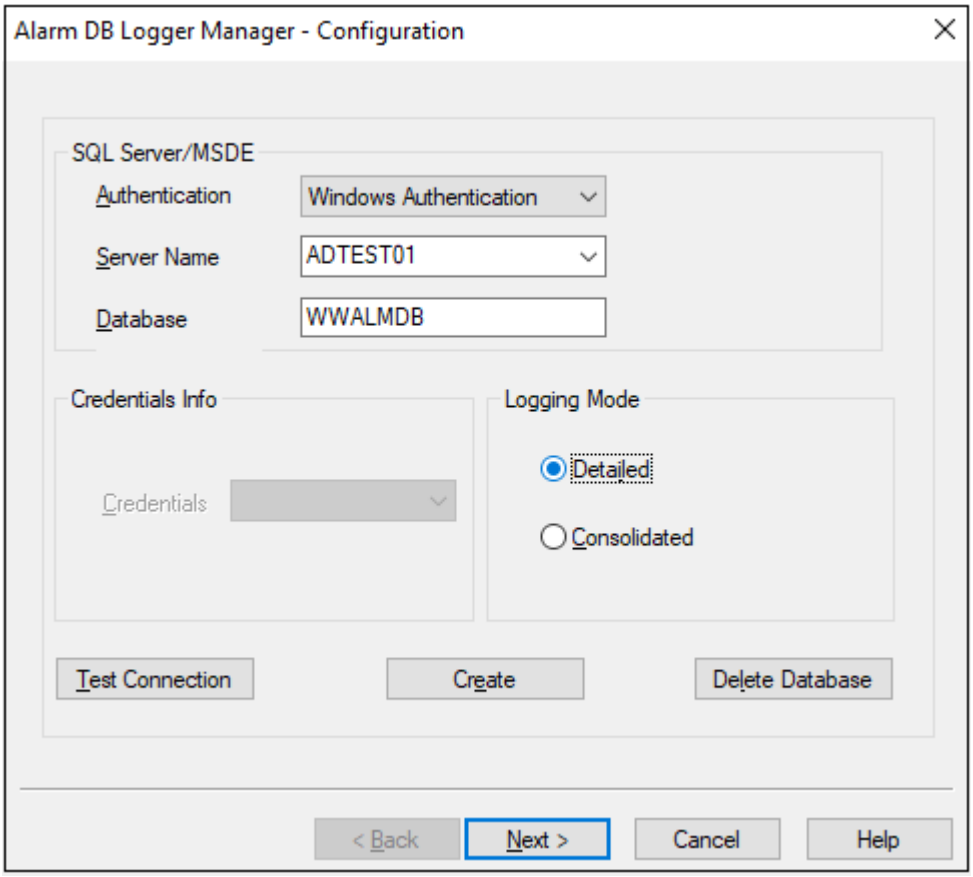
データベース接続の確立

Alarm DB Logger とアラーム データベース間の接続を確立する必要があります。

Alarm DB Logger は、SQL Server および Windows 認証と連動します。SQL Server は混在モードに設定する必要があります。

データベース接続を設定する

1. Alarm DB Logger Manager を開きます。以下の手順を実行します。
   a. ［ツール］ビューで［**アプリケーション**］を展開します。
   b. ［**Alarm DB Logger Manager**］をダブルクリックします。
2. ［**設定**］を選択します。［**Alarm DB Logger Manager - 設定**］ウィザードが表示されます。



3. ［**SQL Server/MSDE]**セクションで以下の手順を実行します。
   a. [**認証**] リストで、認証方法(SQL Server 認証または Windows 認証 (既定)) を選択します。

   注 : Windows 認証は、SQL 認証よりも優れたアプリケーション セキュリティを提供できます。この理由から、Windows 認証から SQL 認証に切り替えると、Windows 認証を使用することを推奨するポップアップ ダイアログが表示されます。この警告を無視して SQL 認証を使用する場合は、［**OK**］を選択します。同様のメッセージが OCMC Log Viewer に記録されます。

   b. ［**サーバー名**］ボックスに、アラーム データベースがインストールされているコンピュータのノード名を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

c. ［データベース］ボックスに、InTouch アラーム データベースの名前を入力します。

4. ［資格情報］セクションの［資格情報］ドロップダウンから認証に使用する資格情報を選択します。

注記:
- ［資格情報］ドロップダウンは、SQL Server 認証タイプが選択されている場合にのみ有効になります。Windows 認証では、現在ログインしているユーザーの資格情報が使用され、ユーザー名とパスワード フィールドが無効になります。
スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「資格情報マネージャの操作」を参照してください。

5. ［ロギング モード］領域で、レコードの保存方法を設定します。以下のいずれかを実行します。
   a. 各アラーム条件（アラーム、確認、および平常状態に戻った）に対して個別のレコードを保存する場合は、［詳細］を選択します。
   b. 各アラーム（アラーム、確認、および平常状態に戻った）のすべての状態を、各変化にタイムスタンプを付けて単一のレコードに保存する場合は、［統合］を選択します。
6. 必要に応じて、［作成］を選択して、データベースを作成します。
7. ［接続テスト］を選択して、アラーム データベースへの接続性を確認します。データベースに正常に接続できたかどうかを示すメッセージが表示されます。
8. ［次へ］を選択して、ログに記録するアラームを設定します。詳細な手順については、「ログに記録するアラームを設定する」を参照してください。

ログに記録するアラームを設定する

Alarm DB Logger Manager 設定ウィザードの 2 ページ目では、InTouch または Galaxy アラーム プロバイダからのアラームを選択するための 1 つまたは複数のクエリーを定義します。また、データベース クエリーの一部であるアラーム優先度の値の範囲も選択します。

リモート ノードには次のクエリー構文を使用します。

\\ノード名\Provider!AlarmGroup

ローカル ノードには次のクエリー構文を使用します。

\Provider!AlarmGroup

例:

\\ProdSvr\InTouch!$System

次に、Galaxy アラーム プロバイダを使用した例を示します。このクエリーは、特定の領域からのすべてのアラームを取得します。

\Galaxy!Area

注：Alarm DB Logger Manager では、最大 50 のクエリーおよび 3072 文字がサポートされます。

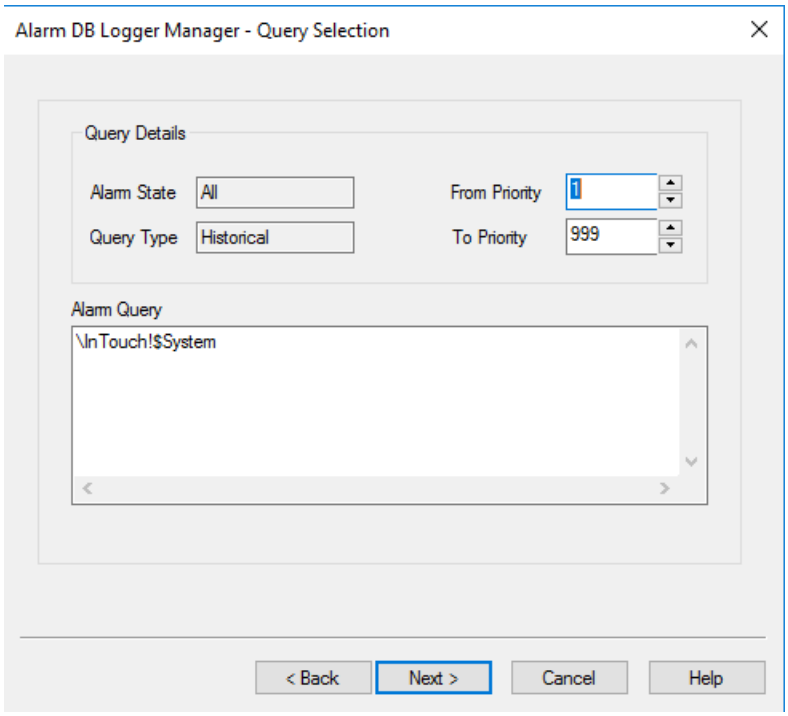
ログに記録するアラームを設定する

1. Alarm DB Logger Manager を開いて、設定ウィザードを起動します。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm DB Logger Manager**］をダブルクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

c. ［設定］を選択します。［**Alarm DB Logger Manager - 設定**］ウィザードが表示されます。

2. **[次へ]** を選択します。［**Alarm DB Logger Manager - クエリーの選択**］ページが表示されます。



読み取り専用の［**アラーム状況**］ボックスに、ログ対象のアラーム状況が表示されます。読み取り専用の［**クエリー タイプ**］ボックスには、クエリーのタイプが表示されます。

3. ［優先度始点］ボックスに、アラームの優先度範囲の開始値を入力します。
4. ［優先度終点］ボックスに、アラームの優先度範囲の終了値を入力します。
5. ［アラーム クエリー］ボックスに、アラーム データベースに対してデータを保存または取得するために使用するアラーム クエリーを入力します。

注： 128 文字のエリア名は、ヒストリアンではサポートされていません。

6. ［次へ］を選択して、ロギング周期を設定します。「ログの間隔を設定する」を参照してください。

ログの間隔を設定する

Alarm DB Logger Manager 設定ウィザードの 3 ページ目では、クエリーの詳細設定を行います。イベントをアラーム データベースにログすることができます。また、アラーム レコードを内部のアラーム メモリからデータベースに書き込む頻度を設定をすることで、アラーム ログのパフォーマンスを調整することもできます。

ログの間隔は、SQL Server の再接続速度と同じではありません。ログの間隔はアラームを読み取る間隔です。一方、再接続速度は、SQL Server に関連付けられている接続試行のタイムアウトによって異なります。

ログの間隔の設定が低過ぎると、システム パフォーマンスに悪影響が生じます。

ログの間隔を設定する

1. Alarm DB Logger Manager を開いて、設定ウィザードを起動します。以下の手順を実行します。

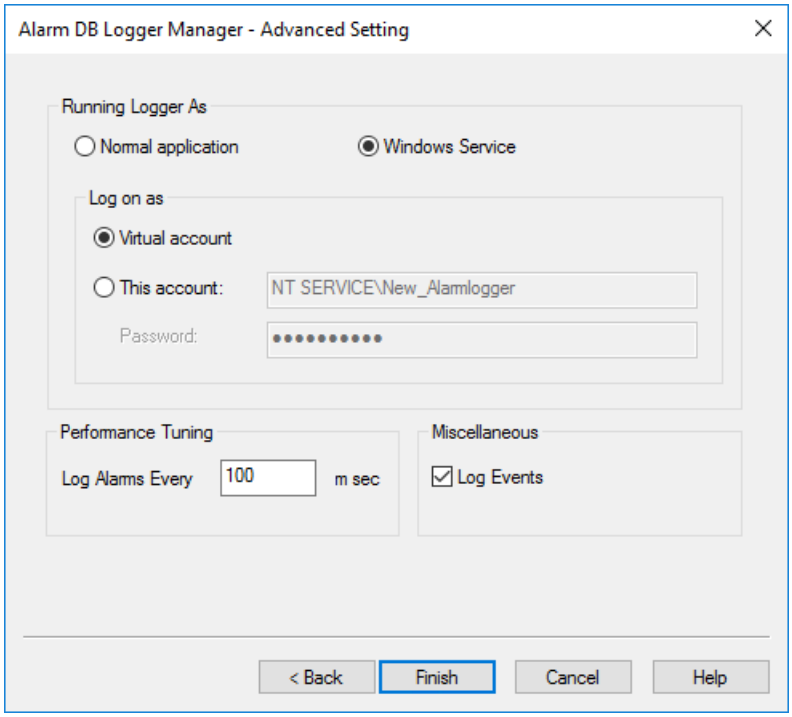
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. ［ツール］ビューで［アプリケーション］を展開します。

b. ［**Alarm DB Logger Manager**］をダブルクリックします。

c. ［設定］を選択します。［**Alarm DB Logger Manager - 設定**］ウィザードが表示されます。

2. **[次へ]** を選択します。［**Alarm DB Logger Manager - クエリーの選択**］ページが表示されます。
3. **[次へ]** を選択します。［**Alarm DB Logger Manager - 詳細設定**］ページが表示されます。



4. InTouch イベント レコードをアラーム データベースに保存する場合は、［イベント ログ］チェックボックスをオンにします。Application Server Galaxy から取得されたイベントを保存することもできます。
5. ［パフォーマンスの調整］領域に、アラーム レコードをアラーム データベースに書き込む間隔をミリ秒単位で入力します。
6. **[完了]** を選択します。

サービスとしての Alarm DB Logger の設定

Windows サービスとして実行するように Alarm DB Logger を設定できます。管理者権限で Alarm DB Logger を実行する場合にセキュリティの脅威にさらされる可能性を削減するために、Alarm DB Logger をサービスとして実行するユーザー アカウントの許可は非対話型に設定されています。

1. コンピュータに管理者としてログインします。
2. Alarm DB Logger Manager を開いて、設定ウィザードを起動します。以下の手順を実行します。

a. ［ツール］ビューで［アプリケーション］を展開します。

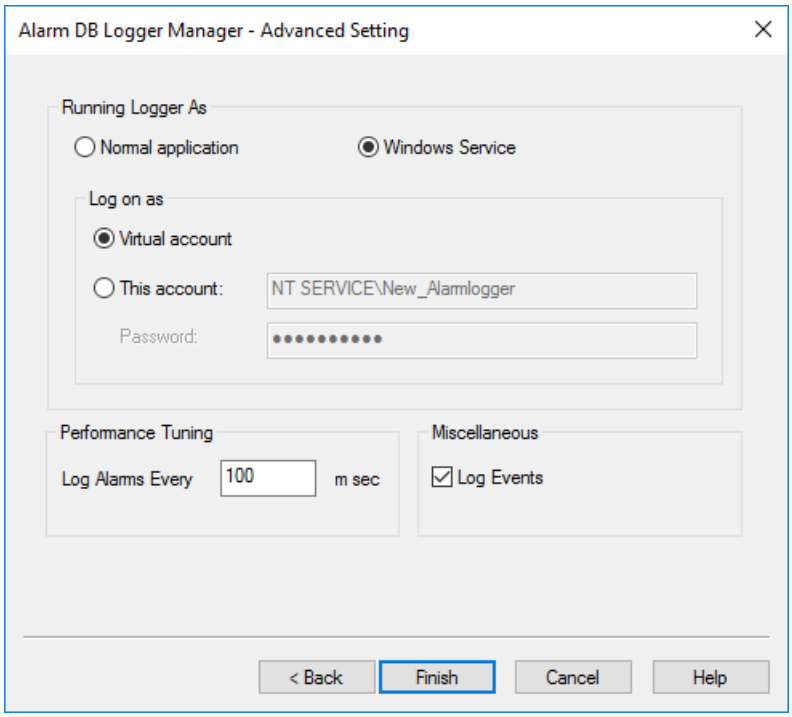
b. ［**Alarm DB Logger Manager**］をダブルクリックします。

c. ［設定］を選択します。［**Alarm DB Logger Manager - 設定**］ウィザードが表示されます。

3. **[次へ]** を選択します。［**Alarm DB Logger Manager - クエリーの選択**］ページが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. **[次へ]** を選択します。［**Alarm DB Logger Manager - 詳細設定**］ページが表示されます。



5. ［**ロガーの実行**］領域で、［**標準アプリケーション**］または［**Windows サービス**］のいずれかを選択します。

   資格情報は［**Alarm DB Logger Manager - 設定**］ウィザード ページで選択した認証方法に応じて異なります。

6. ［**ログオンに使用するアカウント**］領域で以下の条件に基づいてログイン アカウントを選択し、［**完了**］を選択します。

認証方法	ロガーの実行	ログオンに使用するアカウント	ユーザー名	パスワード
SQL Server 認証	Windows サービス	仮想アカウント	--	--
Windows 認証	Windows サービス	仮想アカウント	--	--
Windows 認証	Windows サービス	指定アカウント	<ユーザー名>	<パスワード>

注：仮想アカウントの詳細については、仮想アカウントの使用を参照してください。

デフォルトでは、［**指定アカウント**］を選択すると、現在ログインしているユーザーのユーザー名が表示されます。パスワードを入力する必要はありません。ユーザー名を変更した場合、パスワードを入力する必要があります。

設定を完了するには、適切な権限のあるユーザー アカウントを指定する必要があります。

次回 Alarm DB Logger を起動すると、設定の詳細が自動的に入力されます。

仮想アカウントの使用

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

仮想アカウントを選択するとセキュリティを強化できます。このオプションを選択すると、Alarm DB Logger が New_AlarmLogger という NT Service アカウントで起動します。

仮想アカウントは SQL 認証と Windows 認証の両方で有効になります。［LocalSystem］で設定された以前のシステムの場合、アカウントは次のように移行されます。

- 以前のシステムで［指定アカウント］または［ローカル システム アカウント］オプションが［LocalSystem］に設定されている場合、［指定アカウント］にはユーザーとして［LocalSystem］が設定されます。
- 以前のシステムで［指定アカウント］が［NT Service\New_AlarmLogger］として設定されている場合、［仮想アカウント］に移行されます。

アラーム データベースのビュー

SQL Server データベース ビューのセットを使用すると、過去および現在のアラームやイベントの発生状況について簡単にクエリーを実行できます。

データベース ビューとは、基礎となる複数のデータベース テーブルの情報を組み合わせた 1 つの論理的なテーブルのことです。データベース ビューは仮想データベース テーブルと呼ばれる場合もあります。これは、実際のテーブルと同じように標準の SQL ステートメントを使ってクエリーを実行できるからです。ビューがクエリーされると、レコード（または行）のセットが返されます。各行には、レコードのデータを含む情報のカラムがいくつかあります。

アラーム データベースのビューを使用すると、複雑なクエリーを実行できます。たとえば、工場の特定領域である一定の時間帯に起こった HiHi アラームすべてのアラーム レコードを取り出すことができます。または、あるアラーム プロバイダ ノードでログしたデータ変化イベントをすべて取り出すこともできます。

ビューの文字列はすべて Unicode です。

アラーム履歴ビュー

v_AlarmHistory ビューには、選択した時間範囲内に発生したすべての履歴アラームとアラーム変化のイベントが表示されます。クエリーで、（カラム EventStamp または EventStampUTC で）開始と終了の日付と時刻を指定します。返されるレコードには、アラームの発生、アラームの確認、アラームの有効化、アラームの無効化、アラームから通常状態に戻ったイベントが含まれます。

以下の表は、ビューの列について説明しています。

列名	データ型	説明
イベントスタンプ	日付時刻	（データベースのローカル時間における）アラーム イベントの日付と時刻
AlarmState	nChar	アラーム状態（UNACK、UNACK_RTN、ACK、ACK_RTN、LATCHED のいずれか）
TagName	nChar	TIC101 など、アラームを生成したオブジェクト名
説明	nVarchar	アラームの説明文字列。デフォルトはオブジェクトの説明（または HMI のコメント）。 または、確認レコードの確認コメント。
領域	nChar	アラームの領域、またはグループ名

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

列名	データ型	説明
タイプ	nChar	Hi、HiHi、ROC、PV.HiAlarm など、アラーム タイプ
値	nChar	アラーム発生時に有効だったアラーム値
値の確認	nChar	アラーム発生時のアラームしきい値
優先度	整数型	アラーム優先度
カテゴリ	nChar	アラーム クラス、またはアラーム カテゴリ。Value、Dev、ROC、Process、Batch、System など。
プロバイダ	nChar	アラーム プロバイダ。node/InTouch や GalaxyName など。
オペレータ	nChar	ユーザーの名前
ドメイン名	nChar	ドメインの名前
ユーザーフルネーム	nChar	オペレータであるユーザーのフル ネーム。（Kenichi Suzuki など）
UNACKDuration	浮動小数点	アラームが最後に変化（アラームまたはサブステート）してから確認されるまでの時間（該当する場合）
ユーザー 1	浮動小数点	ユーザー定義のフィールド数 1
ユーザー 2	浮動小数点	ユーザー定義のフィールド数 2
User3	nChar	ユーザー定義のフィールド、文字列
EventStampUTC	日付時刻	アラーム イベントの UTC 日付/時刻
ミリ秒	Small Int	0.1 ミリ秒単位のイベント スタンプ
オペレータ ノード	nvarchar(32)	ユーザーがアラームを確認したノードの名前

v_AlarmHistory2 ビューは v_AlarmHistory ビューの一種です。

列名	データ型	説明
イベントスタンプ	日付時刻	（データベースのローカル時間における）アラーム イベントの日付と時刻
AlarmState	nChar	アラーム状態（UNACK、UNACK_RTN、ACK、ACK_RTN、ACK_ALM、UNACK_ALM、LATCHED のいずれか）
TagName	nChar	TIC101 など、アラームを生成したオブジェクト名
説明	nVarchar	アラームの説明文字列。デフォルトで、オブジェクトの説明か、InTouch のコメントに設定できます。または、確認レコードの確認コメント。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

列名	データ型	説明
領域	nChar	アラームの領域、またはグループ名
タイプ	nChar	Hi、HiHi、ROC、PV.HiAlarm など、アラーム タイプ
値	nChar	アラーム発生時に有効だったアラーム値
値の確認	nChar	アラーム発生時のアラームしきい値
優先度	整数型	アラーム優先度
カテゴリ	nChar	アラーム クラス、またはアラーム カテゴリ。Value、Dev、ROC、Process、Batch、System など。
プロバイダ	nChar	アラーム プロバイダ。node/InTouch や GalaxyName など。
オペレータ	nChar	演算子の名前: JoeR (ある場合)。
ドメイン名	nChar	ドメインの名前
ユーザーフルネーム	nChar	オペレータであるユーザーのフル ネーム。（Kenichi Suzuki など）
AlarmDuration	浮動小数点	アラームが発生してから通常状態に戻るまでの時間
ユーザー 1	浮動小数点	ユーザー定義のフィールド数 1
ユーザー 2	浮動小数点	ユーザー定義のフィールド数 2
User3	nChar	ユーザー定義のフィールド、文字列
EventStampUTC	日付時刻	アラーム イベントの UTC 日付/時刻
ミリ秒	Small Int	0.1 ミリ秒単位のイベント スタンプ

クエリーの例 - アラーム履歴ビュー

この例では、アラーム履歴ビューからすべてのレコードを選択します。

`* FROM v_AlarmHistory` を選択します。

この例では、アラーム履歴ビューから、優先度が 100 より大きいすべてのレコードを選択します。

`* FROM v_AlarmHistory WHERE Priority >100` を選択します。

イベント履歴ビュー

v_EventHistory データベース ビューには、選択した時間範囲内に発生したすべての履歴イベントのリストが表示されます。クエリー クライアントは開始と終了の日付と時刻の範囲を指定します。返されるレコードには、アラーム以外のイベントもすべて含まれます。

列名	データ型	説明
イベントスタンプ	日付時刻	イベントの日付と時刻
TagName	nChar	Pump1 など、イベントを生成したオブジェクト名

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

列名	データ型	説明
説明	nVarChar	イベントを説明する文字列。デフォルトで、オブジェクトの説明か、InTouch のコメントに設定できます。
領域	nChar	イベントの領域、またはグループ名
タイプ	nChar	「ユーザー データの変更」や「開始」など、イベントのタイプ
値	nChar	新規値（該当する場合）
CheckValue	nChar	古い値（該当する場合）
カテゴリ	nChar	Value、Process、Batch、System など、イベント カテゴリまたはクラス
プロバイダ	nChar	イベントを生成したオブジェクト（node/InTouch など）、またはユーザー変更のビュー エンジン名
オペレータ	nChar	演算子 1 の名前:カエルの数(ある場合)。
ドメイン名	nChar	ドメインの名前
ユーザーフルネーム	nChar	オペレータであるユーザーのフル ネーム。（Kenichi Suzuki など）
ユーザー 1	浮動小数点	ユーザー定義のフィールド数 1
ユーザー 2	浮動小数点	ユーザー定義のフィールド数 2
User3	nChar	ユーザー定義のフィールド、文字列
EventStampUTC	日付時刻	イベントの UTC 日付/時間
ミリ秒	Small Int	0.1 ミリ秒単位のイベント スタンプ

アラーム イベント履歴ビュー

v_AlarmEventHistory データベース ビューには、選択時間内に発生したすべてのイベントとアラームの履歴リストが表示されます。クエリー クライアントは開始と終了の日付と時刻を指定します。返される値には、すべてのアラームとイベントが含まれます。このビューでは、アラーム ビューとイベント ビューを 1 つにまとめて、両方のビューのレコードが合わせて表示されます。

列名	データ型	説明
イベントスタンプ	日付時刻	イベントの日付と時刻
AlarmState	nChar	アラーム状態（UNACK、UNACK_RTN、ACK、ACK_RTN、LATCHED のいずれか）イベントには適用されません。
TagName	nChar	TIC101 など、アラームを生成したオブジェクト名

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

列名	データ型	説明
説明	nVarchar	アラーム/イベントの説明文字列。デフォルトで、オブジェクトの説明か、InTouch のコメントに設定できます。 または、確認レコードの確認コメント。
領域	nChar	アラームの領域、またはグループ名
タイプ	nChar	Hi、HiHi、ROC、PV.HiAlarm、ユーザー データの変更など、アラームやイベントのタイプ
値	nChar	アラーム発生時に有効だったアラーム値
値の確認	nChar	アラーム発生時のアラームのしきい値、またはイベントの古い値
優先度	整数型	アラーム優先度
カテゴリ	nChar	Value、Process、Batch、System など、アラームやイベントのクラス、またはアラーム カテゴリ
プロバイダ	nChar	node/InTouch や GalaxyName など、アラーム プロバイダ
オペレータ	nChar	確認したユーザー、またはデータを変更したユーザー名
ドメイン名	nChar	ドメインの名前
ユーザーフルネーム	nChar	オペレータであるユーザーのフル ネーム。 （Kenichi Suzuki など）
UNACKDuration	浮動小数点	アラーム変化が最後に発生してから確認までのミリ秒数
ユーザー 1	浮動小数点	ユーザー定義のフィールド数 1
ユーザー 2	浮動小数点	ユーザー定義のフィールド数 2
User3	nChar	ユーザー定義のフィールド、文字列
EventStampUTC	日付時刻	イベントの UTC 日付/時間
ミリ秒	Small Int	0.1 ミリ秒単位のイベント スタンプ

v_AlarmEventHistory2 ビューは v_AlarmEventHistory ビューの一種です。

列名	データ型	説明
イベントスタンプ	日付時刻	イベントの日付と時刻
AlarmState	nChar	アラーム状態（UNACK、UNACK_RTN、ACK、ACK_RTN、LATCHED のいずれか）イベントには適用されません。
TagName	nChar	TIC101 など、アラームを生成したオブジェクト名

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

列名	データ型	説明
説明	nVarchar	アラーム/イベントの説明文字列。デフォルトで、オブジェクトの説明か、InTouch のコメントに設定できます。 または、確認レコードの確認コメント。
領域	nChar	アラームの領域、またはグループ名
タイプ	nChar	Hi、HiHi、ROC、PV.HiAlarm、ユーザー データの変更など、アラームやイベントのタイプ
値	nChar	アラーム発生時に有効だったアラーム値
値の確認	nChar	アラーム発生時のアラームのしきい値、またはイベントの古い値
優先度	整数型	アラーム優先度
カテゴリ	nChar	Value、Process、Batch、System など、アラームやイベントのクラス、またはアラーム カテゴリ
プロバイダ	nChar	node/InTouch や GalaxyName など、アラーム プロバイダ
オペレータ	nChar	確認したユーザー、またはデータを変更したユーザー名
ドメイン名	nChar	ドメインの名前
ユーザーフルネーム	nChar	オペレータであるユーザーのフル ネーム。 （Kenichi Suzuki など）
AlarmDuration	浮動小数点	アラームが発生してから通常状態に戻る（RTN）までのミリ秒数
ユーザー 1	浮動小数点	ユーザー定義のフィールド数 1
ユーザー 2	浮動小数点	ユーザー定義のフィールド数 2
User3	nChar	ユーザー定義のフィールド、文字列
EventStampUTC	日付時刻	イベントの UTC 日付/時間
ミリ秒	Small Int	0.1 ミリ秒単位のイベント スタンプ

クエリーの例 - アラーム イベント履歴ビュー

次の例では、タグ変数 MyTag1 を持ち、アラーム状況が ACK_RTN または ACK のレコードすべてについて、TagName、Area、および Type の各カラムのデータを選択します。結果はアラーム プロバイダ順に並べます。

```
[TagName]、[Area]、[TYPE FROM v_AlarmEventHistory を選択します。
   WHERE TagName='MyTag1'
      AND (AlarmState='ACK_RTN' OR AlarmState='ACK')
      ORDER BY Provider(プロバイダー別の順序)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム データベースのストアド プロシージャ

SQL Server のストアド プロシージャのセットを使用すると、情報を簡単に検索してアラームとイベントの分析やレポートを実行できます。

ストアド プロシージャとは、SQL ステートメントの集まりで、特定の関数を実行し、データベースからデータを返すものです。ストアド プロシージャでは、入力パラメータを使用して、実行方法や、結果セットの形式でデータを返す方法を指定できます。

返される結果はレコードのセットで、SQL Server のデータベース ビューと類似した SQL レコード セットのように見えます。ストアド プロシージャを使用するとパフォーマンスを向上できます。これはストアド プロシージャが、組み込み SQL ステートメントと共にデータベース内に存在するため、クライアントへ戻す処理が必要なくなるからです。

ストアド プロシージャの詳細（ストアド プロシージャの定義の表示方法を含む）については、Microsoft SQL Server のマニュアルを参照してください。

ストアド プロシージャの呼び出し

ストアド プロシージャは、次のように EXECUTE ステートメントを使用して呼び出します。

```
sp_AlarmCounter @StartDate='2007-01-01', @EndDate='2007-03-31', @Tagname = 'tag1', @Type
= 'LO', @Provider = 'WW21353\InTouch', @Comment = 'SSAADD'を実行する
```

この例では、StartDate および EndDate パラメータが必須で、残りのパラメータはオプションです。特定のパラメータの値を指定しなかった場合は、結果セットをフィルタする際にそのパラメータは使用されません。

ストアド プロシージャに日付／時刻変数が含まれる場合は、SQL Server のマニュアルで指定されている有効な形式であればどれでも使用できます。

AlarmCounter データベース ストアド プロシージャ

このストアド プロシージャは、一定の周期内に固有のアラームが発生した回数を返します。固有のアラームを識別するには、Tagname、Provider、Alarm Type、Category という 4 つのパラメータを使用します。

発生数は、アラームの発生のみを計算するもので、確認や平常状態に戻った場合は含まれません。このため、アラームが発生して確認され、平常状態に戻った場合は、「3」ではなく「1」と認識されます。

クエリーでは、開始と終了の日付と時刻を指定します。このパラメーターには、TagName、Class、Type、Provider、Comment の 5 つのオプションパラメーターがあります。

このビューによって回答される質問の例: ある期間中に、プロバイダ Node1|InTouch（Provider）にあるオブジェクト TIC101（TagName）の値がアラーム（Category）になり、そのアラーム タイプが HiHi（Type）であった回数を特定する場合。

注：AlarmCounter ストアド プロシージャは詳細ロギング モードにのみ適用され、統合ロギングでは使用できません。

sp_AlarmCounter

列名	データ型	説明
TagName	Nchar	TIC101 など、アラームを生成したオブジェクト名
グループ名	Nchar	アラームの領域、またはグループ名
AlarmType	Nchar	Hi、HiHi、ROC、PV.HiAlarm など、アラーム タイプ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

列名	データ型	説明
AlarmClass	Nchar	Value、Process、Batch などのアラーム クラス、またはアラーム カテゴリ
AlarmCount	整数型	特定期間中のアラーム発生数。 開始日付および時刻より前にアラームが発生した場合、そのアラームは発生数に含まれません。
優先度	整数型	アラーム優先度
プロバイダ	Nchar	node/InTouch や GalaxyName など、アラーム プロバイダ
コメント	Nchar	アラーム コメント

次にクエリーの例を示します。

```
EXECUTE sp_AlarmCounter @StartDate='2007/01/01 23:23:23', @EndDate='2007/03/31 23:23:23',
@Tagname = '$NewAlarm'
```

EventCounter データベース ストアド プロシージャ

このデータベース ストアド プロシージャは、ある期間内で、特定タグ変数での特定タイプのイベント数を表示します。クエリー クライアントは、その期間の最初と最後を指定する必要があります。このパラメーターには、TagName、Provider、Comment の 3 つの任意のパラメーターがあります。このカウンタは、アラーム以外のイベントのみに適用されます。このビューは、各イベントの頻度を示します。たとえば、ポンプが特定期間内に何回オンになったかを調べます。TagName、Provider、Category、および Type を使ってイベントを認識し、発生数を計算します。

```
sp_EventCounter
```

列名	データ型	説明
TagName	Nchar	TIC101 など、アラームを生成したオブジェクト名
領域	Nchar	アラームの領域、またはグループ名
タイプ	Nchar	イベントのタイプ
カテゴリ	Nchar	Value、Process、Batch などのアラーム クラス、またはアラーム カテゴリ
EventCount	整数型	ある期間中に発生した特定タグ変数に対するこのタイプのイベント数
プロバイダ	Nchar	アラーム プロバイダ。node/InTouch や GalaxyName など。
コメント	Nchar	イベントのコメント

次にクエリーの例を示します。

EXECUTE sp_EventCounter @StartDate='2007/01/01 23:23:23', @EndDate='31.03.07 23:23:23', @Tagname = '$NewAlarm'

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

レコードされたアラームの表示

Alarm DB View ActiveX コントロールを使用すると、アラーム データベースからのデータを視覚化できます。このコントロールを使用して、ランタイムに InTouch アプリケーションから生成されたすべてのアラームおよびイベント情報を表示できます。

このコントロールでは、以下の内容を設定できます。

- コンテキスト メニュー機能
- 表示モード
- リスト制御オプション
- 異なるプロパティの色
- フォント タイプ、スタイル、サイズ
- データベースの指定（サーバー名、ユーザー ID、パスワード）
- クエリー フィルタ
- カラムの管理
- ソート

ランタイム ユーザーは、アプリケーション実行中にオプションを変更し、表示中のデータを選択できます。以下の操作を行えます。

- カラム内の情報のソート
- 表示の更新
- クエリーの実行
- カラム幅のサイズ変更

ActiveX コントロールの詳細については、「ActiveX コントロール」を参照してください。

Alarm DB View コントロールの設定

Alarm DB View コントロールを設定する場合、以下の操作を行えます。

- アラーム データベースへの接続を設定します。
- グリッドの外観を設定する
- 表示フォントを選択する
- 表示外観を設定する
- アプリケーションが実行している間、ユーザーがアクセスできる機能を設定する
- 表示するアラームを設定する
- カスタム フィルタを作成する
- アラーム レコードのタイプの色を設定する
- 表示されるように時間形式を設定する
- 表示されるアラーム レコードのソート順を設定する

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

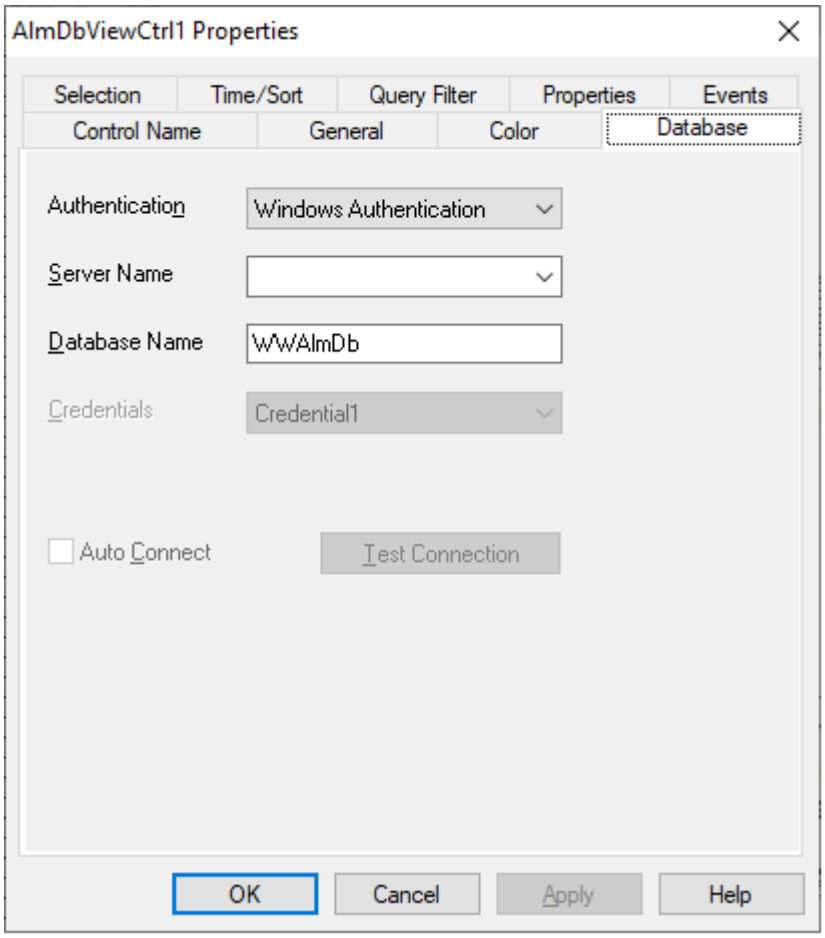
Alarm DB View と Alarm Database への接続

Alarm DB View コントロールとアラーム データベースの間の接続を設定する必要があります。

アラーム データベースへの適切な読み取り専用アクセスのあるアカウントを使用します。システム管理者アカウントではありません。

アラーム データベースとの接続の設定

1. Alarm DB View コントロールを右クリックし、[**プロパティ**] を選択します。[**AlmDbViewCtrl のプロパティ**] ダイアログ ボックスが表示されます。
2. [**データベース**] タブを選択します。



3. アラーム データベースへの接続を設定します。以下の手順を実行します。
   a. [**認証**] リストで、認証方法(**SQL Server 認証** または **Windows 認証** (既定)) を選択します。

   注 : Windows 認証は、SQL 認証よりも優れたアプリケーション セキュリティを提供できます。この理由から、Windows 認証から SQL 認証に切り替えると、Windows 認証を使用することを推奨するポップアップ ダイアログが表示されます。この警告を無視して SQL 認証を使用する場合は、[**OK**] を選択します。同様のメッセージが SMC Log Viewer に記録されます。

   b. [**サーバー名**] ボックスに、アラーム データベースがインストールされているコンピュータのノード名を入力します。
   c. [**データベース名**] ボックスにデータベース名を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

d. ［資格情報］ボックスで認証に使用する資格情報を選択します。

注:
-［資格情報］ドロップダウンは、SQL Server 認証タイプが選択されている場合にのみ有効になります。Windows 認証では、現在ログインしているユーザーの資格情報が使用され、ユーザー名とパスワード フィールドが無効になります。
スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「資格情報マネージャの操作」を参照してください。

---

4. ［自動接続］チェック ボックスをオンにし、InTouch アプリケーションが実行を開始したときに、Alarm DB View コントロールが自動的にアラーム データベースに接続されるようにします。

   ［自動接続］チェック ボックスをオフにした場合は、Connect() メソッドを明示的に呼び出して、Alarm DB View コントロールがアラーム データベースに接続されるよう設定する必要があります。
   Connect メソッドの詳細については、「Connect()」を参照してください。

5. ［接続テスト］を選択して、アラーム データベースへの接続性を確認します。接続が正常に確立されたことを示すメッセージが表示されます。

6. 適用 を選択します。

グリッドの外観プロパティの設定

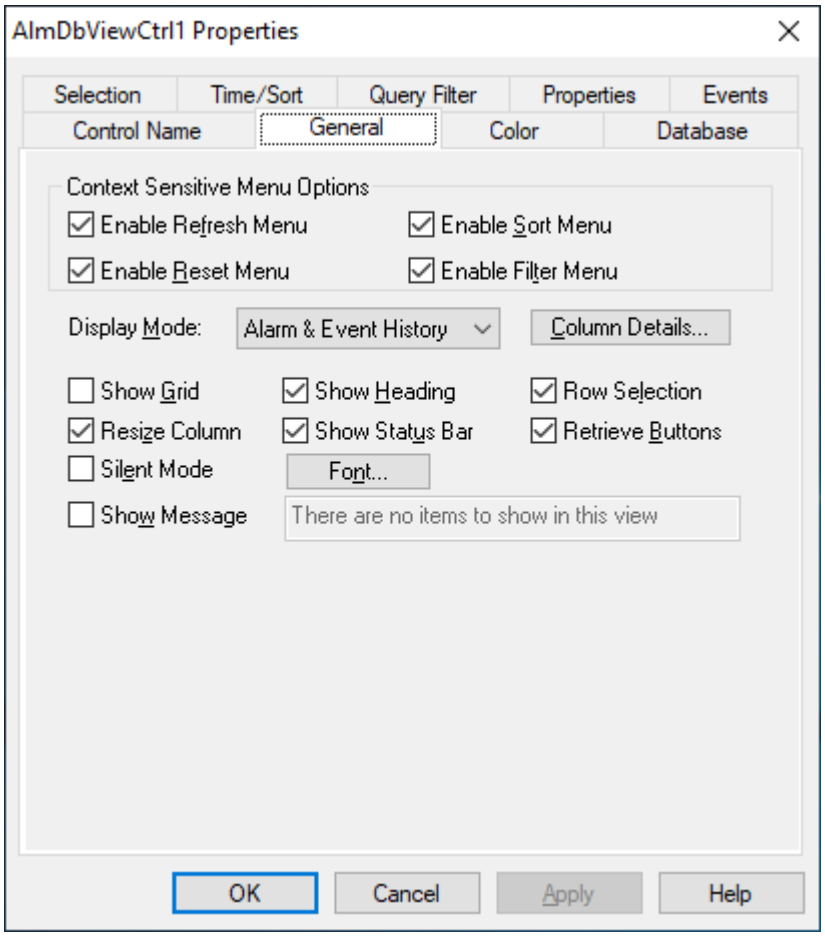
Alarm DB View コントロールの表示外観を決定するプロパティを設定できます。

グリッドの外観を設定する

1. Alarm DB View コントロールを右クリックし、［プロパティ］を選択します。［**AlmDbViewCtrl** のプロパティ］ダイアログ ボックスが表示されます。

2. [全般] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 一般的な外観を設定します。以下の手順を実行します。
   - グリッドを表示する、[**グリッドの表示**] チェック ボックスをオンにします。
   - [**サイレント モード**] チェック ボックスをオンにして、ランタイムにコントロールでエラー メッセージが表示されないようにします。
   - アラーム クエリーがレコードを返さない場合にメッセージを表示する、[**メッセージの表示**] チェックボックスをオンにします。ボックスに表示するメッセージを入力します。
   - [**見出しの表示**] チェック ボックスをオンにして、コントロールの見出しを表示します。
   - ステータス バーを表示する、[**ステータス バーの表示**] チェック クスをオンにします。
4. [**適用**] を選択します。

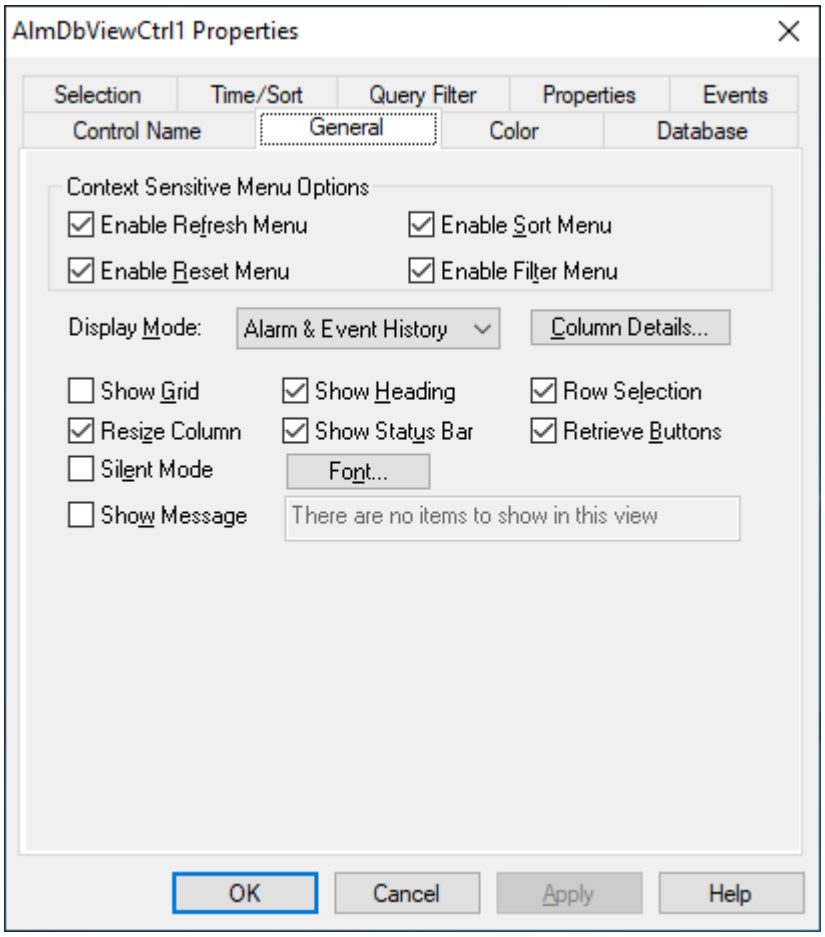
表示フォントの設定

Alarm DB View コントロールに表示されるすべてのテキストのフォントを選択できます。

フォントのプロパティを設定する

1. Alarm DB View コントロールを右クリックし、[**プロパティ**] を選択します。[**AlmDbViewCtrl のプロパティ**] ダイアログ ボックスが表示されます。
2. [**全般**] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. フォントの選択 Windows 標準の［フォント］ダイアログ ボックスが表示されます。フォントを設定して、［**OK**］を選択します。
4. [適用] を選択します。

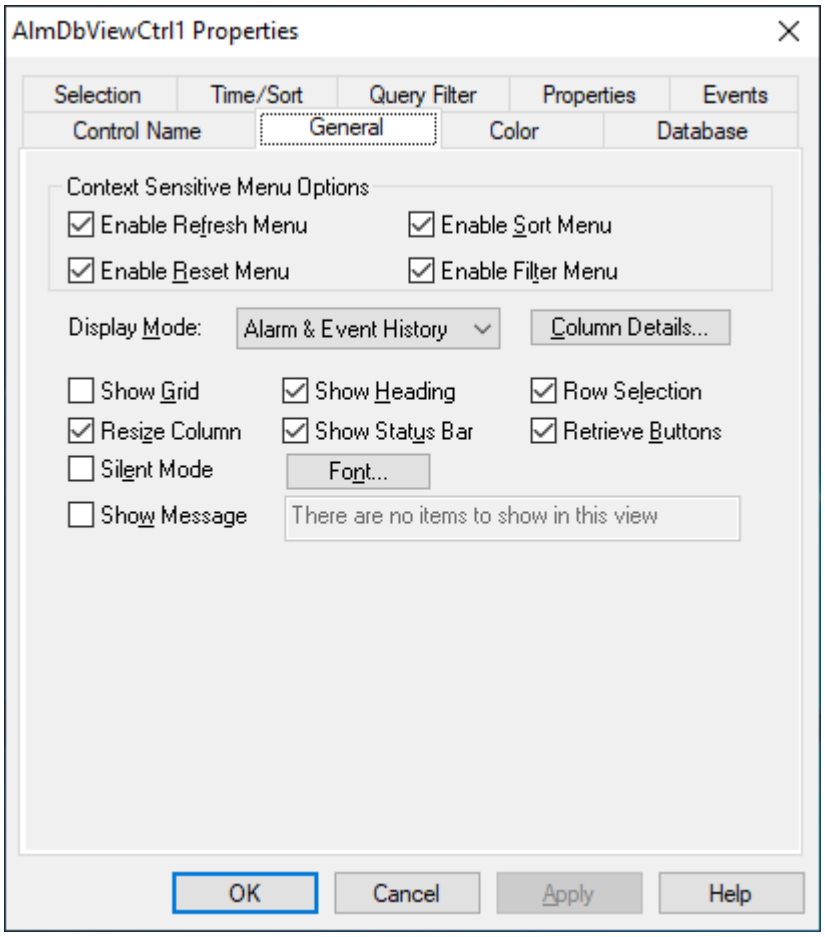
アラーム データまたはイベント データの選択

アラーム レコード、イベント レコード、またはその両方を Alarm DB View コントロールに表示するかどうかを設定できます。

データ タイプを選択する

1. Alarm DB View コントロールを右クリックし、［プロパティ］を選択します。［**AlmDbViewCtrl** のプロパティ］ダイアログ ボックスが表示されます。
2. [全般] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［表示モード］リストで、リストの履歴レコードのタイプを選択します。
   a. ［アラーム／イベント履歴］を選択し、アラームおよびイベントの履歴データベース レコードを表示します。
   b. ［アラーム履歴］を選択し、履歴アラーム レコードだけを表示します。
   c. ［イベント履歴］を選択し、履歴イベント レコードだけを表示します。
4. [適用] を選択します。

表示カラムの選択と設定

Alarm DB View コントロールの場合、以下の操作を行えます。

- カラムを選択して、順序を設定します。
- カラム幅をピクセル単位で設定します。
- カラムの名前を変更します。

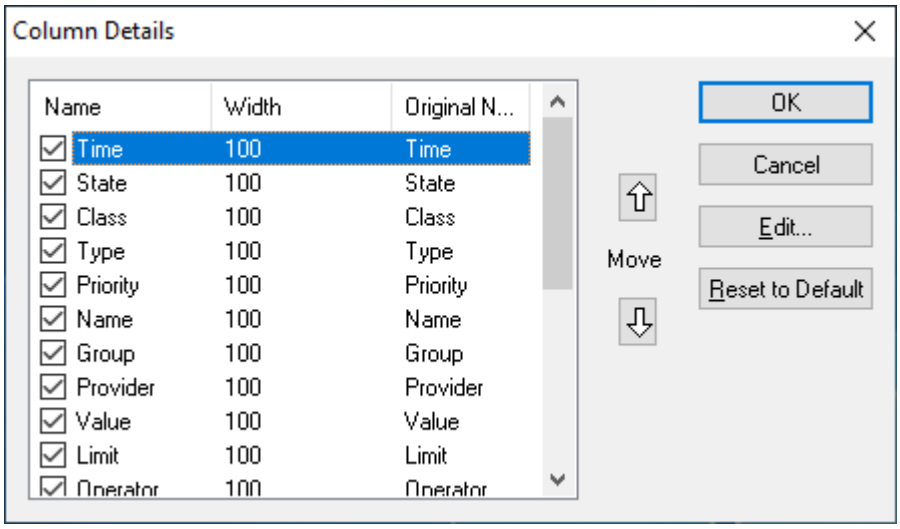
少なくとも、カラムを 1 つ選択する必要があります。

重要：表示モードが変更されると、カラム名はデフォルトのカラム名にリセットされます。カラム名を変更する前に、表示モードを最初に選択することをお勧めします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

表示カラムの詳細を設定する

1. Alarm DB View コントロールを右クリックし、［**プロパティ**］を選択します。［**AlmDbViewCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. ［**全般**］タブを選択します。
3. **列の詳細**を選択します。［**カラムの詳細**］ダイアログ ボックスが表示されます。



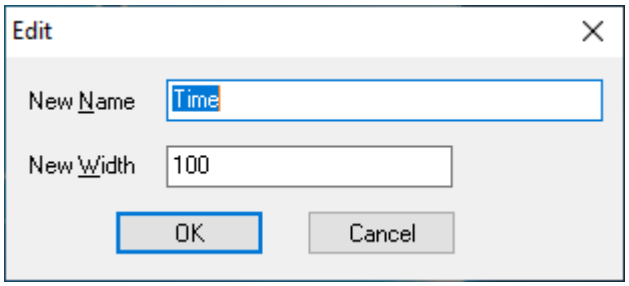
4. ［**名前**］カラムで、Alarm DB View コントロールに表示するカラム名の横のチェック ボックスをオンにします。リストから少なくとも 1 つ以上のカラムを選択する必要があります。

列名	説明
Time	アラームが発生した時刻を表示します。
状態	アラームの状態を表示します。
クラス	アラーム カテゴリを表示します。
Type	アラーム タイプを表示します。
優先度	アラーム優先度を表示します。
名前	アラームまたはイベントの原因となったタグ変数またはソースを表示します。
グループ	アラーム グループ名を表示します。
プロバイダ	アラーム プロバイダ名を表示します。
値	アラームが発生したときのタグの値を表示します。
しきい値	タグ変数のアラームしきい値を表示します。
Operator	アラーム状況に関連付けられているログオンしたオペレータの ID を表示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

列名	説明
オペレータ フル ネーム	ログオンしたオペレータのフル ネームを表示します。
オペレータ ノード	アラーム状況に関連付けられているログオンしたオペレータ ノードを表示します。 ターミナル サービス環境で、オペレータ ノードはオペレータがターミナル サービス セッションを確立したクライアント コンピュータの名前です。ノード名を取得できない場合、ノードの IP アドレスが代わりに使用されます。
オペレータ ドメイン	アラーム状況に関連付けられているログオンしたオペレータ ドメインを表示します。
アラーム コメント	タグ変数のコメントを表示します。このコメントは、データベースでタグ変数のアラームが定義されたときに［**アラーム コメント**］ボックスに入力されたものです。 アラームに対して確認コメントが指定されると、新しいコメントはこのコメント カラムで更新されます。
ユーザー 1	アラームに対応する［**ユーザー定義番号 1**］の数値を表示します。
ユーザー 2	アラームに対応する［**ユーザー定義番号 2**］の数値を表示します。
ユーザー 3	アラームに関連付けられたユーザー定義文字列プロパティの文字列値を表示します。
間隔	オペレータの選択に基づいて、未確認期間またはアラーム期間を表示します。
UTC 時間	アラームの時刻を UTC で表示します。

5. カラム名を選択し、上下の矢印を使用して、カラムを並べ替えます。［**カラムの詳細**］ダイアログ ボックスの一番上に表示されるカラム名は、アラーム コントロールの一番左のカラムの名前です。
6. カラムの名前または幅を変更する、カラムを選択して、［**編集**］を選択します。［編集］ダイアログ ボックスが表示されます。



a. ［**新しい名前**］ボックスに、新しいカラム名を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

b. ［新しい幅］ボックスに、カラム幅を入力します。カラム幅は、1 ～ 999 ピクセルの範囲で設定できます。

c. **[OK]** を選択します。

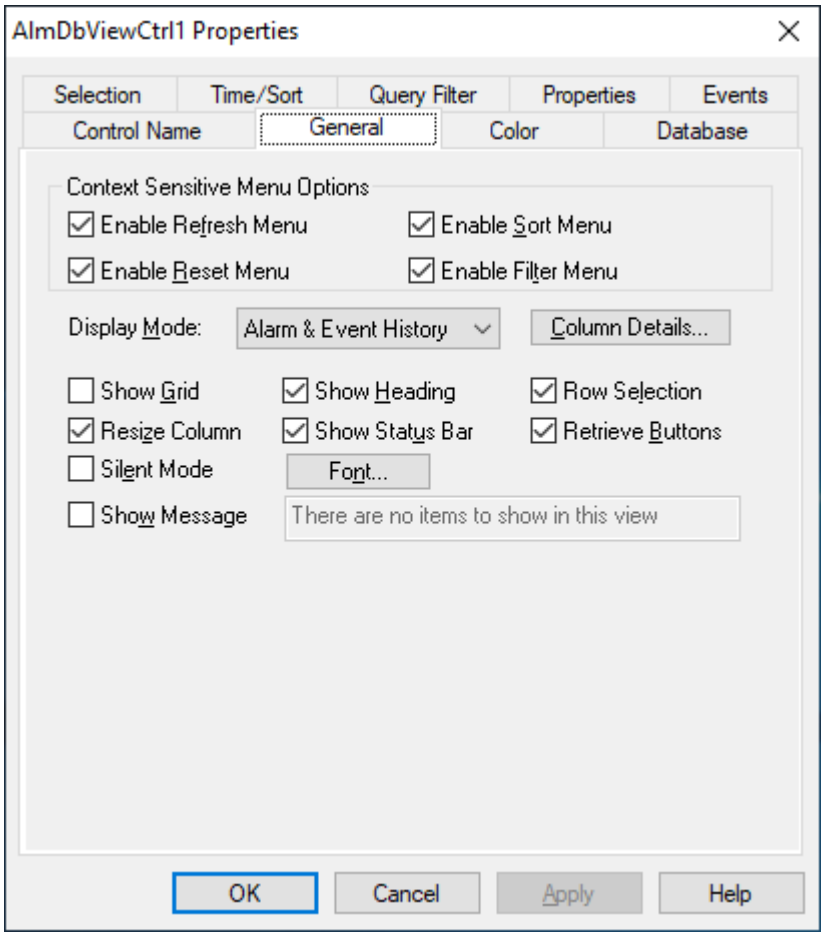
7. デフォルトのカラムの詳細設定に戻すには、［デフォルトにリセット］を選択します。
8. ［カラムの詳細］ダイアログ ボックスの［**OK**］を選択します。
9. [適用] を選択します。

ランタイム時にユーザーがアクセスできる機能の管理

このコントロールで利用できるショートカット メニュー機能を有効または無効にできます。

ランタイム機能を設定する

1. Alarm DB View コントロールを右クリックし、［プロパティ］を選択します。［**AlmDbViewCtrl** のプロパティ］ダイアログ ボックスが表示されます。
2. [全般] タブを選択します。



3. ［コンテキスト メニューのオプション］領域で、ランタイムで使用できるメニュー コマンドを設定します。
   - ランタイムでコントロールのショートカット メニューの［リフレッシュ］オプションを有効にする、［[リフレッシュ]オプションをオン］チェック ボックスをオンにします。［リフレッシュ］

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メニューは、データベースへのコントロールを更新します。正常に接続された場合は、1 から MaxRecords プロパティで定義された値までの範囲のレコードが表示されます。

- ランタイムでコントロールのショートカット メニューの［**リセット**］メニュー オプションを有効にする、［**［リセット］オプションをオン**］チェック ボックスをオンにします。［**リセット**］メニューを使用すると、すべてのカラムがデザイン時に保存した設定に戻ります。
- ランタイムでコントロールのショートカット メニューの［ソート］オプションを有効にする、［**［ソート］オプションをオン**］チェック ボックスをオンにします。このメニューを使用すると、ユーザー定義のカラム ソートを設定するために使用される［**セカンダリ ソート**］メニューが表示されます。
- ランタイムでコントロールのショートカット メニューの［**フィルタ**］オプションを有効にする、［**フィルタ メニューを有効化**］チェック ボックスをオンにします。このメニューを使用すると、ユーザー定義のフィルタ条件を設定するために使用される［フィルタ］メニューが表示されます。

4. カラムのサイズを変更できるようにする、［**カラムのサイズ変更**］チェック ボックスをオンにします。
5. アラーム レコードの選択を有効にする、［**行の選択**］チェック ボックスをオンにします。
6. コントロールの右側の取得ボタンを表示する、［**取得ボタン**］チェック ボックスをオンにします。
7. [**適用**] を選択します。

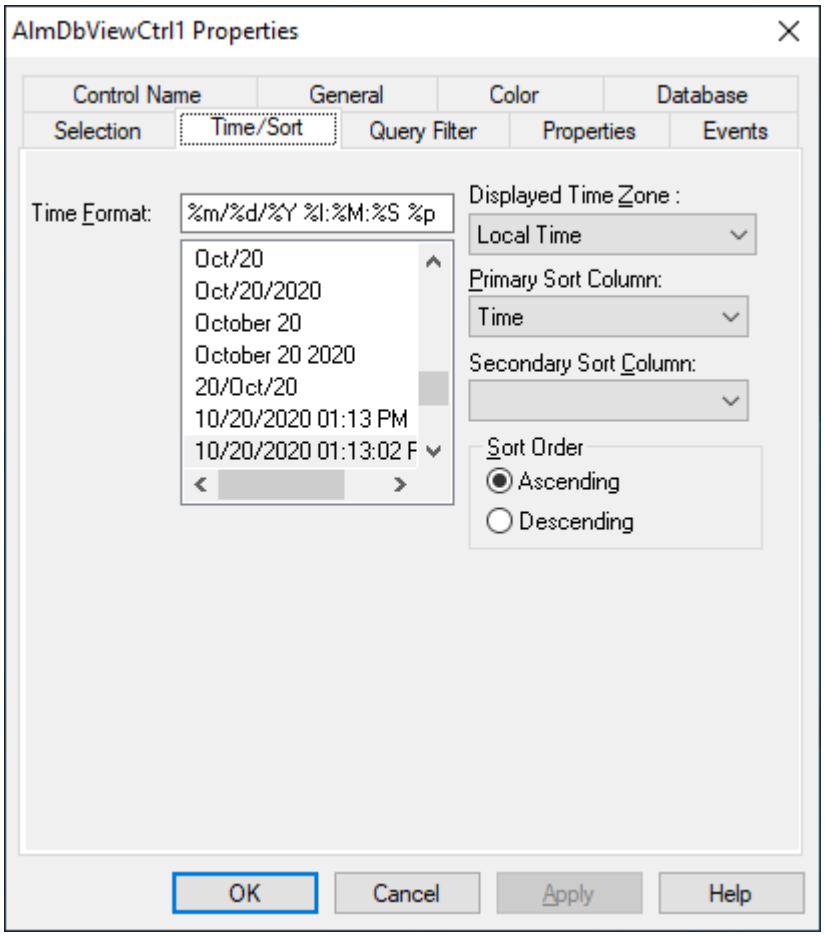
アラーム レコードに対して表示される時間形式とタイム ゾーンの設定

Alarm DB View コントロールに表示されるアラーム レコードの時間形式とタイム ゾーンを設定できます。

時間の形式を設定する

1. Alarm DB View コントロールを右クリックし、［**プロパティ**］を選択します。［**AlmDBViewCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. ［**時間／ソート**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［時間の形式］リストで、時間形式を選択します。

文字列	説明	例
d	日付を表す 2 桁の数	31
b	月を表す 3 文字の略語	8 月
Y	年を表す 4 桁の数	2007
m	月を表す 2 桁の数	11
/、	日付を区切るための記号	06/2007
y	年を表す 2 桁の数	07
#x	日付と曜日	2002 年 8 月 9 日 金曜日
B	月の名前	9 月
-、	日付を区切るための記号（-）	06-07
をクリックします。	日付を区切るための記号（.）	06.07

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列	説明	例
、	日付を区切るための記号（,）	8 月 09, 2007
H	24 時間形式の時間	22:15
:	4:41 など時間を区切るための記号	
M	分	00:41
p	AM または PM の表示	
S	秒	04:41 PM
s	秒の端数	16:41:07.390
I	AM/PM で表す 12 時間形式の時間	04:41 PM

カスタム テキストと書式設定文字を使用して、リスト ボックスに独自の形式の文字列を手動で入力できます。以下に、時間形式の文字列のサンプルを示します。

時間形式の文字列	表示
%d %b	09 8 月
%m/%d/%Y	08/09/2002
%#x	2002 年 8 月 9 日 金曜日
%Y-%m-%d	2002-08-09
%m/%d/%Y %H:%M %p	08/09/2002 16:56 PM
%m/%d/%Y %H:%M %p	2002/09/08 16:56:38.07
%I:%M %p	04:56 PM

4. ［タイム ゾーン表示］リストで、タイム ゾーンを選択します。

タイム ゾーン	説明
[GMT]	アラーム時刻スタンプでは、グリニッジ標準時が使用されます。
ローカル時間	アラーム時刻スタンプは、Alarm DB View コントロールをホストするクライアントのローカル時間で表示されます。
標準時間	アラーム時刻スタンプは、アラーム プロバイダのローカル時間で表示されます。

5. [適用] を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム データベースからのデータ期間の選択

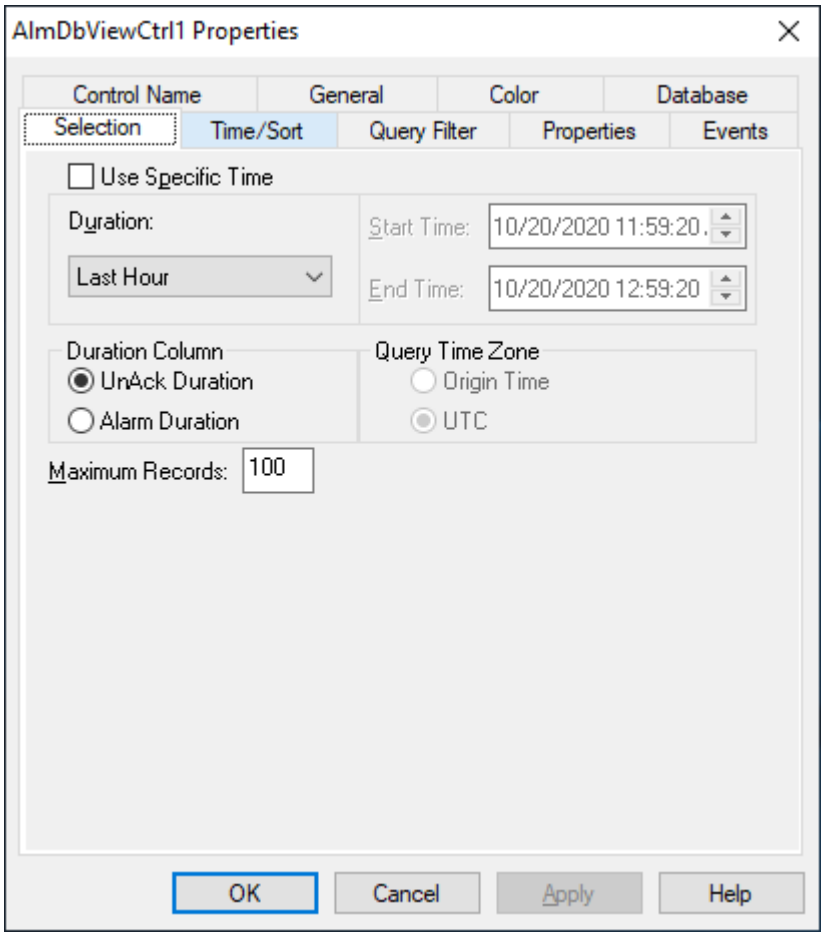
選択した時間に基づいてレコードを選択するようにクエリー値を設定できます。また、表示するレコードの最大数、アラーム クエリーの開始時間と終了時間、クエリー タイム ゾーンを設定することもできます。

アラーム データベース クエリーによって抽出されるレコードの数は、Alarm DB View コントロールにレコードが表示されるまでの時間に影響します。Alarm DB View コントロールでは、アラーム データベースから抽出された 50000 件のレコードを表示するのに約 30 秒かかります。

Alarm DB View コントロールのパフォーマンスを向上させるには、［**最大レコード数**］の値を減らして、レコードがすばやく表示されるようにします。また、［**期間の設定**］フィールドでレコードの取得間隔を短くして、Alarm DB View コントロールのパフォーマンスを向上させることもできます。

データの期間を選択する

1. Alarm DB View コントロールを右クリックし、［**プロパティ**］を選択します。［**AlmDBViewCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. ［**選択**］タブを選択します。



3. UTC タイム ゾーンを使用してデータを照会する事前定義された時間間隔を使用する、［**期間の設定**］リストから間隔を選択します。
4. 特定の開始時間と終了時間を使用する、［**指定時間を使用**］を選択して詳細を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. ［**開始時間**］ボックスに、アラーム レコードを取得する開始時間を入力します。文字列は YYYY/MM/DD HH:MM:SS 形式にします。1970 年 1 月 1 日 0 時 00 分から 2038 年 1 月 18 日 19 時 14 分 07 秒の間なら、どのタイム ゾーンのどの日付でも使用できます。

b. ［**終了時間**］ボックスに、アラーム レコードの取得を停止する終了時間を入力します。文字列は YYYY/MM/DD HH:MM:SS 形式にします。1970 年 1 月 1 日 0 時 00 分から 2038 年 1 月 18 日 19 時 14 分 07 秒の間なら、どのタイム ゾーンのどの日付でも使用できます。

c. ［**クエリー タイム ゾーン**］領域で、［**UTC**］または［**標準時間**］のどちらかを選択します。UTC 時間はグリニッジ標準時（Greenwich Mean Time）です。国際標準時または Zulu とも呼ばれます。標準時間は、オペレータのタイム ゾーンの現在の時間です。詳細については、「クエリー タイム ゾーン」トピックを参照してください。

5. ［**間隔カラム**］領域で、表示される間隔のタイプを設定します。間隔はミリ秒単位で測定されます。
   - 最後のアラームの移行（アラームまたはサブステート）から確認されるまでの時間を表示する、［**未確認間隔**］を選択します。
   - アラームの最初の発生から平常状態に戻るまでに経過した時間を表示する、［**アラーム間隔**］を選択します。

     LATCHED 状態の未確認間隔とアラーム間隔の詳細については、「LATCHED アラームのアラーム間隔と未確認間隔」トピックを参照してください。
6. ［**最大レコード数**］ボックスに、1 つのインスタンスで、コントロールから表示するレコード数を入力します。また、最大レコードの有効な範囲は、1 ～ 1000 です。
7. [**適用**] を選択します。

クエリー タイム ゾーン

［クエリー タイム ゾーン］オプションは、［**標準時間**］と［**UTC**］です。

- 標準時間は、ユーザーのタイム ゾーンのローカル時間です。
- UTC 時間はグリニッジ標準時（Greenwich Mean Time）です。国際標準時または Zulu とも呼ばれます。

［指定時間を使用］が選択されている場合、2 つのクエリー タイム ゾーン オプションを選択できます。［指定時間を使用］が選択されていない場合、［期間の設定］で選択された事前定義された時間間隔によって UTC タイム ゾーンを使用してデータが常に照会されます。

夏時間切り替えの間の問題を避けるために、［クエリー タイム ゾーン］では常に UTC を使用することが推奨されます。代わりに「ローカル時間」を使用すると、夏時間から標準時間へ移行する間にアラーム レコードが失われる可能性があります。

異なるタイム ゾーンの設定を持つ複数の異なるコンピュータを実行していて、それらのコンピュータがすべて同じアラーム データベースにログを記録している場合、各レコードは UTC のタイムスタンプを取得することに加えて、そのタイムスタンプを対応する標準時間に変換するために必要なタイム ゾーン オフセットと夏時間の調整を取得します。その結果、データベースの各エントリは 2 つのタイムスタンプを持ちます。ログ記録を行ったコンピュータからの UTC 時間と標準時間です。これにより、取得が高速になります。テーブル エントリで、UTC 時間は「TransitionTime」として、また標準時間は「EventStamp」として識別されます。

LATCHED アラームのアラーム間隔と未確認間隔

このセクションでは、LATCHED 状態のアラーム間隔と未確認間隔の計算について説明します。アラーム間隔は、アクティブ アラームと非アクティブ アラームの間の期間です。未確認間隔は、UNACK アラー

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ムと ACK アラームの間の期間です。LATCHED 状態のアラーム間隔と未確認間隔は、2 つの方法で計算できます。

オプション 1: アラームが確認されて通常状態に戻される

アラーム状態	名称	アラーム間隔	未確認間隔
UNACK	Tt	-	-
ACK	Ta	-	Tf-Tt
LATCHED	Tf	Tf-Tt	Tf-Tt
ACKRTN	Td	Tf-Tt	-

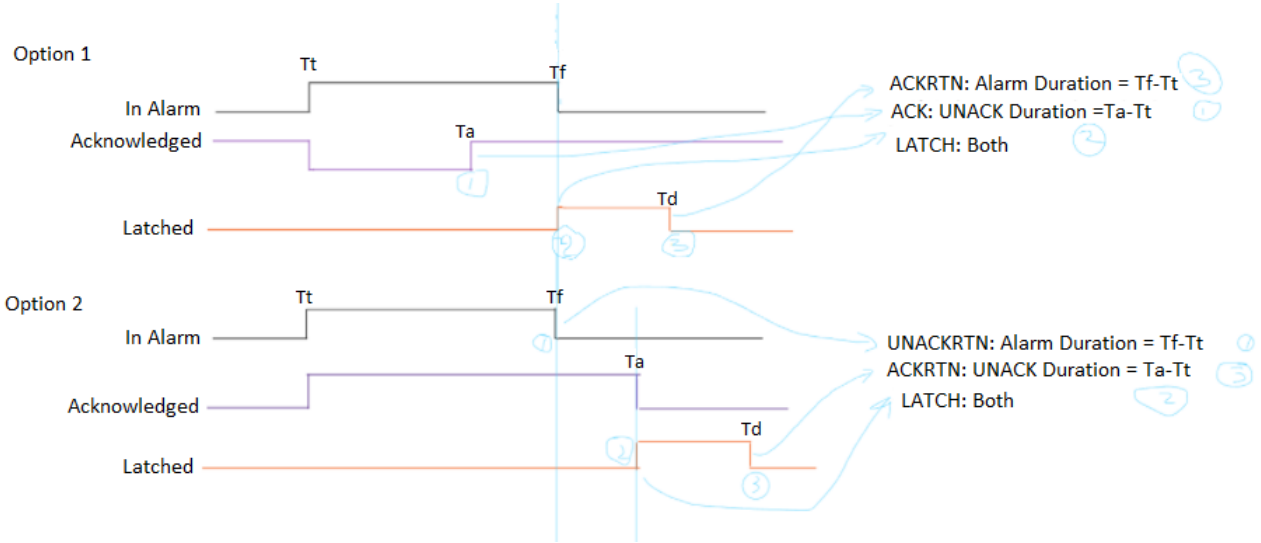
アラームが確認されると、未確認間隔は ACK 状態と LATCHED 状態の間の時間差（Ta-Tt）になります。アラーム間隔はありません。アラーム状態が LATCHED に変化すると、アラーム間隔が LATCHED と UNACK の時間差（Tf-Tt）で更新されます。未確認間隔は変化しません。LATCHED 状態が ACKRTN になると、アラーム間隔は同じ（Tf-Tt）で、未確認間隔はありません。

オプション 2: アラームが通常状態に戻されて確認される

アラーム状態	名称	アラーム間隔	未確認間隔
UNACK	Tt	-	-
UNACKRTN	Tf	Tf-Tt	-
LATCHED	Ta	Tf-Tt	Tf-Tt
ACKRTN	Td	-	Tf-Tt

アラームが返されると、アラーム間隔は UNACKRTN と UNACK の間の時間差（Tf-Tt）になります。アラーム状態が LATCHED になると、アラーム間隔は同じです。未確認間隔は、LATCHED と UNACK の時間差（Ta-Tt）になります。アラームが ACKRTN になると、未確認間隔は同じで、アラーム間隔はありません。

以下の図は、LATCHED アラームのアラーム間隔と未確認間隔の両方のオプションの間隔図を示します。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カスタム フィルタの作成とフィルタ設定の使用

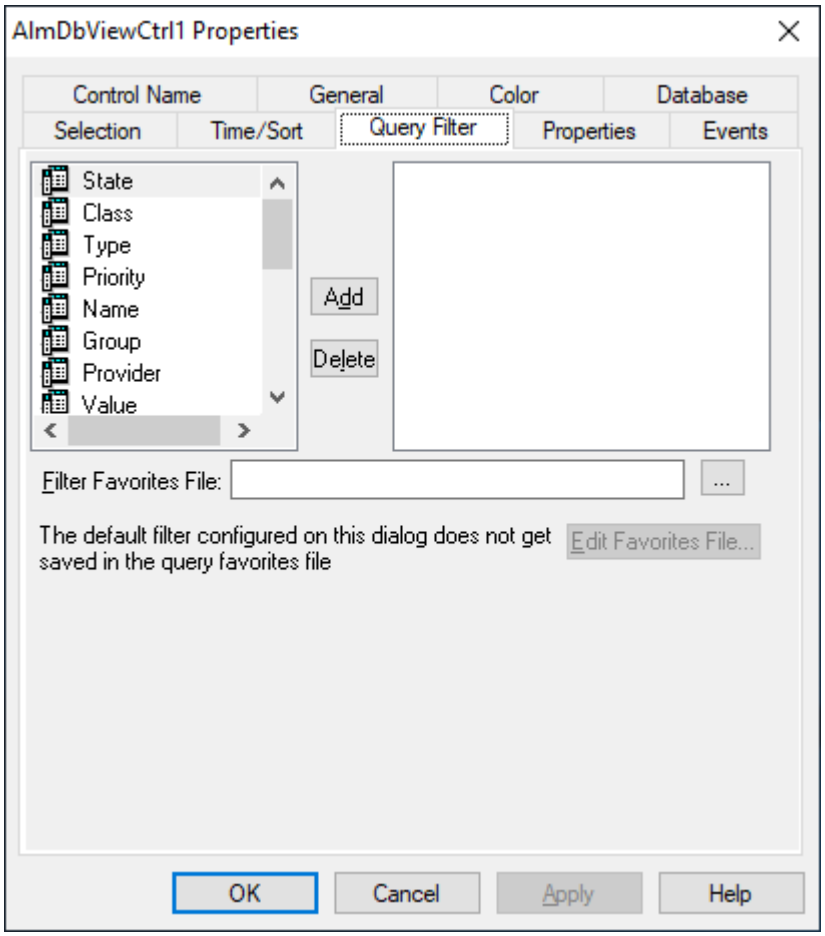
クエリー結果に含めるレコードを選択できます。たとえば、レコードの日付またはアラームの状態によってフィルタを選択できます。クエリー結果を制限または拡大するために複数のフィールドを選択できます。

カスタム フィルタを定義しないと、すべてのレコードが返されます。

言語を切り替えるために文字列を翻訳している場合、それらの文字列をフィルタ条件として使用することはできません。翻訳した文字列はアラーム データベースの外部に保存されます。

カスタム フィルタを作成する

1. **Alarm DB View** コントロールを右クリックし、［プロパティ］を選択します。［**AlmDBViewCtrl** のプロパティ］ダイアログ ボックスが表示されます。
2. ［クエリー フィルタ］タブを選択します。



3. 左ペインでフィルタ フィールドを選択し、［追加］を選択して、これらフィールドをフィルタに含めます。フィルタは右ペインに表示されます。フィルタ フィールドの詳細については、次の表を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

フィールド名	フィルタ クエリー
状態	アラーム状態。詳細については、「状況カラムの値」を参照してください。
クラス	アラーム クラス
Type	アラーム タイプ
優先度	アラーム優先度
名前	アラーム名
グループ	アラーム グループ名
プロバイダ	アラーム プロバイダ
値	アラーム値。[値] カラムの値は英数字値として表示されます。クエリー フィルタのこれらの値の比較は、文字列の比較として行われます。
しきい値	アラームしきい値。値は英数字です。クエリー フィルタの値の比較は、文字列の比較として行われます。
Operator	オペレータ
オペレータ フル ネーム	オペレータ フル ネーム
オペレータ ノード	アラームと関連のあるオペレータ ノード名
オペレータ ドメイン	アラームと関連のあるオペレータ ドメイン名
コメント	アラーム コメント
ユーザー **1**	アラーム ユーザー定義数値 1
ユーザー **2**	アラーム ユーザー定義数値 2
ユーザー **3**	アラーム ユーザー定義文字列値
期間	未確認とアラーム期間ゼロに等しい値で設定された間隔カラムでは、クエリーで null 値を持つレコードは生成されません。

4. フィルタ ペインからフィールドを削除する、削除するフィールドを選択し、[**削除**] を選択します。フィルタの削除は元に戻せません。メッセージが表示されたら、[**はい**] を選択します。
5. 各フィルタ フィールドの条件を設定します。詳細については、「カラム フィルタ条件の定義, カラム フィルタ条件の定義」を参照してください。
6. フィルタのオペレータとグループ化を設定します。詳細については、「アラーム カラムをグループ化する, アラーム カラムをグループ化する」を参照してください。
7. クエリー設定ファイルを設定します。以下の手順を実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

a. ［**フィルタ設定ファイル**］ボックスに、ネットワーク パスとファイル名を入力するか、省略記号ボタンを選択してファイルを参照します。

b. **フィルタ設定**ファイルを編集する、［**設定ファイルの編集**］ボタンを選択します。［**フィルタ設定**］ウィンドウが開き、設定ファイルのフィルタを追加、変更、または削除できます。完了したら、［**OK**］を選択して変更を保存し、ウィンドウを閉じます。

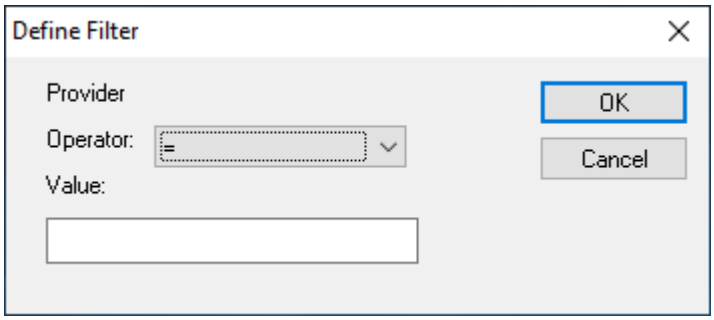
8. **[適用]** を選択します。

カラム フィルタ条件の定義

クエリーに含める各カラム フィルタについて、フィルタ条件を設定する必要があります。たとえば、特定のオペレータについてのアラームだけを表示することもできます。

カラム フィルタを定義する

1. フィールドを右クリックし、［**フィルタの編集**］を選択します。［**フィルタの定義**］ダイアログ ボックスが表示されます。



2. ［**演算子**］リストで、必要な演算子を選択します。
3. ［**値**］ボックスに、一致する必要のある条件を入力します。［**値**］ボックスでは、選択したクエリーについて処理できない値は受け入れられません。「**Like**」および「**Not Like**」のフィルタ演算子が英数字のカラム名に使われている場合、［**値**］ボックスでは次のワイルドカード文字が使用できます。

文字	検索
%	ゼロ文字以上の任意の文字列
_	任意の 1 文字
[ ]	指定範囲（[a-f] など)、またはセット([abcdef] など）内の任意の 1 文字
[^]	指定範囲にない（[^a-f] など)、またはセット内にない([^abcdef] など）任意の 1 文字

次の［値］ボックスしきい値がさまざまなフィールドに適用されます。

フィールド	しきい値
すべてのフィールド	ユーザー 1、ユーザー 2、および優先度を除く、すべての英数字

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

フィールド	しきい値
優先度	1 ～ 999 の整数値が使用できます。
ユーザー 1、ユーザー 2	負の数、正の数、または小数のみを使用できます。

4. **[OK]** を選択します。

アラーム カラムをグループ化する

複数のフィールドが定義されると、カラムはブール演算子を使用して結合されます。

- AND 演算子は、選択したフィールドのすべての値に一致するレコードを返します。
- OR 演算子は、選択した任意のフィールドの値に一致するレコードを返します。

AND/OR 演算子を使用してフィルタ選択条件を使用するには、対応するフィールドをグループ化する必要があります。フィルタ ペインのアイテムに対しては 1 つのフィルタ式のみを作成できます。複数の式が必要な場合、そのアイテムをフィルタ ペインに再び追加する必要があります。

デフォルトで、グループ化されたフィールドには AND 演算子があります。

AND 演算子と OR 演算子は親ノードです。各親ノードの下にある選択されたフィールドは子ノードです。フィールド親ノードを子ノードにドラッグすることはできません。

アラーム カラムをグループ化する

1. フィールドを右クリックし、［**グループ**］を選択します。
2. フィールドを別のフィールドにドラッグします。

クエリー フィルタのコピーまたは移動

Alarm Pareto コントロールの複数のインスタンスがあり、複数のインスタンスについて同じフィルタを使用する場合は、定義済みフィルタをあるインスタンスからコピーまたは切り取りして、別のフィルタに貼り付けます。

フィルタをコピーする

1. Alarm Pareto コントロールの最初のインスタンスで、フィルタを定義します。
2. フィルタを右クリックし、［**コピー**］を選択します。フィルタを移動する、［**切り取り**］を選択します。
3. Alarm Pareto コントロールの最初のインスタンスを終了します。
4. Alarm Pareto コントロールの次のインスタンスを開き、［**クエリー フィルタ**］タブを選択します。
5. 右ペインに矢印を位置付けします。選択したフィルタを右クリックし、［**貼り付け**］を選択します。

状況カラムの値

フィルタ クエリーに［状況］カラムを追加する場合、［**フィルタの定義**］ダイアログ ボックスの［**値**］メニューから値を割り当てることができます。使用できる値を以下の表で説明します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値	説明
確認	すべてのシステム確認のクエリーを作成します。
確認済みｱﾗｰﾑ	すべての確認済みアラームに対するクエリーを作成します。
未確認ｱﾗｰﾑ	すべての未確認アラームに対するクエリーを作成します。
確認/復帰	通常状態に戻ったすべての確認済みアラームのクエリーを作成します。
未確認/復帰	通常状態に戻った未確認アラームのクエリーを作成します。
すべての UNACK レコード	すべての未確認レコードのクエリーを作成します。
すべての ACK レコード	すべての確認レコードのクエリーを作成します。
すべての ALM レコード	すべてのアラーム レコードのクエリーを作成します。
すべての RTN レコード	通常状態に戻ったすべてのアラームのクエリーを作成します。

注：拡張サマリ アラーム モードのタグがアラームの作成に使用され、メイン アラームが確認されたときにそのアラームも通常状態に戻ると、2 つのレコードが作成されます。新しいアラームはすでに通常状態に戻っているため、最初のレコードは「確認され、通常状態に戻った」レコードです。2 番目のレコードは確認され、メイン アラームの確認に対応します。以前の ACK_ALM は ACK に変更されています。

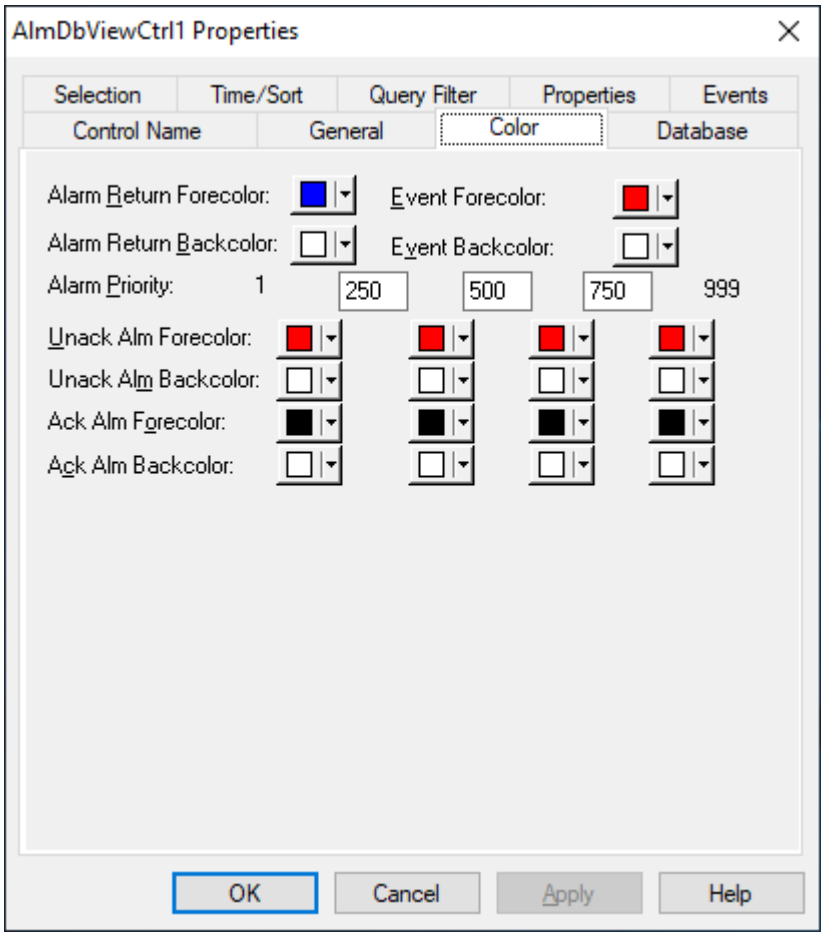
さまざまなタイプのアラーム レコードの色の設定

アラーム レコードとイベント レコードの色を設定できます。

アラーム表示の色を設定する

1. Alarm DB View コントロールを右クリックし、［**プロパティ**］を選択します。［**AlmDBViewCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. **[マルチステート色]**タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［色］ボックスを選択してパレットを開き、以下のそれぞれの色を設定します。

[オプション]	説明
アラーム復帰前景色	正常に復帰するアラームのレコード前景色を設定します。
アラーム復帰背景色	正常に復帰するアラームのレコード背景色を設定します。
イベントの前景色	イベント レコードの前景色を設定します。
イベント背景色	イベント レコードの背景色を設定します。

4. ［アラーム優先度］ボックスに、アラーム表示の変化境界値を入力します。アラームの優先度に基づいて異なる色が表示されるように、境界値を割り当てることができます。デフォルトのアラーム優先度の最小値は 1、最大値は 999 です。

   たとえば、優先度範囲 250 ～ 499 はオレンジ、優先度範囲 1 ～ 249 は赤になるように **Unack Alm Forecolor** を設定したとします。優先度 254 のアラームが発生した場合、そのアラームはオレンジ色のフォントで表示されます。優先度 12 のアラームが発生した場合、そのアラームは赤色のフォントで表示されます。

5. ［色］ボックスを選択してパレットを開き、以下のそれぞれの色を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

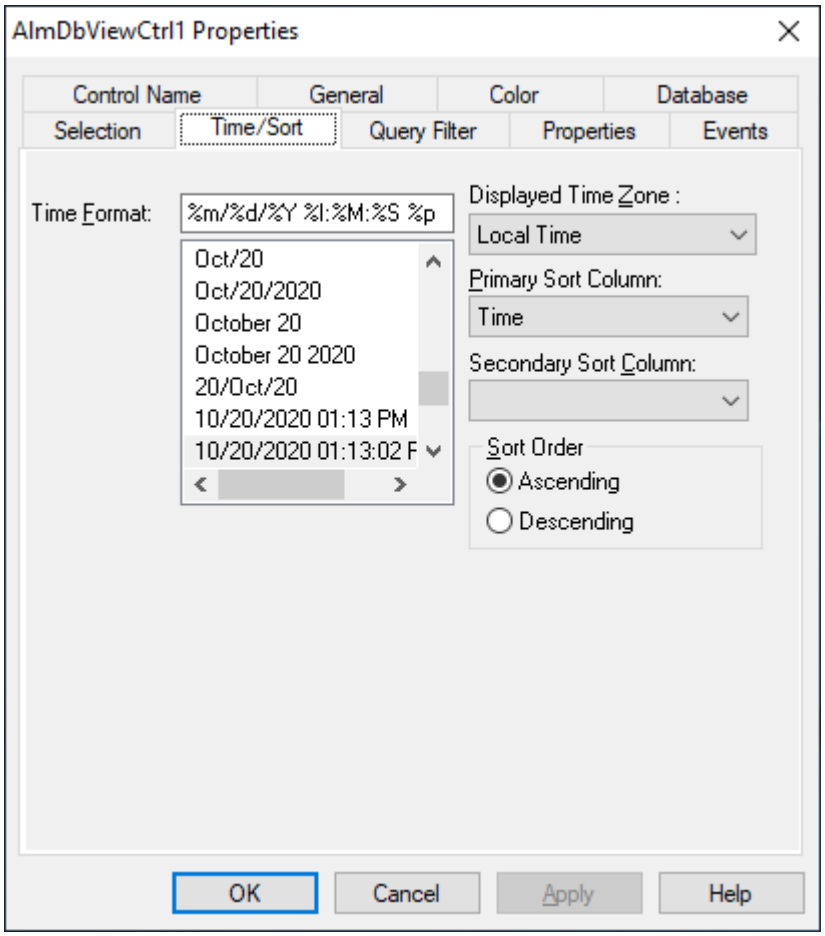
[オプション]	説明
未確認アラーム前景色	未確認アラームのそれぞれの色優先度範囲の前景色を設定します。
未確認アラーム背景色	未確認アラームのそれぞれの色優先度範囲の背景色を設定します。
確認アラーム前景色	確認アラームのそれぞれの色優先度範囲の前景色を設定します。
確認アラーム背景色	確認アラームのそれぞれの色優先度範囲の背景色を設定します。

6. **[適用]** を選択します。

アラーム レコードのソート順の設定

アラーム レコードがコントロールでソートされる方法を制御できます。

1. Alarm DB View コントロールを右クリックし、［**プロパティ**］を選択します。［**AlmDBViewCtrl** のプロパティ］ダイアログ ボックスが表示されます。
2. ［**時間／ソート**］タブを選択します。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［メイン ソート カラム］リストで、プライマリ ソート カラムの名前を選択します。［カラムのソート］リストに表示されるのは、表示されるカラムのみです。目的のカラムが見つからない場合は、［一般］タブを選択して、［カラムの詳細］からカラムを選択します。
4. ［セカンダリ ソート カラム］リストで、セカンダリ ソート カラム名を選択します。
5. ［昇順］または［降順］を選択して、ソート順を設定します。
6. **[OK]** を選択します。

ランタイムでの Alarm DB View コントロールの使用

コントロールが設定される方法に基づいて、以下の操作を行えます。

- データを取得およびリフレッシュする
- データをソートする
- データをフィルタする
- 行を選択する
- カラムをドラッグしてカラム順序を変更する
- すべてのカラムをデザイン時に保存した設定にリセットする

レコードのソート

表示でレコードをソートできます。すべての行をソートする、見出しを選択します。

コントロールを右クリックし、［ソート］を選択して、［ソートのカスタマイズ］ダイアログ ボックスを表示します。このダイアログ ボックスで、昇順または降順で 1 つまたは複数のカラムをソートできます。

ソートするカラムを指定する、カラム名の横にあるチェック ボックスをオンにします。［ソート順］矢印ボタンを使用して、カラムを並べ替えることができます。

複数のアラーム イベントが同じタイムスタンプを持つ場合、これらのイベントは期待される順序で表示されない可能性があります。

たとえば、目的のソート順がアラーム状態を基準にした降順である場合:

1. ［日付］と［状況］の両方のチェックボックスをオンにします。
2. ［状態］行を選択します。
3. 上向きの［ソート順］矢印を選択します。
4. ［ソート タイプ］領域で、［降順］を選択します。
5. **[OK]** を選択します。

ステータス バー情報の理解

ステータス バーには、コントロールの現在の状況が示されます。

- 左フレームには、接続されるサーバー名とデータベースが表示されます。
- 中央のフレームには、クエリーによって返されたレコード合計数のうち、表示されているレコードの数が表示されます。
- フレームの右側には、サーバーとの接続状況が表示されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Alarm DB View ActiveX プロパティの使用

スクリプトを使用して Alarm DB View コントロールのプロパティの値を直接設定するか、InTouch タグまたは I/O 参照に割り当てることができます。プロパティの設定の詳細については、「ActiveX コントロールのスクリプト」を参照してください。

色の値の設定の詳細については、「ActiveX コントロールの色の設定」を参照してください。

AckAlmBackColor プロパティ

確認されたアラーム背景色を取得または設定します。この設定は、確認済みアラームの個別の範囲の色設定を上書きします（AckAlmBackColorRange1～AckAlmBackColorRange4）。

タイプ

整数

デフォルト

白

構文

```
Object.AckAlmBackColor [= color]
```

値

color
背景色を決定する値または定数

AckAlmBackColorRange1 プロパティ

確認されたアラーム背景色を取得または設定します。優先度が 1～ ColorPriorityRange1 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

白

構文

```
Object.AckAlmBackColorRange1 [= color]
```

値

color
背景色を決定する値または定数

AckAlmBackColorRange2 プロパティ

確認されたアラーム背景色を取得または設定します。優先度が ColorPriorityRange1～ColorPriorityRange2 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

白

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Object.AckAlmBackColorRange2 [= color]
```

値

color
背景色を決定する値または定数

AckAlmBackColorRange3 プロパティ

確認されたアラーム背景色を取得または設定します。優先度が ColorPriorityRange2～ColorPriorityRange3 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

白

構文

```
Object.AckAlmBackColorRange3 [= color]
```

値

color
背景色を決定する値または定数

AckAlmBackColorRange4 プロパティ

確認されたアラーム背景色を取得または設定します。優先度が ColorPriorityRange3～999 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

白

構文

```
Object.AckAlmBackColorRange4 [= color]
```

値

color
背景色を決定する値または定数

AckAlmForeColor プロパティ

確認されたアラーム前景色を取得または設定します。この設定は、確認済みアラームの個別の範囲の色設定を上書きします（AckAlmForeColorRange1～AckAlmForeColorRange4）。

タイプ

整数

デフォルト

黒

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Object.AckAlmForeColor [= color]
```

値

color
テキストの色を決定する値または定数

AckAlmForeColorRange1 プロパティ

確認されたアラーム前景色を取得または設定します。優先度が 1～ColorPriorityRange1 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

黒

構文

```
Object.AckAlmForeColorRange1 [= color]
```

値

color
テキストの色を決定する値または定数

AckAlmForeColorRange2 プロパティ

確認されたアラーム前景色を取得または設定します。優先度が ColorPriorityRange1～ColorPriorityRange2 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

黒

構文

```
Object.AckAlmForeColorRange2 [= color]
```

値

color
テキストの色を決定する値または定数

AckAlmForeColorRange3 プロパティ

確認されたアラーム前景色を取得または設定します。優先度が ColorPriorityRange2～ColorPriorityRange3 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

黒

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Object.AckAlmForeColorRange3 [= color]
```

値

color
テキストの色を決定する値または定数

AckAlmForeColorRange4 プロパティ

確認されたアラーム前景色を取得または設定します。優先度が ColorPriorityRange3～999 の範囲で状況が「ACK_ALM」のレコードをこの色でコントロールに表示します。

タイプ

整数

デフォルト

黒

構文

```
Object.AckAlmForeColorRange4 [= color]
```

値

color
テキストの色を決定する値または定数

AckRtnBackColor プロパティ

正常に復帰する確認済みアラームの背景色を取得または設定します（ACK_RTN）。

タイプ

整数

デフォルト

白

構文

```
Object.AckRtnBackColor [= color]
```

値

color
背景色を決定する値または定数

AckRtnForeColor プロパティ

正常に復帰する確認済みアラームのテキスト色を取得または設定します（ACK_RTN）。

タイプ

整数

デフォルト

青

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Object.AckRtnForeColor [= color]
```

値

color
テキストの色を決定する値または定数

AlmRtnBackColor プロパティ

返されたアラーム背景色を取得または設定します。この色は状況が「ALM_RTN」のレコードに適用されます。

タイプ

整数

デフォルト

白

構文

```
Object.AlmRtnBackColor [= color]
```

値

color
背景色を決定する値または定数

AlmRtnForeColor プロパティ

返されたアラーム前景色を取得または設定します。この色は状況が「ALM_RTN」のレコードに適用されます。

タイプ

整数

デフォルト

青

構文

```
Object.AlmRtnForeColor [= color]
```

値

color
テキストの色を決定する値または定数

Authentication プロパティ

［データベース］タブで選択された認証タイプを返します。

タイプ

メッセージ

構文

```
AType = AlarmDBCtrl1.Authentication;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値

AType
メッセージ値（'SQL 認証' または 'Windows 認証'）。

AuthenticationMode プロパティ

認証モードを設定します。

タイプ

整数型

構文

```
Object.AuthenticationMode = [Int]
```

値

Int
0 または 1 の整数値。0 は SQL 認証を示し、1 は Windows 認証を示します。

AutoConnect プロパティ

ランタイムでコントロールがデータベースに自動的に接続されるかどうかを決定する値を取得または設定します。

データ タイプ

整数型

デフォルト

False

構文

```
Object.AutoConnect [= Integer]
```

値

Integer
ランタイムでコントロールがデータベースに接続されるかどうかを指定する整数式
True = データベースに接続します。
False =（デフォルト）データベースに接続しません。

備考

データベースに接続するには、Connect() メソッドを明示的に呼び出す必要があります。

ColorPriorityRange1 プロパティ

アラームが表示される優先度範囲の境界を設定します。このプロパティの値は 1 より大きく、ColorPriorityRange2 の値よりも小さい値を指定する必要があります。

タイプ

整数

デフォルト

250

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Object.ColorPriorityRange1 [= integer または priority]
```

ColorPriorityRange2 プロパティ

アラームが表示される優先度範囲の境界を設定します。このプロパティには、ColorPriorityRange1 より大きく、ColorPriorityRange3 よりも小さい値を指定する必要があります。

タイプ

整数

デフォルト

500

構文

```
Object.ColorPriorityRange2 [= integer または priority]
```

ColorPriorityRange3 プロパティ

アラームが表示される優先度範囲の境界を設定します。このプロパティの値には、ColorPriorityRange2 より大きく、999 よりも小さい値を指定する必要があります。

タイプ

整数

デフォルト

750

構文

```
Object.ColorPriorityRange3 [= integer または priority]
```

ColumnResize プロパティ

カラムのサイズ変更ができるかどうかを決定する値を返すか、または設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.ColumnResize [= Discrete]
```

値

Discrete
True = （デフォルト）ランタイムにカラムのサイズを変更できます。
False = カラムのサイズは変更できません。

ConnectStatus プロパティ

接続状況を返します。このプロパティは読み取り専用です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データタイプ

メッセージ

構文

```
Object.ConnectStatus
```

値

接続中 = コントロールはデータベースに接続されています。
未接続 = コントロールはデータベースに接続されていません。
処理中 = コントロールをデータベースに接続しています。

例

コントロール名は AlmDbView1、タグ変数はメッセージ型タグ変数です。

```
Tagname = #AlmDbView1.ConnectStatus;
```

CustomMessage プロパティ

アラーム レコードをアラーム データベースから取得できないとき、Alarm DB View コントロールが表示するメッセージを取得または設定します。

タイプ

メッセージ

デフォルト

表示するアイテムがありません。

構文

```
Object.CustomMessage [= string]
```

DatabaseName プロパティ

接続するデータベースを指定します。

タイプ

メッセージ

構文

```
Object.DatabaseName [= text]
```

DisplayMode プロパティ

コントロールの表示モードを返します。表示モードによって、アラームのみ、イベントのみ、またはアラームとイベントの両方が表示されます。このプロパティは読み取り専用です。

タイプ

文字列型

デフォルト

Alarms & Events History（アラーム／イベント履歴）

構文

```
Object.DisplayMode
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

可能な値は以下のとおりです。
Alarms & Events History（アラーム／イベント履歴）
Alarms History（アラーム履歴）
Events History（イベント履歴）

例

コントロール名は AlmDbView1、タグ変数はメッセージ型タグ変数です。

```
tag = #AlmDbView1.DisplayMode;
```

DisplayedTimeZone プロパティ

表示されるタイム ゾーンを取得または設定します。

タイプ

文字列型

デフォルト

ローカル時間

構文

```
Object.DisplayedTimeZone [= message]
```

備考

可能な値は以下のとおりです。

GMT - アラーム タイム スタンプでは、グリニッジ標準時が使用されます。

Local Time - アラーム時刻スタンプは、Alarm DB View コントロールをホストするクライアントのローカル時間で表示されます。

Origin Time - アラーム時刻スタンプは、アラーム プロバイダのローカル時間で表示されます

Duration プロパティ

開始時間と終了時間を設定する期間を取得または設定します。

タイプ

メッセージ

デフォルト

Last Hour

構文

```
Object.Duration [= text]
```

値

text
期間を含む文字列式。このプロパティには、以下の文字列の 1 つを持つ必要があります。
Last Minute（1 分前）
Last 5 Minutes（5 分前）
Last 15 Minutes（5 分前）
Last Half Hour（30 分前）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Last Hour（1 時間前）
Last 2 Hours（2 時間前）
Last 4 Hours（2 時間前）
Last 8 Hours（2 時間前）
Last 12 Hours（2 時間前）
Last Day（1 日前）
過去 2 日間
過去 3 日間
先週
過去 2 週間
Last 30 days（30 日前）
Last 90 days（30 日前）

EndTime プロパティ

終了時間を返すか、設定します。

タイプ

メッセージ

構文

```
Object.EndTime [= text]
```

値

text
終了時間を評価する文字列式。返される文字列は常に MM/DD/YYYY HH:MM:SS の形式です。同じ形式は、文字列の値を設定するためにも必要になります。このプロパティは、1970 年 1 月 1 日 0 時 00 分から 2038 年 1 月 18 日の 19 時 14 分 07 秒の間であれば、どのタイム ゾーンでも対応できます。

EventBackColor プロパティ

イベント アラーム背景色を取得または設定します。この色は状況が「EVT_EVT」のコントロールに表示されるレコードに適用されます。

タイプ

整数

デフォルト

白

構文

```
Object.EventBackColor [= color]
```

値

color
背景色を決定する値または定数

EventForeColor プロパティ

イベント アラーム前景色を取得または設定します。この色は状況が「EVT_EVT」のコントロールに表示されるレコードに適用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タイプ

整数

デフォルト

赤

構文

```
Object.EventForeColor [= color]
```

値

color
テキストの色を決定する値または定数

FilterFavoritesFile プロパティ

フィルタ設定ファイルを取得または設定します。このファイルは、[**フィルタ設定**] ダイアログ ボックスでフィルタ設定を読み取りまたは書き込みするために使用されます。

タイプ

文字列型

デフォルト

Null

構文

```
Object.FilterFavoritesFile [= String]
```

FilterMenu プロパティ

ショートカット メニューに [**フィルタ**] メニュー項目を表示するかどうかを決定する値を取得または設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.FilterMenu [= Discrete]
```

値

True の場合、[**フィルタ**] メニュー項目が表示されます（デフォルト）。

False の場合、[**フィルタ**] メニュー項目は表示されません。

FilterName プロパティ

現在のフィルタ名を返します（フィルタ名が存在する場合）。

タイプ

文字列（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

デフォルト

Null

構文

```
Object.FilterName [= String]
```

FromPriority プロパティ

コントロールの優先度始点の値を取得または設定します。

タイプ

整数

デフォルト

1

構文

```
Object.FromPriority [= integer]
```

備考

このプロパティを使用して、表示されるアラーム レコードをフィルタできます。たとえば、このプロパティを 760 に設定すると、プロパティが 760～ToPriority プロパティ値のアラームだけが表示されます。

GroupExactMatch プロパティ

GroupExactMatch プロパティが True の場合、GroupName プロパティ値と正確に一致するアラーム グループ名を持つアラームのみが表示されます。False の場合、グループ名で指定する必要があるのは、フィルタするアラーム グループ名の一部だけです。

タイプ

論理型

デフォルト

False

構文

```
Object.GroupExactMatch [= discrete]
```

備考

このプロパティを GroupName プロパティと共に使用して、Alarm DB View コントロールをフィルタします。

例

次に例を示します。

```
#AlarmDBViewCtrl1.GroupName = "Group"
#AlarmDBViewCtrl1.GroupExactMatch = 0;
#AlarmDBViewCtrl1.Refresh();
```

GroupName プロパティ

現在の Alarm DB View コントロール用のアラーム グループ名を取得または設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タイプ

文字列型

デフォルト

（なし）

構文

```
Object.GroupName [= GroupName]
```

備考

このプロパティを「GroupA」に設定して、表示に対して再度クエリーを実行すると、GroupA アラームグループに属するタグ変数のみが表示されます。

MaxRecords プロパティ

取得する最大レコード数を返すか、設定します。

タイプ

整数

デフォルト

100

構文

```
Object.MaxRecords [=integer]
```

値

integer
指定の時間に取得されるレコード数を指定する整数式。最大レコード数は 1～1000 までの範囲です。最高のパフォーマンスを得るためには、この値をできるだけ小さくしてください。

Password プロパティ

データを取得するための SQL Server パスワードを返すか、設定します。

タイプ

メッセージ

構文

```
Object.Password [= text]
```

値

text
パスワードを評価する文字列式

PrimarySort プロパティ

アラーム表示をソートするために使用されるプライマリ カラム名を取得または設定します。

タイプ

メッセージ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

デフォルト

(なし)

構文

```
Object.PrimarySort [= message]
```

ProviderExactMatch プロパティ

ProviderExactMatch プロパティが True の場合、ProviderName プロパティ値と正確に一致するアラーム プロバイダ名を持つアラームだけが表示されます。False の場合、プロバイダ名で指定する必要があるのは、フィルタするアラーム プロバイダ名の一部だけです。

タイプ

論理型

デフォルト

False

構文

```
Object.ProviderExactMatch [= discrete]
```

備考

このプロパティを ProviderName プロパティと共に使用して、Alarm DB View コントロールをフィルタします。

例

次に例を示します。

```
#AlarmDBViewCtrl1.ProviderName = "Provider"
#AlarmDBViewCtrl1.ProviderExactMatch = 0;
#AlarmDBViewCtrl1.Refresh();
```

ProviderName プロパティ

現在の Alarm DB View コントロール用のアラーム プロバイダ名フィルタを取得または設定します。

タイプ

文字列型

デフォルト

(なし)

構文

```
Object.ProviderName [= ProviderName]
```

備考

あるタグ変数が Provider1 アラーム プロバイダに属する場合、このプロパティを「Provider1」に設定し、表示のクエリーを再実行すると、Provider1 アラーム プロバイダに属するタグ変数だけが表示されます。

QueryTimeZoneName プロパティ

特定の時間がクエリーに使用されるとき、タイム ゾーンを取得または設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タイプ

論理型

デフォルト

False

構文

```
Object.QueryTimeZone [= Discrete]
```

値

True = GMT

False = 標準時間。アラーム プロバイダのローカル時間です。

RefreshMenu プロパティ

ショートカット メニューに［リフレッシュ］メニュー項目を表示するかどうかを決定する値を取得または設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.RefreshMenu [= Discrete]
```

値

True の場合、［リフレッシュ］メニュー項目が表示されます（デフォルト）。

False の場合、［リフレッシュ］メニュー項目は表示されません。

ResetMenu プロパティ

ショートカット メニューに［リセット］メニュー項目を表示するかどうかを決定する値を取得または設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.ResetMenu [= Discrete]
```

値

True の場合、［リセット］メニュー項目が表示されます（デフォルト）。

False の場合、［リセット］メニュー項目は表示されません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

RowCount プロパティ

コントロールで表示されているレコード数を返します。このプロパティは読み取り専用です。

タイプ

整数

構文

```
Object.RowCount
```

例

コントロール名は AlmDbView1、タグ変数は整数型タグ変数です。

```
tagname = #AlmDbView1.RowCount;
```

RowSelection プロパティ

ランタイムで行を選択できるかどうかを決定する値を返すか、設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.RowSelection [= Discrete]
```

値

論理型

True = 行を選択できます（デフォルト）。

False = 行を選択できません。

備考

行を選択できない場合、Click イベントまたは DoubleClick イベントは生成されません。

SecondarySort プロパティ

アラーム表示をソートするために使用されるセカンダリ カラム名を取得または設定します。

タイプ

メッセージ

デフォルト

（なし）

構文

```
Object.SecondarySort [= text]
```

ServerName プロパティ

データを取得するためにコントロールが接続するサーバー名を返すか、設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タイプ

メッセージ

構文

```
Object.ServerName [= text]
```

ShowFetch プロパティ

取得ボタンを表示するかどうかを決定する値を返すか、設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.ShowFetch [= Discrete]
```

値

Discrete
True = 取得ボタンが表示されます（デフォルト）。
False = 取得ボタンが表示されません。

ShowGrid プロパティ

グリッド線を表示するかどうかを決定する値を返すか、設定します。

タイプ

論理型

デフォルト

False

構文

```
Object.ShowGrid [= Discrete]
```

値

Discrete
True = グリッド線が表示されます。
False = グリッド線が表示されません（デフォルト）。

ShowHeading プロパティ

カラム見出しを表示するかどうかを決定する値を返すか、設定します。

タイプ

論理型

デフォルト

True

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Object.ShowHeading [= Discrete]
```

値

Discrete
True = カラム見出しが表示されます（デフォルト）。
False = カラム見出しが表示されません。

ShowMessage プロパティ

アラーム データベースにレコードがないとき、「表示するアイテムがありません。」に対するカスタマイズしたメッセージを表示するかどうかを決定します。

タイプ

論理型

デフォルト

False

構文

```
Object.ShowMessage [= discrete]
```

ShowStatusBar プロパティ

ステータス バーを表示するかどうか決定する値を返すか、または設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.ShowStatusBar [= Discrete]
```

値

Discrete
True = ステータス バーが表示されます（デフォルト）。
False = ステータス バーは表示されません。

SilentMode プロパティ

コントロールをサイレント モードにするかどうかを決定する値を取得または設定します。

タイプ

論理型

デフォルト

False

構文

```
Object.SilentMode [= Discrete]
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値

True = サイレント モードをオンにします。

False =サイレント モードをオフにします（デフォルト）。

SortMenu プロパティ

ショートカット メニューに［ソート］メニュー項目を表示するかどうかを決定する値を返すか、設定します。

タイプ

論理型

デフォルト

True

構文

```
Object.SortMenu [= Discrete]
```

値

論理型式

True = ［ソート］メニュー項目が表示されます（デフォルト）。

False = ［ソート］メニュー項目は表示されません。

SortOrder プロパティ

ソートされるカラムに従って、アラームのソート順を取得または設定します（プライマリ ソート カラム）。

タイプ

論理型

デフォルト

True

構文

```
Object.SortOrder [= discrete]
```

値

論理値式

True = 昇順

False = 降順

SpecificTime プロパティ

コントロールが StartTime プロパティと EndTime プロパティを使用するかどうか、または Duration プロパティの値に基づいて開始時間と終了時間を計算するかどうかを決定する値を返すか、設定します。

タイプ

論理型

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

デフォルト

False

構文

```
Object.SpecificTime [= Discrete]
```

値

True = StartTime プロパティと EndTime プロパティを使用します。

False = "Duration" プロパティに基づいて StartTime と EndTime が計算されます（デフォルト）。

StartTime プロパティ

開始時間を返すか、設定します。

タイプ

メッセージ

構文

```
Object.StartTime [= text]
```

値

text
開始時間を評価する文字列式。返される文字列は常に MM/DD/YYYY HH:MM:SS の形式です。同じ形式は、文字列の値を設定するためにも必要になります。このプロパティは、1970 年 1 月 1 日 0 時 00 分から 2038 年 1 月 18 日の 19 時 14 分 07 秒の間ならば、どのタイム ゾーンでも対応できます。

Time プロパティ

表示に使用される時間形式を取得および設定します。

タイプ

メッセージ

デフォルト

%m/%d/%Y %I:%M:%S %p

構文

```
Object.Time [= message]
```

備考

時間形式文字列の詳細については、「アラーム レコードに対して表示される時間形式とタイム ゾーンの設定」を参照してください。

ToPriority プロパティ

コントロールの優先度終点の値を取得または設定します。

タイプ

整数

デフォルト

999

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Object.ToPriority [= integer]
```

備考

このプロパティを使用して、表示されるアラーム レコードをフィルタします。たとえば、このプロパティを 900 に設定すると、プロパティが FromPriority プロパティ値～ 900 のアラームだけが表示されます。

TotalRowCount プロパティ

現在のクエリーに対するレコードの合計数を返します。このプロパティは読み取り専用です。

タイプ

整数

構文

```
Object.TotalRowCount
```

備考

行カウントは現在のクエリーで返された行数であり、取得されたレコード数が MaxRecords プロパティよりも少ない場合を除いて、通常は MaxRecords プロパティと同じになります。たとえば、特定の条件下で 950 のレコードがあり、MaxRecords プロパティが 100 の場合、最終ページのレコード数は 50 になり、行数も 50 になります。同じ例で、TotalRowCount プロパティは必ず 950 になります。

例

コントロール名は AlmDbView1、タグ変数は整数型タグ変数です。

```
tagname = #AlmDbView1.TotalRowCount;
```

UnAckAlmBackColor プロパティ

未確認のアラーム背景色を取得または設定します。この色は状況が「UNACK_ALM」のコントロールに表示されるすべてのレコードに適用されます。UnAckAlmBackColorRange1 プロパティ～ UnAckAlmBackColorRange4 プロパティの値による設定を上書きします。

タイプ

整数

デフォルト

白

構文

```
Object.UnAckAlmBackColor [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

UnAckAlmBackColorRange1 プロパティ

未確認のアラーム背景色を取得または設定します。この色は優先度が 1～ColorPriorityRange1 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

タイプ

整数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

デフォルト

白

構文

```
Object.UnAckAlmBackColorRange1 [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

UnAckAlmBackColorRange2 プロパティ

未確認のアラーム背景色を取得または設定します。この色は優先度が ColorPriorityRange1～ ColorPriorityRange2 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

タイプ

整数

デフォルト

白

構文

```
Object.UnAckAlmBackColorRange2 [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

UnAckAlmBackColorRange3 プロパティ

未確認のアラーム背景色を取得または設定します。この色は優先度が ColorPriorityRange2～ ColorPriorityRange3 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

タイプ

整数

デフォルト

白

構文

```
Object.UnAckAlmBackColorRange3 [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

UnAckAlmBackColorRange4 プロパティ

未確認のアラーム背景色を取得または設定します。この色は優先度が ColorPriorityRange3～999 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タイプ

整数

デフォルト

白

構文

```
Object.UnAckAlmBackColorRange4 [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

UnAckAlmForeColor プロパティ

未確認のアラーム テキスト色を取得または設定します。この色は状況が「UNACK_ALM」のコントロールに表示されるすべてのレコードに適用されます。UnAckAlmForeColorRange1 プロパティ～UnAckAlmForeColorRange4 プロパティの値による設定を上書きします。

タイプ

整数

デフォルト

赤

構文

```
Object.UnAckAlmBackColor [= color]
```

値

color
テキストの色を決定する値または定数

UnAckAlmForeColorRange1 プロパティ

未確認のアラーム前景色を取得または設定します。この色は優先度が 1～ColorPriorityRange1 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

タイプ

整数

デフォルト

赤

構文

```
Object.UnAckAlmForeColorRange1 [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

UnAckAlmForeColorRange2 プロパティ

未確認のアラーム前景色を取得または設定します。この色は優先度が ColorPriorityRange1～ ColorPriorityRange2 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

タイプ

整数

デフォルト

赤

構文

```
Object.UnAckAlmForeColorRange2 [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

UnAckAlmForeColorRange3 プロパティ

未確認のアラーム前景色を取得または設定します。この色は優先度が ColorPriorityRange2～ ColorPriorityRange3 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

タイプ

整数

デフォルト

赤

構文

```
Object.UnAckAlmForeColorRange3 [= color]
```

値

color
特定オブジェクトの色を決定する値または定数

UnAckAlmForeColorRange4 プロパティ

未確認のアラーム前景色を取得または設定します。この色は優先度が ColorPriorityRange3～999 の範囲で状況が「UNACK_ALM」のコントロールに表示されるレコードに適用されます。

タイプ

整数

デフォルト

赤

構文

```
Object.UnAckAlmForeColorRange4 [= color]
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値

color
特定オブジェクトの色を決定する値または定数

UnAckOrAlarmDuration プロパティ

間隔カラムは［未確認間隔］または［アラーム間隔］のいずれかを表示します。FALSE（0）は未確認間隔、TRUE（1）はアラーム間隔です。

タイプ

整数

デフォルト

False

構文

```
Object.UnAckOrAlarmDuration [= integer]
```

UserID プロパティ

データを取得するためにコントロールが SQL Server に接続する際に使用するユーザー ID を返すか、設定します。

タイプ

メッセージ

構文

```
Object.UserID [= text]
```

値

text
ユーザー ID を評価する文字列式

Alarm DB View ActiveX メソッドの使用

Alarm DB View ActiveX メソッドを使用すると、以下の操作を行えます。

- データベース接続を制御します。
- アラーム データベースからレコードを取得します。
- アラーム情報を取得します。
- 表示の外観をリセットします。
- アラーム レコードをソートする
- ［コンテキスト］メニューを表示します。
- フィルタ設定にアクセスします。

メソッドの呼び出しの詳細については、「ActiveX コントロールのスクリプト」を参照してください。

アラーム データベース接続の制御

アラーム データベースに接続するには、Connect() メソッドを使用します。接続を解除するには、Disconnect() メソッドを使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Connect()

コントロールをデータベースに接続し、接続が成功すると 1～MaxRecords の範囲にあるレコード セットを表示します。

構文

```
Object.Connect()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.Connect();
```

Disconnect()

コントロールをデータベースから接続解除します。

構文

```
Object.Disconnect()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.Disconnect();
```

アラーム データベースからレコードを取得

以下のメソッドを使用して、データベースからレコードを取得し、表示を更新します。

- SelectQuery()
- GetPrevious()
- GetNext()
- Refresh()

SelectQuery()

現在の表示を .xml ファイルで指定されたクエリー名に設定します。

構文

```
Object.SelectQuery("QueryName");
```

パラメータ

QueryName
フィルタ設定ファイルで定義されたクエリー名

例

この例では、現在 AlmDbView1 コントロールに関連付けられているフィルタ設定ファイルの「HighPriority」と呼ばれるクエリーによって定義されるフィルタ条件を適用しています。

```
#AlmDbView1.SelectQuery("HighPriority");
```

GetPrevious()

データベースから以前のレコード セットを取得します（存在する場合）。

構文

```
Object.GetPrevious();
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.GetPrevious();
```

GetNext()

データベースから次のレコード セットを取得します（存在する場合）。

構文

```
Object.GetNext()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.GetNext();
```

Refresh()

データベースからのコントロールを更新し、接続が成功すると 1～MaxRecords の範囲にあるレコード セットを表示します。

構文

```
Object.Refresh
```

備考

Refresh() メソッドを呼び出すことによって Alarm DB View コントロールの表示の更新を初期化した後、更新が完了するまで、RowCount プロパティと TotalRowCount プロパティの値は -1 に変化します（つまり、関連するすべてのレコードがデータベースから取得されます）。更新が完了すると、両方のプロパティは現在の正しい行数で更新されます。

Refresh() メソッドは非同期で機能します。すぐに呼び出しスクリプトにコントロールを返し、バックグラウンドで作業を続行します。つまり、Refresh() を呼び出した後ですぐに RowCount と TotalRowCount の値をクエリーすると、更新がまだ完了していない時点で値がクエリーされているため、-1 が返される可能性が非常に高いです。正しい値を取得する方法の 1 つは、スクリプトを使用して、いずれかのプロパティの値が -1 から異なる値にいつ変わるかを判断します。これで、正しい値が使用できるようになったことがわかります。

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.Refresh();
```

アラーム情報を取得します。

次のメソッドを使用すると、データベースから特定のアラームに関するレコードを取得できます。

- GetItem() メソッド, GetItem() メソッド
- GetSelectedItem() メソッド

GetItem() メソッド

指定された行とカラムのデータを文字列として返します。

構文

```
オブジェクト。GetItem(Integer、Message)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

整数型
コントロールの特定の行を評価する整数式です。

メッセージ型
コントロールのカラム名を評価する文字列式です。

例

コントロール名は AlmDbView1 です。タグはメッセージ型タグとして定義されます。

```
tag = #AlmDbView1.GetItem(1, "Group");
```

GetSelectedItem() メソッド

選択した行、指定したカラムのデータを文字列で返します。

構文

```
Object.GetSelectedItem(Message)
```

パラメータ

Message
コントロールのカラム名を評価する文字列式

例

コントロール名は AlmDbView1 です。タグはメッセージ型タグとして定義されます。

```
tag = #AlmDbView1.GetSelectedItem("State");
```

アラーム レコードのソート

以下のメソッドを使用すると、アラーム レコードをソートして、カラムのサイズ変更をリセットできます。

- SortOnCol() メソッド
- ShowSort() メソッド, ShowSort() メソッド
- Reset() メソッド

SortOnCol() メソッド

表示されるアラーム レコードで、プライマリ ソートを実行します

構文

```
Object.SortOnCol(Message, Integer)
```

パラメータ

Message
コントロールのカラム名を評価する文字列式

整数型
使用するソート順。0 = 昇順、1 = 降順です。

例

コントロールの名前は AlmDbView1 です。

```
tag = #AlmDbView1.SortOnCol("名前",1);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ShowSort() メソッド

SortMenu プロパティが有効である場合、[**セカンダリ ソート**] ダイアログ ボックスを表示します。

構文

```
オブジェクト。ShowSort()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.ShowSort();
```

Reset() メソッド

すべてのカラムを、デザイン時に保存した設定に戻します。

構文

```
Object.Reset()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.Reset();
```

[コンテキスト] メニューを表示する

ショートカット メニューを表示するには、ShowContext() メソッドを使用します。

ShowContext() メソッド ShowContext()めそっど

RefreshMenu、ResetMenu、SortMenu の 3 つのプロパティのうち 1 つを有効にすると、ショートカット メニューが表示されます。

構文

```
オブジェクト。ShowContext()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.Show Replicas();
```

フィルタ設定にアクセスする

ShowFilter() メソッドを使用すると、[**ファイル設定**] ダイアログ ボックスを表示できます。

ShowFilter() メソッド

[**フィルタ設定**] ダイアログ ボックスを表示します。

構文

```
Object.ShowFilter()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.ShowFilter();
```

その他の情報の表示

AboutBox() メソッドを使用して、[**バージョン情報**] ダイアログ ボックスを表示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

AboutBox() メソッド

［バージョン情報］ダイアログ ボックスを表示します。

構文

```
オブジェクト。AboutBox()
```

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.AboutBox();
```

メソッドとプロパティを使用するときのエラー処理

SilentMode プロパティを使用すると、コントロールをサイレント モードにするかどうかを決定できます。コントロールがサイレント モードである場合、エラー メッセージは表示されません。エラーを確認するには、GetLastError() メソッドを呼び出し、エラー メッセージを返します。

GetLastError() メソッド

Alarm DB View コントロールがサイレント モードである場合、最後のエラー メッセージを返します。

構文

```
Object.GetLastError()
```

例

コントロール名は AlmDbView1 であり、タグ名は変数または文字列として定義されています。

```
Tagname = #AlmDbView1.GetLastError();
```

Alarm DB View ActiveX イベントを使用したスクリプトのトリガ

マウス クリックやダブルクリックなどの Alarm DB View コントロール イベントに、QuickScript を割り当てることができます。イベントが発生すると、QuickScript が実行されます。

Alarm DB View コントロールは、以下のイベントをサポートしています。

- クリック
- ダブルクリック
- 終了
- 起動

Click イベントは ClicknRow というパラメータを 1 つ持ち、これはランタイム時にクリックされた行を識別します。

DoubleClick イベントは DoubleClicknRow というパラメータを 1 つ持ち、これはランタイム時にダブルクリックされた行を識別します。

Click イベントおよび DoubleClick イベントはゼロを基準としています。Click イベントまたは DoubleClick イベント（あるいはその両方）が発行されるとき、表示されるバー カウントは 0 から始まります。

注 : ユーザー インターフェイス メソッドが StartUp イベントから呼び出される場合、コントロールはまだ表示されていないため、Alarm DB View コントロールでは無視されます。これらのメソッドには、ShowSort()、ShowContext()、GetSelectedItem()、GetNext()、GetPrevious()、および AboutBox() があります。

ActiveX イベントのスクリプトの詳細については、「ActiveX コントロールのスクリプト」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

複数のタグにわたるアラーム分散の分析

Alarm Pareto ActiveX コントロールを使用すると、指定の生産システムで最も頻繁に発生するアラームとイベントを分析できます。また、発生の間の期間で、アラームの頻度を分析できます。

Alarm Pareto コントロールには分析機能があるので、生産システムの最も大きな問題を明らかにできます。Alarm Pareto コントロールは、最も重大な改良を行うために、作業の重点をどこに置けばよいのか知るのに役立ちます。

Alarm Pareto コントロールでは、アラーム操作時間を表す棒グラフが表示されます。

ActiveX コントロールの詳細については、「ActiveX コントロール」を参照してください。

Alarm Pareto ActiveX コントロールの設定

Alarm Pareto コントロールを設定するとき、次の指定ができます。

- アラーム データベースへの接続
- パレート コントロールの外観（色とフォントを含む）
- ランタイムにユーザーがアクセスできる機能
- グラフに表示されるアラームおよび表示方法

アラーム データベースとの接続の設定

Alarm Pareto コントロールとアラーム データベースの間の接続を設定する必要があります。

管理者アカウントではなく、アラーム データベースへの適切な読み取り専用アクセスのあるアカウントを使用することが推奨されます。

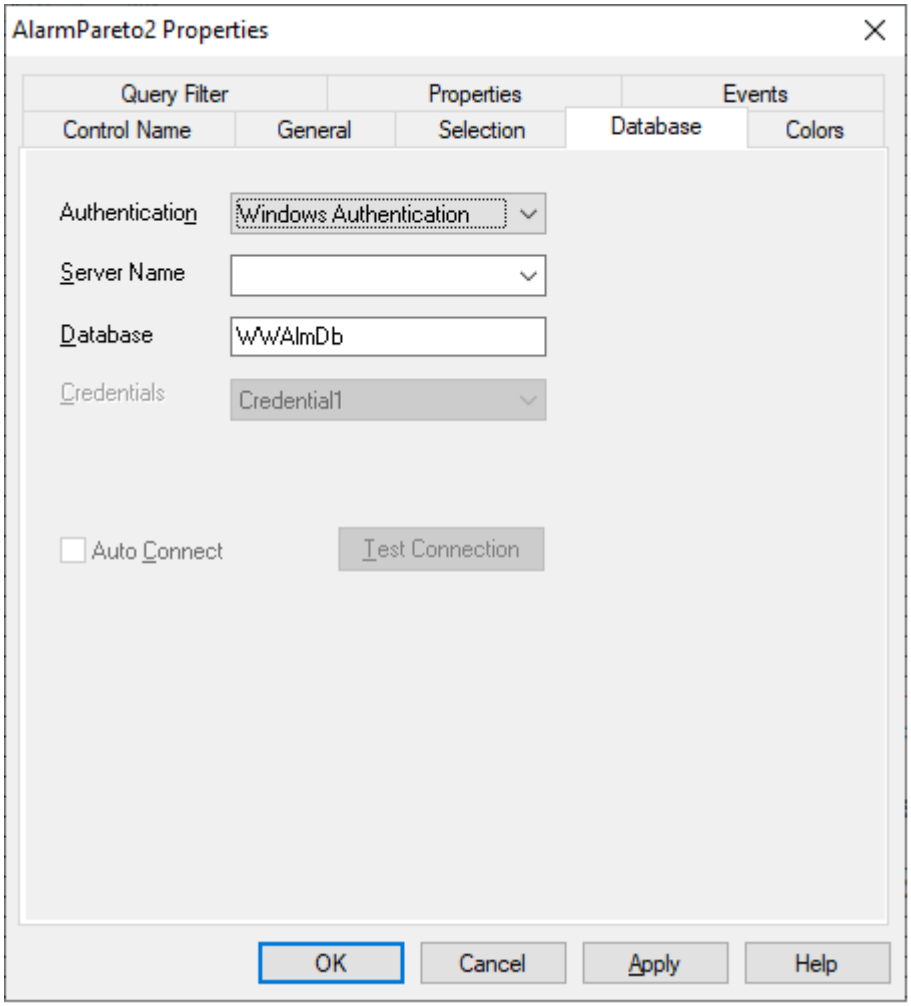
アラーム データベースとの接続の設定

1. Alarm Pareto コントロールを右クリックし、［**プロパティ**］を選択します。

   ［**AlarmPareto のプロパティ**］ダイアログ ボックスが表示されます。

2. [**データベース**] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 接続を設定します。以下の手順を実行します。
   a. [**認証**] リストで、認証方法(**SQL Server 認証** または **Windows 認証** (既定)) を選択します。

   注 : Windows 認証は、SQL 認証よりも優れたアプリケーション セキュリティを提供できます。この理由から、Windows 認証から SQL 認証に切り替えると、Windows 認証を使用することを推奨するポップアップ ダイアログが表示されます。この警告を無視して SQL 認証を使用する場合は、[**OK**] を選択します。同様のメッセージが OCMC Log Viewer に記録されます。

   b. [**サーバー名**] ボックスに、アラーム データベースがインストールされているコンピュータのノード名を入力します。
   c. [**データベース名**] ボックスにアラーム データベースの名前を入力します。
   d. [**資格情報**] ドロップダウンから認証に使用する資格情報を選択します。

注:
- [**資格情報**] ドロップダウンは、SQL Server 認証タイプが選択されている場合にのみ有効になります。Windows 認証では、現在ログインしているユーザーの資格情報が使用され、ユーザー名とパスワード フィールドが無効になります。
スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マネージャから取得されます。詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「資格情報マネージャの操作」を参照してください。

4. WindowViewer が起動したときに、Alarm Pareto コントロールをアラーム データベースに自動的に接続する場合は、［**自動接続**］チェック ボックスをオンにします。

   ［**自動接続**］チェック ボックスをオンにしない場合は、Connect() メソッドを明示的に呼び出して、Alarm Pareto コントロールがアラーム データベースに接続されるよう設定する必要があります。Connect メソッドの詳細については、「Connect() メソッド」を参照してください。

5. ［**接続テスト**］を選択して、アラーム データベースへの接続性を確認します。接続が正常に確立されたことを示すメッセージが表示されます。

6. **[適用]** を選択します。

Alarm Pareto コントロールの外観と色を設定する

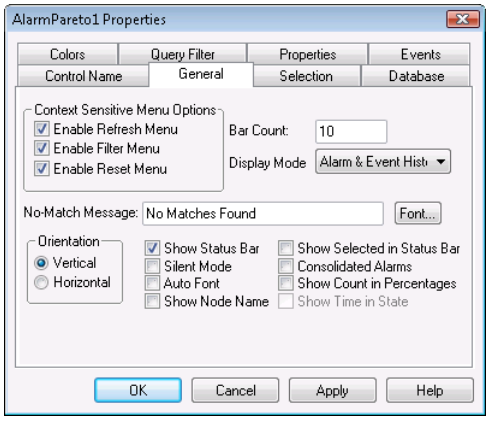
Alarm Pareto コントロールの外観を設定できます。以下の操作を実行できます。

- ステータス バーを含めます。
- パレート グラフの棒の方向を設定します。
- グラフの棒の説明を含めます。
- Alarm Pareto グラフの色を選択します。

Alarm Pareto コントロールの外観を設定する

1. Alarm Pareto コントロールを右クリックし、［**プロパティ**］を選択します。［**AlarmPareto のプロパティ**］ダイアログ ボックスが表示されます。

2. [**全般**] タブを選択します。



3. 一般的なオプションを設定します。以下の手順を実行します。

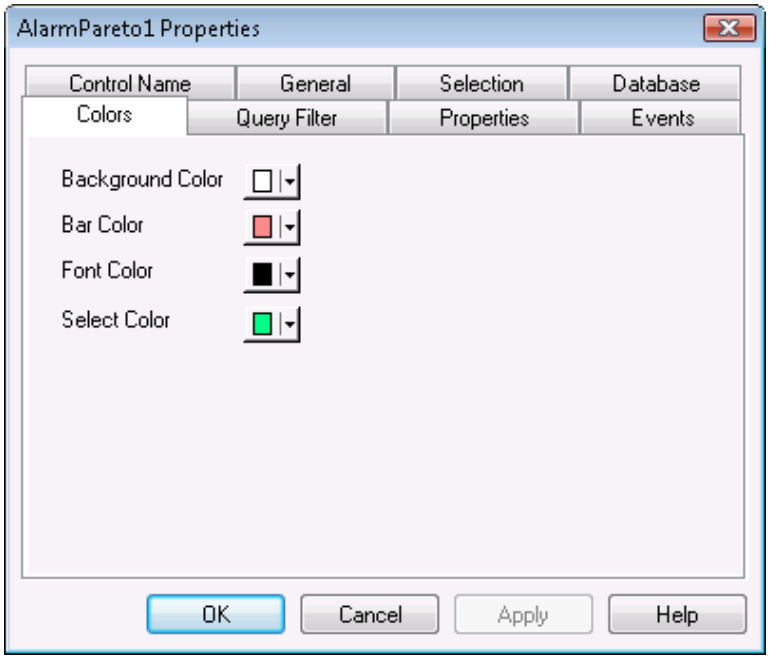
プロパティ	説明
バー カウント	Alarm Pareto コントロールで表示されるバーの数を設定します。
表示モード	このリストには、利用できる表示オプションが示されます。オプションは、［アラーム／イベント履歴］、［アラーム履歴］、および［イベント履歴］です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ	説明
一致しないメッセージ	Alarm Pareto コントロールで処理されるデータがないときを示すメッセージを設定します。
垂直方向	垂直スケールでバーを表示します。
水平方向	水平スケールでバーを表示します。
ステータス バーの表示	ステータス バーを有効化します。
サイレント モード	Alarm Pareto コントロールでは、ランタイム エラー メッセージは表示されません。チェックボックスをオフにすると、アラーム表示でエラー メッセージが表示されます。エラー メッセージは、常にロガーに送信されます。
自動フォント	利用できるスペースでは選択したバーのテキストを正しく表示できない場合、［自動フォント］によって、テキストが非表示になり、バーを選択したときのみテキストが表示されます。
ノード名を表示	棒グラフ上のノード名を表示します。
選択をステータス バーで表示	ステータス バーに選択したバーの説明を示します。
統合アラーム	アラーム状況を 2 つの状況に統合します。例えば、アナログタグが Hi、HiHi、Normal の 3 つの状態のアラーム を持つ場合、Hi と HiHi のアラーム状態は 1 つの状態として分類されます。
パーセント値でカウントを表示	合計カウントに対するカウントのパーセント値でバーを表示します。
状態を時間で表示	各タグ変数がアラーム状況にある時間に基づいて Alarm Pareto コントロールを表示します。 このオプションが有効なのは、表示モードが「アラーム履歴」に設定されている場合のみです。

4. **[適用]** を選択します。
5. **[イベント色]**タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. それぞれの色ボックスを選択すると、カラー パレットが開きます。
7. 以下の各グラフ プロパティに割り当てる色を選択します。

プロパティ	説明
背景色	Alarm Pareto グラフの背景色を設定します。
バーの色	グラフのバーの色を設定します。
フォントの色	グラフに表示されるテキストの色を設定します。
選択の色	選択したバーの色を設定します。

8. **[適用]** を選択します。

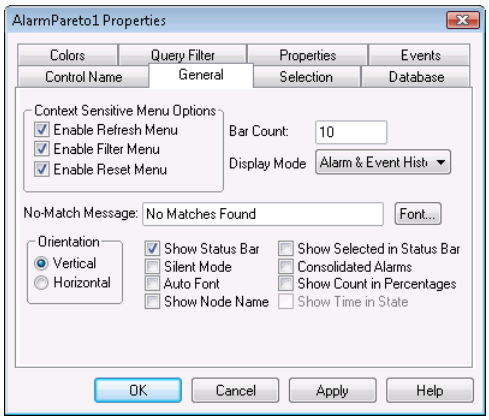
フォントの表示の設定

Alarm Pareto グラフに表示されるテキストにフォント プロパティを割り当てできます。

フォントのプロパティを設定する

1. Alarm Pareto コントロールを右クリックし、［**プロパティ**］を選択します。［**AlarmPareto のプロパティ**］ダイアログ ボックスが表示されます。
2. [**全般**] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



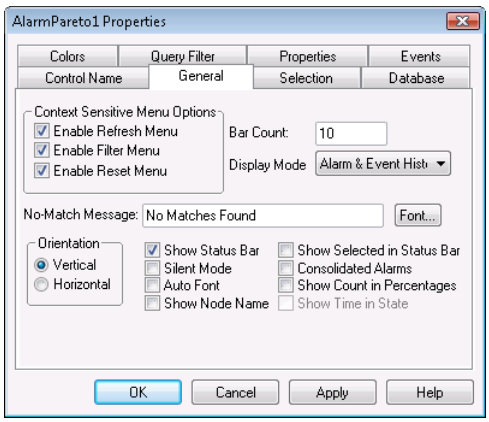
3. フォントを選択します。Windows 標準の［フォント］ダイアログ ボックスが表示されます。フォントを設定して、［**OK**］を選択します。
4. ［**OK**］を選択します。

ランタイムにユーザーがアクセスできる機能の設定

Alarm Pareto グラフにはショートカット メニューが含まれています。ランタイムの間にユーザーがトレンドを右クリックすると、メニューが表示され、そのオプションで、トレンドに表示されたデータを動的に更新できます。

ランタイム機能の設定

1. Alarm Pareto コントロールを右クリックし、［プロパティ］を選択します。［**AlarmPareto** のプロパティ］ダイアログ ボックスが表示されます。
2. ［全般］タブを選択します。



3. ［コンテキスト メニューのオプション］領域で、メニュー コマンドを設定します。
   a. ［リフレッシュ メニューを有効化］チェック ボックスをオンにすると、Alarm Pareto トレンドに表示されたデータをランタイム ユーザーがリフレッシュして、範囲 1 から MaxRecords プロパティで定義された値までのレコードを表示できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

b. ［フィルタ メニューを有効化］チェック ボックスをオンにすると、ユーザーが［フィルタ設定］ダイアログ ボックスを表示して、Alarm Pareto トレンドのデータベース クエリー値を含むファイルを選択できます。

c. ［リセット メニューを有効化］チェック ボックスをオンにすると、ユーザーが、ランタイムの Alarm Pareto グラフを WindowMaker で指定された元の値に戻せます。ユーザーが行ったランタイム変更はすべて元のデザイン時の値に戻ります。

4. ［適用］を選択します。

分析するアラームの設定

Alarm Pareto グラフを使用して、分析するアラームを設定します。以下を指定します。

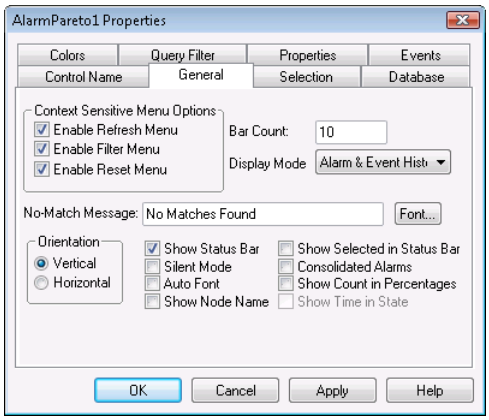
- データベース データのタイプ（アラーム データまたはイベント データ）
- レコードを選択する期間
- データをフィルタする条件

アラーム データまたはイベント データの選択

アラーム　レコード、イベント レコード、またはその両方を Alarm Pareto グラフに表示するかどうかを設定できます。

データ タイプを選択する

1. **Alarm Pareto** コントロールを右クリックし、［プロパティ］を選択します。［AlarmPareto のプロパティ］ダイアログ ボックスが表示されます。
2. ［全般］タブを選択します。



3. ［表示モード］リストで、レコードのタイプを設定します。以下のいずれかを実行します。
   - ［アラーム／イベント履歴］を選択し、アラームおよびイベントの履歴データベース レコードを表示します。
   - ［アラーム履歴］を選択し、履歴アラーム レコードだけを表示します。
   - ［イベント履歴］を選択し、履歴イベント レコードだけを表示します。
4. ［適用］を選択します。

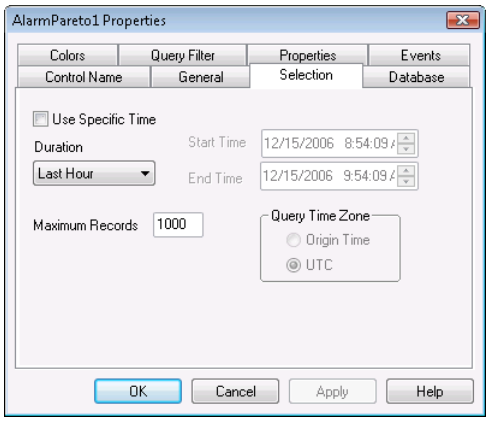
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

期間の選択

選択した時間に基づいてレコードを選択するようにクエリー値を設定できます。また、表示するレコードの最大数、アラーム クエリーの開始時間と終了時間、クエリー タイム ゾーンを設定することもできます。

データの期間の選択

1. **Alarm Pareto** コントロールを右クリックし、［**プロパティ**］を選択します。［**AlarmPareto のプロパティ**］ダイアログ ボックスが表示されます。
2. ［**選択**］タブを選択します。



3. UTC を使用してデータを問い合わせる定義済み間隔を使用するには、［**期間の設定**］リストで間隔を選択します。
4. 特定の開始時間と終了時間を使用するには、［**指定時間を使用**］を選択して詳細を設定します。
   a. ［**開始時間**］ボックスに、アラーム レコードを取得する開始時間を入力します。文字列は YYYY/MM/DD HH:MM:SS 形式にします。1970 年 1 月 1 日 0 時 00 分から 2038 年 1 月 18 日 19 時 14 分 07 秒の間なら、どのタイム ゾーンのどの日付でも使用できます。
   b. ［**終了時間**］ボックスに、アラーム レコードの取得を停止する終了時間を入力します。文字列は YYYY/MM/DD HH:MM:SS 形式にします。1970 年 1 月 1 日 0 時 00 分から 2038 年 1 月 18 日 19 時 14 分 07 秒の間なら、どのタイム ゾーンのどの日付でも使用できます。
   c. ［**クエリー タイム ゾーン**］領域で、［**UTC**］または［**標準時間**］のどちらかを選択します。UTC 時間はグリニッジ標準時（Greenwich Mean Time）です。国際標準時または Zulu とも呼ばれます。標準時間は、オペレータのタイム ゾーンの現在の時間です。
5. ［**最大レコード数**］ボックスに、1 つのインスタンスで、コントロールから表示するレコード数を入力します。最大レコードの有効な範囲は、0 ～ 1000000 です。
6. ［**適用**］を選択します。

フィルタ設定を使用したカスタム フィルタの作成

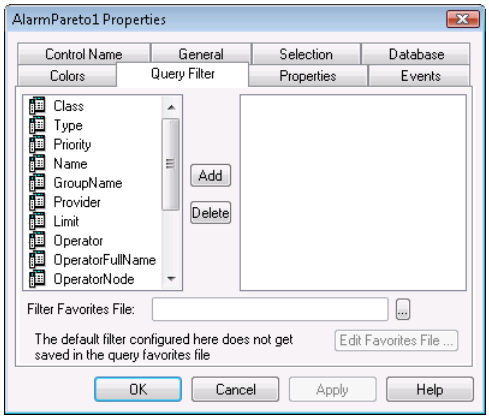
クエリー結果に含めるレコードを選択できます。たとえば、レコードの日付またはアラームの状態によってフィルタを選択できます。クエリー結果を制限または拡大するために複数のフィールドを選択できます。

カスタム フィルタを定義しないと、すべてのレコードを問い合わせるデフォルトのフィルタが使用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カスタム フィルタを作成する

1. **Alarm Pareto** コントロールを右クリックし、［**プロパティ**］を選択します。［**AlarmPareto** のプロパティ］ダイアログ ボックスが表示されます。
2. ［**クエリー フィルタ**］タブを選択します。



3. 左ペインでフィルタ フィールドを選択し、［**追加**］を選択して、これらフィールドをフィルタに含めます。フィルタは右ペインに表示されます。フィルタ フィールドの詳細については、次の表を参照してください。

フィールド名	フィルタ クエリー
クラス	アラーム クラス
Type	アラーム タイプ
優先度	アラーム優先度
名前	アラーム名
GroupName	アラーム グループ名
プロバイダ	アラーム プロバイダ
しきい値	アラームしきい値。値は英数字です。 クエリー フィルタの値の比較は、文字列の比較として行われます。
Operator	オペレータ
オペレータ フル ネーム	オペレータ フル ネーム
オペレータ ノード	アラームと関連のあるオペレータ ノード名
オペレータ ドメイン	アラームと関連のあるオペレータ ドメイン名
コメント	アラーム コメント

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

フィールド名	フィルタ クエリー
ユーザー 1	アラーム ユーザー定義数値 1
ユーザー 2	アラーム ユーザー定義数値 2
ユーザー 3	アラーム ユーザー定義文字列値
期間	未確認とアラーム期間ゼロに等しい値で設定された間隔カラムでは、クエリーで null 値を持つレコードは生成されません。

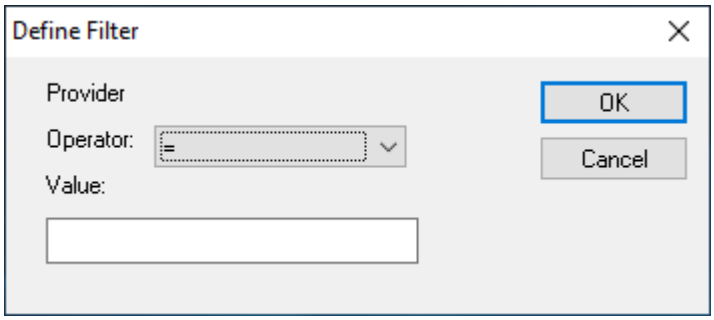
4. フィルタ ペインからフィールドを削除するには、削除するフィールドを強調表示し、［**削除**］を選択します。フィルタの削除は元に戻せません。メッセージが表示されたら、［**はい**］を選択します。
5. 各フィルタ フィールドの条件を設定します。詳細については、「カラム フィルタ条件の定義, カラム フィルタ条件の定義」を参照してください。
6. フィルタのオペレータとグループ化を設定します。詳細については、「アラーム カラムをグループ化する, アラーム カラムをグループ化する」を参照してください。
7. フィルタ設定ファイルを設定します。
   a. ［フィルタ設定ファイル］ボックスに、ネットワーク パスとファイル名を入力するか、省略記号ボタンを選択してファイルを参照します。
   b. **フィルタ設定**ファイルを編集する、［**設定ファイルの編集**］ボタンを選択します。［**フィルタ設定**］ウィンドウが開き、設定ファイルのフィルタを追加、変更、または削除できます。完了したら、［**OK**］を選択して変更を保存し、ウィンドウを閉じます。
8. **[適用]** を選択します。

カラム フィルタ条件の定義

クエリーに含める各カラム フィルタについて、フィルタ条件を設定する必要があります。たとえば、特定のオペレータについてのアラームだけを表示することもできます。

カラム フィルタを定義する

1. フィールドを右クリックし、［**フィルタの編集**］を選択します。［**フィルタの定義**］ダイアログ ボックスが表示されます。



2. ［**演算子**］リストで、必要な演算子を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［値］ボックスに、一致する必要のある条件を入力します。［値］ボックスでは、選択したクエリーについて処理できない値は受け入れられません。「**Like**」および「**Not Like**」のフィルタ演算子が英数字のカラム名に使われている場合、［値］ボックスでは次のワイルドカード文字が使用できます。

文字	検索
%	ゼロ文字以上の任意の文字列
_	任意の 1 文字
[ ]	指定範囲（[a-f] など)、またはセット([abcdef] など）内の任意の 1 文字
[^]	指定範囲にない（[^a-f] など)、またはセット内にない([^abcdef] など）任意の 1 文字

次の［値］ボックスしきい値がさまざまなフィールドに適用されます。

フィールド	しきい値
すべてのフィールド	ユーザー 1、ユーザー 2、および優先度を除く、すべての英数字
優先度	1 ～ 999 の整数値が使用できます。
ユーザー 1、ユーザー 2	負の数、正の数、または小数のみを使用できます。

4. **[OK]** を選択します。

アラーム カラムをグループ化する

複数のフィールドが定義されると、カラムはブール演算子を使用して結合されます。

- AND 演算子は、選択したフィールドのすべての値に一致するレコードを返します。
- OR 演算子は、選択した任意のフィールドの値に一致するレコードを返します。

AND/OR 演算子を使用してフィルタ選択条件を使用するには、対応するフィールドをグループ化する必要があります。フィルタ ペインのアイテムに対しては 1 つのフィルタ式のみを作成できます。複数の式が必要な場合、そのアイテムをフィルタ ペインに再び追加する必要があります。

デフォルトで、グループ化されたフィールドには AND 演算子があります。

AND 演算子と OR 演算子は親ノードです。各親ノードの下にある選択されたフィールドは子ノードです。フィールド親ノードを子ノードにドラッグすることはできません。

アラーム カラムをグループ化する

1. フィールドを右クリックし、［グループ］を選択します。
2. フィールドを別のフィールドにドラッグします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クエリー フィルタのコピーまたは移動

Alarm Pareto コントロールの複数のインスタンスがあり、複数のインスタンスについて同じフィルタを使用する場合は、定義済みフィルタをあるインスタンスからコピーまたは切り取りして、別のフィルタに貼り付けます。

フィルタをコピーする

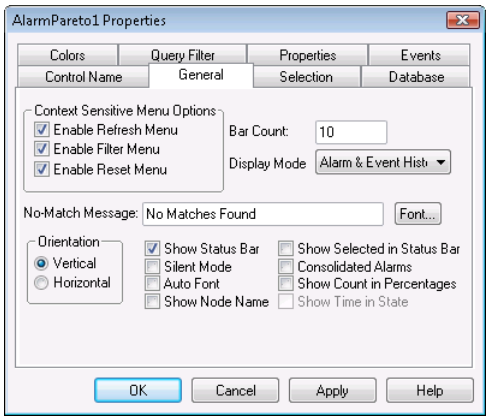
1. Alarm Pareto コントロールの最初のインスタンスで、フィルタを定義します。
2. フィルタを右クリックし、［**コピー**］を選択します。フィルタを移動する、［**切り取り**］を選択します。
3. Alarm Pareto コントロールの最初のインスタンスを終了します。
4. Alarm Pareto コントロールの次のインスタンスを開き、［**クエリー フィルタ**］タブを選択します。
5. 右ペインに矢印を位置付けします。選択したフィルタを右クリックし、［**貼り付け**］を選択します。

分析結果の表示の設定

アラームの表示をクエリー結果に設定できます。

表示を設定する

1. **Alarm Pareto** コントロールを右クリックし、［**プロパティ**］を選択します。［AlarmPareto のプロパティ］ダイアログ ボックスが表示されます。
2. ［**全般**］タブを選択します。



3. ［**統合アラーム**］チェック ボックスをオンにし、マルチ状況アラームから 1 つのタイム スタンプ レコードへとアラームを結合します。
4. ［**パーセント値でカウントを表示**］チェック ボックスをオンにして、グラフに表示された各バーの合計のパーセント値を追加します。
5. ［**適用**］を選択します。

ランタイムでの Alarm Pareto コントロールの使用

ランタイムに Alarm Pareto コントロールを右クリックし、ショートカット メニューを表示します。次の表に、ショートカット メニューに表示される、可能性のあるすべてのオプションをリストします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メニュー オプション	説明
最新の情報への更新	表示をリフレッシュします。
フィルタ	フィルタを編集して、Pareto コントロールが受け取ったデータを変更できます。このメニュー項目は、フィルタ設定ファイルが設定されている場合のみ利用できます。
リセット	グラフをデフォルトのクエリーにリセットします。

ステータス バーに表示される情報について

Alarm Pareto コントロールのステータス バーに表示されるものは次のとおりです。

- コントロールとアラーム データベースの間のデータベース接続のステータス
- グラフに表示されたデータをリフレッシュする、グラフの更新ステータス

Alarm Pareto ActiveX プロパティの使用

スクリプトを使用して Alarm Pareto コントロールのプロパティの値を直接設定するか、InTouch タグまたは I/O 参照に割り当てることができます。プロパティの設定の詳細については、「ActiveX コントロールのスクリプト」を参照してください。

次の表に Alarm Pareto プロパティを示します一覧表示しています。

プロパティ名	説明
認証	［データベース］タブで選択された認証タイプを返します。
AuthenticationMode	認証タイプを決定する値を設定します（SQL 認証の場合は 0、Windows 認証の場合は 1）。
AutoConnect	コントロールが実行モードになるとすぐにコントロールがデータベースに接続されるかどうかを決定する値を返すか、設定します。
AutoFont	利用できるスペースでは選択したバーのテキストを正しく表示できない場合、［自動フォント］によってテキストが非表示になり、バーを選択したときのみ選択したバーのテキストが表示されます。
BackGndColor	Alarm Pareto グラフの背景色を設定します。
BarColor	Alarm Pareto コントロールのバーの色を設定します。
BarCount	Alarm Pareto コントロールで表示されるバーの数を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ名	説明
BarSelectColor	Alarm Pareto コントロールで選択したバーの色を設定します。
接続済み	Alarm Pareto コントロールがデータベースに接続されているかどうか決定します。
ConsolidatedAlarms	アラーム状態を 2 つの状態に統合します。たとえば、アナログ タグ変数に 3 つの状況アラームがある場合、Hi、HiHi、および Normal、また Hi および HiHi アラーム状態は 1 つの状態と分類されます。
DatabaseName	Alarm Pareto コントロールが接続する必要のあるデータベースの名前を設定します。
DisplayMode	表示モードを設定します。表示モード オプションは、［アラーム/イベント履歴］、［アラーム履歴］、または［イベント履歴］です。
間隔	コントロールが「開始時間」と「終了時間」を設定する間隔を返すか、 または設定します。
EnableRefresh	Alarm Pareto コントロールをリフレッシュするコンテキスト メニューを有効または無効にします。
EnableReset	Alarm Pareto コントロールをリセットするコンテキスト メニューを有効または無効にします。
EnableSilentMode	サイレント モードを有効または無効にします。 サイレント モードが無効な場合、Alarm Pareto コントロールにはエラー メッセージ ボックスが表示されます。サイレント モードが有効な場合、エラー メッセージ ボックスは表示されません。 エラー情報がロガーに書き込まれます。
EndTime	終了日付および時間を返すか、または設定します。
FilterMenu	フィルタを編集して Pareto コントロールが受け取ったデータを変更できる、コンテキスト メニューを有効または無効にします。このプロパティが有効なのは、フィルタ メニュー ファイルが設定されている場合のみです。
FilterFavoritesFile	フィルタ設定ファイルを文字列として指定します。
フォント	コントロール内のレコードと見出しのフォントを設定します。
FontColor	Alarm Pareto コントロールでレコード表示のフォントの色を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ名	説明
HorizontalChart	水平バーとしてグラフを表示します。 HorizontalChart が無効な場合、グラフには垂直バーが表示されます。
MaxRecords	指定された時間で取得できる最大レコード数を指定する値を返すか、または設定します。
NoMatchMessage	Alarm Pareto コントロールで処理されるデータがないときに表示されるメッセージを設定します。
QueryTimeZone	タイム ゾーンを［UTC］または［標準時間］のどちらかに設定します。
ServerName	現在のサーバー名を返します。
ShowCountPercentage	選択した場合、合計カウントのパーセント値として各バーのカウントを表示します。選択しない場合、各バーの実際のカウントを表示します。
ShowNodeName	Alarm Pareto コントロールが、バーにノード名その他の情報を表示するよう設定します。
ShowSelectedInStatusBar	ステータス バーで選択したバーの情報の表示を有効または無効にします。
ShowStatusBar	ステータス バーを表示するかどうか決定する値を返すか、または設定します。
ShowTimeinState	タグがアラーム状態にとどまった時間を基礎にして、Alarm Pareto コントロールがバーを表示するかどうかを決定する値を返すか、設定します。 無効な場合、アラームが発生した回数を基礎にして、コントロールはバーを表示します。
SpecificTime	コントロールが「開始時間」プロパティと「終了時間」プロパティを使用するかどうか、または「期間の設定」プロパティを基に開始時間と終了時間を計算するかどうかを決定する値を返すか、または設定します。
StartTime	開始日付および時間を返すか、または設定します。
ﾕｰｻﾞｰ	SQL Server に接続するコントロールとして使用されるユーザーを返すか、設定します。

Alarm Pareto ActiveX メソッドの使用

Alarm Pareto ActiveX メソッドを使用して次のことができます。

- データベース接続を制御します。
- データベースからレコードを取得します。
- 特定のパレート バーについての情報を取得します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メソッドの呼び出しの詳細については、「ActiveX コントロールのスクリプト」を参照してください。

データベース接続の制御

アラーム データベースに接続するには、Connect() メソッドを使用します。

Connect() メソッド

Alarm Pareto コントロール プロパティの［**データベース**］タブで設定されたデータベースに接続します。

構文

```
Object.Connect()
```

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.Connect();
```

データベースからレコードを取得します。

次の関数を使用して、データベースからレコードを取得します。

- Refresh() メソッド
- SelectQuery() メソッド

Refresh() メソッド

データベースのコントロールをリフレッシュします。接続が正常に行われた場合は、1 から MaxRecords プロパティで定義される値の範囲にあるレコード セットが表示されます。

構文

```
Object.Refresh()
```

例

```
#AlarmPareto1.Refresh();
```

SelectQuery() メソッド

クエリー設定ファイルとして設定されるフィルタを選択します。

構文

```
Object.SelectQuery(Filter)
```

パラメータ

フィルタ
クエリー フィルタの名前

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.SelectQuery("MyFilter");
```

特定のパレート バーの情報を取得

次の関数を使用して、特定のパレート バーの情報を取得します。

- GetItemAlarmName() メソッド
- GetItemAlarmType() メソッド
- GetItemCount() メソッド

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- GetItemTotalTime() メソッド
- GetItemEventType() メソッド
- GetItemProviderName() メソッド

GetItemAlarmName() メソッド

特定のバーについてアラーム名を取得します。

構文

```
Object.GetItemAlarmName(BarIndex)
```

パラメータ

BarIndex
バーのインデックス

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.GetItemAlarmName(1);
```

GetItemAlarmType() メソッド

特定のバーについてアラームのタイプを取得します。

構文

```
Object.GetItemAlarmType(BarIndex)
```

パラメータ

BarIndex
バーのインデックス

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.GetItemAlarmType(1);
```

GetItemCount() メソッド

バーのアラーム数を取得します。

構文

```
Object.GetItemCount(BarIndex)
```

パラメータ

BarIndex
バーのインデックス

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.GetItemCount(1);
```

GetItemTotalTime() メソッド

アラーム状況のタグ変数の合計時間（秒単位）を取得します。

構文

```
Object.GetItemTotalTime(BarIndex)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

BarIndex
バーのインデックス

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.GetItemTotalTime(1);
```

GetItemEventType() メソッド

特定のバーについてイベントのタイプを取得します。

構文

```
Object.GetItemEventType(BarIndex)
```

パラメータ

BarIndex
バーのインデックス

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.GetItemEventType(1);
```

GetItemProviderName() メソッド

特定のバーについて生成されたアラームからプロバイダ名を取得します。

構文

```
Object.GetItemProviderName(BarIndex)
```

パラメータ

BarIndex
バーのインデックス

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.GetItemProviderName(1);
```

その他の情報の表示

AboutBox() メソッドを使用して、[**バージョン情報**] ダイアログ ボックスを表示します。

AboutBox() メソッド

[**バージョン情報**] ダイアログ ボックスを表示します。

構文

```
オブジェクト。AboutBox()
```

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.AboutBox();
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メソッドとプロパティを使用する際のエラー処理

Alarm Pareto コントロールでは、［**サイレント モード**］オプションに基づいてランタイム エラー メッセージが処理されます。詳細については、「Alarm Pareto コントロールの外観と色を設定する」を参照してください。

［**サイレント モード**］を選択すると、Alarm Pareto コントロールで、ランタイム エラー メッセージが表示されません。チェックボックスをオフにすると、アラーム表示でエラー メッセージが表示されます。すべての Alarm Pareto エラー メッセージは、Logger に送信されます。

Alarm Pareto ActiveX イベントを使用したスクリプトのトリガ

マウス クリックやダブルクリックなどの Alarm Pareto コントロール イベントに、QuickScript を割り当てることができます。イベントが発生すると、QuickScript が実行されます。

Alarm Pareto コントロールは、以下のイベントをサポートしています。

- クリック
- ダブルクリック
- 停止
- 起動

Click イベントは ClicknBarIndex というパラメータを 1 つ持ち、これはランタイム時にクリックされたバーのインデックスを識別します。

DoubleClick イベントは DoubleClicknBarIndex というパラメータを 1 つ持ち、これはランタイム時にダブルクリックされたバーのインデックスを識別します。

Click イベントおよび DoubleClick イベントはゼロを基準としています。Click イベントまたは DoubleClick イベント（あるいはその両方）が発行されるとき、表示されるバー カウントは 0 から始まります。

注意：ユーザー インターフェイス メソッドが StartUp イベントから呼び出される場合、コントロールはまだ表示されていないため、Alarm Pareto コントロールでは無視されます。これらのメソッドには、AboutBox() と Refresh() が含まれます。

ActiveX イベントのスクリプトの詳細については、「ActiveX コントロールのスクリプト」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 11 スクリプト

InTouch スクリプト言語（QuickScript）を使用すると、より堅固なアプリケーションを構築できます。スクリプトには 8 種類あり、多くの組み込みスクリプト関数を使用できます。

スクリプトは、いつ実行されるか、および継続的に実行されている他のアプリケーション プロセスから独立して実行されているかどうか、などにより分類できます。スクリプトは通常、次の 2 つの異なる方法で実行されます。

- イベントが発生したときに、イベント ベースのスクリプトが 1 回実行されます。たとえば、オペレータがキーを押すか、またはタグ値が変更された後、イベント ベースのスクリプトを実行できます。
- 条件が満たされたときに、時間ベースのスクリプトが定期的に実行されます。たとえば、ウィンドウが開いているか、またはボタンを押し続けているときに、時間ベースのスクリプトを実行できます。

同じトリガで実行するように、複数のイベント ベースのスクリプトおよび時間ベースのスクリプトを設定できます。たとえば、キーが押されたときにあるスクリプトを実行し、同じキーを押し続けている間に他のスクリプトを 5 秒ごとに定期的に実行するように設定できます。

スクリプト言語で条件付きステートメント、ループ、およびローカル変数を使用すると、アプリケーションで複雑な効果を作成できます。

条件スクリプトでは、同期的または非同期的にスクリプトを実行できます。

- 同期スクリプトが実行されている場合、すべての InTouch アニメーションおよびタグ値の更新が停止します。スクリプトが停止すると、アニメーションおよびタグ値の更新が再開します。
- 非同期スクリプトが実行されると、スクリプトが実行されている期間、すべての InTouch アニメーションおよびタグ値の更新が続行されます。

組み込みスクリプト関数には、数学関数、三角関数、文字列関数、その他などがあります。これらの関数を使用すると、アプリケーション開発の時間を節約できます。

InTouch スクリプトには、OLE（Object Linking and Embedding）オブジェクトおよび ActiveX コントロールを含めることができます。

基本的なスクリプトの概念

スクリプトを開始する前に、以下の内容を理解する必要があります。

- スクリプトは、アプリケーションに何かを実行するように指示する命令のセットです。
- QuickScript は、InTouch HMI スクリプト言語です。
- 関数は、他のスクリプトから呼び出すことのできるスクリプトです。InTouch HMI には、すぐに使用できる事前定義済みの関数のセットが含まれています。
- クイック関数は QuickScript で記述され、クイック関数ライブラリに保存される再使用可能な関数です。クイック関数を作成するには、簡単に QuickScript を作成して、名前を付けます。クイック関数は、別のスクリプトにより呼び出すか、アニメーション リンク式から呼び出すことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スクリプトのタイプ

InTouch では、スクリプトは、その実行条件に基づいて分類されます。たとえば、オペレータがキーボードの特定のキーを押したときにスクリプトを実行する場合は、「キー スクリプト」を作成します。

スクリプト タイプを選択した後、スクリプトを実行させる条件または状況を詳細に定義できます。たとえば、キーが押されたときでなく、キーが離されたときにスクリプトを実行することもできます。

スクリプトのタイプを以下に示します。

- **アプリケーション** スクリプトは、WindowViewer の実行中に継続して実行するか、WindowViewer が起動または終了したときに一度実行します。
- **ウィンドウ** スクリプトは、InTouch ウィンドウが開いているときに定期的に実行するか、InTouch ウィンドウが開くとき、または閉じるときに一度実行します。
- **キー** スクリプトは、特定のキーやキーの組み合わせが押されたとき、または離されたときに、一度または定期的に実行します。
- **条件**スクリプトは、特定の条件が満たされた場合、または満たされなかった場合に、一度または定期的に実行します。
- **データ変化**スクリプトは、特定のタグまたは式の値が変化したときに一度実行します。
- **アクション** スクリプトは、オペレータが InTouch HMI グラフィック オブジェクトをクリックしたときに、一度または定期的に実行します。アクション スクリプトは、プッシュボタンに使用されることがあります。
- **ActiveX イベント** スクリプトは、ActiveX コントロールの選択など、ActiveX イベントが発生したときに一度実行します。

高度なスクリプトの概念

高度なスクリプト機能を使用すると、InTouch HMI の基本的なスクリプト機能よりも洗練された関数を作成できます。

OLE オブジェクトや ActiveX コントロールを使用すると、ネイティブ コンピュータ システム関数にアクセスして、Manufacturing Engineering Module などの他のプログラムと対話できます。

OLE オブジェクト OLE おぶじぇくと

カスタム スクリプトでは、OLE オブジェクトを呼び出すことができます。OLE オブジェクトを使用すると、ネイティブ コンピュータ システム関数にアクセスして、Manufacturing Engineering Module などの他のプログラムと対話できます。

たとえば、OLE を使用すると、以下の操作を行えます。

- ランダム数を生成する
- ユーザー インターフェイス ダイアログ ボックスを作成する
- Windows の日時プロパティ パネルを開く
- レジストリへ読み込むまたは書き込む
- ウィンドウを最小化する

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ActiveX コントロールを使用したスクリプト

InTouch HMI の［ウィザード］メニューには、いくつかの ActiveX コントロールが提供されています。InTouch HMI は Windows オペレーティング環境に基づいているため、InTouch HMI ではほとんどすべての ActiveX コントロールを使用できます。

スクリプトの作成と編集

新しいスクリプトを作成する手順は、スクリプトのタイプに応じて異なります。通常は、スクリプト エディタを開いて条件タイプを選択し、ステートメントを入力して、スクリプトを保存します。

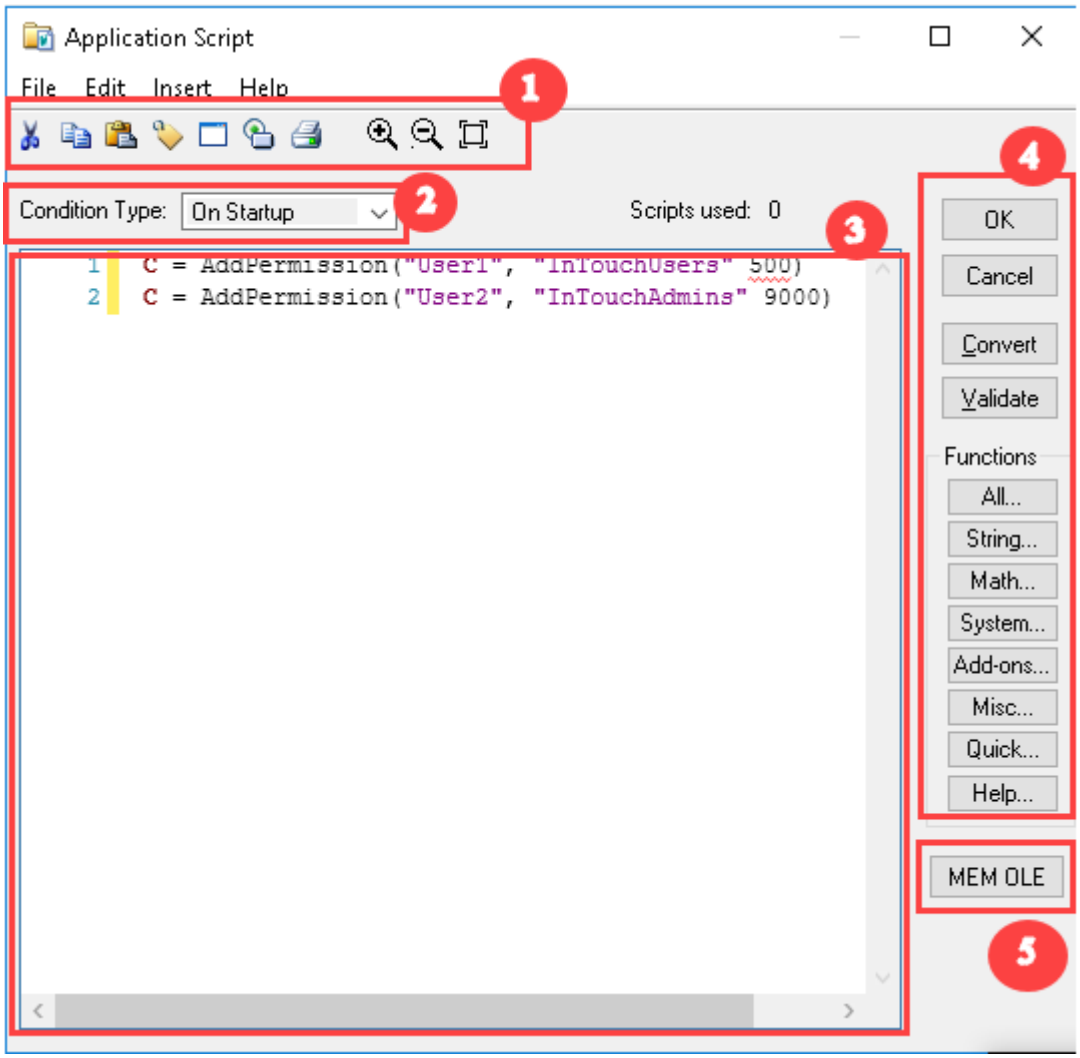
各タイプのスクリプトの作成の詳細については、以下のセクションを参照してください。

- アプリケーション スクリプトの設定
- ウィンドウ スクリプトの設定
- キー スクリプトの設定
- 条件スクリプトの設定
- データ変化スクリプトの設定
- アクション スクリプトの設定
- ActiveX イベント スクリプトの設定

InTouch スクリプト エディタの操作

InTouch WindowMaker 内でスクリプトを作成および編集するには、InTouch HMI スクリプト エディタを使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



領域	説明
1	ツールバー
2	条件定義エリア ［**実行条件**］ボックスは、記述しているスクリプトのタイプで使用できる実行条件を提供します。
3	スクリプト テキスト ウィンドウ
4	コマンド ボタン
5	組み込みスクリプト関数ボタン
6	MEM OLE ボタン 右下隅の［**MEM OLE**］ボタンは、MEM（Manufacturing Engineering Module）が InTouch HMI と共にインストールされている場合にのみ表示されます。このボタンを選択すると、MEM を使用してスクリプトを作成できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

これはアプリケーション スクリプトの例です。各タイプのスクリプトには固有のバージョンのスクリプト ダイアログ ボックスがあり、そのスクリプトのタイプに固有のオプションおよび選択肢があります。

エディタのタイトル バーは、使用しているスクリプトのタイプを示します。スクリプトのタイプの詳細については、「スクリプトのタイプ」を参照してください。

コンテキスト依存のポップアップ ウィンドウにはテキスト、等価、および数学演算のボタンがあります。これらのボタンを選択すると、キーワード、関数、またはシンボルをスクリプトのカーソル位置に挿入できます。

InTouch スクリプトの色インジケータ

InTouch スクリプト エディタは、さまざまなテキストの色を使用してさまざまなスクリプト要素を示します。次の表は、スクリプト要素に関連付けられたテキストの色を示します。

要素	色
キーワード	青 入力中にハイライトされる構文
コメント（単一行と複数行）	緑 入力中にハイライトされる構文
文字列	紫 入力中にハイライトされる構文
関数名、定数、演算子、セミコロン、dim 変数、エイリアス変数など	黒 属性名と予約語の説明を参照してください。
属性、InTouch タグ、参照文字列	えび茶、太字
予約語	赤、非太字
.NET タイプ名	青緑、非太字

InTouch スクリプト エディタのオートコンプリート機能

これらの機能を使用すると、タグ、ドットフィールド、メソッド、プログラミングが可能なコンストラクト、およびその他のスクリプト要素の自動ワード コンプリートが有効になります。スクリプト エディタのオートコンプリート機能 : InTouch スクリプト エディタのオートコンプリートには、InTouch スクリプトを記述する際に使用できる多くの機能が含まれています。これらの機能を使用すると、タグ、ドットフィールド、メソッド、プログラミングが可能なコンストラクト、およびその他のスクリプト要素の自動ワード コンプリートが有効になります。スクリプト エディタのオートコンプリート機能:

- 選択可能なリスト ボックスでオートコンプリート タグ 参照が提供されます。
- 定義、キーワード、スクリプト要素、オブジェクトと属性の名前、およびプログラミングが可能なコンストラクトを含むコンテキスト固有の推奨候補を始めとするメソッド パラメータがオートコンプリート リスト ボックスに表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

これらの機能は、メソッド パラメータおよびスクリプト構文の便利なマニュアルとしてだけでなく、強力な入力方法としても機能します。

CtrL + Space キーを押すと、スクリプト内で選択した場所で使用可能なすべてのオートコンプリート オプションおよび変数が表示されます。コンテキストは、オートコンプリート ポップアップ ウィンドウの下部に表示されるアイコンから識別することができます。

アイコン	意味
	メソッド
	キーワード
	演算子
	変数
	タグ
	ウィンドウ
	ActiveX インスタンス

InTouch スクリプト エディタのオートコンプリート推奨候補の受け入れ

オートコンプリート リスト ボックスから（端点やタブなしで）エディタのキャレットにアイテムを挿入するには、次のいずれかの操作を実行します。

- アイテムをダブルクリックします。
- アイテムをハイライト（選択）して、**Enter** キーまたは **Tab** キーを押します。

スペース、ピリオド、カンマ、開き括弧または閉じ括弧、またはプログラミング言語で使用するその他の句読点（:、 ;、 [、]、 =、 <、 >、 -、 +、 /、 *）を入力すると、オートコンプリート リスト ボックスでハイライトされたアイテムが追加文字と共にエディタ キャレットの位置に挿入されます。

ｽｸﾘﾌﾟﾄのｵｰﾄｺﾝﾌﾟﾘｰﾄを無効化

スクリプトを記述する際にオートコンプリートの推奨候補を表示しない場合、スクリプトのオートコンプリート機能を無効にすることができます。

WindowMaker の設定画面からスクリプトのオートコンプリートを無効にする

1. ［ファイル］メニューの［設定］を選択し、［**WindowMaker**］を選択します。

   WindowMaker の設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［スクリプトのオートコンプリートを無効化］チェックボックスをオンにします。

   スクリプト エディタにオートコンプリートの推奨候補が表示されなくなります。

スクリプト ウィンドウからスクリプトのオートコンプリートを無効にする

1. スクリプト ウィンドウを起動します。
2. ［編集］メニューで［スクリプトのオートコンプリートを無効化］を選択します。

   スクリプト エディタにオートコンプリートの推奨候補が表示されなくなります。

InTouch スクリプト記述での複数レベルの「元に戻す」および「やり直す」機能

スクリプトの変更のシーケンスを選択的に元に戻すことができます。元に戻すことのできる変更の数は、使用可能なメモリの量に応じて異なります。

- メイン メニューの［編集］オプションの［元に戻す］、または **Ctrl + Z** キーの組み合わせを使用して編集を元に戻します。コンテキスト メニュー オプションを使用して、変更を元に戻すことややり直すこともできます。
- メイン メニューの［編集］オプションの［やり直す］、または **Ctrl + Y** キーの組み合わせを使用して編集をやり直しますコンテキスト メニュー オプションを使用して、変更を元に戻すことややり直すこともできます。

元に戻した変更は、やり直すことができます。やり直す操作は、元に戻す操作の逆の操作です。

一般的に、1 つの「元に戻す」操作は、一連の入力または削除操作で構成されます。これらの操作は、オートコンプリート リストでの操作やマウス カーソルの移動、またはスクリプト内の別の場所での選択によって中段されることがあります。

スクリプト エディタを閉じた場合やスクリプト エディタ内で別のスクリプトに切り替えた場合、保留中の「元に戻す」操作および「やり直す」操作は失われます。

InTouch スクリプト エラーの視覚的表示

InTouch スクリプト テキストのエラーは、赤い「波線」の下線でマークされます。

エラーの上にマウス カーソルを置くと、エラー メッセージがツールヒントとして表示されます。ツールヒントのエラー メッセージには、スクリプト検証ボタンを選択したときに表示されるものと同じ情報が表示されます。

場合によっては、複数のエラーに下線が表示されることがあります。エラーによっては、エラー箇所でコンパイラの処理が中断されるので、複数のエラーに常に下線が表示されるわけではありません。

2 つの文字列を連結してパスを形成するために「\」文字が使用されている場合、スクリプト エディタでは「\」の下に赤い下線が表示されます。これは例外で、無視できます。パスが実行時に正常に構築されます。

```
1  TagComment = InfoIntouchAppDir() + "\";
```

InTouch スクリプトの編集の管理

InTouch スクリプト エディタでは、編集を検索して管理するためのツールおよび視覚的リファレンスが提供されます。

スクリプト行番号

スクリプト エディタの左側のマージンには行番号が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- スクリプト エディタがズームされていない場合、最大 4 桁の行番号が表示されます。
- スクリプト内の特定の行に移動するには、右クリックのコンテキスト メニューの［**移動**］オプションを使用します。
- 現在の行のテキストを削除するには、**Ctrl + L** キーの組み合わせを押します。
- テキストを別の行に移動するには、ドラッグ アンド ドロップします。

変更バー

作業中のスクリプトの変更の視覚的な参照として、スクリプト テキスト ウィンドウの左側のマージンに表示される黄色い変更バーによって追加、行の挿入、および編集が示されます。

タグ定義

InTouch スクリプト エディタに新しいタグ名を入力して、**Ctrl + T** キーの組み合わせを押すか、メイン メニューの［**編集**］を選択してタグを定義できます。**タグ名ディクショナリ**が表示されるので、タグ名定義を完了できます。

検索と置換

InTouch スクリプト エディタは、カスタマイズ可能な検索および置換機能を提供します。検索および置換機能の使用方法の詳細については、「スクリプト内での検索」を参照してください。

編集用にスクリプトを開く

既存のスクリプトを開く手順は、スクリプト タイプにより若干異なります。

アプリケーション スクリプトを開く

1. ［**スクリプト**］ペインで［**アプリケーション**］をダブルクリックします。

   ［アプリケーション スクリプト］ダイアログが表示されます。

2. ［**実行条件**］リストで、編集するスクリプトのタイプを選択します。

ウィンドウ スクリプトを開く

1. ［**ウィンドウ**］ペインでウィンドウ名を右クリックし、［**ウィンドウ スクリプト**］を選択します。

   または、スクリプトが関連付けられているウィンドウを開きます。ウィンドウの空白領域を右クリックして、［**ウィンドウ スクリプト**］を選択します。

2. ［**実行条件**］リストで、スクリプトを実行させる条件を選択します。

ActiveX イベント スクリプトを開く

- 以下のいずれかを実行します。
  - ［**スクリプト**］ペインで［**ActiveX イベント**］を展開してスクリプト名をダブルクリックします。
  - スクリプトが関連付けられている ActiveX コントロール インスタンスをダブルクリックします。［**イベント**］タブを選択して、スクリプト名を含むセルをダブルクリックします。

アクション スクリプトを開く

1. アクション スクリプトが関連付けられているグラフィック要素を含むウィンドウを開きます。アクション スクリプトの一般的な用法は、ボタンのアクションのスクリプト記述です。
2. アクション スクリプトが関連付けられているグラフィック要素をダブルクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［タッチ押しボタン］領域で、［スクリプト ボタン］を選択します。スクリプト エディタが表示されます。
4. ［実行条件］リストで、スクリプトを実行させるアクションを選択します。

キー スクリプト、条件スクリプト、またはデータ変化スクリプトを開く

1. ［スクリプト］ペインでスクリプト カテゴリを展開してスクリプト名をダブルクリックします。
   - スクリプト エディタが表示されます。［参照］ボタンを選択して、スクリプト名を選択します。
2. 必要に応じて、［実行条件］リストで、スクリプトを実行させる条件を選択します。

スクリプトへの変更の保存または破棄

スクリプト エディタでの作業中または作業終了時に手動または自動でスクリプトを保存できます。すべてのスクリプトを破棄することもできます。

リストア オプションは、ウィンドウ スクリプトとアプリケーション スクリプトでは使用できません。

注意：変更の保存および破棄は、現在表示されている条件タイプだけではなく、1 つのスクリプト タイプのすべての条件タイプに適用されます。

変更を保存して、スクリプトを開いたままにする

- ［スクリプト］メニューで、［保存］を選択します。

変更を保存して、スクリプトを閉じる

- ［OK］を選択します。

変更を破棄して、スクリプトを開いたままにする

- ［復元］を選択します。

変更を破棄して、スクリプトを閉じる

- ［キャンセル］を選択します。

テキストのコピー、切り取り、および貼り付け

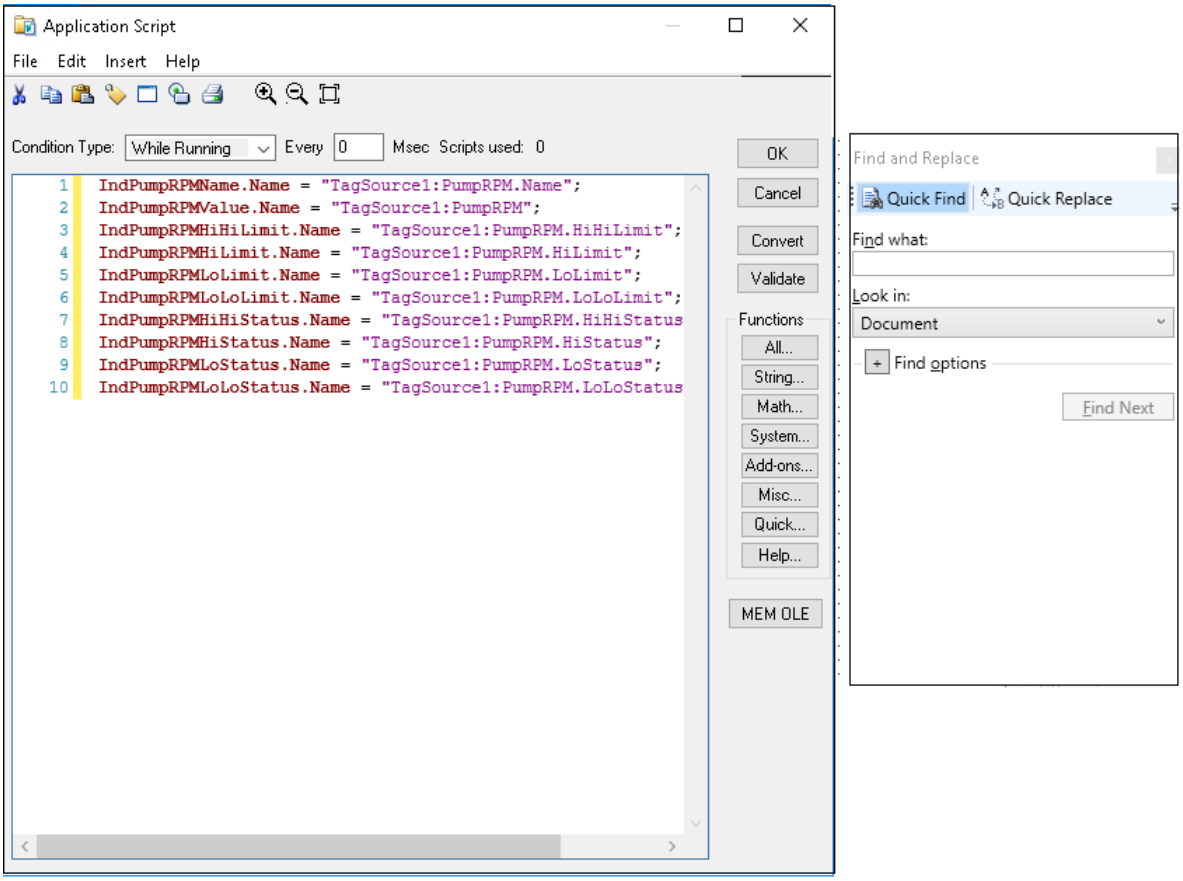
スクリプト エディタでは、その他の Windows アプリケーションと同じ方法で、テキストをコピーしたり、切り取ったり、貼り付けたりすることができます。標準のキーボード ショートカット Ctrl+C、Ctrl+X、および Ctrl+V、またはツールバー ボタンを使用します。

行を選択して、その行のテキストをスクリプト内の別の場所にドラッグすることもできます。

スクリプト内での検索

InTouch スクリプト エディタでは、開いているスクリプト内で検索文字列およびカスタマイズ可能な検索オプションのセットに基づいて検索と置換を行うことができます。［検索］と［置換］は、スクリプト エディタのメニュー バーの［編集］からアクセスできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



また、**Ctrl + F**（クイック検索）および **Ctrl + H**（クイック置換）のキーボード ショートカットを使用して、[**検索と置換**] ダイアログ ボックスを表示することもできます。

- デフォルトの検索オプションを使用したシンプルな検索では、[**クイック検索**] タブを選択して、[**検索する文字列**] フィールドに単語または語句を入力します。

  [**次を検索**] を選択して検索を開始します。スクリプト内で検索文字列に一致する単語または語句の背景が黄色くハイライトされます。

- デフォルト オプションを使用したシンプルな置換操作では、[**クイック置換**] タブを選択し、[**検索する文字列**] フィールドに単語または語句を入力して、[**次の文字列に置換**] フィールドに置換する文字または語句を入力します。

  [**次を検索**] を選択して置換操作を開始します。スクリプト内で、[**検索する文字列**] に入力した検索文字列に一致する単語または語句の背景が明るい青でハイライトされます。一致する単語または単語が見つかった後、3 つの置換オプションを使用できます。

  - [**次を検索**] を選択すると、一致する現在の単語または語句を無視して、スクリプト内の次の一致が継続されます。
  - [**置換**] を選択すると、一致する現在の単語または語句が [**次の文字列に置換**] フィールドに入力した文字列で置き換えられます。
  - [**すべて置換**] を選択すると、一致するすべての単語または語句が [**次の文字列に置換**] フィールドに入力した文字列で置き換えられます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

検索または置換の設定

［**検索と置換**］ダイアログ ボックスには、検索オプションを設定する一連のオプションがあります。

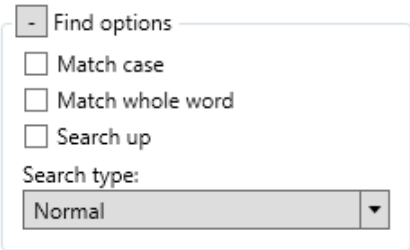
検索

［**検索**］フィールドには、スクリプト全体を検索するオプション（［**Document**］（**ドキュメント**））またはスクリプトの選択範囲のみを検索するオプション（**Selection**（**選択**））があります。デフォルト設定は［**Document**］（**ドキュメント**）です。

スクリプトの選択した部分内だけを検索する場合、検索を実行する前にマウスを使用してスクリプト内の検索領域を選択します。選択したスクリプト行の背景が青でハイライトされます。この領域が検索対象になります。検索結果には、選択したスクリプト領域内の一致項目のみが表示されます。

検索オプション

［**検索オプション**］セクションは展開と折り畳みが可能です。以下のオプションを選択または選択解除して検索結果を絞り込むことができます。



- **大文字と小文字を区別**

  このオプションを選択すると、［**検索する文字列**］に入力した文字列の内容と大文字／小文字が一致するインスタンスだけが検索結果に表示されます。たとえば、［**大文字と小文字を区別**］オプションを選択した状態で Triangle4 を検索した場合、Triangle4 が返されますが、triangle4 は返されません。

- **全単語一致のみ**

  このオプションを選択すると、［**検索する文字列**］に入力した文字列の語句に完全に一致するインスタンスだけが検索結果に表示されます。たとえば、LogicBit を検索する場合、LogicBit が返されますが、LogicBits は返されません。

- **Search up**（**上方向に検索**）

  このオプションを選択すると、スクリプト内の現在の位置からスクリプトの最初に向かって検索が実行されます。デフォルトでは、スクリプト内の現在の位置からスクリプトの最後に向かって検索が実行されます。

Search type（検索のタイプ）

［**Search type**］（**検索のタイプ**）フィールドを使用すると、検索のタイプに基づいてスクリプト検索を実行できます。

- **通常**

  これは、デフォルトの検索タイプで、検索文字列の文字とスクリプト内のテキストが一致する必要があります。

- 正規表現による検索

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

正規表現は、スクリプトを検索するときに 1 つまたは複数の文字列を表します。正規表現には、検索する文字列に一致する文字のテンプレートとして機能する通常の文字が含まれます。

- ワイルドカード文字による検索

  ワイルドカード検索では、スクリプト内を検索するときに 1 つまたは複数の文字を表すためにアスタリスク（*）や疑問符（?）などのキーボード文字を使用します。

- 頭文字による検索

  頭文字検索では、単語の最初で一致する 1 文字が検索され、その後に各大文字またはアンダースコアの後の文字列が検索されます。

- 簡易略記法による検索

  簡易略記法検索では、検索パターン文字列の間にホワイトスペース以外の文字を含めることによって「頭文字による検索」オプションが拡張されます。

構文に準拠する文字列、および選択した検索のタイプでサポートされる文字を入力する必要があります。

ワイルドカード文字による検索

ワイルドカード検索では、スクリプトの文字列を検索するときにいくつかのリテラル文字または空の文字列として解釈可能な 1 つのキーボード文字を使用します。

実際の文字がわからない場合や、完全な検索文字列の入力を省略する場合、1 つまたは複数の文字の代わりとしてワイルドカード文字を使用できます。

ワイルドカード文字	使用方法
アスタリスク（*）	検索文字列内の 1 つのアスタリスクは文字の任意のシーケンスに一致します。アスタリスクは、ゼロ個以上の文字の代わりとして使用します。 **使用例** • Logic* 「Logic1」および「LogicTest」が検索されますが、「ALogicTest」は検索されません。 • *Test* 「LogicTest1」および「PumpTestABC」が検索されますが、「LogicTst1」は検索されません。
疑問符（?）	検索文字列内の 1 つの疑問符は、検索文字列内の特定の位置にある任意の文字を表します。 **使用例** • LogicTest? 「LogicTest1」および「LogicTestA」が検索されますが、「ALogicTest1」は検索されません。 • LogicTest?2

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	「LogicTest12」が検索されますが、「LogicTest13」は検索されません。

正規表現による検索

正規表現は、英数字および特殊文字（メタ文字）を使用して、一致する 1 つまたは複数の文字列を表します。正規表現は、検索するスクリプトテキストと比較する文字パターンとして機能します。

正規表現は、算術表現と同様に構成されます。さまざまなメタ文字と演算子を使用して複数の小さい表現を組み合わせて大きな表現を作成できます。

正規表現の要素は、個々の文字、文字のセット、文字の範囲、または複数の文字の間の選択です。これらの要素の組み合わせも要素として使用できます。

スクリプトの正規表現

正規表現	説明	例
をクリックします。	1 つの任意の文字に一致します（改行を除く）。	「s.e」は 「step」の「ste」および「transfer」の「sfe」に一致しますが、「across」の「acro」には一致しません。
*	先行する式のゼロ以上の発生に一致します（可能な限り多くの文字に一致します）。	a*r 「rack」の「r」、「ark」の「ar」、および「aardvark」の「aar」に一致します。
.*	任意の文字列にゼロ回以上一致します（ワイルドカード *）。	「c.*e」は「racket」の「cke」、「comment」の「comme」、および「「code」の「code」に一致します。
+、	先行する式の 1 つ以上の発生に一致します（可能な限り多くの文字に一致します）。	e.+e 「feeder」の「eede」に一致しますが、「ee」には一致しません。
.+	任意の文字列に 1 回以上一致します（ワイルドカード ?）。	「e.+e」は「feeder」の「eede」に一致しますが、「ee」には一致しません。
*?	先行する式のゼロ以上の発生に一致します（可能な限り少ない文字に一致します）。	e.*?e 「feeder」の「ee」に一致しますが、「eede」には一致しません。
+?	先行する式の 1 つ以上の発生に一致します（可能な限り少ない文字に一致します）。	e.+?e 「enterprise」の「ente」および「erprise」に一致しますが、「enterprise」という単語には一致しません。
^	行または文字列の先頭にある一致文字列に一致します。	^car 行の最初にある「car」という単語にのみ一致します。
\r?$	一致文字列を行の末尾にアンカーします。	End\r?$ 行の末尾にある「end」という単語にのみ一致します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

[abc]	セット内の 1 つの任意の文字に一致します。	b[abc] 「ba」、「bb」、および「bc」に一致します。
[a-f]	文字の範囲に含まれる任意の文字に一致します。	be[n-t]「between」の「bet」、「beneath」の「ben」、および「beside」の「bes」に一致しますが、「below」には一致しません。
()	括弧で囲まれた表現をキャプチャして暗示的に番号を設定します。	([a-z])X\1 「aXa」と「bXb」に一致しますが、「aXb」には一致しません。「\1」は最初の式グループ「[a-z]」を参照します。
(?!abc)	一致を検証します。	real (?!ity) 「realty」および「really」の「real」に一致しますが、「reality」には一致しません。また、「realityreal」の（最初の「real」ではなく）2 番目の「real」も検索されます。
[^abc]	指定した文字のセットに含まれない任意の文字に一致します。	be[^n-t] 「before」の「bef」、「behind」の「beh」、および「below」の「bel」に一致しますが、「beneath」には一致しません。
|	記号の前または後のいずれかの表現に一致します。	(sponge|mud)bath 「spongebath」および「mudbath」に一致します。
|\^	円記号の後の文字をエスケープします。	
{x},	先行する文字またはグループの発生数を指定します。	x(ab){2}x 「xababx」に一致します。 「x(ab){2,3}x」は「xababx」と「xababababx」に一致しますが、「xabababababx」には一致しません。
\p{X}	Unicode 文字クラスのテキストに一致します。ここで「X」は Unicode 番号を表します。	\p{Lu} 「Thomas Doe」の「T」と「D」に一致します。
\b	単語境界に一致します。	\bin 「inside」の「in」に一致しますが、「pinto」には一致しません。
\r?\n	改行（新しい行の前のキャリッジ リターン）	End\r?\nBegin 「End」が行の最後の文字列で、次の行の最初の文字列が「Begin」の場合にのみ「End」と「Begin」に一致します。
\w	任意の英数字に一致します。	a\wd 「add」と「a1d」に一致しますが、「a d」には一致しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

(?[^\r\n])\s	ホワイトスペース文字に一致します。	Public\sInterface 「Public Interface」という語句に一致します。
\d	任意の数字に一致します。	\d 「3456」の「3」、「23」の「2」、および「1」の「1」に一致します。
\uXXXX（「XXXX」は Unicode 文字値を指定します）	1 つの Unicode 文字に一致します。	\u0065 文字「e」に一致します。
\b(\w+|]\w-[0-9\]]\w*)\b	識別子に一致します。	「type1」に一致しますが、「&type1」および「#define」には一致しません。
((\".+?\")|('.+?'))	引用符内の文字列に一致します。	一重引用符または二重引用符に囲まれた文字列に一致します。
\b0[xX]([0-9a-fA-F])\b	16 進数文字に一致します。	「0xc67f」に一致しますが、「0xc67fc67f」には一致しません。
\b[0-9]\.\[0-9]+\b	整数と「d」に一致します。	「1.333」に一致します。

優先順位

正規表現は左から右に評価され、優先順位に従います。

次の表に正規表現演算子の優先順位を高い順から示します。

演算子	説明
\	エスケープ
()、(?:)、(?=)、[]	括弧と角括弧
*、+、?、{*n*}、{*n*,}、{*n*,*m*}	修飾子
^、$、*任意のメタ文字*	アンカーとシーケンス
|	代替

文字は代替演算子よりも優先順位が高いので、「m|food」は「m」または「food」に一致します。

頭文字による検索

頭文字検索では、単語の最初に一致する 1 文字が検索され、各大文字またはアンダースコアの後の文字列が検索されます。

例：「LB」では「LogicBits」のインスタンスが検索されます。

簡易略記法による検索

簡易略記法検索では、検索パターン文字列の間にホワイトスペース以外の任意の数の文字を含めることによって「頭文字による検索」オプションが拡張されます。

例：「ta」では「Triangle」のインスタンスが検索されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

コード要素の挿入

リストからさまざまなコード要素を選択してスクリプトに自動的に挿入することができます。この機能を利用すると時間を節約し、入力エラーのリスクを軽減することができます。コード要素にアクセスするには、メイン メニューの［**挿入**］オプションを使用するか、オートコンプリート ポップアップ ウィンドウのリストから目的の要素を選択します。

オートコンプリート ポップアップ ウィンドウとスクリプト エディタのメニュー項目の両方から挿入できるコード要素を以下に示します。

- 関数
- タグ名とドットフィールド（ドットフィールドの場合、タグ名の後にピリオドを入力してオートコンプリート ポップアップ ウィンドウを開きます）
- 変数
- ウィンドウ名。
- ActiveX インスタンス
- キーワード
- オペレータ

オートコンプリート ポップアップ ウィンドウの下部には、使用可能な要素に直接アクセスできる一連のアイコンが表示されます。

スクリプト関数のヘルプへのアクセス

特定のスクリプト関数のヘルプを検索する場合、スクリプト エディタから直接アクセスできます。

特定のスクリプト関数のヘルプを表示する

1. スクリプト エディタの右下隅で、［**ヘルプ**］を選択します。

   関数のリストが表示されます。

2. 関数のヘルプを表示するには、リストに含まれる関数名を選択します。

   対応するヘルプ トピックが表示されます。

正確な構文を確認するためのスクリプトの検証

スクリプトを保存すると、その構文が正しいかどうかがスクリプト エディタによって自動的に検証されます。エラーが検出された場合、詳細を示すメッセージが表示されます。スクリプトを保存する前に、すべての構文エラーを修正する必要があります。また、スクリプトを編集している間、手動で検証を開始することもできます。

スクリプト構文を手動で確認する

- 「**検証**」を選択します。

スクリプト エラーのハイライト表示

InTouch スクリプト エディタには、スクリプト エラーをハイライトする機能もあります。詳細については、「InTouch スクリプト エラーの視覚的表示」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スクリプトの印刷

スクリプト エディタからスクリプトを個別に印刷したり、WindowMaker の印刷機能を使用して、特定のタイプのスクリプトをすべて印刷したりできます。

スクリプト エディタからスクリプトを個別に印刷したり、WindowMaker の印刷機能を使用して、特定のタイプのスクリプトをすべて印刷したりできます。

個々のスクリプトを印刷する

1. スクリプト エディタでスクリプトを開きます。
2. ツールバーの［**スクリプトの印刷**］をクリックします。スクリプトは、Windows のデフォルトのプリンタに印刷されます。

特定のタイプのすべてのスクリプトを印刷する

1. クイック アクセス ツールバーの［**印刷**］を選択します。

   ［**アプリケーション情報の印刷**］ダイアログ ボックスが表示されます。
2. ウィンドウ スクリプトを印刷するには、以下の手順を実行します。
   a. ［**ウィンドウ**］を選択します。
   b. 印刷するウィンドウを選択します。

      ［**全て**］を選択すると、アプリケーションにあるすべてのウィンドウの情報が印刷されます。

      ［**選択**］を選ぶと、特定のウィンドウの情報のみが印刷されます。［**ウィンドウの印刷**］ダイアログ ボックスが開きます。印刷したいウィンドウを選択し、［**OK**］を選択します。

      ［**バッチ**］を選択すると、.csv ファイルで指定したウィンドウの情報のみが印刷されます。
   c. ウィンドウに関連付けられているスクリプトを印刷するには、［**ウィンドウ スクリプト**］を選択します。
3. その他のタイプのスクリプトを印刷するには、適切なチェック ボックスをオンにします。すべてのスクリプトを印刷するには、［**すべてのスクリプト**］を選択します。
4. ［**次へ**］を選択します。［**出力の送信先の選択**］ダイアログ ボックスが表示されます。
5. 以下のいずれかを実行します。
   - ［**プリンタに出力を送信する**］を選択します。
   - ［**テキスト ファイルに出力を送信する**］を選択します。
6. ［**参照**］を選択して、プリンタを選択するか、ファイルを検索します。
7. ［**印刷**］を選択します。

すべてのスクリプトを印刷する

1. ［**すべてのスクリプト**］を選択すると、アプリケーションで使用されているすべてのスクリプトが印刷されます。

   ［**すべてのスクリプト**］チェック ボックスをオフにすると、印刷対象を選択したタイプのスクリプトだけに制限できます。その場合は、印刷するスクリプト タイプの横のチェック ボックスをオンにします。
2. ［**次へ**］を選択します。［**出力の送信先の選択**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. タグ名ディクショナリの内容を印刷するか、または出力結果をテキスト ファイルや .html ファイルに送信するかのオプションを選択します。
4. ［印刷］を選択します。

スクリプトの削除

スクリプトを削除する手順は、スクリプト タイプにより異なります。以下のセクションを参照してください。

- アプリケーション スクリプトの設定
- ウィンドウ スクリプトの設定
- キー スクリプトの設定
- 条件スクリプトの設定
- データ変化スクリプトの設定
- アクション スクリプトの設定
- ActiveX イベント スクリプトの設定

スクリプトを表示するズーム オプションの調整

スクリプトを表示するズーム オプション（ズーム イン、ズーム アウト、ズーム解除など）を調整できます。

ズームイン

1. スクリプト エディタを開きます。
2. スクリプト エディタ ツールバーで［**ズーム イン**］を選択します。

   または、［**編集**］を選択して、［**ズーム イン**］を選択します。

ズーム アウト

1. スクリプト エディタを開きます。
2. スクリプト エディタ ツールバーで［**ズーム アウト**］を選択します。

   または、［**編集**］を選択して、［**ズーム アウト**］を選択します。

ズーム解除するには

1. スクリプト エディタを開きます。
2. スクリプト エディタ ツールバーで［**ズーム解除**］を選択します。

   または、［**編集**］を選択して、［**ズーム解除**］を選択します。

スクリプト トリガ

すべての InTouch HMI スクリプトは、スクリプト トリガにより実行されます。各スクリプト タイプには、そのスクリプトを起動する 1 つまたは複数のトリガがあります。

スクリプト エディタでは、スクリプトを実行するために使用するスクリプト トリガを選択できます。スクリプト トリガは、スクリプトがいつ、またどのように実行されるかに基づいて選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザー アクション、内部状態、およびタグ名の値の変更に基づいてさまざまなトリガを設定できます。ユーザー アクションには、キーを押す、グラフィック要素を選択するなどがあります。内部状態トリガには、WindowViewer を含むことができます。

スクリプトは、以下のようなアクションにより起動されます。

- WindowViewer の起動およびシャットダウン。「アプリケーション スクリプトの設定」を参照してください。
- ウィンドウを開く、閉じる。「ウィンドウ スクリプトの設定」を参照してください。
- キーおよびキーの組み合わせを押す「キー スクリプトの設定」を参照してください。
- タグ名や式の値など、特定の条件を満たす。「条件スクリプトの設定」を参照してください。
- タグ名の値またはタグ名フィールド値の変更。「データ変化スクリプトの設定」を参照してください。
- グラフィック オブジェクトを選択する。「アクション スクリプトの設定」を参照してください。
- コントロールの選択など、ActiveX コントロール内で発生するイベント。「ActiveX イベント スクリプトの設定」を参照してください。

また、スクリプトの実行を一時停止することもできます。デフォルトでは、WindowViewer が起動されると、ロジックが実行され、スクリプトが実行されます。スクリプトの実行は、ロジックを中断することによってランタイムで一時停止できます。一時停止した後、スクリプト実行は再開できます。詳細については、「ランタイムでのスクリプト実行の一時停止」を参照してください。

スクリプト トリガのタイプ

InTouch HMI では、スクリプトは 7 つのタイプに分割されています。スクリプトの各タイプには、スクリプトを起動するために選択できるトリガが 1 つまたは複数あります。

- アプリケーション スクリプトには、3 つのトリガ（起動時、終了時、および実行中）があります。各トリガは、異なるスクリプトを実行できます。
- ウィンドウ スクリプトには、3 つのトリガ（開く時、閉じる時、表示中）があります。各トリガは、異なるスクリプトを実行できます。
- キー スクリプトには、3 つのトリガ（離した時、押した時、押している間）があります。. 各トリガは、異なるスクリプトを実行できます。
- 条件スクリプトには 4 つのトリガ（True の時、True の間、False の時、および False の間）があります。各トリガは、異なるスクリプトを実行できます。
- データ変化スクリプトは、特定のタグ変数または式の値が変化したときに実行します。
- アクション スクリプトは、オペレータが InTouch HMI グラフィック オブジェクトを選択したときに、一度または定期的に実行します。
- ActiveX イベント スクリプトは、ActiveX コントロールの選択など、ActiveX イベントが発生したときに一度実行します。

複数のトリガの使用

ほとんどのスクリプト タイプでは、複数のトリガを使用して、さまざまなスクリプトを各トリガに関連付けることができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

たとえば、WindowViewer が起動したときにスクリプトを実行して、WindowViewer の実行中に別のスクリプトを定期的に実行するように、アプリケーション スクリプトを設定できます。

［**実行条件**］リストでトリガを選択して、トリガの既存のスクリプトを表示します。

定期的なスクリプトの実行

定期的に実行するスクリプトは、トリガ後すぐに実行するのではなく、指定期間後に実行します。

たとえば、特定のキーが押されている間 5000 ミリ秒ごとに実行するようにキー スクリプトを設定すると、キーが押されて、押された状態のままで 5 秒経過してからそのスクリプトは実行され、その後は 5 秒ごとに実行されます。

アプリケーション スクリプトの設定

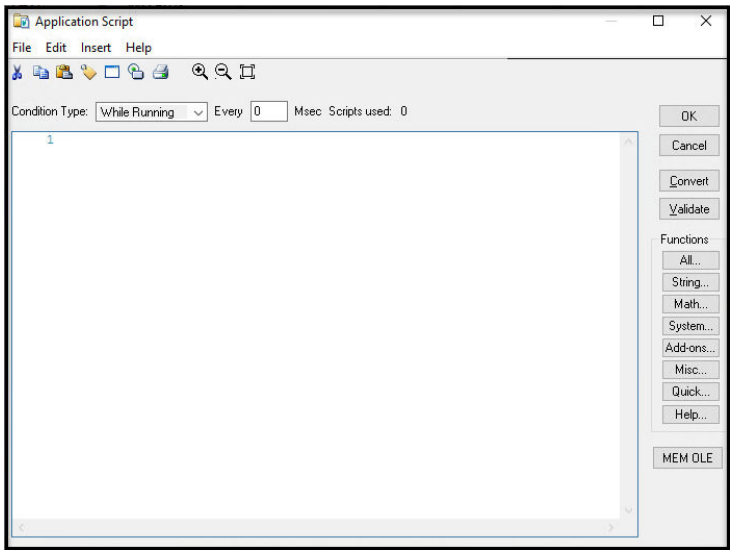
アプリケーション スクリプトは、InTouch HMI アプリケーション全体にリンクされます。アプリケーション スクリプトを使用すると、以下の操作を実行できます。

- WindowViewer が起動されたときにスクリプトを一度実行する
- WindowViewer の実行中にスクリプトを定期的に実行する
- WindowViewer がシャットダウンしたときにスクリプトを一度実行する。

アプリケーション スクリプトを設定する

1. ［**スクリプト**］ペインで［**アプリケーション**］を右クリックし、［**開く**］を選択します。

   ［**アプリケーション スクリプト**］ダイアログ ボックスが表示されます。



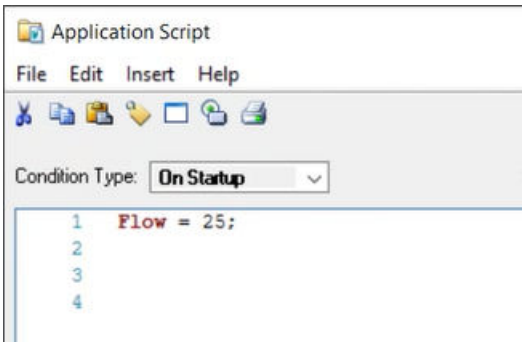
2. ［**実行条件**］リストで、スクリプト実行の条件を選択します。

   - WindowViewer が起動されたときに一度実行するようにスクリプトを設定するには、［**起動時**］を選択します。

     例： InTouch アプリケーションで **Flow** というタグ変数の初期値が 50 で、アプリケーションの起動時にこの値を変更する場合は、［**実行条件**］リストで、［**起動時**］を選択し、次のスクリプトを使用します。

     ```
     Flow = 25;
     ```

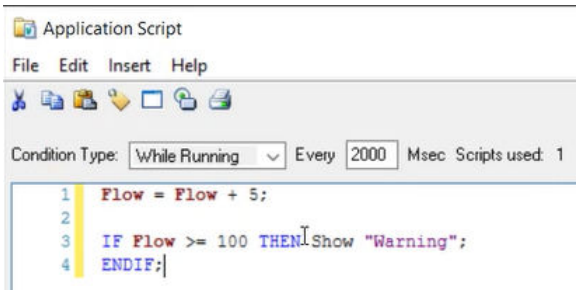
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



- WindowViewer の実行中に定期的に実行するようにスクリプトを設定するには、[**実行中**] を選択します。

  例 : Flow というタグを持つ InTouch アプリケーションを例に考えます。アプリケーションの実行中に **Flow** の値を 2 秒ごとに 5 ずつ増加させ、タグ変数の値が 100 以上になった場合に **Warning** という名前のウィンドウを表示する場合は、[**実行条件**] を [**実行中**] に設定して、次のスクリプトを使用します。

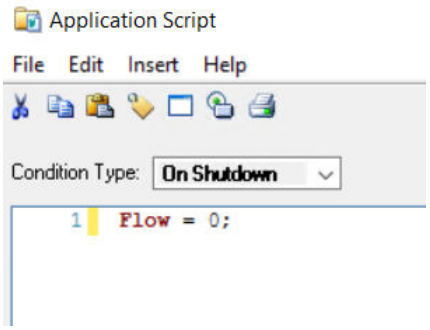
```
Flow = Flow + 5;

If Flow >= 100 THEN Show "Warning"

ENDIF;
```



- WindowViewer がシャットダウンされたときに一度実行するようにスクリプトを設定するには、[**終了時**] を選択します。

  例 : Flow というタグを持つ InTouch アプリケーションを例に考えます。アプリケーションを閉じる時に **Flow** の値を 0 に変更する場合は、[**実行条件**] を [**終了時**] に設定して、次のスクリプトを使用します。

```
Flow = 0;
```



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. 前の手順で［**実行中**］を選択した場合、［**実行周期**］ボックスに 1 ～ 360000 ミリ秒の間隔を入力します。この間隔は、スクリプトが実行される頻度を指定します。
4. スクリプトをウィンドウに入力します。
5. ［**OK**］を選択します。

アプリケーション スクリプトを削除する

1. ［**スクリプト**］ペインで［**アプリケーション**］を右クリックし、［**開く**］を選択します。

   ［**アプリケーション スクリプト**］ダイアログ ボックスが表示されます。
2. ［**実行条件**］リストで、削除するスクリプトの条件を選択します。

   スクリプトが、［**アプリケーション スクリプト**］ダイアログ ボックスのメイン セクションに表示されます。
3. ［**編集**］メニューの［**クリア**］を選択します。

   スクリプトがメイン セクションからクリアされ、関連付けられたスクリプトが削除されます。

アプリケーション スクリプトの制限事項

WindowViewer の起動時または終了時に実行されるアプリケーション スクリプトには、他のオブジェクトとの対話に関する制限事項があります。

起動時のアプリケーション スクリプトは、以下の操作を行うためには使用できません。

- ActiveX メソッド、プロパティ、またはイベントの参照
- コントロールおよび I/O タグ変数またはリモート リファレンスに対する読み取りまたは書き込み
- データ変化スクリプトおよび条件スクリプトの実行

終了時のアプリケーション スクリプトは、以下の操作を行うためには使用できません。

- コントロールおよび I/O タグ変数またはリモート リファレンスに対する読み取りまたは書き込み
- その他のアプリケーションの起動

ウィンドウ スクリプトの設定

ウィンドウ スクリプトは、特定のウィンドウにリンクされるスクリプトです。**GetWindowName** スクリプト関数を使用して、ランタイム環境でのウィンドウの読み込みに必要なスクリプト作成を減らすことができます。ウィンドウ スクリプトを使用すると、以下の操作を実行できます。

- InTouch ウィンドウが開く時にスクリプトを一度実行する。
- InTouch ウィンドウが表示中にスクリプトを定期的に実行する。
- InTouch ウィンドウが閉じる時にスクリプトを一度実行する。

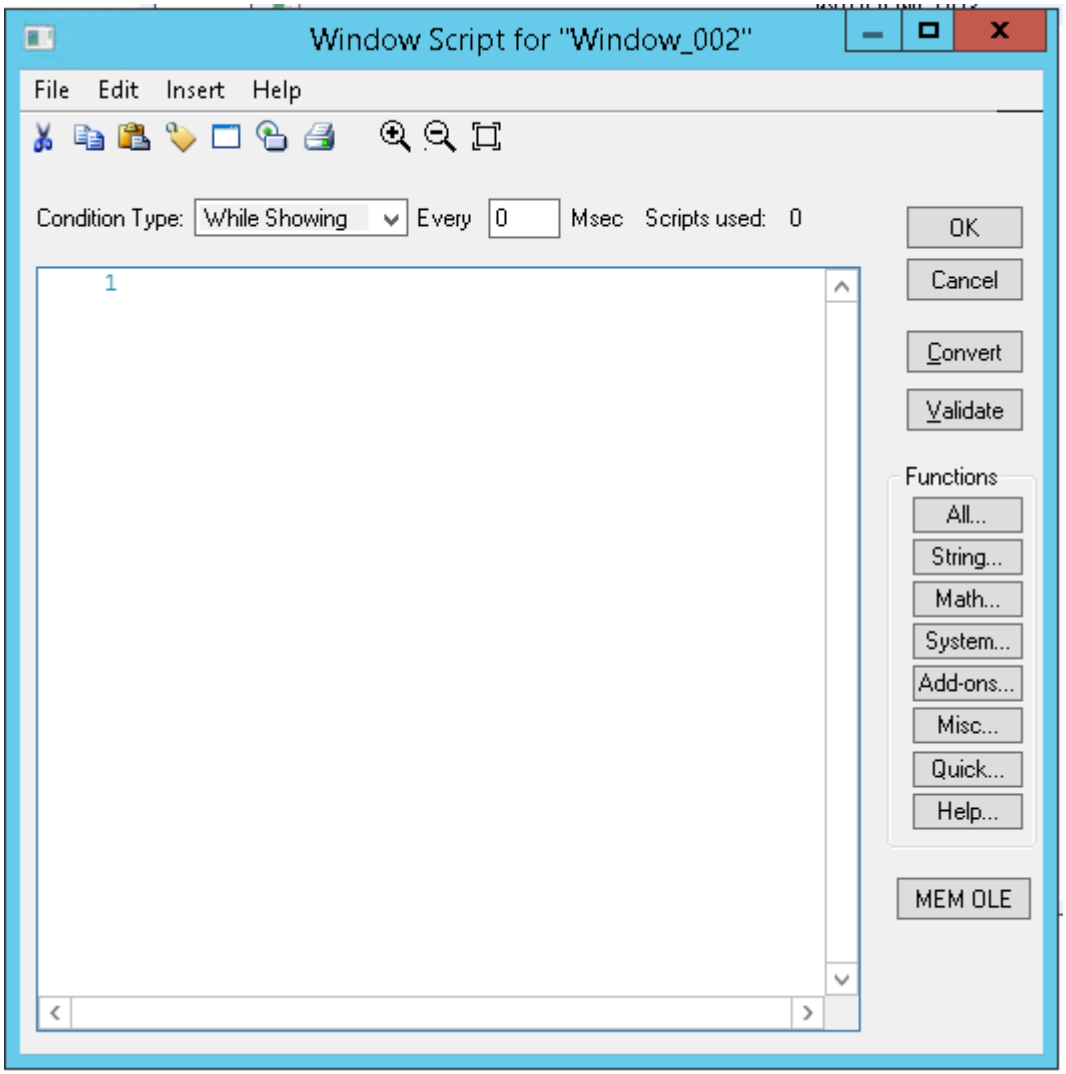
注意 : InTouch ウィンドウを開くことは、「InTouch ウィンドウを表示する」とも言います。また、InTouch ウィンドウを閉じることは、「InTouch ウィンドウを非表示にする」とも言います。

ウィンドウ スクリプトを設定する

1. ［**ウィンドウ**］ペインでウィンドウを右クリックし、［**ウィンドウ スクリプト**］をクリックします。

   ［**ウィンドウ スクリプト - ウィンドウ名**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［**実行条件**］リストで、以下のいずれかを実行します。
   - 関連付けられたウィンドウが起動したときに一度実行するようにスクリプトを設定するには、［**開く時**］を選択します。
   - 関連付けられたウィンドウが開いている間に定期的に実行するようにスクリプトを設定するには、［**表示中**］を選択します。
   - 関連付けられたウィンドウが閉じられたときに一度実行するようにスクリプトを設定するには、［**閉じる時**］を選択します。
3. 前の手順で［**表示中**］を選択した場合、［**実行周期**］ボックスに 1 ～ 360000 ミリ秒の間隔を入力します。
4. スクリプトをウィンドウに入力します。
5. ［**OK**］を選択します。

ウィンドウ スクリプトを削除する

1. ［**ウィンドウ**］ペインでウィンドウを右クリックし、［**ウィンドウ スクリプト**］を選択します。［**ウィンドウ スクリプト - ウィンドウ名**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**実行条件**］リストで、削除するスクリプトのスクリプト トリガを選択します。スクリプトが、［**ウィンドウ スクリプト - ウィンドウ名**］ダイアログ ボックスのメイン セクションに表示されます。
3. ［**編集**］メニューの［**クリア**］を選択します。

重要 : ［閉じる時］スクリプトを使用して、I/O タグ名に対する読み取りおよび書き込みを行わないでください。I/O 値の更新は、ウィンドウが非表示になる前に必ずしも完了するわけではありません。
ウィンドウが閉じるときに I/O タグ変数に対する読み取りまたは書き込みを行うには、データ変更スクリプトを設定して、［閉じる時］スクリプトからスクリプトをアクティブ化します。

キー スクリプトの設定

キー スクリプトは、特定のキーまたはキーの組み合わせを押す操作にリンクされるスクリプトです。キー スクリプトを使用すると、以下の操作を実行できます。

- キーまたはキーの組み合わせが押されたときにスクリプトを一度実行します。
- キーまたはキーの組み合わせが押されて、離されていないときにスクリプトを定期的に実行します。
- キーまたはキーの組み合わせが離されたときにスクリプトを一度実行します。

キー スクリプトは、スクリプトを開始するキーの名前で識別されます（Ctrl+q など）。

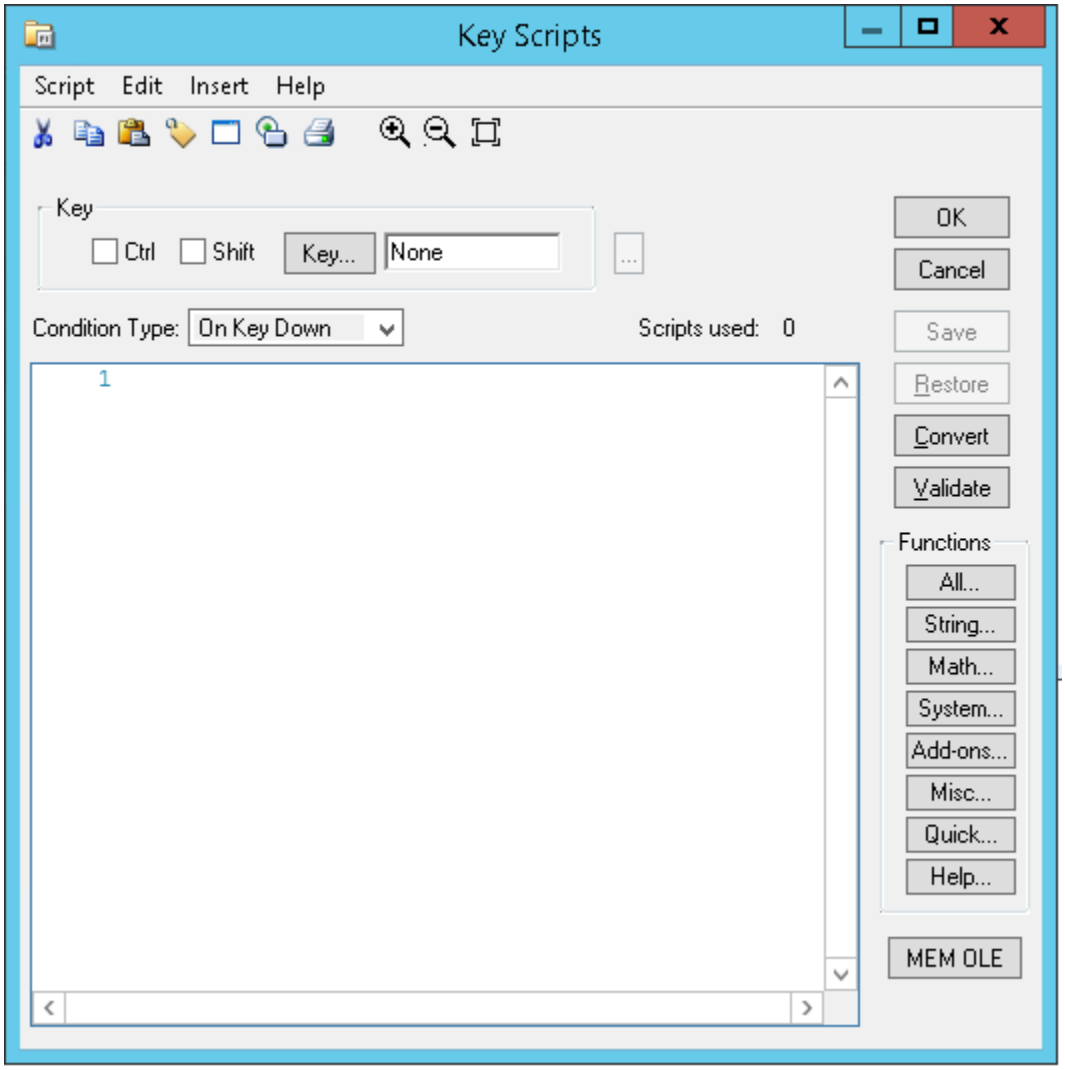
注意 : トリガするために同じキーまたはキーの組み合わせを使用するアクション スクリプトを設定している場合、キー スクリプトは無視され、アクション スクリプトが実行されます。

キー スクリプトを設定する

1. ［**スクリプト**］ペインで以下のいずれかを実行します。
   - 新しいキー スクリプトを設定するには、［**キー**］を右クリックして、［**新規**］を選択します。

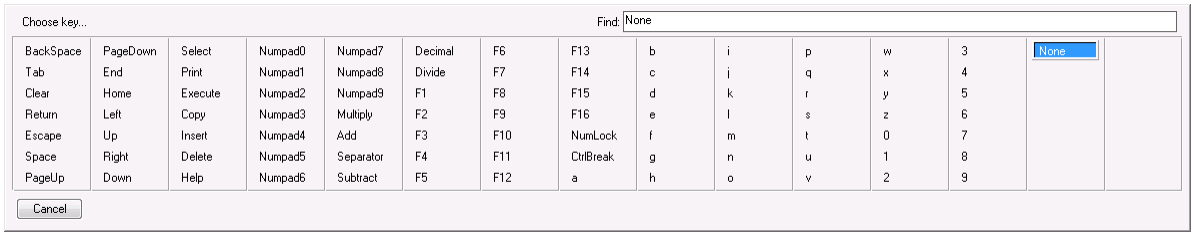
     ［**キー スクリプト**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



- 既存のキー スクリプトを設定するには、［**キー**］を展開し、スクリプト名を右クリックして、［**編集**］を選択します。［**キー スクリプトの編集**］ダイアログ ボックスが表示されます。

2. ［**キー**］を選択して、［**キーの選択**］ダイアログ ボックスからキーを選択します。



3. ［**Ctrl**］または［**Shift**］、あるいはこれら両方のチェック ボックスをオンにして、コントロール キーやシフト キーの組み合わせを選択したキーに割り当てます。
4. ［**実行条件**］リストで、以下のいずれかを実行します。
   - 関連付けられたキーまたはキーの組み合わせが押されたときに一度実行するようにスクリプトを設定するには、［**押した時**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 関連付けられたキーまたはキーの組み合わせが押されている間に定期的に実行するようにスクリプトを設定するには、［**押している間**］を選択します。
- 関連付けられたキーまたはキーの組み合わせが離されたときに一度実行するようにスクリプトを設定するには、［**離した時**］を選択します。

5. 前の手順で［**押している間**］を選択した場合、［**実行周期**］ボックスに 1 ～ 360000 ミリ秒の間隔を入力します。
6. スクリプトをウィンドウに入力します。
7. ［**OK**］を選択します。

キーに関連付けられているすべてのキー スクリプトを削除する

- ［**スクリプト**］ペインで［**キー**］を展開し、キー スクリプト名を右クリックして、［**削除**］を選択します。メッセージが表示されたら、［**はい**］を選択します。

キーに関連付けられているキー スクリプトを削除する

1. ［**クラシック ビュー**］を使用して、［**スクリプト**］ペインで、［**キー**］を展開し、キー スクリプト名を右クリックして、［**編集**］を選択します。［**キー スクリプトの編集**］ダイアログ ボックスが表示されます。
2. ［**実行条件**］リストで、削除するスクリプトのスクリプト トリガを選択します。スクリプトが、［**キー スクリプトの編集**］ダイアログ ボックスのメイン セクションに表示されます。
3. ［**編集**］メニューの［**クリア**］を選択します。スクリプトがメイン セクションからクリアされ、関連付けられたスクリプトが削除されます。

条件スクリプトの設定

条件スクリプトは、特定の論理条件が満たされたときにトリガされます。条件スクリプトを使用すると、以下のようにスクリプトを実行できます。

- 条件が満たされたときに一度実行する
- 条件が満たされなかったときに一度実行する
- 特定の条件が満たされている間定期的に実行する
- 特定の条件が満たされていない間定期的に実行する

条件スクリプトは、スクリプトを開始する条件構文により識別されます（tag1>=13 など）。

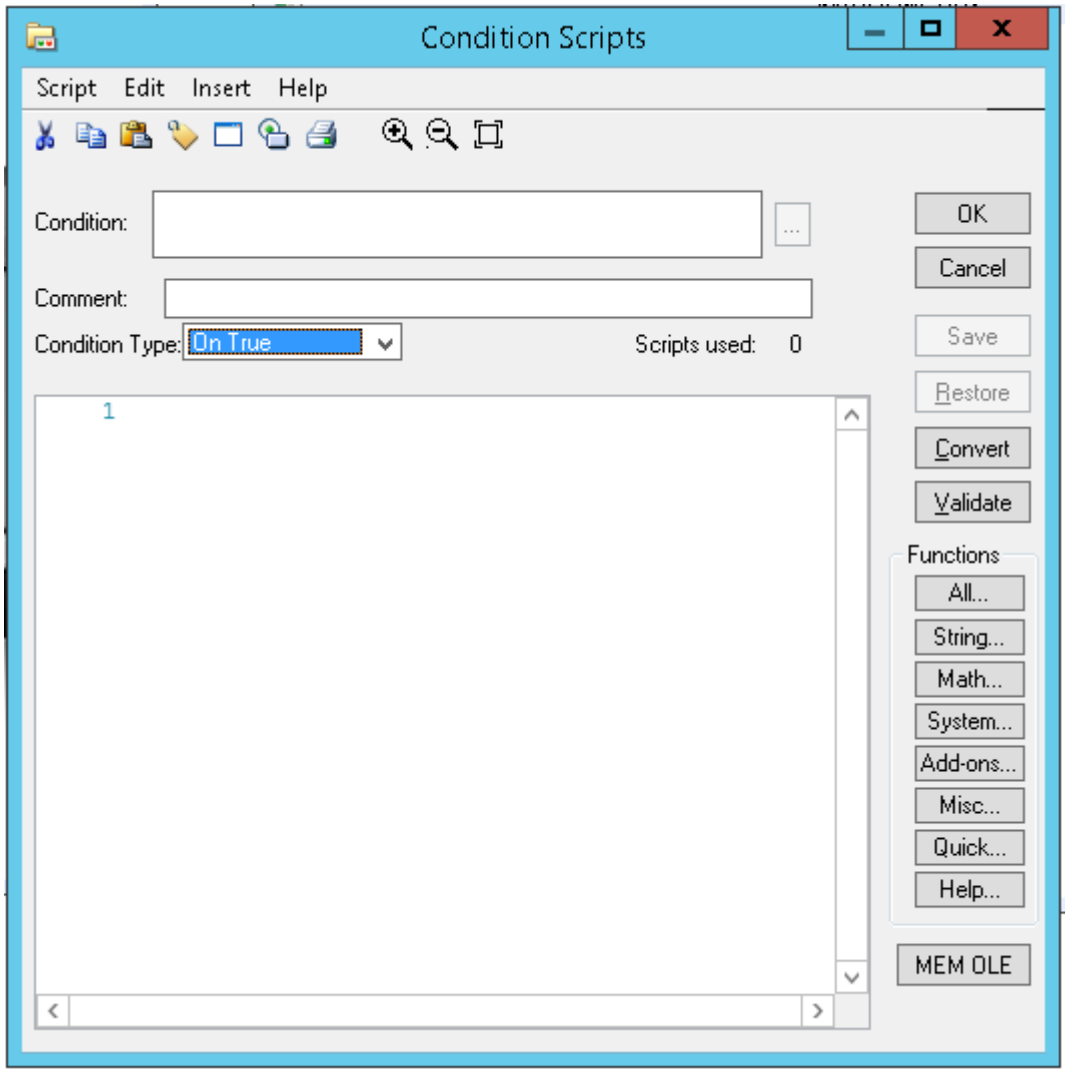
注意：True の時の条件タイプが割り当てられているスクリプトは、条件が False から True に変わったときにだけ実行されます。False の時の条件タイプが割り当てられているスクリプトは、条件が True から False に変わったときにだけ実行されます。

条件スクリプトを設定する

1. ［**スクリプト**］ペインで、［**条件**］を右クリックして、［**新規**］を選択します。

   ［**実行条件**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



- 既存の条件スクリプトを編集するには、［**条件**］の横にあるプラス記号を選択し、条件スクリプト名を右クリックして、［**編集**］を選択します。［**条件スクリプトの編集**］ダイアログ ボックスが表示されます。

2. ［**条件**］ボックスに、条件として使用する式を入力します。

   入力できる式の最大の長さは 1024 文字です。

3. ［**コメント**］ボックスにコメントを入力することもできます。
4. ［**実行条件**］リストで、以下のいずれかを実行します。
   - 条件が False になったときに一度実行するようにスクリプトを設定するには、［**False の時**］を選択します。
   - 条件が False の間に定期的に実行するようにスクリプトを設定するには、［**False の間**］を選択します。
   - 条件が True になったときに一度実行するようにスクリプトを設定するには、［**True の時**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 条件が True の間に定期的に実行するようにスクリプトを設定するには、［**True の間**］を選択します。

5. 前の手順で［**False の間**］または［**True の間**］を選択した場合、［**実行周期**］ボックスに 1 ～ 360000 ミリ秒の間隔を入力します。

   注意：条件が True でなくなった場合、条件付き WindowViewer タイマーは停止します。たとえば、マウス ボタンが離された場合、While Mouse Button Down イベントはトリガされません。キーが離された場合、キー スクリプトは停止します。

6. スクリプトを入力するか、ウィンドウで既存のスクリプトを変更します。
7. ［**OK**］を選択します。

条件が関連付けられているすべての条件スクリプトを削除する

- ［**スクリプト**］ペインで［**条件**］を展開し、条件スクリプト名を右クリックして、［**削除**］を選択します。メッセージが表示されたら、［**はい**］を選択します。

条件が関連付けられている個々の条件スクリプトを削除する

1. ［**スクリプト**］ペインで［**条件**］を展開し、キー スクリプト名を右クリックして、［**編集**］を選択します。［**条件スクリプトの編集**］ダイアログ ボックスが表示されます。
2. ［**実行条件**］リストで、削除するスクリプトのスクリプト トリガを選択します。スクリプトが、［**条件スクリプトの編集**］ダイアログ ボックスのメイン セクションに表示されます。
3. ［**編集**］メニューの［**クリア**］を選択します。スクリプトがメイン セクションからクリアされ、関連付けられたスクリプトが削除されます。

データ変化スクリプトの設定

データ変化スクリプトを使用すると、特定のタグ名またはドット フィールドの変化が定義済みのデッド バンドを超えた場合にスクリプトを一度実行できます。

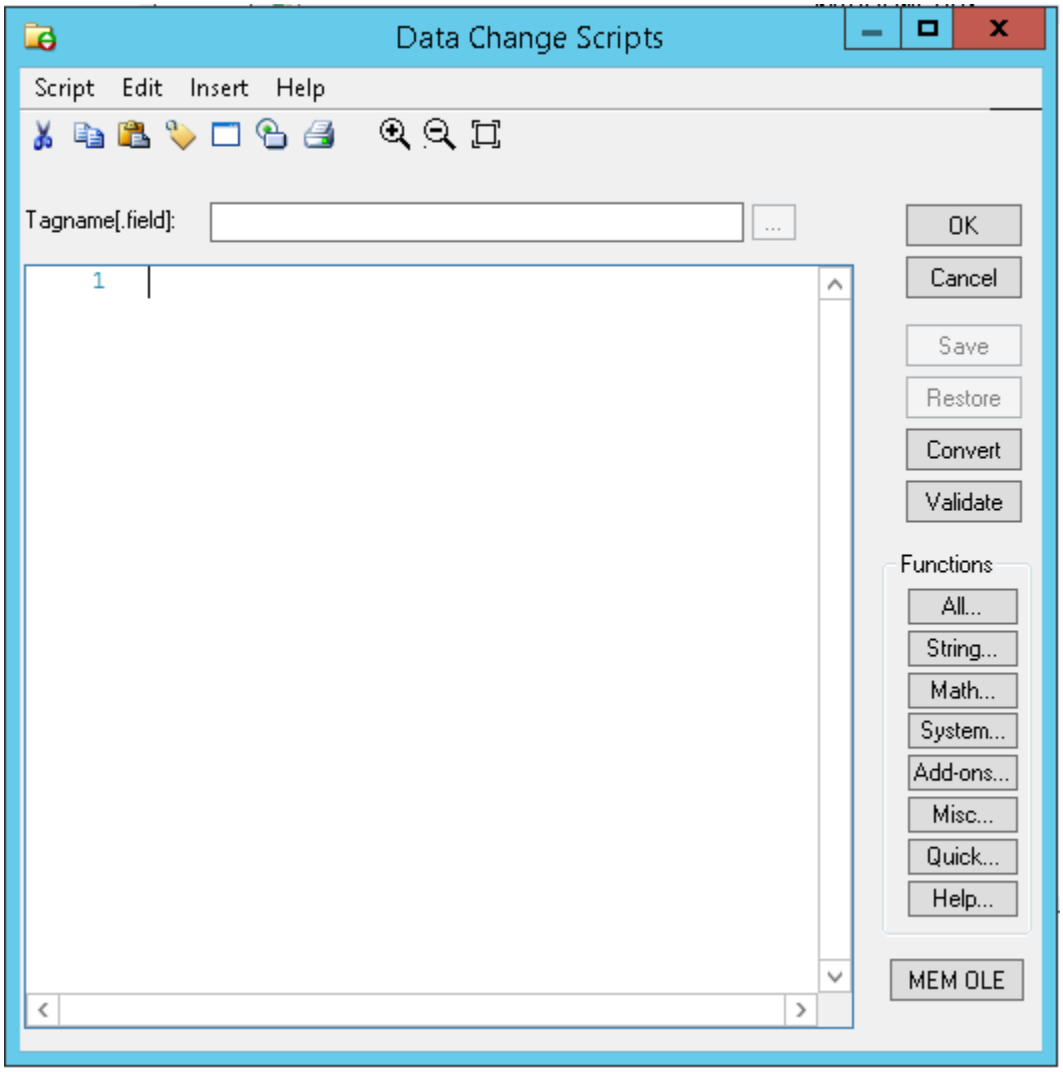
データ変化スクリプトは、スクリプトを開始するタグ名またはタグ名フィールドで識別されます（Tag1 または Tag2.HiHiLimit など）。

データ変化スクリプトを設定する

1. ［**スクリプト**］ペインで、［**データ変化**］を右クリックして、［**新規**］を選択します。

   ［**データ変化スクリプト**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. 新しいスクリプトを作成するには、［**タグ名［.フィールド]**］ボックスに、タグ名またはタグ名フィールドを入力します。

   既存のスクリプトを編集するには、［**タグ名［.フィールド]**］ボックスの右側にある参照ボタンを選択して、表示されるリストからスクリプトを選択します。

3. スクリプトをウィンドウに入力します。
4. ［**OK**］を選択します。

データ変化スクリプトを削除する

- ［**スクリプト**］ペインで［**データ変化**］を展開し、データ変化スクリプト名を右クリックして、［**削除**］を選択します。メッセージが表示されたら、［**はい**］を選択します。

アクション スクリプトの設定

アクション スクリプトを使用すると、オペレータ アクションとグラフィック オブジェクトを関連付けることができます。以下の 1 つまたは複数のイベントをグラフィック オブジェクトで設定できます。

- マウスの左、中央、右のボタンをクリックする

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

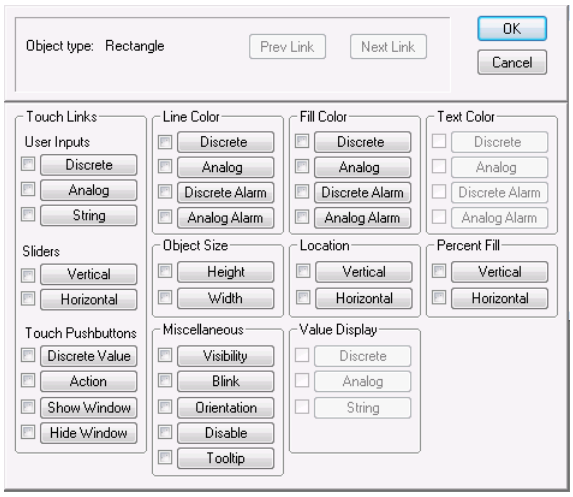
- マウスの左、中央、右ボタンをクリックしたままの状態にする
- 左、中央、右ボタンを離す
- 左、中央、右ボタンをダブルクリックする
- キーおよびキーの組み合わせを押す
- キーおよびキーの組み合わせを押したままの状態にする
- キーおよびキーの組み合わせを離す
- マウス ポインタをオブジェクト上に移動させる

アクション スクリプトを設定できるのは、オブジェクト自体の［**アニメーション リンクの選択**］パネルだけです。

重要：アクション スクリプトと同じキーまたはキーの組み合わせによりトリガされるキー スクリプトが存在する場合、アクション スクリプトが実行され、キー スクリプトは無視されます。

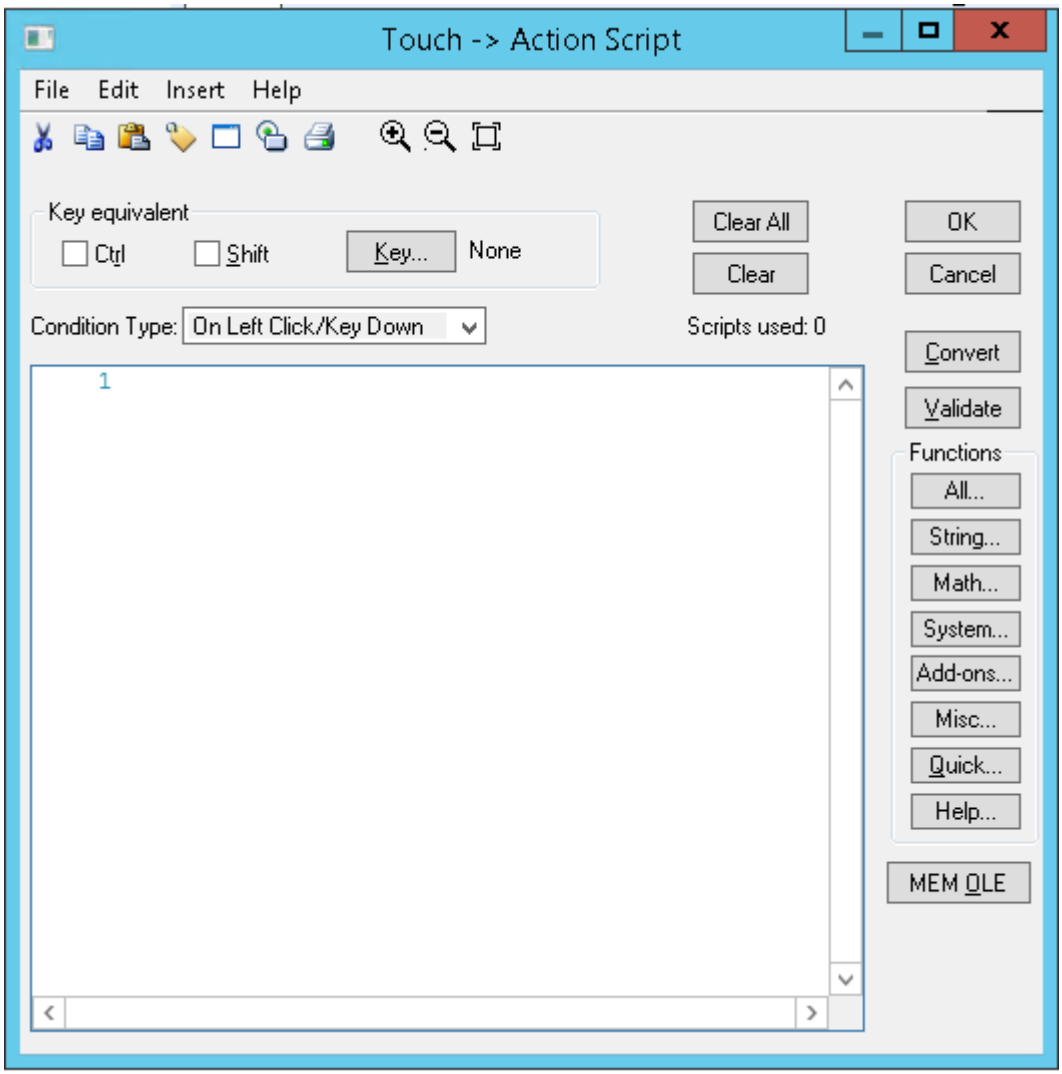
アクション スクリプトを設定する

1. グラフィック オブジェクトをダブルクリックします。**アニメーション リンクの選択**パネルが表示されます。



2. ［**アクション**］を選択します。［**タッチ -> アクション スクリプト**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［実行条件］リストで、以下のいずれかを選択します。

設定するスクリプトの実行条件	選択
左マウス ボタンまたは特定のキーやキーの組み合わせが押されたときに一度実行する	左クリック／ホットキーを押したとき
左マウス ボタンまたは特定のキーやキーの組み合わせが押されたときに定期的に実行する	左クリック／ホットキーを押している間
左マウス ボタンまたは特定のキーやキーの組み合わせが離されたときに一度実行する	左クリック／ホットキーを離したとき
左マウス ボタンがダブルクリックされたときに一度実行する	左ダブルクリック時
右マウス ボタンが押されたときに一度実行する	右クリック時
右マウス ボタンが押されている間に定期的に実行する	右クリックを押している間

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

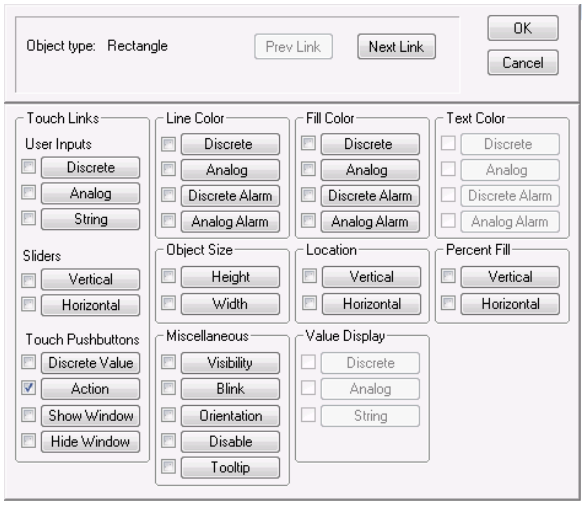
設定するスクリプトの実行条件	選択
右マウス ボタンが離されたときに一度実行する	**右クリックを離した時**
右マウス ボタンがダブルクリックされたときに一度実行する	**右ダブルクリック時**
中央マウス ボタンが押されたときに一度実行する	**中クリック時**
中央マウス ボタンが押されている間に定期的に実行する	**中クリックを押している間**
中央マウス ボタンが離されたときに一度実行する	**中クリックを離した時**
中央マウス ボタンがダブルクリックされたときに一度実行する	**中ダブルクリック時**
マウスがオブジェクト上を移動したときに一度実行する	**マウス ホバー時**

4. ［**左クリック時／ホットキーを押したとき**］、［**左クリック／ホットキーを押している間**］または［**左クリック／ホットキーを離したとき**］を選択した場合、以下を実行します。
   a. ［**キー**］ を選択します。 ［**キーの選択**］ダイアログ ボックスが表示されます。
   b. キーを選択します。
   c. ［**Ctrl**］または［**Shift**］、あるいはこれら両方のチェック ボックスをオンにして、コントロール キーやシフト キーの組み合わせを選択したキーに割り当てます。
5. ［**左／押している間**］または［**右を押している間**］を選択した場合、 ［**実行周期**］ボックスに 1 ～ 360000 ミリ秒の間隔を入力します。
6. ［**マウス ホバー時**］を選択した場合、［**後**］ボックスに、1 ～ 360000 ミリ秒の間隔を入力します。これは、マウスがオブジェクトを通過してから、スクリプトが実行されるまでの時間です。
7. スクリプトをウィンドウに入力します。
8. ［**OK**］を選択します。

InTouch グラフィック オブジェクトに関連付けられているすべてのアクション スクリプトを削除する

1. グラフィック オブジェクトをダブルクリックします。オブジェクト プロパティ パネルが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［**アクション**］チェック ボックスをオフにします。アクション スクリプトは、ランタイムでは実行されなくなります。［**アクション**］ボタンをクリックすると、エディタが開き、前回オブジェクトに対して保存したアクション スクリプトが表示されます。

個々のアクション スクリプトを削除する

1. 削除するアクション スクリプトを持つグラフィック オブジェクトをダブルクリックします。オブジェクト プロパティ パネルが表示されます。
2. **アクション**選択ボタン［**タッチ -> アクション スクリプト**］ダイアログ ボックスが表示されます。
3. ［**実行条件**］リストで、スクリプト トリガを選択します。
4. ［**編集**］メニューの［**クリア**］を選択します。スクリプトがメイン セクションからクリアされ、関連付けられたスクリプトが削除されます。

ActiveX イベント スクリプトの設定

ActiveX イベント スクリプトを使用すると、ActiveX イベントが発生したときにスクリプトを実行できます。ActiveX コントロールにより、以下のイベントが含まれます。

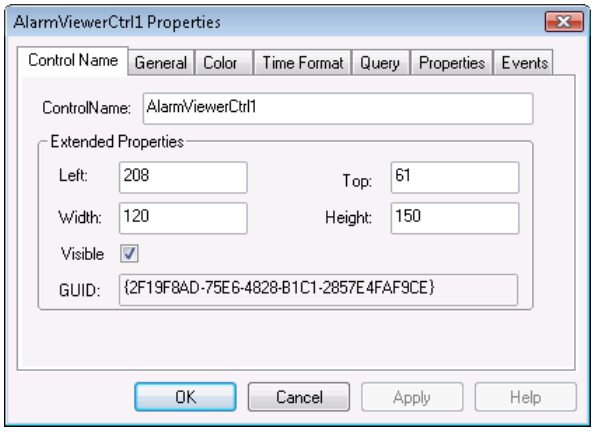
- ActiveX コントロールが起動した：Startup
- ActiveX コントロールが閉じられた：Shutdown
- ユーザーが ActiveX コントロールを選択した：Click
- ユーザーが ActiveX コントロールをダブルクリックした：Doubleclick

ActiveX イベント スクリプトは名前で識別されます。デフォルトでは、InTouch HMI は、コントロール名と、スクリプトが関連付けられているイベントを自動的に追加します（MyActiveXScript (AlarmViewerCtrl1::Click) など）。

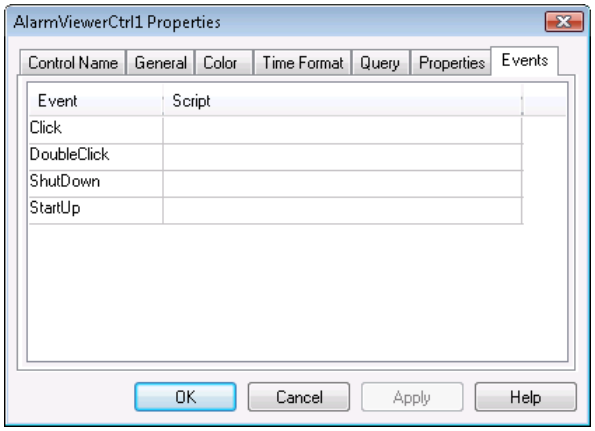
新しい **ActiveX** イベント スクリプトを設定する

1. 設定する ActiveX コントロールをダブルクリックします。ActiveX コントロール プロパティ ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



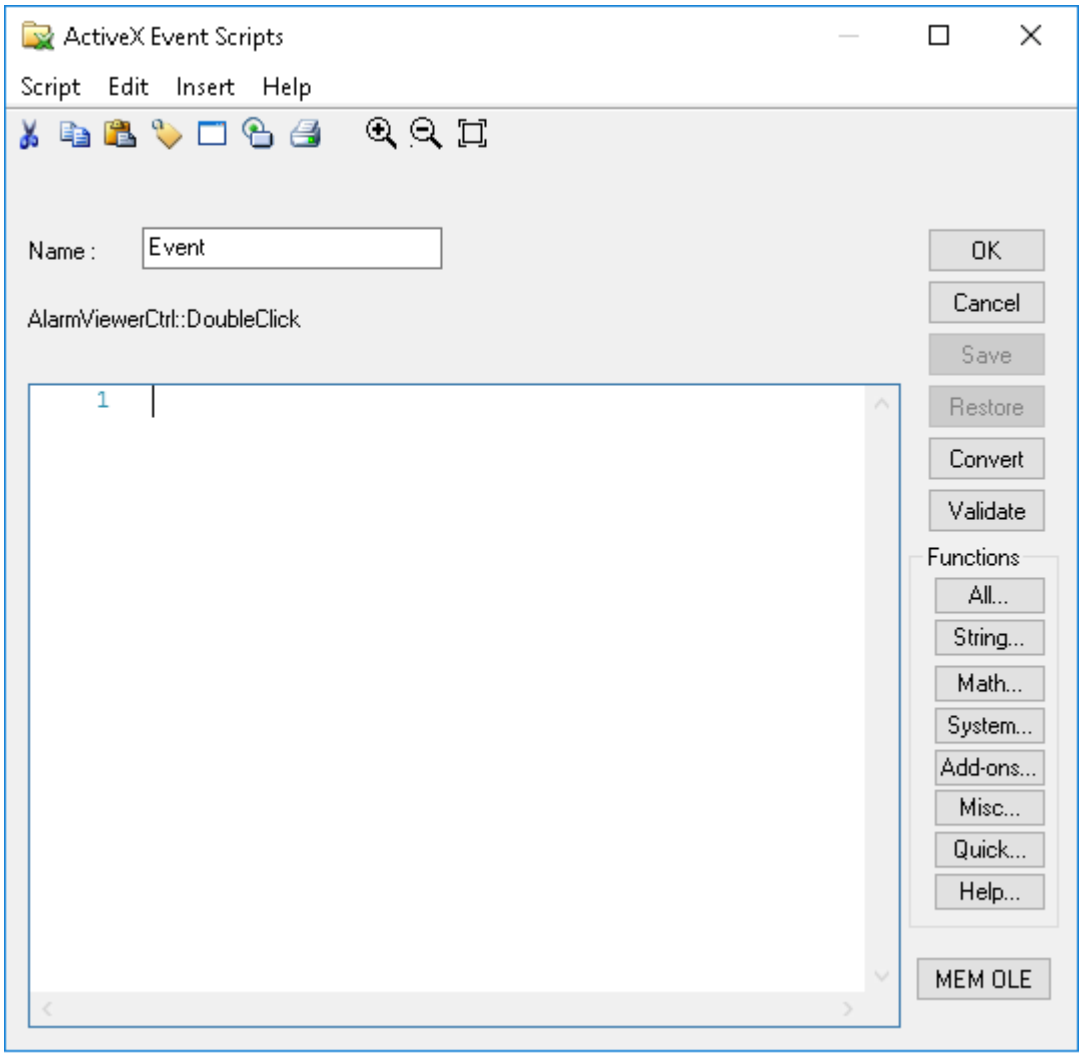
2. ［イベント］タブを選択します。



3. クリック、ダブルクリック、終了、または起動などのイベントを選択します。
4. イベントの［**スクリプト**］セルを選択します。角括弧が表示されます。
5. イベント スクリプトの新しい名前を入力して、［**OK**］を選択します。メッセージが表示されたら、［**OK**］を選択して、新しいスクリプトを作成します。

   ［**ActiveX イベント スクリプト**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



1. ［**スクリプト名**］ボックスで、ActiveX イベント スクリプト名を変更できます。
2. スクリプトをウィンドウに入力します。
3. ［**OK**］を選択します。

既存の **ActiveX** イベント スクリプトを編集する

1. ［**スクリプト**］ペインで［**ActiveX イベント**］を展開し、ActiveX スクリプト名を右クリックして、［**編集**］を選択します。［**ActiveX イベント スクリプト**］ダイアログ ボックスが表示されます。
2. スクリプトを変更したら、［**OK**］を選択します。

既存の **ActiveX** イベント スクリプトを削除する

1. ActiveX コントロールが、削除する ActiveX イベント スクリプトを使用していないことを確認します。スクリプトを使用している ActiveX コントロールが存在する場合、まず以下の手順を実行します。
   a. スクリプトを使用しているすべての ActiveX コントロールの［**イベント**］パネルで、ActiveX イベント スクリプト参照を削除します。
   b. すべてのウィンドウを閉じ、使用タグの検索を更新します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［スクリプト］ペインで［**ActiveX イベント**］を展開し、ActiveX スクリプト名を右クリックして、［**削除**］を選択します。メッセージが表示されたら、［**はい**］を選択します。ActiveX イベント スクリプトが削除されます。

ランタイムでのスクリプト実行の一時停止

デフォルトでは、WindowViewer が起動されると、ロジックが実行され、同期スクリプトが実行されます。スクリプトの実行は、ロジックを中断することによってランタイムで一時停止できます。一時停止した後、スクリプト実行は再開できます。

メニューからランタイムでスクリプト実行を一時停止する

- ［**ロジック**］メニューで、［**ロジック ストップ**］を選択します。同期スクリプトは、実行を停止します。非同期スクリプトは、引き続き実行されますが、新しい非同期スクリプトは起動されません。

スクリプトでランタイムにスクリプト実行を一時停止する

- 値 0 を論理型システム タグ変数 $LogicRunning に書き込みます。同期スクリプトは、実行を停止します。非同期スクリプトは、引き続き実行されますが、新しい非同期スクリプトは起動されません。

ランタイムでスクリプト実行を再開する

- ［**ロジック**］メニューで、［**ロジック スタート**］を選択します。スクリプト実行が再開されます。

スクリプトを使用してランタイムでスクリプト実行を再開する

- 値 1 を論理型システム タグ変数 $LogicRunning に書き込みます。$LogicRunning システム タグ変数は、ロジックが一時停止されたときに実行している非同期スクリプトに含める必要があります。

$LogicRunning システム タグ変数

このシステム タグ変数は、スクリプト実行の監視や制御を行います。

使用法

$LogicRunning

備考

値を 1 に設定すると、スクリプト実行が開始します。値を 0 に設定すると、スクリプト実行が停止します。

このシステム タグ変数は、WindowViewer の［**ロジック**］メニューで、［**ロジック スタート**］または［**ロジック ストップ**］を選択する操作と同じ機能を果たします。

現在実行している非同期スクリプトを停止することはできません。ただし、新しいスクリプトが実行されることは回避できます。

データタイプ

論理型（読み取り／書き込み）

スクリプト言語

InTouch HMI スクリプト言語を使用したスクリプトの作成には、以下の概念、技術、および構文規則を使用します。

- 基本構文規則。「基本構文規則」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 定義済み関数またはカスタム関数の呼び出し。「標準関数の呼び出し」および「カスタム関数（クイック関数）の呼び出し」を参照してください。
- 値の割り当ておよびさまざまな演算子の使用。「値の割り当てと演算子」を参照してください。
- 条件付きステートメントの使用。「条件付きプログラム分岐構造の使用」を参照してください。
- ループの使用。「プログラム ループの使用」を参照してください。
- ローカル変数の使用。「ローカル変数の使用」を参照してください。

スクリプト エディタの一般的な操作の詳細については、「スクリプトの作成と編集」を参照してください。

スクリプト トリガのさまざまなタイプの詳細については、「スクリプト トリガ」を参照してください。

標準スクリプト関数の参照については、「組み込み関数」を参照してください。

基本構文規則

基本構文規則では、InTouch HMI スクリプト言語に関する以下の側面について示しています。

- サブルーチン
- ステートメント
- インデント
- コメント
- タグ リファレンス
- リテラル データ値
- 値式
- 構文確認

サブルーチン

Visual Basic の「Sub」プロシージャなど、同じスクリプト内で個別のサブルーチンを使用するというコンセプトはありません。スクリプトを複数のサブルーチンで構成するには、各サブルーチンにカスタム クイック関数を作成する必要があります。「カスタム スクリプト関数」を参照してください。

ステートメント

- ステートメントには、値の割り当て、関数呼び出し、または制御構造を使用できます。
- スクリプトの各ステートメントは、セミコロン（;）で終了する必要があります。
- 各ステートメントがセミコロンで終了している限り、同じ行で複数のステートメントを使用できます。
- 改行（Enter キーを押す）を使用することで、1 つのステートメントを複数の行に分割できます。

インデント

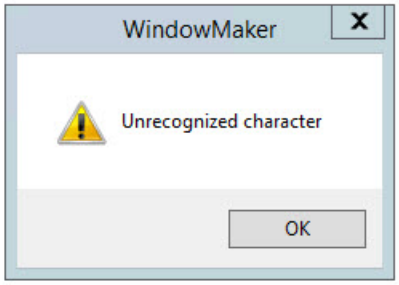
スクリプト コードは、任意の方法でインデントできます。インデントには、機能的な関連性はありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

コメント

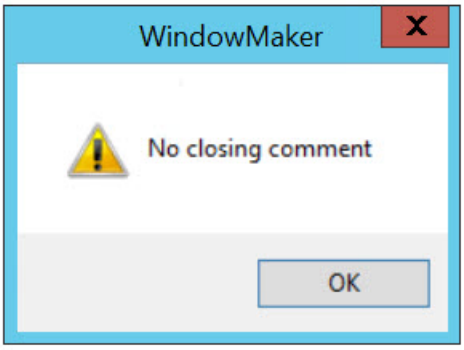
テキストをコメントにするには、括弧 { } で囲みます。コメントが複数の行に及んでも構いません。ネストされたコメントはサポートされません。たとえば、次の例では、最初の } はコメントを閉じ、2 番目の } に対してエラーが表示されます。

```
1  DIM msg AS Message;
2  {
3
4  Comments across multiple lines
5
6  {
7  Nested comments
8  }
9  }
10 msg = $DateString + ":" + $Operator;
```



} がない場合、検証中にエラーが表示されます。

```
1  DIM msg AS Message;
2  {
3
4  Comments across multiple lines
5
6
7  Nested comments
8
9  msg = $DateString + ":" + $Operator;
```



タグ リファレンス

タグ リファレンスはいくつかの方法で作成できます。

- ローカル タグ変数ディクショナリで定義されているタグ変数を参照するには、簡単にタグ変数名を使用します。
- 特定のドット フィールドを参照するには、通常の参照形式（`Tagname.Dotfield`）を使用します。
- リモート ノードのデータ アイテムを参照するには、通常のリモート タグ リファレンス（`AccessName:Item`）を使用します。
- また、範囲が現在のスクリプトに制限されるローカル変数を定義することもできます。「ローカル変数の使用」を参照してください。

リテラル データ値

- 整数値を 10 進数または 16 進数で指定できます。たとえば、`255` または `0xFF` です。
- 浮動小数点値を 10 進数または 16 進数で指定できます。たとえば、`0.001` または `1E-3` です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ブール値を指定するには、FALSE の場合は数値 0、TRUE の場合は数値 1 を使用します。
- 文字列値を指定するには、二重引用符で囲みます。例："これは文字列です"

値式

値式には、リテラル値、タグ リファレンス、および関数呼び出し、または適切な演算子によりリンクされたこれらのすべての組み合わせを含めることができます。「値の割り当てと演算子」を参照してください。

構文確認

スクリプトを保存すると、その構文が正しいかどうかがスクリプト エディタによって自動的に検証されます。［**確認**］ボタンを選択すると、手動でこの確認を開始できます。「正確な構文を確認するためのスクリプトの検証」を参照してください。

標準関数の呼び出し

標準関数は、InTouch HMI で事前に定義されています。「カスタム関数（クイック関数）の呼び出し」を参照してください。

標準関数の呼び出しの構文

事前に定義されたスクリプト関数を呼び出す構文は、関数が結果を返すかどか、およびどのように返すかにより異なります。

関数には、結果を返さない関数や、タグ変数に割り当てたり、式で使用できるオプションの結果を返す関数、またタグ変数に割り当てたり、式で使用する必要がある結果を返す関数があります。

関数タイプを判断するには、関数の説明を参照してください。それぞれの関数の説明には、関数が結果を返すかどうか、結果がオプションかどうかを示す構文リストがあります。

結果を返さない関数を呼び出す

- 関数名のみ（および存在する場合はパラメータ）をステートメントで使用します。以下に例を示します。
  ```
  FunctionName(Parameters);
  ```

結果を割り当てる必要がある関数を呼び出す

- リテラル値または関連するデータ タイプのタグ変数を使用できるスクリプト内のいずれかの場所で、関数名（および存在する場合はパラメータ）を使用します。たとえば、値割り当てでは、以下のようになります。
  ```
  ResultsTagname = FunctionName(Parameters);
  ```
  または、別の標準関数のパラメータとして使用する、入れ子の関数呼び出しでは以下のようになります。
  ```
  OtherStandardFunction(FunctionName(Parameters));
  ```

オプションの結果を返す関数を呼び出す

- 前の手順のいずれかを使用します。

パラメータを関数に渡す

標準の事前定義関数へのパラメータは通常、*値*により渡されます。つまり、式がパラメータに対して必要なデータ タイプを評価する限り、有効な式をパラメータとして渡すことができます。このような式には、リテラル値、タグ リファレンス、関数呼び出し、または適切な演算子によりリンクされたこれらの

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

すべての組み合わせを含めることができます。式および演算子の詳細については、「値の割り当てと演算子」を参照してください。

スクリプトが関数を呼び出すと、式が評価され、結果の値が関数に渡されます。

ただし、関数によっては、タグ *リファレンス*をパラメータとして必要とする場合があります。以下に例を示します。

```
RecipeSelectRecipe(Filename, RecipeName, Number);
```

この例では、RecipeName パラメータはタグ リファレンスでなければなりません（つまり、RecipeName パラメータのリテラル パラメータを使用する必要があります）。式が有効なタグ変数を評価する場合でも、文字列式を代わりに渡すことはできません。

注意：パラメータが 1 つだけの従来の事前定義関数の一部は（たとえば、Ack() 関数）、括弧内にパラメータを渡すという標準構文には従いません。その代わり、パラメータは、スペースにより関数名と区別されます。特定の関数に関して不明な場合は、関数のマニュアル内の構文説明を確認してください。

カスタム関数（クイック関数）の呼び出し

カスタム クイック関数の呼び出しは、事前定義された標準関数の呼び出しとは若干異なります。

- キーワード CALL をクイック関数名の前に付ける必要があります。
- クイック関数により返される結果は、常にオプションです。これらを使用することはできますが、使用しなくても構いません。

結果を返さないクイック関数を呼び出す

- キーワード CALL を前に付けた関数名（存在する場合はパラメータ）をステートメントで使用します。以下に例を示します。

  ```
  CALL QuickFunctionName(Parameters);
  ```

結果を返すクイック関数を呼び出す

- 以下のいずれかを実行します。
  - 結果を返さなかったものとしてクイック関数を呼び出す（前述の手順を参照してください）
  - リテラル値または関連するデータ タイプのタグ変数を使用できるスクリプトのいずれかの場所で、キーワード CALL を前に付けた関数名（存在する場合はパラメータ）を使用します。たとえば、値割り当てでは、以下のようになります。

    ```
    ResultsTagname = CALL QuickFunctionName(Parameters);
    ```

    または、標準関数のパラメータとして使用する、入れ子の関数呼び出しでは以下のようになります。

    ```
    OtherStandardFunction(CALL FunctionName(Parameters));
    ```

注意：クイック関数が別のクイック関数のパラメータとして使用されるように、クイック関数呼び出しを入れ子にすることはできません。たとえば、`Call QF1(Call QF2());` は有効なステートメントではありません。

パラメータをクイック関数に渡す

クイック関数へのパラメータは常に、値によって渡されます。リファレンスによって、パラメータをクイック関数に渡すことはできません。

式がパラメータに対して必要なデータ タイプを評価する限り、有効な式をパラメータとして渡すことができます。このような式には、リテラル値、タグ リファレンス、関数呼び出し、または適切な演算子に

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

よりリンクされたこれらのすべての組み合わせを含めることができます。式および演算子の詳細については、「値の割り当てと演算子」を参照してください。スクリプトが関数を呼び出すと、式が評価され、結果の値が関数に渡されます。

注意：クイック関数が別のクイック関数のパラメータとして使用されるように、クイック関数呼び出しを入れ子にすることはできません。たとえば、`CALL QF1(CALL QF2());` は有効なステートメントではありません。

値の割り当てと演算子

スクリプトでは、値の割り当てを使用して、値をタグ変数に書き込むことができます。値の割り当ての構文は以下のとおりです。

```
Tagname = ValueExpression;
```

このステートメントが実行されると、`ValueExpression` が、`Tagname` により参照されるタグ変数に書き込まれます。`ValueExpression` には、データ タイプがタグ変数データ タイプに一致する任意の有効な式を使用できます。値式には、リテラル値、タグ リファレンス、および関数呼び出し、または適切な演算子によりリンクされたこれらのすべての組み合わせを含めることができます。

「サポートされている演算子」を参照してください。

「演算子の評価順の設定」を参照してください。

「有効な式と無効な式の例」を参照してください。

サポートされている演算子

以下の表では、サポートされているすべての演算子を一覧表示しています。特定の演算子の使用の詳細については、関連するセクションを参照してください。

オペレータ	詳細
+、	加算または連結：+
-、	減算：-
*	乗算：*
/、	除算：/
**	累乗：**
MOD	剰余：MOD
~	補数：~
SHLSHL	左方向桁移動：SHL、および右方向桁移動：SHR
SHRSHR	左方向桁移動：SHL、および右方向桁移動：SHR
&	ビット単位の論理積：&
|	ビット単位の論理和：|
^	ビット単位の排他的論理和：^
AND	論理積：AND

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オペレータ	詳細
または	論理和：OR
NOT	論理否定：NOT
<、	比較：<、>、<=、>=、==、<>
>、	比較：<、>、<=、>=、==、<>
<=	比較：<、>、<=、>=、==、<>
>=	比較：<、>、<=、>=、==、<>
==	比較：<、>、<=、>=、==、<>
<>	比較：<、>、<=、>=、==、<>

注意：数値計算では、計算結果が実数値の範囲内となるように、常に演算数を選択してください。このようにしないと、結果が正しいものとなりません。

加算または連結：+

2 つの数値演算数を加算するか、または 2 つの文字列演算数を連結します。

有効な演算数

加算の場合：任意の整数値または実数値

連結の場合：任意のメッセージ値

戻り値のデータ タイプ

加算の場合：整数または実数

連結の場合：メッセージ

例

```
MessageTag = "Setpoint value: " + Text(SetpointTag, "#.##");
```

減算：-

2 つの演算数と共に使用すると、通常の数値減算が実行されます。

有効な演算数

任意の整数値または実数値

戻り値のデータ タイプ

整数型または実数型

例

TemperatureSetpoint が 70 の場合、この例のスクリプトを実行した後の TemperatureSetpoint は 65 になります。

```
TemperatureSetpoint = TemperatureSetpoint - 5;
```

乗算：*

通常の数値乗算。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効な演算数

任意の整数値または実数値

戻り値のデータ タイプ

整数型または実数型

除算：/

通常の数値除算。ランタイムに 0 で除算しようとすると、結果として 0 が返されます。

有効な演算数

任意の整数値または実数値

戻り値のデータ タイプ

整数型または実数型

累乗：**

左側の演算数（基数）を右側の演算数（べき数）で累乗します。

有効な演算数

整数値または実数値。基数 0 と負のべき数、または負の奇数と分数のべき数を組み合わせて使用することはできません。このような場合、結果として 0 が返されます。

戻り値のデータ タイプ

整数型または実数型

例

`8 ** (1/3)` は 2 を返します（8 の立方根）

剰余：MOD

2 つの整数値による除算の余りを返します。

有効な演算数

任意の整数値

戻り値のデータ タイプ

整数

例

`37 MOD 4` は 1 を返します

補数：~

整数値の 1 の補数を返します。つまり、各 0 ビットと 1 ビット（またはその逆）を変換します。

有効な演算数

任意の整数値

戻り値のデータ タイプ

整数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

左方向桁移動：SHL、および右方向桁移動：SHR

整数値のバイナリ表現を指定ビット位置の数まで右または左に移動します。左側の演算数は移動される値で、右側の演算数はビット位置の数です。桁あふれしたビットは切り捨てます。桁移動によって空いたビット位置は、ゼロに設定されます。

有効な演算数

任意の整数値

戻り値のデータ タイプ

整数

例

`IntTag = IntTag SHL 1;` の結果は、初期タグ変数値 5 に対して繰り返し実行された場合、以下のようになります。

反復回数	バイナリ パターン	タグ変数値
初期値	`0[...]00000101`	5
実行 1	`0[...]00001010`	10
実行 2	`0[...]00010100`	20

ビット単位の論理積：&

2 つの整数のバイナリ表現をビット単位で比較し、以下の表に示すように結果を返します。

最初の演算数のビット	2 つ目の演算数のビット	結果のビット
0	0	0
0	1	0
1	0	0
1	1	1

この演算子を使用すると、ビット パターンの特定の部分を素早く「マスク」（0 に設定）できます。たとえば、以下のステートメントは、`IntTag` タグ変数の上位 24 ビットをマスクします。

```
IntTag = IntTag & 255;
```

表に示すように、演算数ビットのいずれかが 0 の場合、結果のビットは常に 0 となります。255 のバイナリ表現では、下位 8 ビットだけが 1 なので、残りの 24 の 0 ビットにより、結果の対応するすべてのビットが 0 に設定されます。

有効な演算数

任意の整数値

戻り値のデータ タイプ

整数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ビット単位の論理和：|

2 つの整数のバイナリ表現をビット単位で比較し、以下の表に示すように結果を返します。

最初の演算数のビット	2 つ目の演算数のビット	結果のビット
0	0	0
0	1	1
1	0	1
1	1	1

この演算は、「包含的論理和」とも呼ばれます。

有効な演算数

任意の整数値

戻り値のデータ タイプ

整数

ビット単位の排他的論理和：^

2 つの整数のバイナリ表現をビット単位で比較し、以下の表に示すように結果を返します。

最初の演算数のビット	2 つ目の演算数のビット	結果のビット
0	0	0
0	1	1
1	0	1
1	1	0

この演算は、「排他的論理和」とも呼ばれます。

有効な演算数

任意の整数値

戻り値のデータ タイプ

整数

論理積：AND

両方の論理演算数が TRUE の場合 TRUE を返し、そうでない場合 FALSE を返します。この演算子の真理値表を以下に示します。

p	q	p AND q
氏	氏	氏
氏	T	氏

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

p	q	p AND q
T	氏	氏
T	T	T

有効な演算数

任意の論理値

戻り値のデータ タイプ

論理型

論理和：OR

少なくとも一方の論理演算数が TRUE の場合 TRUE を返し、そうでない場合 FALSE を返します。この演算子の真理値表を以下に示します。

p	q	p OR q
氏	氏	氏
氏	T	T
T	氏	T
T	T	T

有効な演算数

任意の論理値

戻り値のデータ タイプ

論理型

論理否定：NOT

論理値が FALSE の場合 TRUE を返し、TRUE の場合 FALSE を返します。この演算子の真理値表を以下に示します。

p	NOT p
氏	T
T	氏

有効な演算数

任意の論理値

戻り値のデータ タイプ

論理型

比較：<、>、<=、>=、==、<>

これらの演算子は、2 つの値を比較して、演算子により指定された条件が満たされる場合 TRUE を返します。演算数には任意のデータ タイプを使用できます。文字列演算数の場合、比較は大文字小文字の区

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

別がないアルファベット順で行われます。たとえば、b は a より大きく、c は b より大きいなどです。論理演算数の場合、TRUE は FALSE より大きいとみなされます。以下の表は、すべての比較演算子およびそれらの条件を示しています。

演算	例	条件
より小さい	a < b	a は b より小さい
より大きい	a > b	a は b より大きい
以下	a <= b	a は b 以下
以上	a >= b	a は b 以上
等しい	a == b	a は b と等しい
等しくない	a <> b	a は b と等しくない

有効な演算数

任意のデータ タイプの値（両方の値のデータ タイプは同じでなければなりません）。

戻り値のデータ タイプ

論理型

演算子の評価順の設定

任意の式で、括弧を使用することで、特定の順序で演算子を評価させることができます。これは、任意の数学計算と同じように機能します。括弧を使用しない場合、式は演算子のデフォルトの優先度規則に基づいて評価されます。優先度が最も高いレベルの演算が最初に実行され、その後で 2 番目、3 番目と続きます。

以下の表は、各演算子の優先度レベルを示しています。同じ行の演算子の優先度レベルは同じです。

-、NOT、~	最高優先度
**	
*、/、MOD	
+、-	
SHL、SHR	
<、>、<=、>=	
==、<>	
&	
^	
|	
AND	
または	
=、	最低優先度

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

暗黙的なデータ タイプ変換

InTouch HMI スクリプト言語は、特定のデータ タイプ間の割り当てで、暗黙的な値変換の機能を提供します。ただし、これにより予測しない結果が生じることがあるので、この機能を使用する場合は十分に注意してください。

以下の表は、特定のタイプの値を異なるタイプのタグ変数に割り当てた場合の結果を示しています。

予測されるデータ タイプ	使用されるデータ タイプ	備考
論理型	整数型	値 0 は FALSE と解釈されます。その他の値は TRUE と解釈されます。
論理型	実数型	値 0 は FALSE と解釈されます。その他の値は TRUE と解釈されます。
整数型	論理型	値 FALSE は 0 に変換されます。値 TRUE は 1 に変換されます。
整数型	実数型	小数点より前の値のみが使用されます。小数点以下はすべて破棄されます。
実数型	論理型	値 FALSE は 0 に変換されます。値 TRUE は 1 に変換されます。
実数型	整数型	値は変更されずに保持されます。

他のデータ タイプ間で変換するスクリプト関数の使用については、「データ タイプの変換」を参照してください。

有効な式と無効な式の例

以下の表は、いくつかの有効な式およびその式の結果と結果のデータ タイプを示しています。

式	結果のデータ タイプ	結果
`37 MOD 4`	整数型	1
`37 MOD 4 == 1`	論理型	TRUE
`NOT (37 MOD 4 == 1)`	論理型	FALSE
`InfoAppActive(InfoAppTitle("xyz")) == 1`	論理型	プロセス「xyz」が実行している場合 TRUE
`"Batch " + Text(IntTag, "000")`	メッセージ型	`IntTag` の値が 10 の場合「Batch 010」

以下の表は、いくつかの無効な式とそれらの式が無効である理由を示しています。

式	問題
`NOT (37 MOD 4)`	NOT では論理演算数が必要です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

式	問題
`NOT 37 MOD 4 == 1`	NOT は他の演算子より優先されるため、InTouch HMI は比較の論理結果ではなく、整数値 37 に NOT を適用しようとします。
`"Batch " + IntTag`	+ 演算子を使用して文字列を連結する場合、両方の演算数が文字列でなければなりません。

条件付きプログラム分岐構造の使用

特定の条件の一致に基づいて、スクリプトの実行パスを動的に制御できます。InTouch HMI は、IF-THEN-ELSE 制御構造をこの目的でサポートしています。

IF-THEN-ELSE 制御構造の基本構文を以下に示します。

構文

```
IF 条件 THEN
   ... ステートメントおよび/または別の IF-THEN-ELSE 構造
[ELSE
   ... ステートメントおよび/または別の IF-THEN-ELSE 構造]
ENDIF;
```

IF-THEN-ELSE 構造を使用する場合、以下の規則に注意してください。

- IF-THEN-ELSE 構造は、THEN セクションおよび ELSE セクションの両方で入れ子にすることができます。
- すべての IF ステートメントには、終了を示す ENDIF ステートメントが必要です。ENDIF ステートメントは、同じ入れ子レベルで直前に入れ子となっている IF ステートメントに適用されます。
- Condition は、有効な論理式でなければなりません。THEN セクションは、Condition が TRUE の場合に実行されます。ELSE セクションは、Condition が FALSE の場合に実行されます。
- ELSE セクションはオプションです。
- 他のいくつかのプログラミング言語では、IF-THEN-ELSE 構造の同じ階層レベルの複数条件を確認でき、すべての条件が FALSE に評価された場合に実行される 1 つの汎用 ELSE セクションを使用できます（Visual Basic の If-ElseIf-Else 構造がこの例です）。ただし、InTouch HMI では行えません。確認するすべての条件に対して、新しい IF-THEN-ELSE 構造を開く必要があります。したがって、コードの 1 つのセクションをすべての条件の ELSE コードとして使用するには、最後の入れ子レベルの IF-THEN-ELSE 構造の ELSE セクションにコードを置く必要があります。

簡単な条件付き構造

以下のスクリプトは、簡単な条件付き構造を示しています。SuccessTag が TRUE の場合、「Success」ウィンドウが開き、FALSE の場合、「Failure」ウィンドウが開きます。

```
IF SuccessTag == 1 THEN
   Show "Success";
ELSE
   Show "Failure";
ENDIF;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

入れ子の条件付き構造

以下のスクリプトは、複数の条件を確認する方法、およびいずれの条件も満たされない場合に実行されるコードがある 1 つの汎用 ELSE セクションを使用する方法を示しています。

```
IF ChoiceTag == 1 THEN
   Show "Procedure 1";
ELSE
   IF ChoiceTag == 2 THEN
      Show "Procedure 2";
   ELSE
      IF ChoiceTag == 3 THEN
         Show "Procedure 3";
      ELSE
         Show "Default Procedure";
      ENDIF;
   ENDIF;
ENDIF;
```

無効なスクリプトの例（ENDIF が見つからない）

Visual Basic を使い慣れている場合、以下のように簡単な IF ステートメントを作成できます。

```
IF OpenThisWindow == 1 THEN Show "This Window";
```

これは、InTouch HMI では機能しません。すべての IF ステートメントには、終了を示す ENDIF ステートメントが必要です。

無効なスクリプトの例（間違った入れ子）

Visual Basic などの言語を使い慣れている場合、以下のように、複数の条件とデフォルトの条件を持つ条件付きステートメントを作成できます。

```
IF ChoiceTag == 1 THEN
   Show "Procedure 1";
ELSE IF ChoiceTag == 2 THEN
   Show "Procedure 2";
ELSE IF ChoiceTag == 3 THEN
   Show "Procedure 3";
ELSE
   Show "Default Procedure";
ENDIF;
```

これは、InTouch HMI では機能しません。それぞれの IF は新しい入れ子レベルを開いているため、対応する ENDIF ステートメントが必要です。この例の正しいバージョンについては、「入れ子の条件付き構造」を参照してください。

プログラム ループの使用

ループを使用すると、コードのセクションを繰り返し実行できます。InTouch HMI は、FOR ループのみをサポートしています。FOR ループは、ループの繰り返しごとに増加または減少する数値ループ変数の値を監視することによって機能します。ループは、ループ変数の値が固定制限に達するまで実行されます。

構文

```
FOR LoopTag = StartExpression TO EndExpression [STEP ChangeExpression]
... ステートメントまたは別の FOR ループ ...
NEXT;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- StartExpression、EndExpression、および ChangeExpression は、合わせて繰り返しの回数を定義します。
- StartExpression は、ループ範囲の開始値を設定します。EndExpression は、ループ範囲の終了値を設定します。
- STEP ChangeExpression はオプションで、各ループ繰り返し中にループ タグ変数を増加または減少させる値を設定します。この値を指定しない場合、デフォルト値 1 が使用されます。

FOR ループを実行すると、InTouch HMI は、以下の処理を実行します。

1. LoopTag を StartExpression の値に設定します。
2. LoopTag が EndExpression より大きいかどうかをテストします。大きい場合、InTouch HMI は、ループを終了します（ChangeExpression が負の場合、InTouch HMI は、LoopTag が EndExpression より小さいかどうかをテストします）。
3. ループ内で、ステートメントを実行します。
4. LoopTag を ChangeExpression の値（値が指定されていない場合は 1）だけ増加します。
5. 手順 2 ～ 4 を繰り返します。

FOR ループを使用する場合、以下の規則に注意してください。

- FOR ループは入れ子にすることができます。入れ子レベルの最大数は、使用できるメモリおよびシステム リソースにより異なります。
- すべての FOR ステートメントには、終了を示す NEXT ステートメントが必要です。NEXT ステートメントは、同じ入れ子レベルで直前に入れ子となっている FOR ステートメントに適用されます。
- LoopTag は、数値タグ変数（またはローカル変数）でなければなりません。
- StartExpression、EndExpression、および ChangeExpression は、数値の結果として評価される有効な式でなければなりません。
- ChangeExpression が正の場合、EndExpression は、StartExpression より大きくなければなりません。ChangeExpression が負の場合、StartExpression は、EndExpression より大きくなければなりません。そうでない場合、ループは開始されません。
- ループを終了するには、EXIT FOR ステートメントを使用します。詳細については、「ループの強制終了」を参照してください。
- ループには時間制限があります。「ループ実行の時間制限」を参照してください。

警告：ループ実行は、他のランタイム プロセスに影響を与えます。詳細については、「他のランタイム プロセスへのループの効果」を参照してください。.

ループの強制終了

ループは、以下のステートメントを呼び出すことによって、いつでも終了できます。

```
EXIT FOR;
```

このステートメントを使用すると、ループ NEXT ステートメントの直後のステートメントでスクリプト実行が続行されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のコード部分では、ループを使用して多くのダミー レコードをデータベース テーブルに挿入します。レコード挿入時にエラーが発生した場合、さらにエラーが発生しないように、ループが中断されます。

```
FOR Counter = 1 TO 1000
   ResultCode = SQLInsert(ConnectionID, "BatchDetails", "BindList1");
   IF ResultCode <> 0 THEN
      LogMessage("Error creating records! Aborting...");
      EXIT FOR;
   ENDIF;
NEXT;
```

他のランタイム プロセスへのループの効果

FOR ループの実行中、WindowViewer の他のすべてのランタイム プロセスは一時停止されます。これには以下のプロセスがあります。

- 画像の更新（アニメーション リンク、値表示、トレンドなど）。つまり、ループが完了するまでオブジェクトの動きは発生しないので、FOR ループをオブジェクト アニメーションに使用することはできません。
- I/O 通信。たとえば、FOR ループの I/O タグ変数の値を変更する場合、最後の実行後の値だけが、I/O デバイスに書き込まれます。
- 非同期クイック関数など、他のスクリプト。

非同期クイック関数で FOR ループを使用することで、他のランタイム プロセスの一時停止を回避することができます。

ループ実行の時間制限

無限ループを回避するため、FOR ループが実行を完了する必要のある時間制限があります。この時間が経過してもループ実行が完了しない場合、WindowViewer は自動的にこのループを終了して、Log Viewer に終了に関するメッセージを書き込みます。

デフォルトの時間制限は 5 秒です。この値は、以下の行をアプリケーション ディレクトリの intouch.ini ファイルに追加することで、カスタマイズできます。

```
LoopTimeout=x
```

x は秒単位での時間制限で置き換えます。

注意：時間制限は、ループの NEXT ステートメントでだけ確認されます。したがって、制限時間を超える場合でも、ループの最初の繰り返しは必ず実行されます。

ループの例

以下のスクリプトは、簡単なループおよび間接タグ変数を使用して、100 のタグ変数（Tag001 ～ Tag100）を値 0 で再初期化します。

```
DIM Counter AS INTEGER;
FOR Counter = 1 TO 100
   IndirectInteger.Name = "Tag" + Text(Counter, "000");
   IndirectInteger.Value = 0;
NEXT;
```

以下のスクリプトは、2 つの入れ子ループおよび間接タグ変数を使用して、1000 のタグ変数（Line01_Tag001 ～ Line10_Tag100）を値 0 で再初期化します。

```
DIM LineCounter AS INTEGER;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
DIM TagCounter AS INTEGER;
FOR LineCounter = 1 TO 10
   FOR TagCounter = 1 TO 100
      IndirectInteger.Name = "Line" + Text(LineCounter, "00") + "_Tag" + Text(TagCounter,
      "000");
      IndirectInteger.Value = 0;
   NEXT;
NEXT;
```

ローカル変数の使用

ローカル変数をスクリプトで宣言して、一時的な結果または中間結果を保存できます。これにより、パフォーマンスが向上し、タグ変数の数を少なく保つことができます。ローカル変数は、スクリプトのタグ変数のように使用できます。ただし、以下のようにいくつかの違いがあります。

- ローカル変数は、宣言されたスクリプトの範囲内でのみ存在します。スクリプト実行が終了すると、その値は失われます。アプリケーションの他のスクリプトから参照することはできません。
- ローカル変数にはドット フィールドがありません。
- ローカル変数は、タグ変数の数には含まれません。

ローカル変数をスクリプトで使用するには、最初に宣言する必要があります。宣言しないと、リファレンスがタグ変数とみなされます。「ローカル変数の宣言」を参照してください。

タグ変数と同じ名前を持つローカル変数を宣言できます。「ローカル変数とタグ変数の名前の競合」を参照してください。

ローカル変数の宣言

ローカル変数は、最初の使用の前に宣言している限り、スクリプトの任意の場所で宣言できます。ローカル変数を宣言するには、以下のステートメントを使用します。

```
DIM LocVarName [AS DataType];
```

`LocVarName` は、ローカル変数の名前です。この名前は、タグ変数の命名規則に従う必要があります。詳細については、「タグ名規則」を参照してください。

`DataType` は、ローカル変数のデータ タイプです。有効な値は、`Discrete`、`Integer`、`Real`、および `Message` です。このオプションを指定しない場合、`Integer` がデフォルトとして使用されます。

宣言する各ローカル変数に対して、個別の `DIM` ステートメントを使用する必要があります。

任意の数のローカル変数を宣言できます。宣言できる数は、使用できるメモリにより制限されます。

例

Integer 変数を宣言する場合:

```
DIM MyLocalIntVar AS Integer;
```

複数の Real 変数を宣言する場合:

```
DIM MyLocalRealVar1 AS Real;
DIM MyLocalRealVar2 AS Real;
```

以下のステートメントは有効 ではありません。

```
DIM MyLocalRealVar1, MyLocalRealVar2 AS Real;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ローカル変数とタグ変数の名前の競合

既存のタグ変数と名前が同じローカル変数を宣言できます。ただし、スクリプトでその名前を参照する場合、タグ変数よりローカル変数が優先されます。たとえば、「iTag」という名前の Integer タグ変数がすでに存在する場合に、以下のスクリプトを実行するとします。

```
DIM iTag as Integer;
iTag = 20;
```

この場合、値はローカル変数にのみ割り当てられます。同じ名前のタグ変数の値は変わりません。

カスタム スクリプト関数

InTouch HMI クイック関数は、他の環境ではマクロ、サブルーチン、またはプロシージャと呼ばれているスクリプトです。

クイック関数について

クイック関数は、他のスクリプトやアニメーション リンクから呼び出すことのできるスクリプトです。クイック関数の主な利点は、コードの重複が減少することです。

値はクイック関数に渡すことができ、その後クイック関数はその値を使用して、結果を返すことができます。

クイック関数は、非同期で実行できます。他のスクリプトとは異なり、メイン プログラム フローに影響を与えることなく、バックグラウンドで実行できます。非同期で実行するクイック関数は、SQL データベース呼び出しなど、時間を費やす操作に対して使用できます。

注意：クイック関数で引数を変更する場合、クイック関数を使用するすべてのスクリプトからそのクイック関数へのすべての呼び出しを削除する必要があるため、クイック関数とその引数は十分注意して計画する必要があります。変更が行われた後で、クイック関数呼び出しをスクリプトに追加する必要があります。「クイック関数の設定」の注記を参照してください。.

クイック関数には 3 つの基本部分があります。

- 名前
- 引数（オプション）
- スクリプト本文（およびオプションの戻り値）

クイック関数は、アニメーション リンクまたは別のスクリプトで CALL 関数を使用することによって実行されます。「クイック関数の呼び出し」を参照してください。

クイック関数の設定

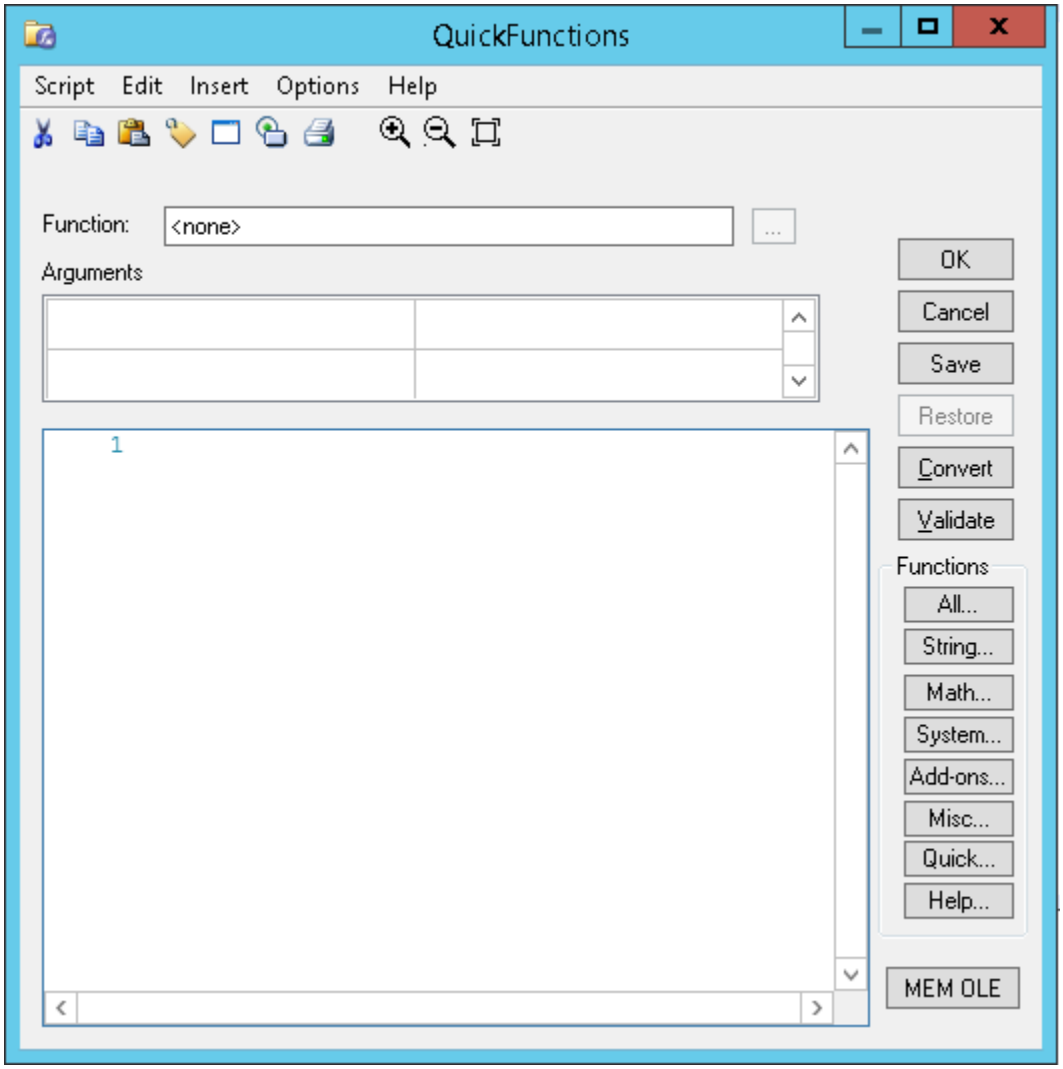
クイック関数は、作成、変更、または削除できます。

クイック関数を作成する

1. ［**スクリプト**］ペインで、［**クイック関数**］を右クリックして、［**新規**］を選択します。

   ［**クイック関数**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［**関数名**］ボックスで、クイック関数の名前を入力します。
3. ［**引数**］領域で、各引数に対して左側に名前を、右側にデータ タイプを入力します。

   引数は、それらが定義されているクイック関数内にのみ存在するローカル変数です。各クイック関数に対して、最大 16 個までの引数を持つことができます。引数名は 31 文字まで持つことができますが、スペースは使用できません。引数名は、アルファベット文字で開始する必要があります。引数名は固有である必要があります。
4. スクリプトをウィンドウに入力します。
5. クイック関数で結果を返すようにするには、スクリプトに RETURN value を追加します。

   Value には、リテラル値、ローカル変数、またはグローバル タグ名や計算式を使用できます。スクリプトは RETURN コマンドで終了し、呼び出し関数で続行します。
6. ［**OK**］を選択します。

クイック関数を変更する

1. ［**スクリプト**］ペインで、［**クイック関数**］を展開し、変更するクイック関数を右クリックして、［**開く**］を選択します。［**クイック関数の編集**］ダイアログボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. スクリプト本文を変更して、［**OK**］を選択します。

注意：InTouch アプリケーションのクイック関数への呼び出しがある場合、引数は変更できません。この場合、最初にこれらの呼び出しを削除し、すべての InTouch ウィンドウを閉じて、使用タグの検索を更新する必要があります。

クイック関数を削除する

1. クイック関数へのすべての呼び出しを削除し、すべての InTouch ウィンドウを閉じて、使用タグの検索を更新します。
2. ［**スクリプト**］ペインで、［**クイック関数**］を展開し、削除するクイック関数を右クリックして、［**削除**］をクリックします。メッセージが表示されたら、［**はい**］を選択します。

クイック関数の呼び出し

スクリプトおよびアニメーション リンクを設定して、クイック関数を呼び出し、可能な戻り値を処理または表示することができます。

クイック関数は、パラメータ値が変更されない場合、呼び出されません。$second をパラメータとして使用すると、クイック関数が少なくとも毎秒実行されます。

詳細については、「カスタム関数（クイック関数）の呼び出し」を参照してください。

非同期クイック関数の作成

クイック関数をメイン プログラム フローとは非同期に（つまり並行して）実行するように定義できます。

非同期クイック関数を作成する

1. スクリプト エディタで、クイック関数を作成します。
2. ［**オプション**］メニューで、［**非同期**］を選択します。

非同期クイック関数の制限事項

以下の内容は実行できません。

- 非同期クイック関数から値を返す
- 同じクイック関数の複数のインスタンスを同時に実行する
- 実行を開始した後で、非同期クイック関数を停止する

以下の操作を実行することは避けてください。

- 4 種類以上のクイック関数を同時に実行する。4 種類以上のクイック関数を同時に実行すると、システム パフォーマンスが著しく低下します。
- 非同期関数をアニメーション リンク（ツール ヒントなど）の式の一部として使用してください。

非同期クイック関数が実行しているかどうかの確認

IsAnyAsyncFunctionBusy() 関数を使用すると、非同期クイック関数が実行しているかどうかを確認できます。この関数は、非同期クイック関数を呼び出す QuickScript に、その他の非同期クイック関数の処理がすべて完了するまで待機させる場合に使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

IsAnyAsyncFunctionBusy() 関数

非同期クイック関数が実行しているかうかを示す論理値を返します。

構文

```
result = IsAnyAsyncFunctionBusy(timeout)
```

引数

result
非同期クイック関数が実行しているかどうかを示す論理値を返します。値の意味は以下のとおりです。

- 0 = 非同期クイック関数は実行していません。
- 1 = 非同期クイック関数は実行しています。

timeout
非同期クイック関数が実行しているかどうかを確認するまでに待機する秒単位の時間です。リテラル整数値、整数型タグ名、または整数式です。

例

非同期クイック関数を使用して複数の SQL データベースに接続する場合、それらの接続を行うために 2 分必要であることを把握していると仮定します。

最初に、非同期クイック関数を実行して SQL データベースに接続します。

次に、QuickScript で IsAnyAsyncFunctionBusy(120) 関数を使用して、クイック関数が完了する前に SQL が接続を確立するために必要な時間を十分与えます。

2 分経過しても接続できず、非同期クイック関数がまだ接続を確立しようとしている場合は、IsAnyAsyncFunctionBusy() 関数によって値 1（True）が返されます。

これで、オペレータに対して SQL の接続が正常に行われなかったことを知らせるエラー メッセージを表示できます。

以下のスクリプトは、上記の例を実装しています。

```
IF IsAnyAsyncFunctionBusy(120) == 1 THEN
   SHOW "SQL Connection Error Dialog";
ENDIF;
```

非同期クイック関数の実行の停止

開始している非同期クイック関数を停止することはできませんが、その後で実行される非同期クイック関数は、スクリプト ロジックを停止することで実行の開始を停止できます。これは、InTouch アプリケーションのすべての QuickScript に影響を与えます。

スクリプト実行の停止の詳細については、「ランタイムでのスクリプト実行の一時停止」を参照してください。

組み込み関数

InTouch QuickScript の関数を使用すると、指定した条件が満たされた場合に、コマンド操作やロジック操作を実行できます。QuickScript の関数は、それ自体で使用できます。また、特定の条件が満たされた場合に実行したり、アニメーション表示リンクで使用したりできます。事前定義関数は、機能に応じてグループに分類されています。グループを選択した後、スクリプトに挿入する事前定義関数を選択します。事前定義された数学関数のグループを選択した後に表示される、［**関数の選択**］ダイアログ ボック

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スを使用します。関数を選択すると、スクリプト内でポインタが現在ある位置に事前定義関数が配置されます。

重要：この章には、32 ビット バージョンの Windows オペレーティング システムでのみ機能するよう設計された従来の InTouch QuickScript 機能が含まれます。これらの機能は、64 ビット バージョンの Windows で実行するよう設計された InTouch QuickScript には含めないでください。この章では、これらの従来の 32 ビット バージョンの機能が識別されます。

アニメーション表示リンクでの強制更新

QuickScript をアニメーション リンクで使用する場合、このアニメーション リンクは、タグ変数が関連付けられている場合だけ更新されます。このタグ変数は、値が変更されたときにトリガとして機能します。アニメーション リンクの更新には、$Second システムタグ変数または $Minute システム タグ変数を使用することが好ましい選択です。

アニメーション表示リンクで強制更新を行う

1. アニメーション リンクをオブジェクト プロパティ ウィンドウで開きます。
2. トリガ タグ変数（たとえば、$Second）を計算に追加します。以下に例を示します。
   - アニメーション リンクが実数型または整数型の場合、$Second/$Second で式を乗算できます。
   - アニメーション リンクが文字列である場合、StringMid($TimeString, 0, 0) を式に追加できます。
   - アニメーション リンクが論理型の場合、($second.00 - $second.00) を式に追加できます。

数学計算

InTouch HMI は、以下の処理を行う関数など、スクリプトやアニメーション リンクで使用できる基本的な数学関数をサポートしています。

- 数値の切り上げ／切り捨ておよび切り捨て
- サインおよびコサインの計算
- 対数および指数の計算
- 平方根の計算

丸め処理、切り捨て、符号の決定

スクリプトでは、以下の関数を使用して、数値の丸め処理、数値の切り捨て、および数値の符号の決定を行うことができます。

使用場所	目的
Abs()	値または式の絶対値を計算します。
Int()	値または式の整数値を計算します。
Round()	値または式を丸めます。
Sgn()	値または式の符号（マイナス、プラス、ゼロ）を決定します。
Trunc()	小数点の前の値または式を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Abs() 関数

指定した数の絶対値を返します。負の数を正の数に変換するために使用できます。

構文

```
result = Abs (number)
```

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

例

`Abs(14)` は 14 を返します。

`Abs(-7.5)` は 7.5 を返します。

Int() 関数

指定した数値以下の整数値を返します。

構文

```
result = Int (number)
```

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

例

`Int(4 7)` は 4 を返します。

`Int(-4.7)` は -5 を返します。

注意：負の実数値の場合、この関数は指定した数値より小さい整数値を返します。たとえば、Int(-4.7) は -4 ではなく、-5 を返します。整数部分を返すには、Trunc() 関数を使用します。「Trunc() 関数」を参照してください。

Round() 関数

数値を指定した精度に丸めます。結果は実数値です。

構文

```
result = Round (number, precision)
```

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

precision
数値が切り上げ／切り捨てられる精度です。リテラル数、アナログ型タグ変数、または数式を使用できます。

例

`Round(4.3, 1)` は 4 を返します。

`Round(4.3, 0.01)` は 4.30 を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

`Round(4.5, 1)` は 5 を返します。

`Round(-4.5, 1)` は -4 を返します。

`Round(106, 5)` は 105 を返します。

`Round(43.7, 0.5)` は 43.5 を返します。

Sgn() 関数

数値の符号を返します。数値、タグ変数、または式が、負、正、またはゼロであるかを決定するときに使用できます。

構文

```
result = Sgn (number)
```

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

例

`Sgn(425)` は 1 を返します。

`Sgn(0)` は 0 を返します。

`Sgn(-37.3)` は -1 を返します。

Trunc() 関数

数値が切り捨てられた値を返します。切り捨てられた値は、小数点の前の部分です。実数値の整数部分を操作するときに使用できます。

構文

```
result = Trunc (number)
```

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

例

`Trunc(4.3)` は 4 を返します。

`Trunc(-4.3)` は -4 を返します。

注意：数値の小数点以下の部分を操作するときにも、この関数を使用できます。指定した数値の少数部分を返すには、Trunc() 関数を以下のように使用します。
result = number - trunc(number);

三角関数の使用

スクリプトでは、以下の関数を使用して、三角関数計算を実行できます。

使用場所	目的
Sin()	角度のサインを計算します。
ArcSin()	値または式のアークサインを計算します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用場所	目的
Cos()	角度のコサインを計算します。
ArcCos()	値または式のアークコサインを計算します。
Tan()	角度のタンジェントを計算します。
ArcTan()	値または式のアークタンジェントを計算します。

注意：InTouch HMI の三角 QuickScript 関数は、角度（0 ～ 360）を使用します。ラジアンを使用するには、パラメータを関数に渡す前に、または結果を関数から受け取った後に、対応する計算を実行する必要があります。

Sin() 関数

数値のサインを返します。三角関数では、数値は角度です。

構文

```
result = Sin (number)
```

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

例

`Sin(90)` は 1 を返します。

`Sin(0)` は 0 を返します。

`Sin(30)` は 0.5 を返します。

`100 * Sin (6 * $second)` は振幅 100、周期 1 分のサイン波を返します。

ArcSin() 関数

数値のアークサインを返します。Sin() 関数の逆関数です。ArcSin() 関数を使用すると、サインがその数値に等しい -90 ～ 90 の角度を計算できます。

構文

```
result = ArcSin (number)
```

パラメータ

number
-1 ～ 1 のリテラル数、アナログ型タグ変数、または数式です。

例

`ArcSin(1)` は 90 を返します。

`ArcSin(0)` は 0 を返します。

`ArcSin(0.5)` は 30 を返します。

Cos() 関数

数値のコサインを返します。三角関数では、数値は角度です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
result = Cos (number)
```

パラメータ

数値型
リテラル数、アナログ型タグ名、または数式です。

例

`Cos(90)` は 0 を返します。

`Cos(0)` は 1 を返します。

`Cos(60)` は 0.5 を返します。

`20 + 50 * Cos(6 * $second)` は切片 20、振幅 50、周期 1 分のサイン波を生成します。

ArcCos() 関数

数値のアークコサインを返します。Cos() 関数の逆関数です。ArcCos() 関数を使用すると、コサインがその数値に等しい 0 ～ 180 の角度を計算できます。

構文

```
result = ArcCos (number)
```

パラメータ

number
-1 ～ 1 のリテラル数、アナログ型タグ変数、または数式です。

例

`ArcCos(1)` は 0 を返します。

`ArcCos(-0.5)` は 120 を返します。

Tan() 関数

指定した数値のタンジェントを返します。三角関数では、数値は角度です。

構文

```
result = Tan (number)
```

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

例

`Tan(45)` は 1 を返します。

`Tan(0)` は 0 を返します。

ArcTan() 関数

数値のアークタンジェントを返します。Tan() 関数の逆関数です。ArcTan() 関数を使用すると、タンジェントがその数値に等しい角度を計算できます。

構文

```
result = ArcTan (number)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

number
リテラル数、アナログ型タグ変数、または数式です。

例

`ArcTan(1)` は 45 を返します。

`ArcTan(0)` は 0 を返します。

円周率の値を返す

スクリプトでは、Pi() 関数を使用して、定数 π を数学計算で使用できます。Pi() 関数は、小数点以下 7 桁と等しいものです。

構文

```
result = Pi ()
```

例

`Pi()` は 3.1415927 を返します。

対数の計算

スクリプトでは、以下の関数を使用して、対数および指数関数で計算を実行できます。

使用場所	目的
Log()	値または式の自然対数を計算します。
Exp()	値または式の指数を計算します。
LogN()	別の値または式の底に対する値または式の対数を計算します。

Log() 関数

指定した正の数値の自然対数を返します。Exp() 関数の逆関数です。

注意：0 および負の数値の自然対数は使用できません。0 または負の数値を Log() 関数に渡すと、-99.0000000 が返されます。

構文

```
result = Log (number)
```

パラメータ

number
正のリテラル数、アナログ型タグ変数、または数式です。

例

`Log(100)` は 4.6051702 を返します。

`Log(1)` は 0 を返します。

Exp() 関数

指定した数値の指数を返します。これは、Log() 関数の逆関数で、べき乗した e と等価です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：-88.72 ～ 88.72 の範囲外の数値を Exp() 関数に渡すと、-99.0000000 が返されます。

構文

```
result = Exp (number)
```

パラメータ

number
-88.72 ～ 88.72 のリテラル数、アナログ型タグ変数、または数式です。

例

`Exp(1)` は 2.7182818 を返します。

`Exp(0)` は 1 を返します。

LogN() 関数

指定した底に対する正の数値の自然対数を返します。これは、対数のべき乗の底の逆関数です。

例

構文

```
result = LogN (number, base)
```

パラメータ

number
正のリテラル数、アナログ型タグ変数、または数式です。

base
1 ではない正のリテラル数、アナログ型タグ変数、または数式です。

例

`LogN(8,2)` は 3 を返します。

`LogN(num,btag)` は底 btag に対する num の対数を返します。

注意：無効な数値を LogN() 関数に渡すと、-99.0000000 が返されます。

平方根の計算

スクリプトでは、Sqrt() 関数を使用して、負ではない数値の平方根を計算できます。

注意：負の数値を Sqrt() 関数に渡すと、-99.0000000 が返されます。

構文

```
result = Sqrt (number)
```

パラメータ

数値型
負ではないリテラル数、アナログ型タグ変数、または数式です。

例

`Sqrt(36)` は 6 を返します。

`Sqrt(perftag)` はタグ変数 perftga に保持されている値の平方根を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列操作

スクリプトおよびアニメーション リンクでは、多くの基本的な文字列関数を使用できます。これらの関数を使用すると、以下の処理を実行できます。

- 文字列の一部を返す
- 文字列の大文字、小文字を変更する
- 文字列のスペースを削除したり、追加したりする
- 文字列で ASCII 値を処理する
- 文字列で検索置換を行う
- 文字列をそれぞれ比較する
- 長さなど文字列のその他の情報を返す

文字列の一部を返す

スクリプトで、StringLeft() 関数、StringMid() 関数、および StringRight() 関数を使用すると、文字列の一部を返すことができます。

StringLeft() 関数

文字列の先頭から指定した文字数を返します。

構文

```
result = StringLeft (string, length)
```

パラメータ

string
リテラル テキスト、メッセージ型タグ変数、または文字列式です。

length
返す文字数です。リテラル数、アナログ型タグ変数、または数式です。

例

`StringLeft("Hello World",5)` は "Hello"を返します。

`StringLeft("Hello World",20)` は "Hello World" を返します。

`StringLeft("Hello World",0)` は "Hello World" を返します。

注意：長さ 0 を StringLeft() 関数に渡すと、文字列全体が返されます。

StringRight() 関数

文字列の末尾から指定した文字数を返します。

構文

```
result = StringRight (string, length)
```

パラメータ

string
リテラル テキスト、メッセージ型タグ変数、または文字列式です。

length
返す文字数です。リテラル数、アナログ型タグ変数、または数式です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

`StringRight("Hello World",5)` は "World" を返します。

`StringRight("Hello World",20)` は "Hello World" を返します。

`StringRight("Hello World",0)` は "Hello World" を返します。

注意：長さ 0 を StringRight() 関数に渡すと、文字列全体が返されます。

StringMid() 関数

文字列の一部を返します。開始点および返す文字数を指定できます。

構文

```
result = StringMid (string, startpos, length)
```

パラメータ

string
リテラル テキスト、メッセージ型タグ変数、または文字列式です。

startpos
文字列の開始位置です。リテラル数、アナログ型タグ変数、または数式です。

length
返す文字数です。リテラル数、アナログ型タグ変数、または数式です。

例

`StringMid("Hello World",5,4)` は "o Wo" を返します。

`StringMid("Hello World",7,50)` は "World" を返します。

`StringMid("Hello World",4,0)` は "lo World" を返します。

注意：長さ 0 を StringMid() 関数に渡すと、開始位置からの文字列全体が返されます。

文字列の大文字、小文字を変更する

スクリプトで、StringLower() 関すおよび StringUpper() 関数を使用すると、指定した文字列を小文字および大文字で返します。指定した文字列に結果を割り当て、大文字と小文字を変換します。

StringLower() 関数

文字列を小文字にして返します。

構文

```
result = StringLower (string)
```

パラメータ

string
リテラル テキスト、メッセージ型タグ変数、または文字列式です。

例

`StringLower("TURBINE")` は "turbine" を返します。

`StringLower("The Value Is 22.2")` は "the value is 22.2" を返します。

`mtag = StringLower(mtag)` は mtag のメッセージ値を小文字に変換します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

StringUpper() 関数

文字列を大文字にして返します。

構文

```
result = StringUpper (string)
```

パラメータ

string
リテラル テキスト、メッセージ型タグ変数、または文字列式です。

例

`StringUpper("abcd")` は "ABCD" を返します。

`StringUpper("The Value Is 22.2")` は "THE VALUE IS 22.2" を返します。

`mtag = StringUpper(mtag)` は mtag のメッセージ値を大文字に変換します。

文字列からのスペースの削除

スクリプトで、StringTrim() 関数を使用すると、先頭および後に続くスペース（空白）を文字列から削除できます。この関数は、たとえば、ユーザー入力後に文字列から不要なスペースを削除するときに使用できます。

StringTrim() 関数

先頭および後に続くスペース（空白）を文字列から削除します。この関数は、たとえば、ユーザー入力後に文字列から不要なスペースを削除するときに使用できます。

構文

```
result = StringTrim (string, trimtype)
```

パラメータ

文字列もじれつ
リテラル テキスト、メッセージ型タグ変数、または文字列式です。

trimtype
削除するスペースを決めるリテラル値、アナログ型タグ変数、または数式です。

- 1 = 先頭のスペース
- 2 = 後に続くスペース
- 3 = 先頭および後に続くスペース

備考

この関数は、先頭および後に続く空白を文字列からすべて削除します。空白はスペース（ASCII 0x20）および ASCII 0x09 ～ 0x0D の制御文字です。

例

アクション スクリプトで、メッセージ型タグ変数 mtag のすべてのスペースを削除するには、以下のスクリプトを使用します。

```
DIM i AS INTEGER;
DIM tmp AS MESSAGE;
mtag = StringTrim(mtag,3); {mtag は調整されます}
FOR i = 1 TO StringLen(mtag) {mtag の文字に対して変数 i を実行}
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
  IF StringMid(mtag, i, 1)<>" " THEN {i-th 文字はスペースではありません} tmp = tmp +
StringMid(mtag, i, 1); {
文字を tmp へ追加します}
  ENDIF;
NEXT;
mtag = tmp; {tmp を mtag へ返します}.
```

その他の例：

`StringTrim(" Joe ",1)` は "Joe " .を返します。

`StringTrim(" Joe ",2)` は " Joe" を返します。

このスクリプトは、mtag 値の左および右側のスペースをすべて削除します。

```
mtag = StringTrim(mtag,3)
```

スペースによる文字列の形式設定

スクリプトで、StringSpace() 関数を使用すると、スペース（空白）を文字列に追加できます。

構文

```
result = StringSpace (number)
```

パラメータ

数値型
リテラル数、数値タグ変数、または数式です。

例

`StringSpace(4)` は 4 つの空白で構成される文字列を返します。

"Pump"+`StringSpace(1)`+"Station" は "Pump Station" を返します。

文字と ASCII コードの変換

スクリプトで、StringChar() 関数および StringASCII() 関数を使用すると、文字列を ASCII コードに、ASCII コードを文字列に変換できます。

これらの関数は、複数バイトの文字セットをサポートしていません。0 ～ 255 の文字だけがサポートされています。

いくつかの数値計算を文字列で実行する場合（たとえば、文字列のコード化など）、ASCII コードを使用すると便利です。

StringChar() 関数

指定した ASCII コードに対応する 1 つの文字を返します。

構文

```
result = StringChar (ASCIICode)
```

パラメータ

ASCIICode
0 ～ 255 のリテラル数、数値タグ変数、または数式です。

備考

この関数は、制御文字を外部装置（ポインタやモデムなど）に渡す、または二重引用符を SQL クエリに渡すときに非常に便利です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

```
StringChar(65) は "A" を返します。
```

このスクリプトは、二重引用符で囲まれた "Hello World" を返します。

```
StringChar(34)+"Hello World"+StringChar(34)
```

以下の文字列は、改行およびライン フィードで分割された "Hello World" を返します。

```
"Hello"+StringChar(13)+StringChar(10)+"World"
```

StringASCII() 関数

文字列の最初の文字の ASCII コードを返します。

構文

```
result = StringASCII (string)
```

パラメータ

string
リテラル文字列、メッセージ型タグ変数、または文字列式です。

例

```
StringASCII("A") は 65 を返します。
StringASCII("hello world") は 104 を返します。
```

文字列で検索置換を行う

シングルバイト文字列セットを使用する言語（英語など）では、スクリプトで StringInString() および StringReplace() 関数を使用して、制限付き検索および置換機能をメッセージ型タグ変数で実行できます。

使用場所	目的
StringInString()	別の文字列で文字列を検索し、その位置を結果として返します。
StringReplace()	指定文字列の特定の文字または単語を他の文字または単語に置換して、新しい文字列を結果として返します。

StringInString() 関数

別の文字列での指定文字列の最初の位置を返します。

構文

```
result = StringInString (string, searchfor, startpos, casesens)
```

パラメータ

string
検索される文字列です。リテラル文字列、メッセージ型タグ変数、または文字列式です。

searchfor
検索対象の文字列です。リテラル文字列、メッセージ型タグ変数、または文字列式です。

startpos
検索の文字列の開始位置です。リテラル値、数値タグ変数、または数式です。

casesens

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

検索で大文字と小文字を区別するかどうかを決定します。0 または 1、論理型タグ変数、またはブール式を指定できます。

0 - 大文字と小文字は区別されません（大文字と小文字は同じとみなされます）。

1 - 大文字と小文字は区別されます（大文字と小文字は別のものとみなされます）。

備考

この機能を使用すると、特定の文字列がメッセージ型タグ変数に含まれているか判別できます。検索の開始位置、および小文字を認識するかどうかを指定できます。

例

このスクリプトは 5 を返します。これは、 "MTX" の最初の "M" が文字列の 5 番目にあるからです。

```
StringInString("DBO MTX-010","MTX",1,0)
```

このスクリプトは 3 を返します。これは、 "MTX" の最初の "M" が文字列の 3 番目にあるからです。

```
StringInString("T-MTX 010 MTX","MTX",1,0)
```

このスクリプトは 11 を返します。これは、8 番目より後における "MTX" の最初の "M" が文字列の 11 番目にあるからです。

```
StringInString("T-MTX 010 MTX","MTX",8,0)
```

このスクリプトは 11 を返します。これは、大文字と小文字をした “MTX” と一致する最初の文字列が 11 番目にあるからです。

```
StringInString("t-mtx 030 MTX", "MTX",1,1)
```

このスクリプトは 0 を返します。これは、 "Mty" が文字列に含まれていないからです。

```
StringInString("t-mtx 030 MTY-Mtx","Mty",1,1)
```

StringReplace() 関数

別の文字列で文字列を検索し、その文字列があった場合、別の文字列で置換します。以下の内容を指定できます。

- 大文字小文字の区別 - 大文字と小文字が同じ文字として扱われるかどうかを指定します。
- 置換するオカレンスの数 - 検索文字列のオカレンスが複数ある場合に役立ちます。
- 単語全体の一致 - 検索文字列が単語全体の場合に使用します。

注意：この関数は、ダブル バイトの文字セットをサポートしていません。

構文

```
result = StringReplace (string, searchfor, replacewith, casesens, numtoreplace,
matchwholewords)
```

パラメータ

string
検索が実行される文字列です。リテラル文字列、メッセージ型タグ変数、または文字列式です。

searchfor
検索対象の文字列です。リテラル文字列、メッセージ型タグ変数、または文字列式です。

replacewith
置換後の文字列です。リテラル文字列、メッセージ型タグ変数、または文字列式です。

casesens

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

検索で大文字と小文字を区別するかどうかを決定します。0 または 1、論理型タグ変数、またはブール式を指定できます。

0 - 大文字と小文字は区別されません（大文字と小文字は同じとみなされます）。

1 - 大文字と小文字は区別されます（大文字と小文字は別のものとみなされます）。

numtoreplace
実行する置換数です。-1 を設定すると、一致した検索文字列のすべてのオカレンスが置換されます。リテラル整数値、整数型タグ変数、または整数式です。

matchwholewords
完全に一致する単語だけを対象とするかどうかを決めます。0 または 1、論理型タグ変数、またはブール式を指定できます。

0 - 文字列の任意の場所で、検索文字列が検索されます。

1 - 完全に一致する単語だけが対象となります。

例

このステートメントは、最初のオカレンスのみを置換して、“MTY 030 MTX” を返します。

```
StringReplace("MTX 030 MTX","MTX","MTY",0,1,0)
```

このステートメントは、すべてのオカレンスを置換して、“MTY 030 MTY” を返します。

```
StringReplace("MTX 030 MTX","MTX","MTY",0,-1,0)
```

このステートメントは、大文字と小文字が一致するすべてのオカレンスを置換して、“MTY 030 mtx” を返します。

```
StringReplace("MTX 030 mtx","MTX","MTY",1,-1,0)
```

このステートメントは、完全に一致するすべてのオカレンスを置換して、“MTY 030 QMTX” を返します。

```
StringReplace("MTX 030 QMTX","MTX","MTY",0,-1,1)
```

文字列の情報を返す

スクリプトで、StringLen() 関数および StringTest() 関数を使用すると、指定した文字列の長さを返し、文字が特定の文字グループに含まれるかテストできます。

StringLen() 関数

非表示文字を含む、指定された文字列の長さを返します。

構文

```
result = StringLen (string)
```

パラメータ

string
リテラル文字列、メッセージ型タグ変数、または文字列式です。

例

```
StringLen("Twelve percent") は 14 を返します。
StringLen("12%") は 3 を返します。
StringLen("The end."+ StringChar(13)) は 9 を返します。
```

StringTest() 関数

文字列の最初の文字が、特定の文字グループに含まれるかどうかをテストします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
result = StringTest (string, group)
```

パラメータ

string
リテラル文字列、メッセージ型タグ変数、または文字列式です。

group
文字をテストするグループの番号です。1 ～ 11 のリテラル数、整数型タグ変数、または整数式です。

1 - 英数字（A～Z、a～z、0～9）

2 - 数字（0～9）

3 - 英字（A～Z、a～z）

4 - 大文字の英字（A～Z）

5 - 小文字の英字（a～z）

6 - !、@、#、$、%、^、&、* などの記号（ASCII 0x21 ～ 0x2F）

7 - ASCII 文字（ASCII 0x00 ～ 0x7F）

8 - 16 進数文字（0～9、A～F、a～f）

9 - 印刷可能文字（ASCII 0x20 ～ 0x7E）

10 - 制御文字（ASCII 0x00 ～ 0x1F および 0x7F）

11 - 空白文字（ASCII 0x09 ～ 0x0D および 0x20）

例

次の文字列は 1 を返します。これは、"A" が英数字だからです。

```
StringTest("ACB123",1)
```

次の文字列は 0 を返します。これは、"A" が小文字ではないからです。

```
StringTest("ABC123",5)
```

文字列の比較

スクリプトで、StringCompare() 関数、StringCompareNoCase() 関数、および StringCompareEncrypted() 関数を使用すると、2 つの文字列を比較できます。

使用場所	目的
StringCompare()	大文字と小文字を区別して比較します。
StringCompareNoCase()	大文字と小文字を区別しないで比較します。
StringCompareEncrypted()	暗号化された文字列を暗号化されていない文字列と比較します。

StringCompare() 関数

2 つの文字列をそれぞれ比較して、ブール値の結果を返します（0 = 文字列は同じです）。各文字の大文字と小文字は区別されるので、たとえば “A” と “a” は別の文字とみなされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
result = StringCompare (string1, string2)
```

パラメータ

string1
リテラル文字列、メッセージ型タグ変数、または文字列式です。

string2
リテラル文字列、メッセージ型タグ変数、または文字列式です。

例

```
StringCompare ("Apple","Apple") は 0 を返します。
StringCompare ("Apple","apple") は 1 を返します。
```

この文字列は、2 つのメッセージ型タグ変数を比較して、論理値結果（0 または 1）を返します。

```
StringCompare (mtag1, mtag2)
```

StringCompareNoCase() 関数

2 つの文字列をそれぞれ比較して、整数値結果を返します。各文字の大文字と小文字は区別されないので、たとえば、 “A” と “a” は同じ文字とみなされます。

以下の整数値結果を返します。

- 両方の文字列が同じ（大文字と小文字は区別されません）場合は 0 です。
- 同じ文字列でない場合は 0 以外です。結果は、異なる文字の ASCII 値の差です（大文字と小文字は区別されません）。

注意 : すべての 0 以外の値は InTouch スクリプトでは TRUE とみなされるので、StringCompareNoCase() 関数の結果は、論理値結果として使用できます。

構文

```
result = StringCompareNoCase (string1, string2)
```

パラメータ

string1
リテラル文字列、メッセージ型タグ変数、または文字列式です。

string2
リテラル文字列、メッセージ型タグ変数、または文字列式です。

例

以下の文字列は、0 を返します。これは、2 つの文字列が同じとみなされるからです。

```
StringCompareNoCase("Apple","apple")
```

以下の文字列は、6 を返します。これは、2 つの文字列が同じとみなされず、最初の異なる文字 "p" の ASCII 値から対応する文字 "v" の ASCII 値を引いた値が -6 だからです。

```
StringCompareNoCase("Apple","Avocado")
```

StringCompareEncrypted() 関数

暗号化された文字列を暗号化されていない文字列と比較し、ブール値結果を返します。この関数は、パスワードの確認に使用できます。パスワード暗号化の詳細については、「オブジェクトのアニメーション化」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
result = StringCompareEncrypted (plain, encrypted)
```

パラメータ

plain
リテラル文字列、メッセージ型タグ名、または文字列式です。

encrypted
暗号化されたメッセージ型タグ名です。

例

このスクリプトは、プレーン テキストと暗号化されたテキストが同じ場合 1 を返し、そうでない場合 0 を返します。パスワードは、暗号化されたユーザー入力の値を含むメッセージ型タグです。PlainTxt は、ユーザー入力と比較されるメッセージ型タグです。

```
StringCompareEncrypted(PlainTxt, Passwd)
```

データ タイプの変換

スクリプトで、変換 QuickScript を使用して、タグ名に含まれる値を他のデータ タイプに変換できます。これにより、文字列データを数学関数で操作、またはデバッグのために Log Viewer に値をログ記録できます。

- Text() 関数
- StringFromIntg() 関数
- StringFromReal() 関数
- StringToIntg() 関数
- StringToReal() 関数
- DText() 関数

Text() 関数

Text() 関数は、指定形式に従って、数値を文字列として返します。これは、特定の方法で値の形式を設定、またはその後の処理のために結果を他の文字列値と組み合わせるときに使用できます。

構文

```
result = Text (number, format)
```

パラメータ

number
リテラル数値、アナログ型タグ変数、または数式です。

format
"#"、"0"、"."、または "," を使用します。
桁を表す場合は "#"、小数点区切りを表す場合は "."、先頭に 0 を配置する場合は "0"、カンマを挿入する場合は "," を使用します。
書式設定でゼロを使用する場合、ゼロの後の文字はゼロを指定する必要があります。また、小数点の右側の文字も常にゼロにする必要があります。たとえば、000.00 は正しいですが、#0#0.0# は間違っています。
この関数は、必要に応じて値を切り上げ／切り捨てます。リテラル文字列、メッセージ型タグ変数、または文字列式です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

`Text(66,"#.00")` は "66.00" を返します。

`Text (1234,"#")` は "1234" を返します。

`Text (123.4, "#,##0.0")` は "123.4" を返します。

`Text (12.3, "0,000.0")` は "0,012.3" を返します。

`Text(3.57,"#.#")` は "3.6" を返します。

このスクリプトは、アナログ型タグ変数 "pressure" に値 1690.2743 が含まれている場合、文字列 "Reactor Pressure is 1690.3 mbar" を返します。

```
"Reactor Pressure is "+Text(pressure,"#.#")+" mbar"
```

StringFromIntg() 関数

スクリプトで、StringFromIntg() 関数を使用すると、整数値を文字列値に変換できます。

この関数は、整数値の文字列値を返し、同時に進数変換を実行します。これは、たとえばテキストと整数値を同時に表示する場合、または整数値を 16 進数値に変換する場合に使用できます。

構文

```
result = StringFromIntg (number, base)
```

パラメータ

number
リテラル整数値、整数型タグ変数、または整数式です。

base
変換する進数です。これは、2 進（2）、10 進（10）、または 16 進（16）など、値を別の進数に変換するときに使用されます。リテラル整数値、整数型タグ変数、または整数式です。

例

`StringFromIntg(26,2)` は "11010"（2 進数）を返します。

`StringFromIntg(26,8)` は "32" を返します。
(8 進数：26 = 3*8 + 2）

`StringFromIntg(26,10)` は "26"（10 進数）を返します。

`StringFromIntg(26,16)` は "1A"（16 進数）を返します。

StringFromReal() 関数

スクリプトで、StringFromReal() 関数を使用すると、実数値を文字列値に変換できます。

以下の内容を実行するように指定することもできます。

- 値を指定した精度に切り上げ／切り捨てる
- 値を指数表記で渡す

これは、たとえばテキストと実数値を同時に表示する場合、または実数値を指数表記で表示する場合に使用できます。

構文

```
result = StringFromReal (number, precision, type)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

number
リテラル値、アナログ型タグ変数、または数式です。

precision
小数点以下の桁数を指定します。リテラル整数値、整数型タグ変数、または整数式です。

type
指数表記を使用するかどうかを指定します。リテラル文字列、メッセージ型タグ変数、または文字列式です。

"f" - 浮動小数点表記を使用します。

"e" - 小文字 "e" による指数表記を使用します。

"E" - 大文字 "E" による指数表記を使用します。

例

`StringFromReal(263.355, 2,"f")` は "263.36" を返します。

`StringFromReal(263.355, 2,"e")` は "2.63e2" を返します。

`StringFromReal(263.55, 3,"E")` は "2.636E2" を返します。

`StringFromReal(0.5723, 2,"E")` は "5.72E-1" を返します。

StringToIntg() 関数

スクリプトで、StringToIntg() 関数を使用すると、文字列に含まれる値を整数値に変換できます。

これは、文字列の先頭に含まれる値をその後の数学演算のために整数型タグ変数に渡すときに使用できます。

構文

```
result = StringToIntg (string)
```

パラメータ

string
リテラル文字列、メッセージ型タグ変数、または文字列式です。

備考

この関数は、文字列の最初の文字を確認します。最初の文字が数値の場合、この数値およびその後の文字を、数値以外の文字が検出されるまで整数値として読み取ります。この関数は、文字列の先頭のスペースを無視します。

例

`StringToIntg("ABCD")` は 0 を返します。

`StringToIntg("13.4 mbar")` は 13 を返します。

`StringToIntg("Pressure is 13.4")` は 0 を返します。

文字列（mtag）から先頭でない最初の整数値を抽出し、その値を整数型タグ変数 itag に保存するには、以下のアクション スクリプトを使用します。

```
DIM i AS INTEGER;
DIM tmp AS INTEGER;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
FOR i = 1 TO StringLen(mtag) {mtag の文字に対して変数 i を実行}
 tmp = StringASCII(StringMid(mtag, i, 1)) - 48; {ASCII 値を検出}
 IF (tmp>=0 AND tmp<10) THEN {ASCII 値が "0" ～ "9" である場合}
 itag = StringToIntg(StringMid(mtag, i, 0)); {その位置の値を itag に設定してループを終了}
 EXIT FOR;
 ENDIF;
NEXT;
```

StringToReal() 関数

スクリプトで、StringToReal() 関数を使用すると、文字列に含まれる値を実数値に変換できます。

これは、文字列の先頭に含まれる値をその後の数学演算のために実数型タグ変数に渡すときに使用できます。

注記: また、この関数は指数表記をサポートし、文字列式 1e+6 を正確に 1000000 に変換します。

構文

```
result = StringToReal (string)
```

パラメータ

string
リテラル文字列、メッセージ型タグ変数、または文字列式です。

備考

この関数は、文字列の最初の文字を確認します。最初の文字が数値の場合、この数値およびその後の文字を、数値以外の文字が検出されるまで実数値として読み取ります。この関数は、文字列の先頭のスペースを無視します。

文字列（メッセージ型タグ変数 mtag）から先頭でない最初の実数値を抽出し、その値を実数型タグ変数 rtag1 に保存するには、以下のスクリプトを使用します。

```
DIM i AS INTEGER;
DIM tmp AS INTEGER;
FOR i = 1 TO StringLen(mtag) {mtag の文字に対して変数 i を実行}
 tmp = StringASCII(StringMid(mtag, i, 1)) - 48; {ASCII 値を検出}
 IF (tmp>=0 AND tmp<10) THEN {ASCII 値が "0" ～ "9" である場合}
 rtag = StringToReal(StringMid(mtag, i, 0)); {その位置の値を rtag に設定してループを終了}
 EXIT FOR;
 ENDIF;
NEXT;
```

例

```
StringToReal("ABCD") は 0 を返します。
StringToReal("13.4 mbar") は 13.4 を返します。
StringToReal("Pressure is 13.4") は 0 を返します。
```

DText() 関数

スクリプトで、DText() 関数を使用すると、ブール値を文字列値に変換できます。この関数を使用すると、カスタマイズしたメッセージ表示アニメーション リンクを使用できます。

この関数は、ブール値の値によって、異なる文字列値を返します。

構文

```
result = Dtext (Boolean, stringtrue, stringfalse)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

Boolean
リテラル ブール値、論理型タグ変数、またはブール式です。

stringtrue
Boolean が true の場合に返される文字列です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

stringfalse
Boolean が false の場合に返される文字列です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

例

このスクリプトは、論理型タグ変数 switch が TRUE の場合 "Running" を返し、それ以外の場合 "Stopped" を返します。

```
DText(switch,"Running","Stopped")
```

このスクリプトは、論理型タグ変数 switch1 の値に応じて、別の論理型タグ変数 switch2 のオン／オフ メッセージを返します。

```
DText(switch1,switch2.OnMsg,switch2.OffMsg)
```

ランタイムでの InTouch ウィンドウの操作

スクリプトで、InTouch ウィンドウの動作および外観を制御できます。また、QuickScript を使用して、個々の InTouch ウィンドウまたは画面全体を印刷するスクリプトを作成することもできます。

ウィンドウ名のプロパティの公開

GetWindowName スクリプト関数を使用して、ランタイム環境でのウィンドウの読み込みに必要なスクリプト作成を減らすことができます。関数を呼び出したウィンドウの名前を取得することができます。

GetWindowName() 関数

関数が呼び出されたウィンドウの名前を取得します。

構文

このスクリプト関数の構文は、以下のとおりです。

```
resultcode = GetWindowName(tagname);
```

Resultcode は、スクリプト関数の成功または失敗を示します。Resultcode は論理型/整数型/実数型のデータ タイプになります。Resultcode はスクリプト関数の成功または失敗によって、1 または 0 のいずれかになります。

- スクリプト関数がウィンドウ コンテキストから呼び出された場合、Resultcode は 1 になります。
- スクリプト関数がウィンドウ コンテキスト以外から呼び出された場合、Resultcode は 0 になります。

注: スクリプト関数の戻り値の設定はオプションです。これは InTouch の既存のスクリプト関数と似ています。

パラメータ

TagName
tagname はこの関数のパラメータです。これはウィンドウ名を取得するためのメッセージ型タグになります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

この関数のパラメータは、以下のどれでも構いません。

- InTouch タグ
- ドット フィールド
- スクリプトのローカル変数
- リモート タグ リファレンス
- Galaxy 参照

スクリプト ブラウザによって読み込まれるデフォルトのテキストは、

GetWindowName(*TagName*) です。

注: "TagName" はメッセージ型タグまたはメッセージ タイプであるリモート タグ リファレンス（RTR）です。

戻り値

GetWindowName スクリプト関数は、次のシナリオの場合、戻り値 1 とウィンドウの名前を返します。

- ウィンドウ スクリプト（すべての条件タイプ）
- 押しボタンの動作スクリプト（すべての条件タイプ）

GetWindowName スクリプト関数は、次のシナリオの場合、戻り値 0 と空の文字列を返します。

- アプリケーション スクリプト
- キー スクリプト
- 条件スクリプト
- データ変化スクリプト
- ActiveX イベント スクリプト
- クイック スクリプト関数（同期および非同期）

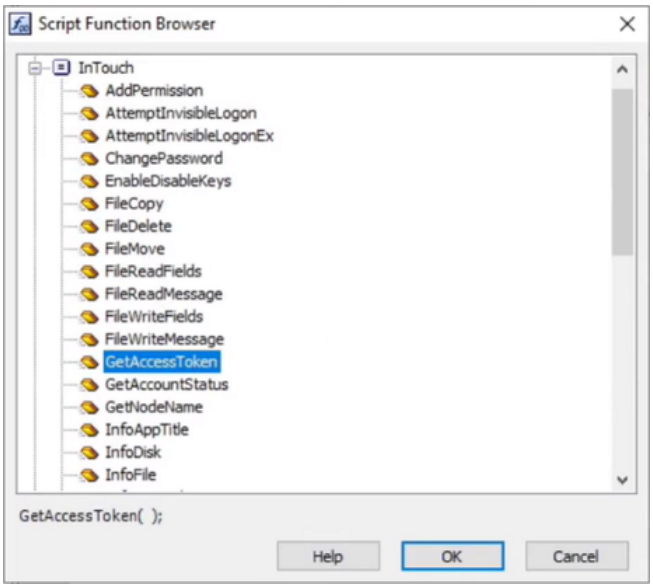
最新のアクセス トークンの公開

GetAccessToken スクリプトと **GetSecureAccessToken** スクリプトを使用することによって、CONNECT アクセス トークンを InTouch に公開し、ランタイムの接続エクスペリエンスでのコントロールとウィジェットのシングル サインオンに使用できます。

GetAccessToken スクリプト関数を使用すると最新のトークン値を返すことができます。
GetSecureAccessToken スクリプトを使用すると最新のトークン値を安全な方法または暗号化された方法で返すことができますアクセス トークンは時間にバインドされ、値は定期的に変更されるので、このスクリプト関数を使うと最新のアクセス トークン値を取得できます。

スクリプト関数ブラウザで、組み込みスクリプト **GetAccessToken** および **GetSecureAccessToken** を表示できます。これはアクセス トークンの値をロガーに記録します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



GetAccessToken() 関数

GetAccessToken () は、認証トークンを提供します。この認証トークンは、シングル サインオン機能をサポートするために、トレンド コントロール、アラーム クライアント コントロール、およびその他の InTouch コントロールのようなコントロールによって受け入れられます。

構文

このスクリプト関数の構文は、以下のとおりです。

```
resultCode = GetAccessToken();
```

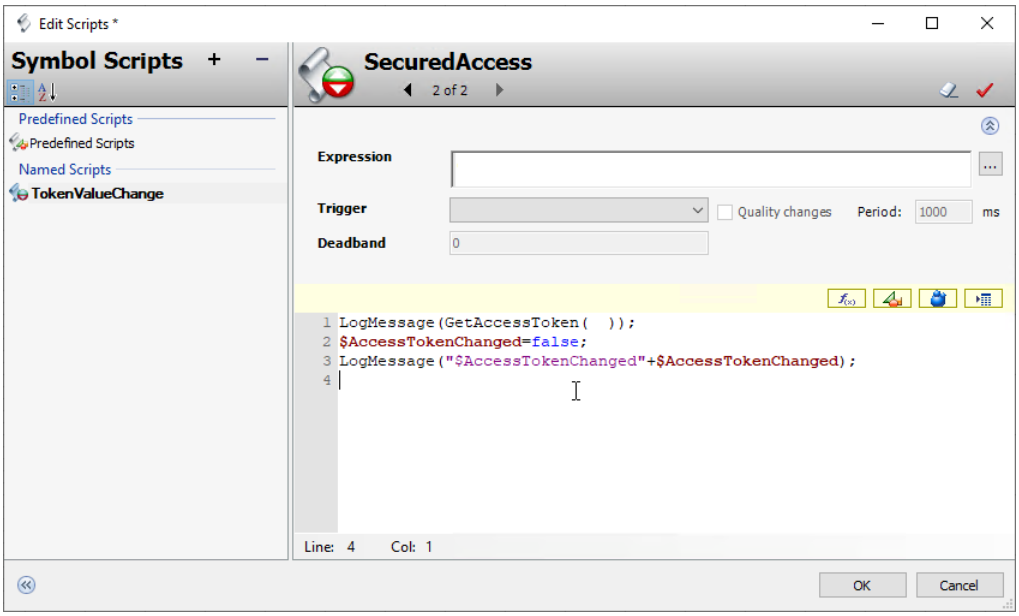
resultcode は、最新のアクセス トークンを示します。

例:

有効期限が切れるたびにトークン値を更新するには、スクリプトを次のように使用できます。

1. ［シンボル スクリプト］ウィンドウで［**スクリプトを追加**］を選択します。
2. スクリプトに名前を付けます。
3. ［**スクリプト関数ブラウザを表示**］アイコンを選択します。
4. ［**スクリプト ブラウザ**］］画面の［**InTouch**］で［**GetAccessToken**］を選択してスクリプト関数を挿入します。手動で入力することもできます。
5. **[OK]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



戻り値

ユーザーが CONNECT を使用して認証されている場合、GetAccessToken スクリプト関数は、トークン値を含むアクセス トークンを返します。

ユーザーが CONNECT を使用して認証されていない場合、GetAccessToken スクリプト関数は、空の文字列を返します。

GetSecureAccessToken() 関数

GetSecureAccessToken () は、セキュアな認証トークンを提供します。この認証トークンは、シングル サインオン機能をサポートするために、トレンド コントロール、アラーム クライアント コントロール、およびその他の InTouch コントロールのようなコントロールによって受け入れられます。

構文

このスクリプト関数の構文は、以下のとおりです。

```
resultCode = GetSecureAccessToken();
```

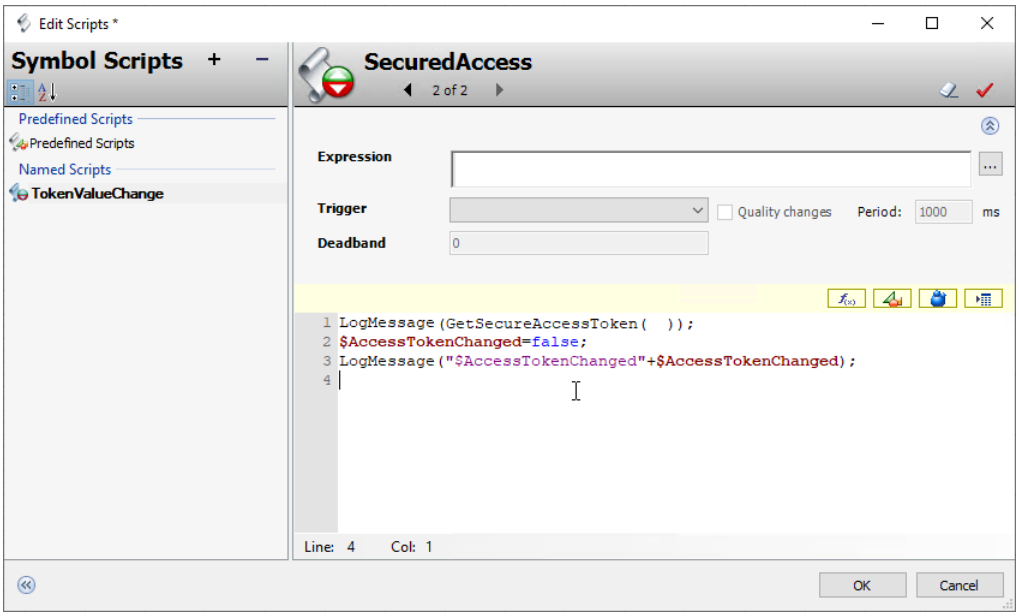
resultcode は、最新のアクセス トークンを示します。

例:

有効期限が切れるたびにセキュアなトークン値を更新するには、スクリプトを次のように使用できます。

1. ［シンボル スクリプト］ウィンドウで［**スクリプトを追加**］を選択します。
2. スクリプトに名前を付けます。
3. ［**スクリプト関数ブラウザを表示**］アイコンを選択します。
4. ［**スクリプト ブラウザ**]］画面の［**InTouch**］で［**GetSecureAccessToken**］を選択してスクリプト関数を挿入します。手動で入力することもできます。
5. **[OK]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



戻り値

ユーザーが CONNECT を使用して認証されている場合、GetSecureAccessToken スクリプト関数は、トークン値を含むアクセス トークンを返します。

ユーザーが CONNECT を使用して認証されていない場合、GetSecureAccessToken スクリプト関数は、空の文字列を返します。

接続ステータスの取得

GetTokenConnectionStatus スクリプト関数を使用して、AVEVA Operations Control 接続エクスペリエンスの接続ステータスを取得できます。

GetTokenConnectionStatus() 関数

AVEVA Operations Control 接続エクスペリエンスの AVEVA Identity Manager と CONNECT の接続のステータスを取得します。

構文

このスクリプト関数の構文は、以下のとおりです。

```
resultcode = GetTokenConnectionStatus();
```

resultcode は、接続ステータスのトークンを示します。

戻り値

- 接続ステータスが完全接続の場合、resultcode は 0 です。アプリケーションは AVEVA Identity Manager と CONNECT の両方に接続されています。
- 接続ステータスが部分接続の場合、resultcode は 1 です。アプリケーションは AVEVA Identity Manager に接続されていて、CONNECT には接続されていません。
- 接続ステータスが完全接続解除の場合、resultcode は 2 です。アプリケーションは AVEVA Identity Manager と CONNECT のいずれにも接続されていません。

例：

int AccessTokenStatus=GetTokenConnectionStatus()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

開いているウィンドウのリストの表示

スクリプトで、OpenWindowsList() 関数を使用すると、現在開いている InTouch ウィンドウのリストを示すダイアログ ボックスを表示できます。

OpenWindowList() 関数

現在開いている InTouch ウィンドウのリストを示すダイアログ ボックスを表示します。

この関数はアニメーション リンクでは使用できません。

構文

```
[result = ]OpenWindowsList();
```

例

このスクリプトは、［**開かれているウィンドウのリスト**］ダイアログ ボックスを開き、現在開いているすべての InTouch ウィンドウを示します。

```
OpenWindowsList()
```

注意：［**メモリ内ウィンドウのキャッシュを使用する**］WindowViewer オプションが有効な場合、閉じたウィンドウが OpenWindowList() 関数によって作成されたリストに表示される場合があります。

ウィンドウが開いているか、閉じているか、存在するかを確認する

スクリプトで、WindowState() 関数を使用すると、InTouch ウィンドウが開いているか、閉じているか、または存在しないかを確認できます。

WindowState() 関数

InTouch ウィンドウが開いているか、閉じているか、または存在しないかを確認します。

構文

```
result = WindowState (windowname)
```

パラメータ

windowname
ウィンドウの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

戻り値

整数値であり、値の意味は以下のとおりです。

0 - InTouch ウィンドウが存在し、現在閉じています。

1 - InTouch ウィンドウが存在し、現在開いています。

2 - InTouch ウィンドウは存在しません。

例

このスクリプトは、InTouch ウィンドウ Main が存在するが、開いていない場合、0 を返します。

```
WindowState("Main")
```

InTouch ウィンドウを開く

スクリプトで、以下の QuickScript 関数のいずれかを使用して、InTouch ウィンドウを開くことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用場所	目的
表示	位置設定で定義された場所に InTouch ウィンドウを開きます。
ShowAt()	指定した場所に InTouch ウィンドウを開きます。開かれたウィンドウは、その場所の中央に配置されます。この関数は、開いているウィンドウを移動するときにも使用できます。
ShowHome	［**WindowViewer のプロパティ**］ダイアログ ボックスの［**起動時ウィンドウ**］タブで指定した InTouch ウィンドウを開いて、他のウィンドウを閉じます。
ShowTopLeftAt()	指定した場所に InTouch ウィンドウを開きます。開かれたウィンドウは、その場所の左上隅に合わせて配置されます。この関数は、開いているウィンドウを移動するときにも使用できます。

Show() 関数

デフォルトの場所に InTouch ウィンドウを開きます。

構文

```
Show windowname
```

パラメータ

windowname
開かれるウィンドウの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

例

このスクリプトは、ウィンドウ Main を開きます。

```
Show "Main";
```

このスクリプトは、wname メッセージ型タグ変数に保存されている名前でウィンドウを表示します。

```
Show wname;
```

ShowAt() 関数

指定した場所に InTouch ウィンドウを開きます。また、すでに開いている InTouch ウィンドウを指定した場所に移動することもできます。場所は、ウィンドウの中央です。

注意：ウィンドウのいずれかの端が画面上にない場合、中央には配置されません。

構文

```
ShowAt (windowname, xpos, ypos)
```

パラメータ

windowname
開かれる、または移動されるウィンドウの名前です。

xpos
ウィンドウの中央が移動されるピクセル単位での水平位置です。リテラル値、アナログ型タグ変数、または数式です。

ypos

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウの中央が移動されるピクセル単位での垂直位置です。リテラル値、アナログ型タグ変数、または数式です。

例

このスクリプトは、x:450、y:130 の位置を中央として配置されるようにウィンドウ Main を開きます。

```
ShowAt("Main",450,130);
```

このスクリプトは、UserDialog というウィンドウを開いて、この関数を呼び出したオブジェクト（ボタンなど）の中央位置に合わせてウィンドウを配置します。

```
ShowAt("UserDialog",$ObjHor,$ObjVer);
```

ShowHome() 関数

［**WindowViewer のプロパティ**］ダイアログ ボックスの［**起動時ウィンドウ**］タブで指定した InTouch ウィンドウを開いて、他のウィンドウを閉じます。

構文

```
ShowHome;
```

ShowTopLeftAt() 関数

指定した場所に InTouch ウィンドウを開きます。開いているウィンドウを移動するときにも使用できます。

構文

```
ShowTopLeftAt (windowname, xpos, ypos)
```

パラメータ

開かれる、または移動されるウィンドウの名前です。

xpos
ウィンドウの左端が移動されるピクセル単位での水平位置です。リテラル値、アナログ型タグ変数、または数式です。

ypos
ウィンドウの上端が移動されるピクセル単位での垂直位置です。リテラル値、アナログ型タグ変数、または数式です。

例

このスクリプトは、x:450、y:130 の位置にウィンドウ左上隅が配置されるようにウィンドウ Main を開きます。

```
ShowTopLeftAt("Main",450,130);
```

ウィンドウの移動とサイズ変更

スクリプトで、WWMoveWindow() 関数を使用すると、開いている InTouch ウィンドウを移動したりサイズを変更したりできます。新しい位置および新しいサイズは、指定ウィンドウが開いている間、一時的に適用されます。

WWMoveWindow() 関数

開いている InTouch ウィンドウを指定位置に移動して、指定サイズに変更します。新しい位置および新しいサイズは、指定ウィンドウが開いている間、一時的に適用されます。

構文

```
WWMoveWindow (windowname, xpos, ypos, xsize, ysize)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

windowname
開かれる、または移動されるウィンドウの名前です。

xpos
ウィンドウの左端が移動されるピクセル単位での水平位置です。リテラル値、アナログ型タグ変数、または数式です。

ypos
ウィンドウの上端が移動されるピクセル単位での垂直位置です。リテラル値、アナログ型タグ変数、または数式です。

xsize
指定したウィンドウのピクセル単位での水平サイズです。リテラル値、アナログ型タグ変数、または数式です。

ysize
指定したウィンドウのピクセル単位での垂直サイズです。リテラル値、アナログ型タグ変数、または数式です。

InTouch ウィンドウを非表示にする

スクリプトで、以下の関数のいずれかを使用して、InTouch ウィンドウを非表示にすることができます。

使用場所	目的
非表示	指定したウィンドウを非表示にします。
HideSelf	現在アクティブなウィンドウを非表示にします。

Hide() 関数

InTouch ウィンドウを非表示にします（閉じます）。

構文

```
Hide windowname;
```

パラメータ

windowname
非表示にされるウィンドウの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

例

このスクリプトは、ウィンドウ UserConfirmation を非表示にします。

```
Hide "UserConfirmation";
```

HideSelf() 関数

現在アクティブな InTouch ウィンドウを非表示にします（閉じます）。

注意：この関数は、アクション QuickScript でのみ使用できます。

構文

```
HideSelf;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

```
HideSelf;
```

ウィンドウの色の変更

スクリプトで、ChangeWindowColor() 関数を使用すると、開いている InTouch ウィンドウの色を変更できます。

ChangeWindowColor() 関数

開いている InTouch ウィンドウの色を変更して、結果コードを返します。

構文

```
Result = ChangeWindowColor (windowname, rValue, gValue, bValue)
```

パラメータ

windowname
色を変更するウィンドウの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

rValue
赤色の輝度です。0 ～ 255 のリテラル整数値、整数型タグ変数、または整数式です。

gValue
緑色の輝度です。0 ～ 255 のリテラル整数値、整数型タグ変数、または整数式です。

bValue
青色の輝度です。0 ～ 255 のリテラル整数値、整数型タグ変数、または整数式です。

戻り値

値の意味は以下のとおりです。

0 - 失敗。ウィンドウが定義されていないか、RGB 値が範囲外です。

1 - 成功

2 - 失敗。ウィンドウは存在しますが、開いていません。

ランタイムでのウィンドウの印刷

スクリプトで、PrintWindow() 関数または PrintScreen() 関数を使用すると、個々の InTouch ウィンドウまたは WindowViewer 画面全体を印刷できます。また、SetWindowPrinter() 関数と組み合わせて使用するプリンタを設定できます。

SetWindowPrinter() 関数

ランタイムで、SetWindowPrinter() 関数と組み合わせて使用するプリンタを設定できます。

注意：この関数で設定されたプリンタは、PrintHT() で使用されるプリンタでもあります。

構文

```
SetWindowPrinter (printername)
```

パラメータ

printername
プリンタの名前です。これは、プロパティ ウィンドウに表示されるネットワーク共有またはプリンタ名のいずれかです。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例では、PRTSRV1 はノード名であり、PRT22SW1 はプリンタに指定された共有名です。

```
SetWindowPrinter("\\PRTSRV1\PRT22SW1");
```

この例では、Epson LX-300 はプリンタの［プロパティ］ウィンドウに表示されるプリンタの名前です。

```
SetWindowPrinter("Epson LX-300");
```

この例では、MyPrinter はインストールされている Windows プリンタの名前または共有ネットワーク プリンタのパスを含むメッセージ型タグ変数です。

```
SetWindowPrinter(MyPrinter);
```

印刷に関する推奨事項

以下のリストは、印刷時に考慮する必要のあるいくつかの問題を示しています。これらの問題は、1 つのウィンドウまたは WindowViewer 画面を印刷するときに該当します。

- 印刷するウィンドウを印刷前に開きます。このようにしないと、Windows コントロールおよび ActiveX コントロールでの印刷が正しく機能しないことがあります。
- 現在アラームを印刷しているプリンタと同じプリンタには印刷できません。
- 印刷する場合、ウィンドウおよびウィンドウのオブジェクトが重ならないようにします。
- 可能な限り、True Type フォントを使用します。デフォルトの InTouch フォント（システム）は True Type フォントではありません。
- 印刷処理を速くするには、背景色を白くする、オブジェクト数を減らす、グラフィックの代わりにテキストを使用するなどの処理を考慮してください。
- WindowViewer は、ウィンドウがプリンタ キューに送信される前に、一定時間待機します。この時間中、WindowViewer は、そのウィンドウの任意の I/O 値をバックグラウンドで更新します。この待機時間を変更するには、intouch.ini ファイルを開いて、以下の行を変更または追加します（ミリ秒単位）：`PrintWindowWait=10000`

PrintWindow() 関数

スクリプトで、PrintWindow() 関数を使用すると、InTouch ウィンドウを印刷できます。

注記: PrintWindow() 関数を含むスクリプトでは、InTouch ウィンドウ内の ListBox、DateTimePicker、CalenderControl、EditBox、CheckBox、RadioButtonGroup、ComboBox、AlarmClient、TrendClient の産業用シンボルのコントロールは印刷できません。

構文

```
[result = ] PrintWindow (windowname, leftmargin, topmargin, width, height, options);
```

パラメータ

windowname
印刷されるウィンドウの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

leftmargin
左マージン オフセット（インチ単位）。リテラル数値、アナログ型タグ変数、または数式です。

topmargin
上マージン オフセット（インチ単位）。リテラル数値、アナログ型タグ変数、または数式です。

width
印刷幅（インチ単位）。縦横比を最大にする場合、この値を 0 に設定します。リテラル数値、アナログ型タグ変数、または数式です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

height
印刷高さ（インチ単位）。縦横比を最大にする場合、この値を 0 に設定します。リテラル数値、アナログ型タグ変数、または数式です。

options
論理値 0 または 1 です。これは、width および height が 0 の場合だけ使用されます。リテラル ブール値、論理型タグ変数、またはブール式です。値は以下のように設定されます。

1 - ウィンドウ サイズの整数の倍数である最大縦横比でウィンドウが印刷されます。

0 - ページに合わせた最大縦横比でウィンドウが印刷されます。

注記: ウィンドウにビットマップが含まれる場合、options を 1 に設定して、ビットマップが拡大しないようにします。

戻り値

0 - 印刷ジョブが正常にキューに送信されなかったか、ウィンドウが存在しません。

1 - 印刷ジョブが正常にキューに送信されました。

PrintScreen() 関数

PrintScreen() 関数を使用すると、WindowViewer 画面全体を印刷するスクリプトを記述できます。

構文

```
PrintScreen (ScreenOption, PrintOption)
```

パラメータ

ScreenOption
印刷する WindowViewer 画面の領域を決定します。リテラル整数値、整数型タグ名、または整数式です。

1 - クライアント領域を印刷します。メニューは印刷しません（デフォルト）。

2 - メニューを含むウィンドウの全体の領域を印刷します。

PrintOption
印刷される画像が印刷に合わせてどのように拡大されるかを決めます。

- 1 - 最適：
  画像は、縦横比を変更せずに水平または垂直のいずれかに合わせて拡大されます（デフォルト）。
- 2 - 垂直：
  画像は、印刷の縦横比を変更せずに垂直に合わせて拡大されます。画像は水平に切り取られることがあります。
- 3 - 水平：
  画像は、縦横比を変更せずに水平に合わせて拡大されます。画像は垂直に切り取られることがあります。
- 4 - ページに合わせる：
  画像は、水平および垂直に合わせて拡大されます。縦横比が変わることがありますが、画像は切り取られません。
- 0 など無効なオプションを指定すると、デフォルトで最適が適用されます。

注意：WindowViewer 画面領域を超えて表示されるポップアップ ウィンドウは、はみ出した部分が切り取られます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

このスクリプトは、メニューを除く、現在の WindowViewer 画面の全領域の出力をプリンタ キューに送信します。出力には、出力寸法に合わせて拡大された画面領域が含まれます。

```
PrintScreen(1,4);
```

上記の例では、関数に渡される最初のパラメーターは *ScreenOption* (1) です。これにより、メニューのないクライアント領域が印刷されます。

渡される 2 番目のパラメーターは *PrintOption* (4) です。これにより、印刷ページの寸法に合わせて領域が拡大されます。

PrintHT() 関数

スクリプトで、ボタンを作成して、PrintHT QuickScript 関数を実行するアクション QuickScript にリンクすると、このボタンを使って履歴トレンド グラフを印刷できます。

トレンド グラフだけではなく、ウィンドウ全体を印刷する場合は、PrintHT() 関数の代わりに PrintWindow() 関数を使用します。

注意 :［印刷］オプションまたは PrintHT() 関数を使用して履歴トレンドを印刷する場合、X 値および Y 値は印刷されません。X および Y 値を印刷するには、PrintWindow() または PrintScreen() を使用します。

構文

```
PrintHT(HistTrendTagname);
```

パラメータ

HistTrendTagname
印刷する履歴トレンドの履歴トレンド タグ変数名です。

Tag Viewer の起動

Tag Viewer はランタイム時にタグを監視し、タグの値を変更できるランタイム アプリケーションです。Tag Viewer およびその使用方法の詳細については、『*AVEVA™ InTouch HMI アプリケーション ランタイム*』を参照してください。

LaunchTagViewer() 関数

Tag Viewer は WindowViewer が実行中で、デザイン時に有効に設定されているときにのみ起動できます。

Tag Viewer の有効化に関する詳細は、『*AVEVA™ InTouch HMI コンポーネントの標準の作成*』の「一般的な WindowViewer プロパティを設定する」参照してください。

構文

```
LaunchTagViewer()
```

備考

LaunchTagViewer() 関数は OnStartup と OnShutdown 以外の、どのスクリプト タイプからでも実行することができます。

Tag Viewer が WindowMaker で有効化されていない場合、関数を呼び出しても Tag Viewer は起動せず、警告メッセージがロガーに表示されます。

Tag Viewer を起動するには、適切なセキュリティ権限が必要です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

日付と時刻情報の操作

スクリプトでシステム タグ変数および QuickScript 関数を使用して、計算でシステムの時間と日付の設定を使用できます。InTouch スクリプトは、複数のタイム ゾーンと夏時間を使用する計算もサポートします。

数値の日付と時刻情報の取得

スクリプトで、さまざまな数値システム タグ変数、および 1 つのスクリプト関数を使用すると、システム時刻および日付に関する情報を取得できます。これらのタグ変数およびスクリプト関数は、他の数学演算で使用できます。以下のシステム タグ変数およびスクリプト関数を使用できます。

使用場所	目的
$Year	現在の年度を返します。
$Month	現在の月を返します。
$Day	現在の日付を返します。
$Hour	現在の時間を返します。
$Minute	現在の分を返します。
$Second	現在の秒を返します。
$Msec	現在のミリ秒を返します。
$Time	ローカル タイム ゾーンで真夜中以降に経過した時間をミリ秒単位で返します。
$Date	ローカル タイム ゾーンで 1970 年 1 月 1 日以降に経過した日数を返します。
$DateTime	ローカル タイム ゾーンで 1970 年 1 月 1 日以降に経過した日数（端数を含む）を返します。
DateTimeGMT()	世界協定時刻（UTC）で 1970 年 1 月 1 日以降に経過した日数（端数を含む）を返します。

$Year システム タグ変数

現在の年を返します。

構文

```
$Year
```

データタイプ

整数型（読み取り専用）

例

このスクリプトは、文字列 "Welcome to xxxx" を文字列 Welcome に割り当てます。ここで、xxxx は現在の年です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
Welcome = "Welcome to " + StringFromIntg($Year,10)
```

$Month システム タグ変数

現在の月を返します。

構文

```
$Month
```

データタイプ

整数型（読み取り専用）

例

このスクリプトは、現在の月が 10 の場合、文字列 "October" を文字列 MonthName に割り当てます。

```
IF $Month==10 THEN
   MonthName="October";
ENDIF;
```

$Day システム タグ変数

現在の日付を返します。

構文

```
$Day
```

データ タイプ

整数型（読み取り専用）

例

このスクリプトは、現在の日付が 2 月 29 日の場合、文字列「今年はうるう年です!」を文字列 Msg2User に割り当てます。

```
IF $Day==29 AND $Month==2 THEN
   Msg2Usr="今年はうるう年です!";
ENDIF;
```

$Hour システム タグ変数

現在の時間を返します。

構文

```
$Hour
```

データタイプ

整数型（読み取り専用）

例

このスクリプトは、現在の時刻が 8 PM で、バックアップが実行されていないかどうか（論理型タグ変数 BackupAlreadyRun を使用）を確認し、そうである場合、RunBackup() という QuickFunction スクリプトを呼び出して、BackupAlreadyRun フラグを TRUE に設定します。

```
IF $Hour==20 AND BackupAlreadyRun==0 THEN
   CALL RunBackup();
   BackupAlreadyRun=1;
ENDIF;
```

$Minute システム タグ変数

現在の分を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
$Minute
```

データタイプ

整数型（読み取り専用）

例

このスクリプトは、現在の時刻が 4:50 PM であるかを確認し、その場合、Shift End という名前のウィンドウを表示します。

```
IF $Minute==50 AND $Hour==16 THEN
   Show "Shift End";
ENDIF;
```

$Second システム タグ変数

現在の秒を返します。

構文

```
$Second
```

データタイプ

整数型（読み取り専用）

例

このスクリプトは、振幅 100 および周期 1 分のサイン波を生成します。

```
100*Sin(6*$Second)
```

このスクリプトは、毎秒ごとに変化する一連の 0 と 1 を生成します。

```
$second.00
```

$Msec システム タグ変数

現在のミリ秒を返します。

注意：デフォルトでは、InTouch は、1000 ミリ秒ごとにすべてのタグ変数を更新します。このため、$Msec システム タグ変数は変化していないように見えます。WindowViewer プロパティで更新速度を増加すると、$Msec タグ変数が更新していることを確認できます。

構文

```
$Msec
```

データタイプ

整数型（読み取り専用）

$Time システム タグ変数

ローカル時刻で真夜中以降に経過した時間をミリ秒単位で返します。

構文

```
$Time
```

データタイプ

整数型（読み取り専用）

例

このスクリプトは、真夜中以降に経過した秒数を返します。

```
$Time/1000
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

$Date システム タグ変数

ローカル時刻で 1970 年 1 月 1 日以降に経過した日数を返します。

構文

```
$Date
```

データ タイプ

整数型（読み取り専用）

例

このスクリプトは、現在の時刻を返します。

```
StringFromTime(($Date*86400)+($Time/1000),3);
```

$DateTime システム タグ変数

ローカル時刻で 1970 年 1 月 1 日以降に経過した日数（端数を含む）を返します。

構文

```
$DateTime
```

データ タイプ

実数型（読み取り専用）

例

このスクリプトは、現在の時刻を返します。

```
StringFromTime($DateTime*86400,3);
```

DateTimeGMT() 関数

世界協定時刻（UTC）で 1970 年 1 月 1 日以降に経過した日数（端数を含む）を返します。

注: この関数は、アニメーション表示リンクでは使用できません。

構文

```
result = DateTimeGMT();
```

戻り値

UTC で 1970 年 1 月 1 日以降の日数。リテラル実数値です。

例

このスクリプトは、UTC での現在の日付および時刻を返します。

```
StringFromTime(DateTimeGMT() * 86400.0, 3);
```

文字列の日付と時刻情報の取得

スクリプトで、日付および時刻情報を文字列として取得できます。これは、画面に日付または時刻を表示、または時刻／日付の文字列全体の計算が必要な場合に役立ちます。

以下のシステム タグ変数およびスクリプト関数を使用できます。

使用場所	目的
$DateString	システム日付を短い形式で返します。
$TimeString	システム時刻を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用場所	目的
UTCDateTime	ローカル コンピュータの UTC 時刻およびタイム ゾーン、あるいはその両方を返します。

$DateString システム タグ変数

ローカル オペレーティング システムの［地域と言語のオプション］の定義に従い、システム日付を短い形式で返します。

構文

```
$DateString
```

データタイプ

文字列（読み取り専用）

例

このスクリプトは、オペレーティング システムの［地域と言語のオプション］の短い日付形式設定に従い 2006/4/28 を返します。

```
$DateString
```

$TimeString システム タグ変数

ローカル オペレーティング システムの［地域と言語のオプション］の定義に従い、システム時刻を返します。

構文

```
$TimeString
```

データタイプ

文字列（読み取り専用）

例

このスクリプトは、オペレーティング システムの［地域と言語のオプション］の時刻形式設定に従い 02:40:37 PM を返します。

```
$TimeString
```

UTCDateTime() 関数

UTC 時刻、UTC 日付および時刻、またはローカル タイム ゾーンを返します。

構文

```
result = UTCDateTime (format)
```

パラメータ

format
返される内容を決めます。以下のいずれかの値のリテラル文字列値、メッセージ型タグ変数、または文字列式です。

UTC_SHORT - UTC 時刻を返します。

UTC_LONG - UTC 日付および時刻を返します。

UTC_LOCAL - ローカル オペレーティング システムのタイム ゾーン設定の設定に従い、タイム ゾーンの名前を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

他の値を指定すると、UTC 日付および時刻がデフォルト形式（ddd mm dd hh:mm:ss yyyy）で返されます。

例

太平洋タイム ゾーンで 2003 年 1 月 6 日、月曜 09:24 AM の場合、UTCDateTime() 関数は、以下のように情報を返します。

このスクリプトは、17:24:05 を返します。

```
UTCDateTime("UTC_SHORT")
```

このスクリプトは、01/06/2003 17:24:05 を返します。

```
UTCDateTime("UTC_LONG")
```

このスクリプトは、太平洋標準時 -8:0: 1 を返します。1

```
UTCDateTime("UTC_LOCAL")
```

このスクリプトは、Mon Jan 06 17:24:05 2003 を返します。

```
UTCDateTime("Invalid")
```

文字列への日付と時刻情報の変換

スクリプトで、解釈や表示をより簡単にするため、日付および時刻情報を文字列に変換できます。以下の関数を使用できます。

使用場所	目的
StringFromTime()	UTC タイムスタンプをローカル時刻に変換し、結果を時刻文字列として返します。
wwStringFromTime()	ローカル タイムスタンプを UTC 時刻に変換して、結果を時刻文字列として返します。
StringFromTimeLocal()	タイムスタンプを時刻文字列として変換します。

StringFromTime() 関数

UTC 時刻のタイムスタンプをローカル時刻に変換して、結果を文字列として返します。この関数は、夏時間を考慮します。

注: この関数は、StringFromGMTTimeToLocal() 関数と同じです。

構文

```
result = StringFromTime (timestamp, format)
```

パラメータ

timestamp
UTC タイム ゾーンで 1970 年 1 月 1 日の真夜中以降に経過した秒数を返します。リテラル整数値、整数型タグ名、または整数式です。

形式
文字列結果の表示方法を決めます。以下の意味を示す 1 ～ 5 のリテラル整数値、整数型タグ名、または整数式です。

1 - ローカル オペレーティング システムの［地域と言語のオプション］で設定された形式に従い日付を表示します。

2 - ローカル オペレーティング システムの［地域と言語のオプション］で設定された形式に従い時刻を表示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3 - 日付および時刻を 24 文字の文字列（ddd mmm dd hh:mm:ss yyyy）で表示します。

4 - 曜日を短い形式で表示します。

5 - 曜日を長い形式で表示します。

例

この例では、ローカル ノードのタイム ゾーンは太平洋標準時（PST、UTC-0800）です。関数に渡される UTC 時間は 1970 年 1 月 2 日金曜日の 12:00:00 AM です。PST は UTC よりも 8 時間遅れているので、関数は以下の結果を返します。

以下のスクリプトは "1/1/70" を返します。

```
StringFromTime(86400,1)
```

以下のスクリプトは "04:00:00 PM" を返します。

```
StringFromTime(86400,2)
```

このスクリプトは "Thu Jan 01 16:00:00 1970" を返します。

```
StringFromTime(86400,3)
```

このスクリプトは "Thu" を返します。

```
StringFromTime(86400,4)
```

このスクリプトは "Thursday" を返します。

```
StringFromTime(86400,5)
```

wwStringFromTime() 関数

ローカル時刻のタイムスタンプを UTC 時刻に変換して、結果を文字列として返します。この関数は、夏時間を考慮します。

構文

```
result = wwStringFromTime (timestamp, format)
```

パラメータ

timestamp
ローカル タイム ゾーンで 1970 年 1 月 1 日の真夜中以降に経過した秒数を返します。リテラル整数値、整数型タグ変数、または整数式です。

format
文字列結果の表示方法を決めます。以下の意味を示す 1 ～ 5 のリテラル整数値、整数タグ変数、または整数式です。

1 - ローカル オペレーティング システムの［地域と言語のオプション］で設定された形式に従い日付を表示します。

2 - ローカル オペレーティング システムの［地域と言語のオプション］で設定された形式に従い時刻を表示します。

3 - 日付および時刻を 24 文字の文字列（ddd mmm dd hh:mm:ss yyyy）で表示します。

4 - 曜日を短い形式で表示します。

5 - 曜日を長い形式で表示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例では、ローカル ノードのタイム ゾーンは太平洋標準時（PST、UTC-0800）です。関数に渡されるローカル時間は 1970 年 1 月 1 日木曜日の 04:00:00 PM です。PST は UTC よりも 8 時間遅れているので、関数は以下の結果を返します。

以下のスクリプトは "1/2/70" を返します。

```
wwStringFromTime(57600,1)
```

以下のスクリプトは "12:00:00 AM" を返します。

```
wwStringFromTime(57600,2)
```

以下のスクリプトは "Fri Jan 02 00:00:00 1970" を返します。

```
wwStringFromTime(57600,3)
```

以下のスクリプトは "Fri" を返します。

```
wwStringFromTime(57600,4)
```

以下のスクリプトは "Friday" を返します。

```
wwStringFromTime(57600.5)
```

StringFromTimeLocal() 関数

タイムスタンプを時刻に変換して、結果を文字列として返します。

構文

```
result = StringFromTimeLocal (timestamp, format)
```

パラメータ

timestamp
1970 年 1 月 1 日の真夜中以降に経過した秒数を返します。リテラル整数値、整数型タグ変数、または整数式です。

format
文字列結果の表示方法を決めます。以下の意味を示す 1 ～ 5 のリテラル整数値、整数タグ変数、または整数式です。

1 - ローカル オペレーティング システムの［地域と言語のオプション］で設定された形式に従い日付を表示します。

2 - ローカル オペレーティング システムの［地域と言語のオプション］で設定された形式に従い時刻を表示します。

3 - 日付および時刻を 24 文字の文字列（ddd mmm dd hh:mm:ss yyyy）で表示します。

4 - 曜日を短い形式で表示します。

5 - 曜日を長い形式で表示します。

例

以下のスクリプトは "1/2/70" を返します。

```
StringFromTimeLocal(86400,1)
```

以下のスクリプトは "12:00:00 AM" を返します。

```
StringFromTimeLocal(86400,2)
```

以下のスクリプトは "Fri Jan 02 00:00:00 1970" を返します。

```
StringFromTimeLocal(86400,3)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以下のスクリプトは "Fri" を返します。

```
StringFromTimeLocal(86400,4)
```

以下のスクリプトは "Friday" を返します。

```
StringFromTimeLocal(86400.5)
```

夏時間状態の確認

スクリプトで、wwIsDaylightSaving() 関数を使用すると、夏時間設定がアクティブになっているかどうかを確認できます。

wwIsDaylightSaving() 関数

夏時間が現在アクティブになっているかどうかを返します。

構文

```
result = wwIsDaylightSaving()
```

戻り値

ブール値の意味は以下のとおりです。

0 - 夏時間はアクティブになっていません。

1 - 夏時間はアクティブになっています。

他のアプリケーションとの対話

スクリプトで、ざまざまな QuickScript を使用して、他の Windows アプリケーションと対話できます。たとえば、以下の操作を実行できます。

- メモ帳などのアプリケーションを起動する
- アプリケーション タイトル名を確認する
- 特定のアプリケーションが実行しているかどうか確認する
- 実行中のアプリケーションをアクティブにする
- キーボード入力をシミュレートする
- アプリケーション ウィンドウを閉じる、最小化する、または最大化する
- コマンドを実行し、DDE をサポートするアプリケーションとデータを交換する

Windows アプリケーションの起動

スクリプトで、StartApp コマンドを使用すると、Windows アプリケーションを起動できます。

構文

```
StartApp appname;
```

パラメータ

AppName
起動するアプリケーションのパスおよびファイル名です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

注意：アプリケーションのパスおよびファイル名を把握している必要があります。アプリケーションが、Windows PATH 環境変数の一部であるディレクトリにある場合、ファイル名のみ（パスなし）を渡す必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

このスクリプトは、Microsoft の電卓を起動します。

```
StartApp "calc"
```

実行中のアプリケーションのアプリケーション タイトルの取得

スクリプトで、InfoAppTitle() 関数を使用すると、指定した実行中のアプリケーションのアプリケーション タイトルまたは Windows タスク リスト名を検索できます。この情報は、たとえば、InTouch スクリプトで、指定したアプリケーションが現在実行中かどうかを確認するとき、またはそれをアクティブにするときに必要です。

注意：この機能では、Internet Explorer の値は返されません。回避策として、Google Chrome ブラウザを使用してください。

InfoAppTitle() 関数 InfoAppTitle()かんすう

指定した実行中のアプリケーションのアプリケーション タイトルまたは Windows タスク リスト名を返します。

構文

```
result = InfoAppTitle (アプリ名)
```

パラメータ

appname
.exe 拡張子なしのアプリケーション名です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

例

このスクリプトは、"Calculator" を返します。

```
InfoAppTitle("calc")
```

このスクリプトは、"Microsoft Excel" を返します。

```
InfoAppTitle("excel")
```

アプリケーションが実行中かどうかを確認する

スクリプトで、InfoAppActive() 関数を使用すると、特定のアプリケーションがすでに実行中かどうかを確認できます。特定のアプリケーションが実行中かどうかを確認するには、そのアプリケーション タイトルまたは Windows タスク リスト名を把握している必要があります。

InfoAppActive() 関数

アプリケーションの実行状態を返します。

構文

```
result = InfoAppActive (apptitle)
```

パラメータ

apptitle
実行状態を照会するアプリケーションのアプリケーション タイトルまたは Windows タスク リストです。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

戻り値

ブール値の意味は、以下のとおりです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

0 - アプリケーションは実行していません。

1 - アプリケーションは実行しています。

例

このスクリプトは、メモ帳アプリケーションを照会し、実行中である場合はアクティブにします。実行していない場合、メモ帳の新しいインスタンスを起動します。このようにして、メモ帳を複数回起動することを避けます。

```
IF InfoAppActive(InfoAppTitle("Notepad"))==1
THEN
   ActivateApp InfoAppTitle( "Notepad" );
ELSE
   StartApp "Notepad";
ENDIF;
```

実行中の Windows アプリケーションをアクティブにする

スクリプトで、ActivateApp() 関数を使用すると、実行中の Windows アプリケーションをアクティブにできます。アクティブにすると、指定アプリケーションが前面に表示され、フォーカスが与えられます。

実行中の Windows アプリケーションをアクティブにするには、以下を実行する必要があります。

- アプリケーション タイトルまたは Windows タスク リスト名を検索する。「実行中のアプリケーションのアプリケーション タイトルの取得」を参照してください。
- Windows アプリケーションが実行中であることを確認する。「アプリケーションが実行中かどうかを確認する」を参照してください。

ActivateApp 関数

すでに実行中の Windows アプリケーションをアクティブ化します。

構文

```
ActivateApp apptitle;
```

パラメータ

apptitle
アクティブ化する実行中のアプリケーションのアプリケーション タイトルまたは Windows タスク リスト名です。

例

このスクリプトは、コマンド プロンプト ウィンドウが開いているかを確認し、開いている場合はアクティブ化します。開いていない場合、コマンド プロンプト ウィンドウを起動します。

```
IF InfoAppActive( InfoAppTitle("cmd")) == 1 THEN
   ActivateApp InfoAppTitle("cmd");
ELSE
   StartApp "cmd";
ENDIF;
```

シミュレートしたキー入力をアプリケーションに送信する

スクリプトで、キーボードの一連のキーを押す操作をシミュレートできます。たとえば、以下の操作を実行するために使用できます。

- 開いているアプリケーションでデータを自動的に入力する
- アプリケーション（InTouch HMI など）を制御する

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SendKeys 関数

一連のキー入力をシミュレートします。

重要：SendKeys() 関数は、64 ビット バージョンの Windows オペレーティング システムでは機能しません。

構文

```
SendKeys sequence;
```

パラメータ

sequence

シミュレートする一連のキー入力です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。
キーボードの通常の文字（英数字など）の他に、コードとして制御キーも指定できます。

```
{BACKSPACE} - Backspace キーをシミュレートします。
{BREAK} - Break キーをシミュレートします。
{CAPSLOCK} - Caps Lock キーをシミュレートします。
{DELETE} - Delete キー（または {DEL}）をシミュレートします。
{DOWN} - 下矢印キーをシミュレートします。
{END} - End キーをシミュレートします。
{ENTER} - Enter キー（または ~）をシミュレートします。
{ESCAPE} - ESC キー（または {ESC}）をシミュレートします。
{F1} ..{F12} - F1 ..F12 キーをシミュレートします。
{HOME} - Home キーをシミュレートします。
{INSERT} - Insert キーをシミュレートします。
{LEFT} - 左矢印キーをシミュレートします。
{NUMLOCK} - Num Lock キーをシミュレートします。
{PGDN} - Page Down キーをシミュレートします。
{PGUP} - Page Up キーをシミュレートします。
{PRTSC} - Print Screen キーをシミュレートします。
{RIGHT} - 右矢印キーをシミュレートします。
{TAB} - Tab キーをシミュレートします。
{UP} - 上矢印キーをシミュレートします。
+ - Shift キーをシミュレートします。
Shift キーと同時に押すキーは括弧で囲みます。
^ : Ctrl キーをシミュレートします。
Ctrl キーと同時に押すキーは括弧で囲みます。
% - Alt キーをシミュレートします。
Alt キーと同時に押すキーは括弧で囲みます。
```

備考

StartApp コマンドまたは ActivateApp() コマンド、あるいはその両方を使用すると、シミュレートしたキー入力送信する前に、別のアプリケーションをアクティブにできます。

例

このスクリプトは、B キーを押す操作をシミュレートしています。

```
SendKeys "b";
```

このスクリプトは、Ctrl キーを押しながら P を押す操作をシミュレートしています。これは、別のアプリケーションで［印刷］ダイアログ ボックスを起動するときに使用できます。

```
SendKeys "^(p)";
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

このスクリプトは、F1（ヘルプ機能を開く）、Tab キー（カーソルを検索フィールドに置く）の順に押し、HAL と入力して、Enter キーを押す（検索を開始する）操作をシミュレートしています。

```
SendKeys "{F1}{TAB}HAL{ENTER}";
```

このスクリプトは、Ctrl、Shift、およびキー 1 を押す操作をシミュレートしています。これは、WindowMaker への切り替えと同じです。この強力な組み合わせは、自動修正（動的） InTouch HMI アプリケーションの開発に使用できます。

```
SendKeys "^(+(1))";
```

Windows アプリケーションを閉じる、最小化する、または最大化する

スクリプトで、WWControl() コマンドを使用すると、別の Windows アプリケーションを閉じたり、最小化したり、最大化したりできます。

Windows アプリケーションを閉じたり、最小化したり、最大化したりするには、以下を実行する必要があります。

- アプリケーション タイトルまたは Windows タスク リスト名を検索する。「実行中のアプリケーションのアプリケーション タイトルの取得」を参照してください。
- Windows アプリケーションが実行中であることを確認する。「アプリケーションが実行中かどうかを確認する」を参照してください。

WWControl() 関数

Windows アプリケーションを復元、最小化、最大化、または終了します。

構文

```
WWControl (apptitle, control);
```

パラメータ

apptitle
元のサイズに戻す、最小化する、最大化する、または閉じる、実行中のアプリケーションのアプリケーション タイトルまたは Windows タスク リスト名です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

control
指定した Windows アプリケーションで実行するアクションを決めます。以下のいずれかの値のリテラル文字列値、メッセージ型タグ名、または文字列式です。

Restore - アプリケーション ウィンドウをアクティブにして、表示します。

Minimize - アプリケーション ウィンドウをアクティブにして、最小化します。

Maximize - アプリケーション ウィンドウをアクティブにして、最大化します。

Close - アプリケーションを閉じます。

例

このスクリプトは、すでに実行している場合、電卓アプリケーションを元のサイズに戻します。

```
WWControl ("Calculator","Restore");
```

このスクリプトは、WindowViewer を閉じます。

```
WWControl (InfoAppTitle("View"),"Close");
```

DDE を使用してコマンドの実行とデータの交換を行う

DDE をサポートするアプリケーションと対話するスクリプトを記述できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用場所	目的
WWExecute()	コマンドを送信および実行します。
WWRequest()	データを DDE アイテムから読み取ります。
WWPoke()	データを DDE アイテムに書き込みます。

WWExecute() 関数

コマンドをアプリケーションに送信し、それを実行して、状態結果を返します。これを使用すると、Excel でマクロを実行できます。

重要：WWExecute() 関数は、64 ビット バージョンの Windows オペレーティング システムでは機能しません。

構文

```
Result = WWExecute (appname, topic, command)
```

パラメータ

appname
コマンドを送信するアプリケーションの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

topic
コマンドを送信するアプリケーション内のトピックの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

command
送信されるコマンドです。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

戻り値

以下を示す -1、0、または 1 の値です。
-1 - コマンドは正常に実行されませんでした。原因として、アプリケーションが実行中でない、トピックが存在しない、またはコマンドでエラーが発生したことが考えられます。
0 - アプリケーションがビジー状態であるため、コマンドは正常に実行されませんでした。
1 - コマンドは正常に実行されました。

例

このスクリプトは、コマンド [Run("Macro1",0)] を Excel に送信することで、Microsoft Excel にマクロ Macro1 を実行するように指示します。

```
Macro="Macro1";
Command="[Run(" + StringChar(34) + Macro + StringChar(34) + ",0)]";
WWExecute("excel","system",Command);
```

WWRequest() 関数

アプリケーションのアイテムからデータを読み取ります。たとえば、これを使用して、Microsoft Excel のスプレッドシート セルの値を読み取ることができます。

重要：WWRequest() 関数は、64 ビット バージョンの Windows オペレーティング システムでは機能しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
Result = WWRequest(appname, topic, item, messagetag)
```

パラメータ

appname
アプリケーションの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

topic
アプリケーション内のトピックの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

item
トピックおよびアプリケーションに属すアイテムの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

messagetag
アイテムの値を受け取るメッセージ型タグ変数名です。メッセージ型タグ変数値は、StringToIntg() 関数または StringToReal() 関数を使用して、整数値または実数値に変換できます。

戻り値

以下を示す -1、0、または 1 の値です。
-1 - データは正常に読み取られませんでした。原因として、アプリケーションが実行中でない、またはトピックやアイテムが存在しないことが考えられます。
0 - アプリケーションがビジー状態であるため、データは正常に読み取られませんでした。
1 - データは正常に読み取られました。

例

このスクリプトは、Microsoft Excel ブック Book1.xls、シート Sheet1 の行 1、列 1 に含まれている値をメッセージ型タグ変数 MTag に読み取り、実数タグ変数 CellValue にこの値を配置します。

```
Result = WWRequest("excel","[Book1.xls]sheet1", "r1c1",Mtag);
CellValue=StringToReal(MTag);
```

英語以外のオペレーティング システムを使用している場合、StringReplace() 関数を使用して、別のデータ タイプに変換する前に *MTag* の内容を変更しなければならないことがあります。たとえば、小数点としてカンマを使用するオペレーティング システムでは、実数型データ タイプに変換する前に *MTag* 内のカンマをすべて小数点に置換しなければならない場合があります。

WWPoke() 関数

データをアプリケーションのアイテムに書き込みます。たとえば、これを使用して、Excel のスプレッドシート セルに値を書き込むことができます。

重要: WWPoke() 関数は、64 ビット バージョンの Windows オペレーティング システムでは機能しません。

構文

```
result = WWPoke (appname, topic, item, string)
```

パラメータ

appname
アプリケーションの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

topic

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーション内のトピックの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

item
トピックおよびアプリケーションに属すアイテムの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

string
書き込まれる値です。リテラル文字列値、メッセージ型タグ名、または文字列式です。StringFromIntg() 関数、StringFromReal() 関数、または Text() 関数を使用すると、整数型または実数型タグ名の値をメッセージ型タグ名に変換できます。

戻り値

以下を示す -1、0、または 1 の値です。
-1 - データは正常に書き込まれませんでした。原因として、アプリケーションが実行中でない、またはトピックやアイテムが存在しないことが考えられます。
0 - アプリケーションがビジー状態であるため、データは正常に書き込まれませんでした。
1 - データは正常に書き込まれました。

備考

WWPoke() 関数または WWRequest() 関数を使用して、InTouch アプリケーション間の異なるノードまたはセッションに対してデータの読み取りや書き込みを実行しないでください。InTouch アプリケーション間でデータを読み取りおよび書き込むには、アクセス名を使用します。「アクセス名の設定」を参照してください。

例

このスクリプトは、実数型タグ名 CellValue の値をメッセージ型タグ名 Mtag に配置し、その値を Microsoft Excel ブック Book1.xls のシート Sheet1 のスプレッドシート セル行 1、カラム 1 に書き込みます。

```
MTag = Text(CellValue,"0");
Result = WWPoke("excel","[Book1.xls]sheet1", "r1c1",Mtag);
```

ファイルを操作する

さまざまなファイル管理およびアクセス操作を使用するスクリプトを記述できます。

使用場所	目的
FileCopy()	ファイルをコピーします。
FileDelete()	ファイルを削除します。
FileMove()	ファイルを移動します。
FileReadFields()、FileWriteFields()	csv データの読み取り／書き込みを行います。
FileReadMessage()、FileWriteMessage()	テキスト データの読み取り／書き込みを行います。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ファイルを管理する

スクリプトでは、ファイルをコピー、削除、または移動できます。

FileCopy() 関数 FileCopy()かんすう

ソース ファイルをコピー先ファイルにコピーして、結果状態を返します。この関数の実行には時間がかかることがあり、以下の複数の段階で実行されます。

1. FileCopy() 関数が呼び出され、ファイル コピー開始が成功または失敗したかを示す結果が返されます。
2. FileCopy() 関数は、バックグランドでコピー操作を実行し、InTouch スクリプトは、ファイルのコピー中も実行を続けます。ファイル コピーの進行状況は、整数型タグを使用して監視できます。
3. FileCopy() 関数は、ファイル コピー操作が成功または失敗したことを示すファイル コピーの結果を返します。

コピー先フォルダが使用できない場合（ネットワークの別のコンピュータなど）、関数はタイムアウトまで最大 10 秒間待機し、メッセージを Logger にポストします。

注意 : FileCopy() 関数は、非同期クイック関数では使用しないでください。

構文

```
result = FileCopy (sourcefile, destfile, progresstag)
```

パラメータ

sourcefile
コピーされるファイルのフル パスおよびファイル名です。リテラル文字列値、メッセージ型タグ名、または文字列式です。このパラメータでワイルカード文字（* および ?）を使用すると、指定条件に一致するファイルだけをコピーできます。パス名には、UNC パス名も使用できます。

destfile
コピー先のフル パスおよびファイル名（またはパス名のみ）です。リテラル文字列値、メッセージ型タグ名、または文字列式です。パス名には、UNC パスも使用できます。

progresstag
ファイル コピーの進行状況を示す値を含む、二重引用符で囲まれた整数型タグの名前です。リテラル文字列値、メッセージ型タグ名（値 "IntTag.Name" を含むメッセージ型タグなど）、または文字列式です。値の意味は以下のとおりです。
0 - FileCopy() 操作は現在進行中です。
1 - FileCopy() 操作は正常に完了しました。
-1 - FileCopy() 操作はエラーで終了しました。

戻り値

以下を示す -1、0、または 1 の値です。
1 - FileCopy() 関数は、正常に呼び出されました。
0 - 別の FileCopy() 操作がすでに実行されているため、FileCopy() 関数を呼び出すときにエラーが発生しました。
-1 - ソース ファイルが存在しないか、コピー先ファイルが読み取り専用であるため、FileCopy() 関数を呼び出すときにエラーが発生しました。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

このスクリプトは、ファイル c:\MyData\output.log をディレクトリ d:\archive にコピーして、ファイルの名前を output.txt に変更します。ファイル コピーの進行状況は、整数型タグ Monitor に書き込まれます。

```
Status=FileCopy("c:\MyData\output.log","d:\archive\output.txt","Monitor");
```

このスクリプトは、c:\ ルート ディレクトリにあり、名前が .txt で終わるすべてのファイルをコピー先ディレクトリ c:\Backup にコピーします。

```
Status=FileCopy("c:\*.txt", "c:\Backup", "Monitor");
```

このスクリプトは、フル パスおよびファイル名がメッセージ型タグ LogFile に含まれるファイルをコピー先ディレクトリ c:\results\ にコピーして、そのファイルの名前を logxxx.txt に変更します。ここで、xxx はタイムスタンプです。

```
Status=FileCopy(LogFile, "c:\results\log" + $DateString + $TimeString + ".txt", "Monitor");
```

FileDelete() 関数 FileDelete()かんすう

個々のファイルを削除します。

構文

```
result = FileDelete (filename)
```

パラメータ

filename
削除するファイルのパス名およびファイル名です。リテラル文字列値、メッセージ型タグ名、または文字列式です。UNC パス名がサポートされています。

備考

FileDelete() 関数にはワイルドカード文字（* および ?）を使用しないでください。また、FileDelete() 関数を非同期クイック関数で使用しないでください。

FileDelete() 関数は、ディレクトリを削除しません。

戻り値

ファイル削除の成功または失敗を示す値です。
1 - ファイルは正常に削除されました。
0 - ファイルは正常に削除されませんでした。原因として、読み取り専用ファイルを削除しようとしたか、ファイルが存在しないことが考えられます。

例

このスクリプトは、ファイル c:\Data.txt を削除して、ファイルが検索あり、正常に削除された場合は 1 を返します。

```
Status=FileDelete("c:\Data.txt");
```

FileMove() 関数 FileMove()かんすう

ソース ファイルを移動先ファイルに移動して、結果状態を返します。また、ファイルの名前を変更するときにも使用できます。この関数の実行には時間がかかることがあり、以下の複数の段階で実行されます。

1. FileMove() 関数が呼び出され、ファイル移動開始が成功または失敗したかを示す結果が返されます。
2. FileMove() 関数は、バックグランドで移動操作を実行し、InTouch スクリプトは、ファイルの移動中も実行を続けます。ファイル移動の進行状況は、整数型タグを使用して監視できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. FileMove() 関数は、ファイル移動操作が成功または失敗したかを示す、ファイル移動の結果を返します。

FileMove() 関数は、非同期クイック関数では使用しないでください。

構文

```
result = FileMove (sourcefile, destfile, progresstag)
```

パラメータ

sourcefile
移動されるファイルのフル パスおよびファイル名です。リテラル文字列値、メッセージ型タグ名、または文字列式です。このパラメータでワイルカード文字（* および ?）を使用すると、指定条件に一致するファイルだけを移動できます。パス名には、UNC パス名も使用できます。

destfile
コピー先のフル パスおよびファイル名（またはパス名のみ）です。リテラル文字列値、メッセージ型タグ名、または文字列式です。パス名には、UNC パスも使用できます。

progresstag
ファイル移動の進行状況を示す値を含む、二重引用符で囲まれた整数型タグの名前です。リテラル文字列値、メッセージ型タグ名（値 "IntTag" を含むメッセージ型タグなど）、または文字列式です。値の意味は以下のとおりです。

0 - FileMove() 操作は現在進行中です。

1 - FileMove() 操作は正常に完了しました。

-1 - FileMove() 操作はエラーで終了しました。

戻り値

以下を示す -1、0、または 1 の値です。
1 - FileMove() 関数は、正常に呼び出されました。
0 - 別の FileMove() 操作がすでに実行されているため、FileMove() 関数を呼び出すときにエラーが発生しました。
-1 - FileMove() 関数を呼び出すときにエラーが発生しました。存在しないファイルを移動しようとしたなどのエラーが考えられます。

例

このスクリプトは、ファイル c:\MyData\output.log をディレクトリ d:\archive に移動して、ファイルの名前を output.txt に変更します。ファイル移動の進行状況は、整数型タグ Monitor に書き込まれます。

```
Status=FileMove("c:\MyData\output.log","d:\archive\output.txt","Monitor");
```

このスクリプトは、c:\ ルート ディレクトリにあり名前が .txt で終わるすべてのファイルを移動先ディレクトリ c:\Backup に移動します。

```
Status=FileMove("c:\*.txt", "c:\Backup", "Monitor");
```

このスクリプトは、フル パスおよびファイル名がメッセージ型タグ LogFile に含まれるファイルを移動先ディレクトリ c:\results\ に移動して、そのファイルの名前を logxxx.txt に変更します。ここで、xxx はタイムスタンプです。

```
Status=FileMove(LogFile, "c:\results\log" + $DateString + $TimeString + ".txt", "Monitor");
```

CSV データの読み取りと書き込み

FileReadFields() 関数および FileWriteFields() 関数を使用すると、一連のタグ変数に対して csv（カンマ区切り形式）ファイルに含まれているデータを読み取りおよび書き込むスクリプトを作成できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

FileReadFields() 関数および FileWriteFields() 関数は、カンマを区切り文字としてのみサポートしています。

FileReadFields() 関数 FileReadFields()かんすう

csv ファイルに含まれている値を一連のタグ変数に読み取ります。この関数を使用すると、一組のタグ変数値をロードできます。

サポートされている区切り文字はカンマだけです。

この関数は、同期呼び出しでのみ使用できます。

構文

```
[result = ] FileReadFields (filename, offset, starttag, numberoffields)
```

パラメータ

filename
データを読み取る csv ファイルの名前です。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

offset
読み取りを開始するファイルの位置（バイト単位）です。リテラル整数値、整数型タグ名、または整数式です。

starttag
最初の読み取りデータ アイテムを受け取る最初のタグ変数の名前です。タグ名は、"MyTag1" のように、二重引用符で囲み、数値で終わる必要があります。リテラル文字列値、メッセージ型タグ変数名（値 "MyTag1" を含むメッセージ型タグ変数など）、または文字列式です。

numberoffields
csv ファイルから読み取るデータ アイテムの名前です。リテラル整数値、整数型タグ名、または整数式です。最初のデータ アイテムは、starttag パラメータで定義されているタグ変数に読み取られ、その後のデータ アイテムは、starttag パラメータに接尾辞として増分値が付けられたタグ変数（MyTag1、MyTag2、MyTag3、....）に読み取られます。

戻り値

データ読み取り後のオプションの新しいファイル オフセット（バイト単位）です。これは、データの次のセットを読み取るときに使用できます。

例

このスクリプトは、値 "Flour" を RecipeTag1、27.23 を RecipeTag2、14 を RecipeTag3、および 1 を RecipeTag4 に読み取り、新しいオフセットを返します。この csv ファイル c:\set.csv には、データ Flour、27.23、14、1 が含まれていて、タグ変数 RecipeTag1:message、RecipeTag2:real、Recipe3:integer、RecipeTag4:discrete が定義されています。

```
FileReadFields("c:\set.csv",0,"RecipeTag1",4);
```

FileWriteFields() 関数 FileWriteFields()かんすう

一連のタグ名に含まれている値を csv ファイルに書き込みます。この関数を使用すると、タグ名の値のセットを保存できます。

サポートされている区切り文字はカンマだけです。

構文

```
[result = ] FileWriteFields (filename, offset, starttag, numberoffields)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

filename
データを書き込む csv ファイルの名前です。ファイルが存在しない場合、新しいファイルが作成されます。リテラル文字列値、メッセージ型タグ名、または文字列式です。

offset
書き込みを開始するファイルの位置（バイト単位）です。ファイルの最後に書き込むには（追加）、-1 を使用します。リテラル整数値、整数型タグ名、または整数式です。

starttag
書き込まれる最初のデータ アイテムを含む最初のタグ名の名前です。タグ名は、"MyTag1" のように、二重引用符で囲み、数値で終わる必要があります。リテラル文字列値、メッセージ型タグ名（値 "MyTag1" を含むメッセージ型タグなど）、または文字列式です。

numberoffields
csv ファイルに書き込むデータ アイテムの名前です。リテラル整数値、整数型タグ名、または整数式です。最初のデータ アイテムは、starttag パラメータで定義されているタグ名からファイルに書き込まれ、その後のデータ アイテムは、starttag パラメータの増分数字の接尾辞付きのタグ名（MyTag1、MyTag2、MyTag3...）から書き込まれます。

戻り値

データ書き込み後のオプションの新しいファイル オフセット（バイト単位）です。これは、データの次のセットを書き込むときに使用できます。

例

一連の InTouch タグは以下のように定義されています。

タグ名	データ型	値
RecipeTag1	メッセージ型	Flour
RecipeTag2	実数型	27.23
RecipeTag3	整数型	14
RecipeTag4	論理型	1

このスクリプトは、RecipeTag1 ～ RecipeTag4 に含まれている値を csv ファイル c:\set.csv に書き込みます。

```
FileWriteFields("c:\set.csv",0,"RecipeTag1",4);
```

これで、ファイル c:\set.csv には以下のデータが含まれます。

```
Flour,27.23,14,1
```

テキスト データの読み取りと書き込み

FileReadMessage() 関数および FileWriteMessage() 関数を使用すると、ファイルに対してテキスト データを読み取りまたは書き込みを行うスクリプトを作成できます。指定バイト数、またはテキストのライン全体（ライン フィードで区切られる）を読み取りまたは書き込みすることができます。

FileReadMessage() 関数 FileReadMessage()かんすう

ファイルから文字列データの指定バイト数（または 1 つの行）を読み取ります。

構文

```
[result = ] FileReadMessage (filename, offset, messagetag, charstoread)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

filename
データを読み取るファイルの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

offset
読み取りを開始するファイルの位置（バイト単位）です。リテラル整数値、整数型タグ名、または整数式です。

messagetag
ファイルから最初のラインまたはバイト数を受け取るメッセージ型タグ名です。産業用グラフィックエディタのスクリプト エディタ内で関数を使用する場合は、タグ名を二重引用符で囲みます。

charstoread
ファイルから読み取るバイト数です。ライン フィード記号（LF）まで読み取る場合、0 に設定します。リテラル整数値、整数型タグ名、または整数式です。

戻り値

読み取り後の新しいバイト位置を含みます。この値は、ファイルから続けて読み取りを行う場合に使用できます。

例

次のスクリプトは、ファイル c:\Data\File.txt のデータの最初のラインをメッセージ型タグ変数 MsgTag に読み取ります。

```
FileReadMessage ("c:\Data\File.txt",0,MsgTag, 0);
FileReadMessage ("c:\Data\File.txt",0,"InTouch:MsgTag", 0);
```

FileWriteMessage() 関数 FileWriteMessage()かんすう

ファイルに文字列データの指定バイト数（または 1 つの行）を書き込みます。

構文

```
[result = ] FileWriteMessage (filename, offset, messagetag, linefeed)
```

パラメータ

filename
データを書き込むファイルの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

offset
書き込みを開始するファイルの位置（バイト単位）です。ファイルの最後にデータ書き込むには（追加）、-1 に設定します。リテラル整数値、整数型タグ名、または整数式です。

messagetag
ファイルに書き込むデータを含むメッセージ型タグ名です。

linefeed
データをファイルに書き込んだ後、ライン フィード記号（LF）を書き込むかどうかを指定します。ライン フィード記号を書き込む場合、1 に設定します。書き込まない場合、0 に設定します。リテラル ブール値、論理型タグ変数、またはブール式です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

戻り値

書き込み後の新しいバイト位置を含みます。この値は、ファイルに続けて書き込みを行う場合に使用できます。

例

次のスクリプトは、メッセージ型タグ名 MsgTag の値をファイル c:\Data\File.txt の最後に書き込みます。

```
FileWriteMessage("c:\Data\File.txt",-1,MsgTag,1);
```

システム関連情報を取得する

スクリプトで、以下のクイック関数を使用すると、システム関連情報を取得できます。

使用場所	目的
GetNodeName()	コンピュータのノード名を取得します。
InfoDisk()	ディスク空き容量情報を取得します。
InfoFile()	ファイル情報を取得します。

コンピュータのノード名を取得する

スクリプトで、GetNodeName() 関数を使用すると、コンピュータのノード名を取得できます。この関数は、たとえばアクセス名を操作するときに、InTouch アプリケーションを動的に保つ場合に使用できます。

GetNodeName() 関数

コンピュータのノード名を返します。

構文

```
GetNodeName (messagetag, nodenum);
```

パラメータ

messagetag
ノード名を含むメッセージ型タグ名です。産業用グラフィック エディタのスクリプト エディタ内で関数を使用する場合は、タグ名を二重引用符で囲みます。

nodenum
ノード名から取得する文字数です。0 ～ 131 のリテラル整数値、整数型タグ名、または整数式です。

例

次のスクリプトは、ノード名を取得して、NodeName メッセージ型タグ変数に割り当てます。

```
GetNodeName(NodeName,131);
GetNodeName("InTouch:NodeName",131);
```

ディスク空き容量情報を取得する

スクリプトで、InfoDisk() 関数を使用すると、ディスク容量情報を取得できます。以下の情報を取得できます。

- ディスク ドライブの合計サイズ（バイトまたはキロバイト単位）
- ディスク ドライブの使用可能な空き容量（バイトまたはキロバイト単位）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

また、トリガ タグ変数を指定すると、(アニメーション リンクで）情報を取得するタイミングや方法を指定することもできます。

InfoDisk() 関数 InfoDisk()かんすう

ローカルまたはネットワーク ディスク ドライブの合計容量または空き容量を返します。

構文

```
result = InfoDisk (drive, infotype, trigger);
```

パラメータ

drive
情報を取得するドライブ文字です。文字列の最初の文字だけが使用されます。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

infotype
情報タイプを指定します。以下のいずれかの値のリテラル整数値、整数型タグ変数、または整数式です。

1 - ディスク ドライブの合計サイズを返します（バイト単位)。

2 - ディスク ドライブの空き容量を返します（バイト単位)。

3 - ディスク ドライブの合計サイズを返します（キロバイト単位)。

4 - ディスク ドライブの空き容量を返します（キロバイト単位)。

trigger
ディスク情報を再計算するトリガとして機能するタグ変数（または式）です。トリガ値が変わると、ディスク情報が再計算されます。論理型またはアナログ型のタグ名、または論理型またはアナログ型の式です。

備考

トリガ タグ変数は、InfoDisk() 関数がアニメーション表示リンクで使用される場合だけ意味があります。この関数がスクリプトで使用される場合、リテラル数値、アナログ型タグ名または数式を指定できます。

例

アニメーション表示リンクで以下のスクリプトを使用すると、ディスク ドライブ C の空き容量を表示して、その情報を毎分更新できます。

```
InfoDisk("C", 4, $Minute)
```

ファイルまたはディレクトリの情報を取得する

スクリプトで、InfoFile() 関数を使用すると、特定のファイルまたはディレクトリに関する情報を取得できます。さまざまなパラメータを使用することで、以下の情報を取得できます。

- ファイルが存在するかどうか
- 指定ファイル名が実際はディレクトリかどうか
- ファイルのサイズ（バイト単位)
- ファイルまたはディレクトリのタイムスタンプ
- ワイルドカード検索に一致するファイル数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InfoFile() 関数 InfoFile()かんすう

ファイルまたはディレクトリのさまざまな情報を返します。

構文

```
result = InfoFile (filename, infotype, trigger)
```

パラメータ

filename
情報を取得するフル ファイル名またはディレクトリ名です。リテラル文字列値、メッセージ型タグ名、または文字列式です。"*" および "?" などのワイルドカード文字を含めることもできます。

infotype
指定ファイルまたはディレクトリについて取得する情報のタイプです。リテラル整数値、整数型タグ名、または整数式です。値と意味は以下のとおりです。

1 - 存在。InfoFile() 関数は、ファイルが存在する場合は 1、ファイルがディレクトリの場合は 2、ファイルまたはディレクトリが存在しない場合は 0 を返します。

2 - サイズ。ファイルのサイズをバイト単位で返します。

3 - 作成タイムスタンプ。InfoFile() 関数は、タイムスタンプを 1970 年 1 月 1 日の真夜中以降に経過した秒数として返します。StringFromTimeLocal() 関数を使用すると、この値をメッセージ タイムスタンプに変換できます。

4 - ワイルドカード検索一致。InfoFile() 関数は、指定したワイルドカード検索に一致するファイル数を返します。

trigger
ファイル情報を再計算するトリガとして機能するタグ名（または式）です。トリガ値が変わると、ファイル情報が再計算されます。論理型またはアナログ型のタグ名、または論理型またはアナログ型の式です。

備考

トリガ タグは、InfoFile() 関数がアニメーション表示リンクで使用される場合だけ意味があります。この関数がスクリプトで使用される場合、リテラル数値、アナログ型タグ名または数式を指定できます。

例

このスクリプトは、ファイル c:\data\log.txt が存在する場合 1 を返します。

```
InfoFile("c:\data\log.txt",1,$minute)
```

このスクリプトは、ファイル c:\data\log.txt のファイル サイズが 14223 のバイトの場合 14223 を返します。

```
InfoFile("c:\data\log.txt",2,$minute)
```

このスクリプトは、ファイル c:\data\log.txt が 2006 年 1 月 26 日 11:14:26 AM に作成された場合 1138245266 を返します。

```
InfoFile("c:\data\log.txt",3,$minute)
```

このスクリプトは、c:\data\ ディレクトリに txt で終了するファイルが 14 個ある場合 14 を返します。

```
InfoFile("c:\data\*.txt",4,$minute)
```

InTouch 関連情報を取得する

スクリプトで、以下の関数を使用すると、InTouch 関連情報を取得できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用場所	目的
InfoInTouchAppDir()	開発している InTouch アプリケーションのディレクトリ情報を取得します。
InTouchVersion()	InTouch バージョン情報を取得します。

InTouch アプリケーション ディレクトリの名前を取得する

スクリプトで、InfoInTouchAppDir() 関数を使用すると、InTouch アプリケーションが実行されているディレクトリの名前を取得できます。この関数は、InTouch アプリケーションに同梱する外部ファイルを検索する場合に役立ちます。

InfoInTouchAppDir() 関数 InfoInTouchAppDir()かんすう

現在の InTouch アプリケーション ディレクトリを返します。

構文

```
result = InfoInTouchAppDir();
```

戻り値

現在実行している InTouch アプリケーションのディレクトリを含めるメッセージ型タグ変数です。

備考

文字数が 131 文字に制限されているため、メッセージ型タグ変数に渡されるとき、またはアニメーション リンクで表示されるときに、アプリケーション ディレクトリ名が切り捨てられることがあります。

例

このスクリプトは、c:\documents and settings\user1\my documents\マイ InTouch アプリケーション\packaging を返します。

```
InfoInTouchAppDir()
```

InTouch バージョンを取得する

スクリプトで、InTouchVersion() 関数を使用すると、現在実行している InTouch アプリケーションのバージョン番号を取得できます。

InTouchVersion() 関数 InTouchVersion()かんすう

完全な InTouch バージョン番号、またはその一部を返します。

構文

```
result = InTouchVersion (infotype);
```

パラメータ

infotype
バージョン情報がどのように返されるかを指定します。リテラル整数値、整数型タグ変数、または整数式の意味は以下のとおりです。

0 - 完全なバージョン番号を返します。

1 - メジャー バージョン番号を返します。

2 - マイナー バージョン番号を返します。

3 - パッチ レベルを返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4 - ビルド レベルを返します。

例

機能	考えられる結果
InTouchVersion(0)	10.5.1626.0521.0045.0012
InTouchVersion(1)	10
InTouchVersion(2)	5
InTouchVersion(3)	0
InTouchVersion(4)	1626

セキュリティ関連スクリプト

さまざまな QuickScript 関数およびシステム タグ変数を使用すると、InTouch アプリケーションでセキュリティを追加および管理できます。セキュリティ機能の詳細については、『*AVEVA™ InTouch HMI 管理*』の「InTouch のセキュリティ保護」を参照してください。

ログオンとログオフ

以下の関数とシステム タグ変数を使用して、ログオンおよびログオフできます。

使用場所	目的
AttemptInvisibleLogon()	認証データをパラメータに提供することで、ユーザーをログオンします。
LogonCurrentUser()	現在 Windows ユーザーにログオンしているユーザーをログオンします（認証モードが "OS" の場合です）。
PostLogonDialog()	［**ログオン**］ダイアログ ボックスを表示します。
Logoff()	現在のユーザーをログオフします。
$PasswordEntered	パスワードを設定します。
$OperatorEntered	有効なユーザー名を設定します。
$OperatorDomainEntered	有効なユーザー ドメイン名を設定します（認証モードが "OS" の場合）。

セキュリティ機能の詳細については、『*AVEVA™ InTouch HMI 管理*』の「InTouch のセキュリティ保護」を参照してください。

パスワードの変更と設定

以下の関数およびシステム タグ変数をパスワードの変更に使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用場所	目的
ChangePassword()Change Password()	現在ログオンしているユーザーに対して、[パスワードの変更] ダイアログ ボックスを呼び出します。
$ChangePassword	現在ログオンしているユーザーに対して、[パスワードの変更] ダイアログ ボックスを呼び出します。

セキュリティ機能の詳細については、『*AVEVA™ InTouch HMI 管理*』の「InTouch のセキュリティ保護」を参照してください。

ユーザーの指定と設定

以下のシステム タグ変数をユーザーの指定および設定に使用できます。

使用場所	目的
$ConfigureUsers	[ユーザーの設定] ダイアログ ボックスを呼び出します。

セキュリティ機能の詳細については、『*AVEVA™ InTouch HMI 管理*』の「InTouch のセキュリティ保護」を参照してください。

セキュリティとその他の情報を管理する

以下のシステム タグ変数および関数をセキュリティの管理に使用できます。

使用場所	目的
$AccessLevel	現在ログイン中のユーザーのアクセス レベルを取得します。
AddPermission()	アクセス レベルを特定のユーザーグループ（ローカル/ドメイン）に割り当てます。
GetAccountStatus()	アカウント情報（パスワードの使用期限、ロック、無効フラグ）を取得します。
$InactivityTimeout	ユーザーが自動的にログオフされるまでに経過する時間を示します。
$InactivityWarning	タイムアウト警告の時間を示します。
InvisibleVerifyCredentials()	OS ユーザーの InTouch アクセス レベル情報を取得します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用場所	目的
IsAssignedRole()	現在ログオンしているユーザーが、指定したユーザー役割であるかどうかを確認します。
QueryGroupMembership()	現在ログオンしているユーザーが指定したユーザー役割のメンバであるかどうかを確認します。

セキュリティ機能の詳細については、『*AVEVA™ InTouch HMI 管理*』の「InTouch のセキュリティ保護」を参照してください。

その他のスクリプト

InTouch スクリプトは、ヒューマン マシン インタラクションと音声を関連付けることができるように、音声出力をサポートしています。また、InTouch スクリプトは、ウィザードのプロパティの取得および設定もサポートしています。

InTouch アプリケーションから音声ファイルを再生する

スクリプトで、イベントおよび条件を特定の音声に関連付けることができます。たとえば、警告ダイアログ ボックスまたは重要な条件を警告音に関連付けることができます。

PlaySound() 関数

wave ファイルの音声または Windows のデフォルト音声を再生します。

構文

```
Playsound (soundname, flag)
```

パラメータ

soundname
音声または wave ファイルの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。音声が名前として定義されている場合、たとえば、MC="c:\test.wav" のように、Win.ini ファイルの [Sounds] セクションで定義されている必要があります。

flag
音声の再生方法を指定します。リテラル整数値、整数型タグ名、または整数式です。値と意味は以下のとおりです。
0 - 音声を一度に同時に再生します（音声再生が終了するまで、スクリプトは実行を待機します）。
1 - 音声を一度に非同期に再生します（スクリプトは、音声再生の終了を待機しません）。
9 - 音声を連続して再生します（PlaySound() 関数が再び呼び出されるまでです）。

例

このスクリプトは、ファイル c:\welcome.wav の音声を一度再生して、再生が終了するまでスクリプト実行を停止します。

```
PlaySound("c:\welcome.wav",0);
```

このスクリプトは、音声 Alert を連続して再生します。win.ini ファイル [Sounds] セクションで、以下のように、音声名 Alert と音声ファイルを関連付ける必要があります。

```
Alert=c:\alert.wav.
PlaySound("Alert",9);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィザードのプロパティの取得と設定

分散アラーム オブジェクトやウィンドウ コントロールなどのいくつかのウィザードには、設定プロパティまたは読み取りプロパティが含まれています。これらのプロパティは、テキスト ボックスの値、またはチェックボックスのオンまたはオフ状態です。

スクリプトで、以下の関数を使用すると、これらのプロパティにアクセスできます。

使用場所	目的
SetPropertyD()、GetPropertyD()	論理型プロパティを設定または読み取ります。
SetPropertyI()、GetPropertyI()	整数型プロパティを設定または読み取ります。
SetPropertyM()、GetPropertyM()	メッセージ型プロパティを設定または読み取ります。

サポートされているプロパティのリストについては、ウィザードの説明を参照してください。

これらのプロパティを設定したり読み取ったりする一般的な方法を以下に示します。

GetPropertyD() 関数

ウィザードの論理型プロパティを読み取り、成功コードを返します。

構文

```
result = GetPropertyD (controlname.property, dtag)
```

パラメータ

controlname
プロパティをサポートするウィザードの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

property
読み取られるウィザードの論理型プロパティです。controlname と組み合わせて、リテラル文字列値、メッセージ型タグ名、または文字列式を使用できます。

dtag
論理型プロパティ値を取得する論理型タグ名です。

戻り値

整数型のエラー コードです。エラー コードの詳細については、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

例

チェックボックス ウィザードが Checkbox1、論理型タグ名が dtag の場合、以下のスクリプト関数を使用すると、このチェックボックスの表示または非表示を確認できます。

```
result=GetPropertyD("Checkbox1.visible",dtag);
```

このスクリプトは、チェックボックス ウィザードが表示可能な場合、dtag を 1 に設定し、表示不可の場合 0 に設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SetPropertyD() 関数

ウィザードの論理型プロパティを設定して、成功コードを返します。

構文

```
result = SetPropertyD(controlname.property, Boolean)
```

パラメータ

controlname
プロパティをサポートするウィザードの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

property
設定されるウィザードの論理型プロパティです。controlname と組み合わせて、リテラル文字列値、メッセージ型タグ名、または文字列式を使用できます。

ブール型
ウィザード プロパティに渡すブール値です。リテラル ブール値、論理型タグ名、またはブール式です。

戻り値

整数型のエラー コードです。エラー コードの詳細については、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

例

チェックボックス ウィザードが Checkbox1、論理型タグ名が dtag の場合、以下のスクリプト関数を使用すると、このチェックボックスの表示または非表示を制御できます。

```
result=SetPropertyD("Checkbox1.visible",dtag);
```

dtag を 0 に設定し、上記のスクリプト関数を呼び出すと、チェックボックス ウィザードは非表示となります。

GetPropertyI() 関数

ウィザードで整数を読み取り、成功コードを返します。

構文

```
result = GetPropertyI (controlname.property, itag)
```

パラメータ

controlname
プロパティをサポートするウィザードの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

property
読み取られるウィザードの整数型プロパティです。controlname と組み合わせて、リテラル文字列値、メッセージ型タグ名、または文字列式を使用できます。

itag
整数型プロパティ値を取得する整数型タグ名です。

戻り値

整数型のエラー コードです。エラー コードの詳細については、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

ラジオ ボタン ウィザードが Radiobutton1、整数型タグ名が itag の場合、以下のスクリプト関数を使用すると、このラジオ ボタン グループで現在選択されているアイテムを確認できます。

```
result=GetPropertyI("Radiobutton1.value",itag);
```

このスクリプトは、最初（2 つ目、3 つ目...）のラジオ ボタンが選択されている場合、itag を 1（2、3...）に設定します。

SetPropertyI() 関数

ウィザードで整数型プロパティを設定して、成功コードを返します。

構文

```
result = SetPropertyI (controlname.property, integer)
```

パラメータ

controlname
プロパティをサポートするウィザードの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

property
設定されるウィザードの整数型プロパティです。controlname と組み合わせて、リテラル文字列値、メッセージ型タグ名、または文字列式を使用できます。

integer
ウィザード プロパティに渡す整数値です。リテラル整数値、整数型タグ名、または整数式です。

戻り値

整数型のエラー コードです。エラー コードの詳細については、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

例

ラジオ ボタン ウィザードが Radiobutton1 の場合、以下のスクリプト関数を使用すると、2 つ目のラジオ ボタンを設定できます。

```
result=SetPropertyI("Radiobutton1.value",2);
```

GetPropertyM() 関数

ウィザードのメッセージ型プロパティを読み取り、成功コードを返します。

構文

```
result = GetPropertyM (controlname.property, mtag)
```

パラメータ

controlname
プロパティをサポートするウィザードの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

property
読み取られるウィザードのメッセージ型プロパティです。controlname と組み合わせて、リテラル文字列値、メッセージ型タグ名、または文字列式を使用できます。

mtag
メッセージ型プロパティ値を取得するメッセージ型タグ名です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

戻り値

整数型のエラー コードです。エラー コードの詳細については、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

例

チェックボックス ウィザードが Checkbox1、メッセージ型タグ名が mtag の場合、以下のスクリプト関数を使用すると、このチェックボックスのキャプションを確認できます。

```
result=GetPropertyM("Checkbox1.caption",mtag);
```

このスクリプトは、mtag をチェックボックスのキャプションに設定します。

SetPropertyM() 関数

ウィザードのメッセージ型プロパティを設定して、成功コードを返します。

構文

```
result = SetPropertyM (controlname.property, message)
```

パラメータ

controlname
プロパティをサポートするウィザードの名前です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

property
設定されるウィザードのメッセージ型プロパティです。controlname と組み合わせて、リテラル文字列値、メッセージ型タグ名、または文字列式を使用できます。

メッセージ型
ウィザード プロパティに渡すメッセージ値です。リテラル文字列値、メッセージ型タグ名、または文字列式です。

戻り値

整数型のエラー コードです。エラー コードの詳細については、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

例

チェックボックス ウィザードが Checkbox1 の場合、以下のスクリプト関数を使用すると、このチェックボックス ウィザードのキャプションを動的に設定できます。

```
result=SetPropertyM("Checkbox1.caption","Start Engine 1");
```

このスクリプトは、チェックボックス Checkbox1 のキャプションを "Start Engine 1" に設定します。

OLE オブジェクトを使用したスクリプト

OLE オブジェクトを使用すると、InTouch HMI アプリケーションの機能を拡張できます。OLE オブジェクトを使用すると、以下の操作を実行できます。

- オペレータ インターフェイスのポップアップ ダイアログ ボックスを作成する。
- コントロール パネルなど、オペレーティング システム の機能にアクセスする。
- Manufacturing Execution Module のデータを InTouch HMI 内での処理に使用できるようにする。Manufacturing Execution Module のマニュアルを参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

OLE オブジェクトの作成、確認および解放

InTouch スクリプトで使用する OLE オブジェクトを作成および確認できます。OLE オブジェクトを使用したら、解放することによって、メモリを確保できます。

以下の関数を使用すると、OLE オブジェクトを作成、確認、および解放できます。

- OLE_CreateObject() 関数
- OLE_IsObjectValid() 関数
- OLE_ReleaseObject() 関数

OLE_CreateObject() 関数

スクリプトで OLE オブジェクトを参照するには、最初に OLE オブジェクトを作成する必要があります。OLE オブジェクトを作成すると、それを参照するポインタを受け取ります。

スクリプトで、OLE_CreateObject() 関数を使用することによって、OLE オブジェクトを作成して、ポインタを割り当てることができます。

構文

```
OLE_CreateObject(%pointer, classname);
```

パラメータ

%pointer
OLE オブジェクトへのポインタの名前です。英数字（A ～ Z、0 ～ 9）およびアンダースコアを含めることができます。大文字と小文字は区別されません。

classname
OLE クラスの名前です。クラス名では大文字と小文字が区別されます。リテラル文字列値、メッセージ型タグ変数、または文字列式です。

備考

同じオブジェクト名を使用して別のオブジェクトを作成すると、そのオブジェクトは更新され、新しい OLE クラスを参照します。これは、古い OLE クラスから解放されます。

例

このスクリプトでは、クラス Wscript.Shell を参照する %WShell と呼ばれる OLE オブジェクト が作成されます。

```
OLE_CreateObject(%WShell, "Wscript.Shell");
```

OLE_IsObjectValid() 関数

スクリプトで、OLE_IsObjectValid() 関数を使用することによって、OLE オブジェクトが有効であるかどうかを確認できます。これは、OLE オブジェクトを操作する場合に必要な手順ではありませんが、OLE オブジェクトを操作する場合に問題が発生しないことを確認するために推奨します。

構文

```
result = OLE_IsObjectValid(%pointer)
```

引数

%pointer
テストされる OLE オブジェクトを参照するポインタです。

result

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以下の内容を示すブール値です。

0 - ポインタが参照している OLE オブジェクトが無効です。

1 - ポインタが参照している OLE オブジェクトが有効です。

例

このスクリプトでは、Wscript.Shell クラスに基づいて OLE クラスが作成され、それを参照するポインタ %WS が作成されます。isvalid は、OLE オブジェクトが正常に作成された場合に TRUE となる論理型タグ変数です。それ以外の場合、FALSE となります。

```
OLE_CreateObject(%WS, "Wscript.Shell");
isvalid = OLE_IsObjectValid(%WS);
```

OLE_ReleaseObject() 関数

OLE オブジェクトをスクリプトで使用した後でこれを解放し、そのポインタを削除して、システム リソースを確保できます。OLE オブジェクトを削除すると、そのポイントを使用して、関連付けられている OLE クラスのプロパティおよびメソッドにアクセスできなくなります。

構文

```
OLE_ReleaseObject(%pointer);
```

引数

%pointer
OLE オブジェクトを参照するポインタ名です。英数字（A ～ Z、0 ～ 9）およびアンダースコアを含めることができます。大文字と小文字は区別されません。

例

このスクリプトでは、ポイント %WShell に関連付けられている OLE オブジェクトが解放され、ポインタ %WShell が削除されます。

```
OLE_ReleaseObject(%WShell);
```

OLE オブジェクトのプロパティとメソッドを使用する

スクリプトでは、ポインタを使用して、OLE プロパティに対して値の読み取りおよび書き込みを行うことができます。また、ポインタを使用して、OLE メソッドを呼び出すこともできます。使用できるプロパティおよびメソッドは、OLE オブジェクトにより異なります。

OLE オブジェクトのプロパティにアクセスする

スクリプトで、ほとんどのプログラム言語で行うように、OLE オブジェクトのプロパティにアクセスできます。プロパティは通常、ドット "." 演算子を使用して、識別されます。

注意 : OLE オブジェクト プロパティをスクリプトで使用する場合、先頭の "%" を含め、リファレンスが 98 文字を超えないようにしてください。OLE ポインタ名はできるだけ短くしてください。

OLE オブジェクト プロパティを読み取る

スクリプトでは、プロパティをタグ変数に割り当てることによって、OLE オブジェクト プロパティを読み取ることができます。OLE オブジェクト プロパティへの直接リファレンスをアニメーション表示リンクで使用することはできません。

構文

```
tagname = %pointer.property;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

%pointer
OLE オブジェクトを参照するポインタです。OLE_CreateObject() 関数を使用して作成するか、プロパティを読み取る前に別のポインタに割り当てる必要があります。

property
読み取るプロパティの名前です。

tagname
値の書き込み先のタグ変数です。

例

このスクリプトでは、System.Random OLE クラスに基づいて OLE オブジェクトが作成され、そのオブジェクトを参照するポインタ %SR が作成され、Math.Random OLE オブジェクトの .NextDouble プロパティの値が実数型タグ変数 randtag に割り当てられます。

ランタイムでは、実数型タグ変数 Randtag は、0 ～ 1 の間のランダムな Double 浮動値を受け取ります。

```
OLE_CreateObject(%SR,"System.Random");
randtag = %SR.NextDouble;
```

OLE オブジェクト プロパティに書き込む

スクリプトでは、値をプロパティに割り当てることで、OLE オブジェクト プロパティに値を書き込むことができます。

構文

```
%pointer.property = value;
```

引数

%pointer
OLE オブジェクトを参照するポインタです。OLE_CreateObject() 関数を使用して作成するか、プロパティに書き込む前に別のポインタに割り当てる必要があります。

property
書き込まれるプロパティの名前です。

value
プロパティに書き込まれる値です。リテラル値、タグ変数、または式となります。アニメーション入力リンクから直接 OLE プロパティに書き込むことはできません。

OLE オブジェクトのメソッドを呼び出す

スクリプトで、OLE オブジェクト メソッドを呼び出すことができます。

構文

```
%pointer.method(parameters);
```

引数

%pointer
OLE オブジェクトを参照するポインタです。OLE_CreateObject() 関数を使用して作成するか、メソッドを呼び出す前に別のポインタに割り当てる必要があります。

method
OLE オブジェクトの一部であるメソッドの名前です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータぱらめーた
メソッドに渡すパラメータのリストです。これらのパラメータは、カンマで区切る必要があります。リテラル値、タグ変数、または式です。

例

このスクリプトでは、OLE クラス Shell.Application に基づいて OLE オブジェクトが作成され、その OLE オブジェクトへのポインタ %sa が作成され、メソッド .MinimizeAll() が呼び出されます。このメソッドでは、デスクトップのすべてのウィンドウが最小化されます。

```
OLE_CreateObject(%SA,"Shell.Application");
%SA.MinimizeAll();
```

注意 : オプション パラメーターは、OLE InTouch HMI スクリプトでは許可されていません。すべてのパラメータを指定する必要があります。

同じ OLE オブジェクトに複数のポインタを割り当てる

スクリプトでは、等号記号を使用することによって、複数のポインタを同じ OLE オブジェクトに割り当てることができます。

構文

```
%newpointer = %pointer
```

引数

%pointer
作成された OLE オブジェクトをすでに参照しているポインタの名前です。

%newpointer
同じ OLE オブジェクトを参照する新しいポインタの名前です。英数字（A ～ Z、0 ～ 9）およびアンダースコアを含めることができます。大文字と小文字は区別されません。

例

このスクリプトでは、Wscript.Shell クラスに基づいて OLE オブジェクトが作成され、このオブジェクトを参照するポインタ %WS が作成されます。ポインタ %WS2 は、%WS に設定されると、同じ OLE オブジェクトを参照します。これは、プロパティに対する読み取りおよび書き込み操作や、同じ OLE オブジェクトのメソッドの呼び出しに使用できます。

```
OLE_CreateObject(%WS,"Wscript.Shell");
%WS2=%WS;
```

注意 : メッセージ型タグ変数とポインタを合わせて使用できます。メッセージ型タグ変数をポインタに割り当てる場合、ID 値を取得できます。これを使用して、同じ OLE オブジェクトを参照する複数のポインタを作成できます。

OLE エラーをトラブルシューティングする

スクリプトで、OLE 関数を使用して、OLE エラーをトラブルシューティングできます。

機能	説明
OLE_GetLastObjectError() 関数	最後の OLE エラーのエラー番号を取得します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

機能	説明
OLE_GetLastObjectErrorMessage() 関数	最後の OLE エラーの情報を取得します。
OLE_ResetObjectError() 関数	最後のエラーをリセットします。
OLE_ShowMessageOnObjectError() 関数	OLE エラー メッセージ ダイアログ ボックスを表示または非表示にします。
OLE_IncrementOnObjectError() 関数	InTouch HMI タグ変数を使用して、OLE エラー数をカウントします。

OLE_GetLastObjectError() 関数

この関数は、最後の OLE エラーのエラー番号を返します。

構文

```
errnum = OLE_GetLastObjectError();
```

引数

errnum
最後の OLE エラーの番号です。

OLE_GetLastObjectErrorMessage() 関数

この関数は、最後の OLE エラーのエラー メッセージを返します。

構文

```
errmsg = OLE_GetLastObjectErrorMessage();
```

引数

errmsg
最後の OLE エラーのエラー メッセージです。

OLE_ResetObjectError() 関数

スクリプトで、OLE_ResetObjectError() 関数を使用すると、最後の OLE エラーをリセットできます。これにより、最後の OLE エラー番号が 0 に設定され、最後の OLE エラー メッセージが空白に設定されます。

これは、OLE 関数のバッチで発生したエラーを特定するために使用できます。

構文

```
OLE_ResetObjectError()
```

OLE_ShowMessageOnObjectError() 関数

デフォルトでは、OLE エラーが発生すると、エラー メッセージ ダイアログ ボックスが表示されます。

スクリプトで、関数 OLE_ShowMessageOnObjectError() を使用すると、エラー メッセージ ダイアログ ボックスを表示するかどうかを指定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
OLE_ShowMessageOnObjectError(Boolean)
```

引数

Boolean
OLE エラー メッセージ ダイアログ ボックスが表示されるかどうかを指定する値です。リテラル ブール値、論理型タグ変数、またはブール式です。値の意味は以下のとおりです。

0 - OLE エラーが発生したとき、OLE エラー メッセージ ダイアログ ボックスは表示されません。

1 - OLE エラーが発生したとき、OLE エラー メッセージ ダイアログ ボックスが表示されます。

例

このスクリプトでは、すべての OLE エラー メッセージ ダイアログ ボックスが非表示となります。OLE エラーが発生したとき、エラー メッセージ ダイアログ ボックスは表示されません。

```
OLE_ShowMessageOnObjectError(0);
```

OLE_IncrementOnObjectError() 関数

スクリプトで、OLE_IncrementOnObjectError() 関数を使用すると、OLE エラーの数のカウンタとして、整数型タグ変数を指定できます。

構文

```
OLE_IncrementOnObjectError(integertag)
```

パラメータ

integertag
カウンタとして機能するタグ変数です。

備考

OLE エラー メッセージ ダイアログ ボックスが表示される場合、OLE エラー メッセージ ダイアログ ボックスが閉じられた後でのみ、カウンタ タグ変数の数が増加します。

例

このスクリプトでは、整数型タグ変数 errorcount がエラー カウンタとして指定され、エラー メッセージ ダイアログ ボックスが非表示にされ、無効な OLE クラス名に基づいて OLE オブジェクトが作成されます。ただし、この操作はエラーとなり、タグ変数値 errorcount の値は 1 に増加します。

```
errorcount = 0;
OLE_IncrementOnObjectError(errorcount);
OLE_ShowMessageOnObjectError(0);
OLE_CreateObject(%WS,"InVaLiD.cLaSs.nAmE");
```

OLE を使用して行える操作

以下のスクリプトを使用すると、OLE オブジェクトを使用してアプリケーションに追加できる強力な機能について理解できます。

ランダム数を生成する

スクリプトで、以下のコマンドを使用すると、0 ～ 255 までのランダム数を生成できます。

```
OLE_CreateObject(%SR,"System.Random");
randtag = (%SR.NextDouble)*255;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザー インターフェイス ダイアログ ボックスを作成する

スクリプトで、以下のコマンドを使用すると、ユーザー インターフェイス ダイアログ ボックスを作成できます。

```
dim DlgBody as message;
dim DlgTitle as message;
dim Style as integer;
dim Result as integer;
DlgBody = "Do you want to open the valve ’MR-3-FF’?";
DlgTitle = "Confirm Opening Valve MR-3-FF";
Style = 48;
OLE_CreateObject(%WS,"Wscript.Shell");
result = %WS.Popup(DlgBody,1,DlgTitle,Style);
```

この例では、以下のユーザー インターフェイス ダイアログ ボックスが作成されます。

Style タグ変数は、ダイアログ ボックスに表示されるアイコンおよびボタンを決定します。以下の値を使用します。

アイコン	スタイル	値
（アイコンなし）	アイコンなし	0
	エラー アイコン	16
	疑問符アイコン	32
	警告アイコン	48
	情報アイコン	64

特定のボタンを使用するには、以下のいずれかの値を Style 値に追加します。

値	スタイル
0	［**OK**］ボタンのみ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値	スタイル
1	［**OK**］ボタンおよび［**キャンセル**］ボタン
2	［**中止**］ボタン、［**再試行**］ボタン、および［**無視**］ボタン
3	［**はい**］ボタン、［**いいえ**］ボタン、および［**キャンセル**］ボタン
4	［**はい**］ボタンおよび［**いいえ**］ボタン
5	［**再試行**］ボタンおよび［**キャンセル**］ボタン
6	［**キャンセル**］ボタン、［**再試行**］ボタン、および［**継続**］ボタン

Result タグ名には、ユーザーが選択したボタン番号が含まれます。これは、InTouch スクリプトの条件分岐に使用できます。以下の結果コードを使用できます。

結果値	意味
1	［**OK**］ボタンが押された
2	［**キャンセル**］ボタンが押された
3	［**中止**］ボタンが押された
4	［**再試行**］ボタンが押された
5	［**無視**］ボタンが押された
6	［**はい**］ボタンが押された
7	［**いいえ**］ボタンが押された
10	［**再試行**］ボタンが押された
11	［**継続**］ボタンが押された

Windows の日時プロパティ パネルを開く

スクリプトで、以下のコマンドを使用すると、Windows の［日付と時刻のプロパティ］パネルを開くことができます。

```
OLE_CreateObject(%WP,"Shell.Application");
%WP.SetTime();
```

同様のタスクを実行するには、異なるメソッドを呼び出して、これらを参照されている OLE オブジェクトに渡します。

メソッド	開くパネル
TrayProperties()	**トレイ** プロパティ
FileRun()	［**ファイル実行**］ダイアログ ボックス

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メソッド	開くパネル
FindFiles()	［ファイルの検索］ダイアログ ボックス
FindComputer()	［コンピュータの検索］ダイアログ ボックス
ShutdownWindows()	［**Windows** のシャットダウン］パネル

レジストリの読み込みとレジストリへの書き込み

スクリプトで、以下の操作を行うことによって、OLE を使用して、Windows レジストリに対して読み取りや書き込みを行うことができます。

- Windows クラス Wscript.Shell に基づいて OLE オブジェクトを作成する
- OLE オブジェクトの RegRead() メソッドおよび RegWrite() メソッドを使用する

たとえば、以下のコマンドは、インストールされている InTouch HMI のバージョンを直接レジストリ キーから読み取り、その値を rkey メッセージ タグ変数に保存します。

```
OLE_CreateObject(%WS,"Wscript.Shell");
rkey = %WS.RegRead("HKLM\SOFTWARE\Wonderware\InTouch\Installation\Version");
```

これらのコマンドでは、現在ログオンしているユーザーにファイル拡張子が非表示になっているかどうかを決めるレジストリ キーに値 1 が書き込まれます。

```
OLE_CreateObject(%WS,"Wscript.Shell");
%WS.RegWrite("HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\HideFileExt"
,1,"REG_DWORD");
```

ウィンドウを最小化する

スクリプトで、以下のコマンドを使用すると、デスクトップのすべてのウィンドウを最小化できます。

```
OLE_CreateObject(%WA,"Shell.Application");
%WA.MinimizeAll();
```

以下のメソッドを呼び出すと、同様のタスクを実行できます。

メソッド	ウィンドウの配置
TileHorizontally()	すべてのウィンドウを水平に整列します
TileVertically()	すべてのウィンドウを垂直に整列します
CascadeWindows()	すべてのウィンドウをカスケード表示します
UndoMinimizeALL()	すべてのウィンドウのサイズを元に戻します

ActiveX コントロールをスクリプト化する

ActiveX コントロールを使用すると、タグ名および I/O 参照の読み取りおよび書き込みを行えます。スクリプトでは、ActiveX コントロールを参照できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

また、ActiveX コントロールに対してイベントが発生したときに実行するスクリプトを作成することもできます。これらのスクリプトは、他のアプリケーションで再使用およびインポートすることができます。

ActiveX コントロール メソッドを呼び出す

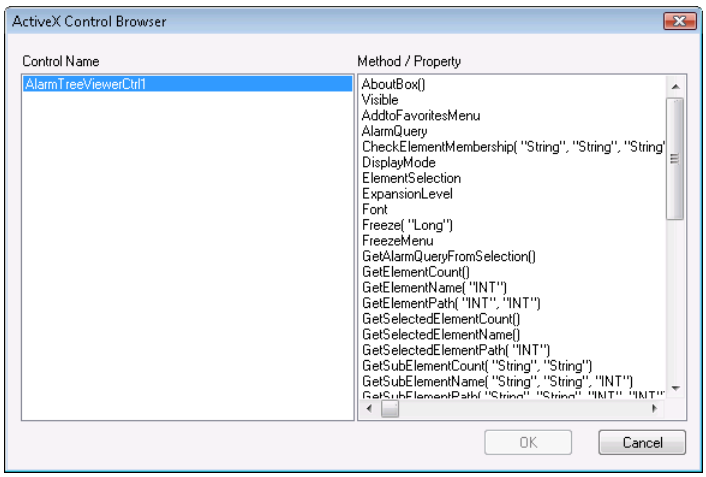
スクリプトでは、ActiveX コントロールのメソッドを呼び出して、ActiveX コントロールによりサポートされているアクションを実行できます。ActiveX メソッドは、任意のタイプの InTouch QuickScript または ActiveX イベント スクリプトから呼び出すことができます。

注意 : ActiveX イベントが発生したときに ActiveX メソッドを呼び出すには、いくつかの必要条件があります。「ActiveX イベント スクリプトの設定」を参照してください。

ActiveX コントロール メソッドを呼び出す

1. スクリプト ダイアログ ボックスの［**挿入**］メニューで［**ActiveX**］を選択します。

   ［**ActiveX コントロール ブラウザ**］ダイアログ ボックスが表示されます。



2. 左ペインから ActiveX コントロールの名前を選択します。右ペインには、ActiveX コントロールによりサポートされているプロパティおよびメソッドの名前が含まれています。
3. 右ペインから使用するメソッドの名前を選択し、続いて［**OK**］を選択します。メソッド名およびデフォルトのパラメータが、スクリプト ウィンドウのカーソル位置に挿入されます。
4. メソッド パラメータは括弧内に設定します。

InTouch HMI から ActiveX コントロール プロパティにアクセスする

スクリプトでは、ActiveX コントロール プロパティに対して読み込みおよび書き込みを実行して、ActiveX コントロールと InTouch タグ変数および表示リンク間でデータを交換できます。

データの読み取りや書き込みを行うために ActiveX コントロール プロパティを設定する

スクリプトでは、ActiveX コントロールに対するデータの読み取りや書き込みを行うことができます。特定の ActiveX コントロールに関連付けられている ActiveX コントロール プロパティを使用できます。

以下の 2 つの方法があります。

- ActiveX コントロール プロパティを InTouch HMI QuickScript または ActiveX イベント スクリプトで使用する。プロパティ値は、スクリプトが実行されるたびに読み取りまたは書き込みされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

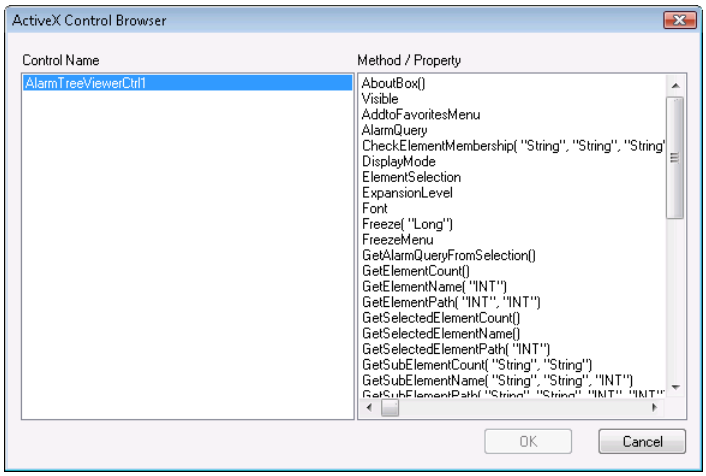
- ActiveX コントロール プロパティを InTouch HMI タグ変数または I/O リファレンスに直接リンクする。プロパティ値は、更新周期のたびに読み取りまたは書き込みされます。

ActiveX コントロール プロパティの読み取りや書き込みを行うためにスクリプトを設定する

スクリプトでは、ActiveX コントロール プロパティを設定して、InTouch HMI タグ変数またはその他の式に対して値の書き込みや読み取り行うことができます。

ActiveX コントロール プロパティに対してデータの読み取りや書き込みを行う

1. スクリプト ウィンドウを開き、［**挿入**］をポイントして［**ActiveX**］を選択します。［**ActiveX コントロール ブラウザ**］ダイアログ ボックスが表示されます。



2. 左ペインから ActiveX コントロールの名前を選択します。右ペインには、選択した ActiveX コントロールのプロパティおよびメソッドの名前が含まれます。
3. 右ペインから、使用するプロパティの名前を選択します。プロパティ名が、スクリプト ウィンドウのカーソル位置に挿入されます。
4. プロパティ名をタグ変数に割り当てるか、必要に応じて使用します。
5. ［**OK**］を選択します。

例

以下のスクリプトでは、ActiveX コントロール インスタンス AlarmViewerCtrl1 の ToPriority プロパティが整数型タグ変数 topri に読み込まれます。

```
topri = #AlarmViewerCtrl1.ToPriority;
```

以下のスクリプトでは、値 MS Comic が ActiveX コントロールの Font プロパティである AlarmViewerCtrl1 に書き込まれます。この例では、AlarmViewer ActiveX コントロールの表示フォントが動的に変更されます。

```
#AlarmViewerCtrl1.Font = "MS Comic";
```

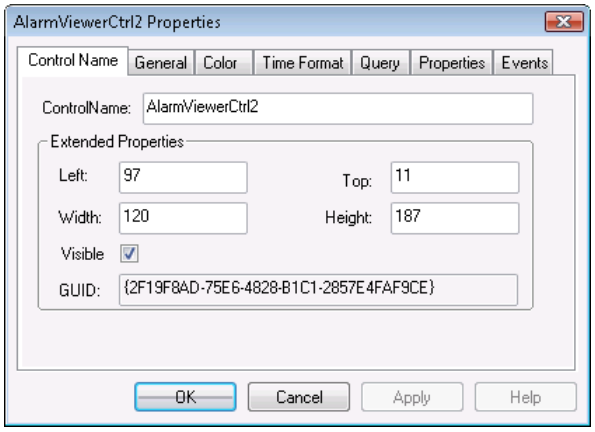
タグ変数または I/O リファレンスに ActiveX コントロール プロパティをリンクする

ActiveX コントロール プロパティを InTouch HMI タグ変数または I/O リファレンスにリンクできます。

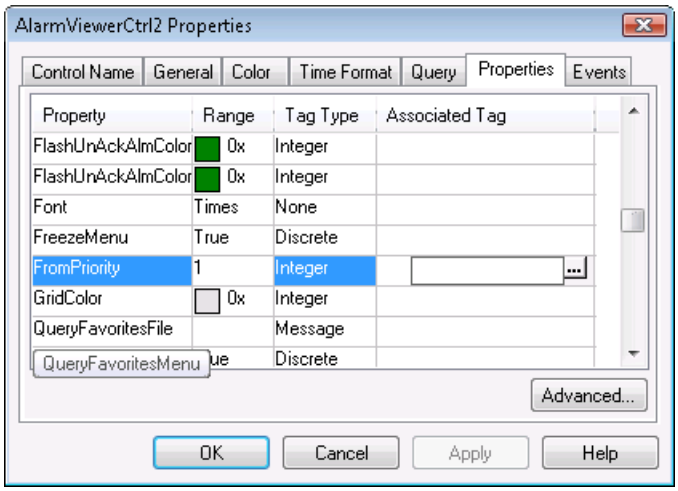
タグ変数または I/O リファレンスに ActiveX コントロール プロパティをリンクする

1. ActiveX コントロールをダブルクリックします。ActiveX コントロールのプロパティ ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［プロパティ］タブを選択して、右側にスクロールします。
3. 一覧からプロパティを選択します。



4. タグ変数または I/O リファレンスを割り当てます。以下のいずれかを実行します。
   - タグ変数 または I/O リファレンスを［**関連タグ変数**］カラムに直接入力します。
   - 角括弧の間の［**関連タグ変数**］カラムの参照ボタンを選択します。［**タグの選択**］ダイアログ ボックスが表示されます。タグを選択して［**OK**］を選択します。
5. ［**OK**］を選択します。

ActiveX イベント スクリプトの作成と再使用

ActiveX コントロールでは、特定のアクションへの関連付けに使用できる、コントロールのシングルクリックなどのイベントがサポートされています。これらのアクションは、ActiveX イベント スクリプトに保存されます。

ActiveX イベント スクリプトを作成する

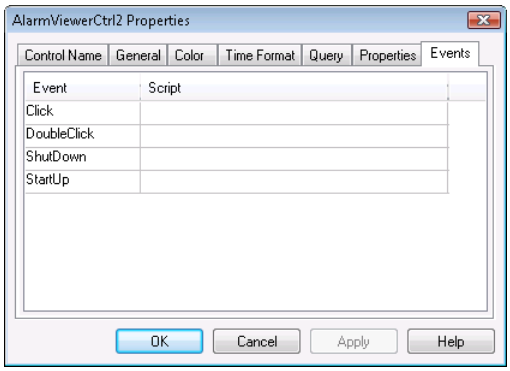
ActiveX コントロールの選択など、特定の ActiveX コントロール イベントが発生するたびに実行されるイベント スクリプトを作成または再使用できます。

ActiveX イベント スクリプトを作成する

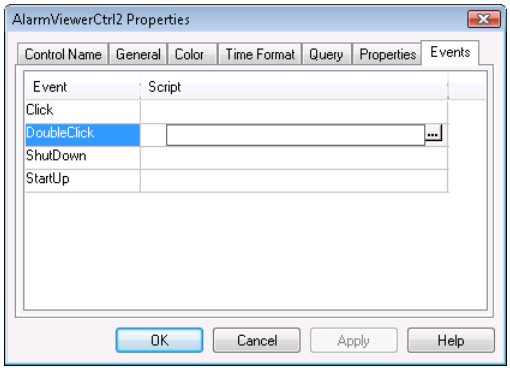
1. ActiveX コントロールをダブルクリックします。プロパティ ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［イベント］タブを選択します。



3. 関連付けるイベントを選択します。［**スクリプト**］カラムに括弧と参照ボタンが表示されます。



4. 対応する行の［**スクリプト**］カラムで、その括弧内の領域を選択します。
5. 新しい名前を入力して［**OK**］を選択します。メッセージが表示されたら、［**OK**］を選択します。［**ActiveX イベント スクリプト**］ダイアログ ボックスが表示されます。
6. 必要に応じて、スクリプトを作成します。

ActiveX イベント スクリプトを再使用する

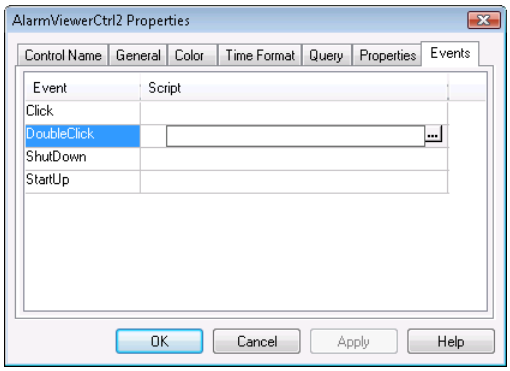
同じ ActiveX コントロールの親およびイベントにより作成された場合、ActiveX イベント スクリプトは再使用できます。

たとえば、アプリケーションに AlarmViewer ActiveX コントロールが複数ある場合、これらは、DoubleClick イベントのイベント スクリプトを共有できます。

ActiveX イベント スクリプトを再使用する

1. ActiveX コントロールをダブルクリックします。プロパティ ダイアログ ボックスが表示されます。
2. ［イベント］タブを選択します。
3. 関連付けるイベントを選択します。［**スクリプト**］カラムに括弧と参照ボタンが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. 対応する行の［**スクリプト**］カラムで、参照ボタンを選択します。［**ActiveX スクリプトの選択**］ダイアログ ボックスが表示されます。
5. ActiveX スクリプトを選択して、［**OK**］をクリックします。
6. ［**OK**］をもう一度選択します。

セルフリファレンス ActiveX イベント スクリプトを作成する

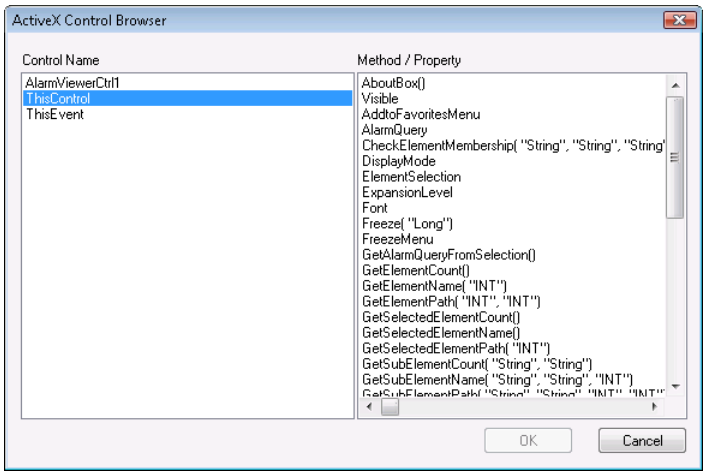
ActiveX イベント スクリプトを使用する場合、絶対的な ActiveX コントロール名ではなく、それら自身を参照するようにスクリプトを設定できます。これは、再使用される ActiveX イベント スクリプトを作成する場合に役立ちます。ActiveX イベント スクリプトは、以下のいずれかを参照できます。

- イベントを生成した特定の ActiveX コントロール（ThisControl）
- スクリプトを呼び出した特定のイベント（ThisEvent）

特定のイベントを参照すると、ActiveX コントロールでは、他のパラメータを ActiveX コントロール スクリプトに渡すことができます。

セルフリファレンス ActiveX イベント スクリプトを作成する

1. 特定の ActiveX イベントの ActiveX イベント スクリプトを作成します。「ActiveX イベント スクリプトを作成する」を参照してください。
2. ［**ActiveX イベント スクリプト**］ダイアログ ボックスで、［**挿入**］を選択して、［**ActiveX**］を選択します。［**ActiveX コントロール ブラウザ**］ダイアログ ボックスが表示されます。



3. 左ペインで、以下のいずれかを実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**ThisControl**］を選択して、このコントロール（およびこのスクリプトを再使用できるその他のコントロール）と一緒に使用できるプロパティおよびメソッドを参照します。
- ［**ThisEvent**］を選択して、セルフリファレンス イベントと一緒に使用できる ActiveX コントロールのプロパティおよびメソッドを参照します。

4. 右ペインで、いずれかのプロパティまたはメソッドをハイライト表示して、［**OK**］を選択します。選択したプロパティまたはメソッドが、スクリプト ウィンドウにコピーされます。
5. スクリプトを設定します。
6. ［**OK**］を選択します。

たとえば、次のステートメントは、ClicknRow イベント パラメータの値を ClickedRow タグ変数に書き込みます。

```
ClickedRow = ThisEvent.ClicknRow;
```

ActiveX イベント スクリプトをインポートする

ActiveX イベント スクリプトを他の InTouch HMI アプリケーションからインポートして、現在開発中のアプリケーションで再使用できます。

他のアプリケーションから **ActiveX** イベント スクリプトをインポートする

1. WindowMaker を開きます。
2. ［ファイル］メニューで［インポート］、［表示］の順に選択してから［ウィンドウとスクリプト］を選択します。

   ［フォルダを開く］ダイアログ ボックスが表示されます。
3. インポートする ActiveX イベント スクリプトを含む InTouch HMI アプリケーションを参照します。
4. ［**OK**］を選択します。
1. ［**ActiveX** イベント スクリプト］チェックボックスをオンにして［インポート］を選択します。すべての ActiveX イベントが現在の InTouch HMI アプリケーションにインポートされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 12 ActiveX コントロール

ActiveX コントロールは、InTouch アプリケーションに追加の機能を提供するスタンドアロン ソフトウェア コンポーネントです。ActiveX コントロールには、コントロールの動作を変更するための、アプリケーションの実行中に変更可能なプロパティ、メソッド、およびイベントがあります。

InTouch アプリケーションでは、複数のタイプの ActiveX コントロールを使用できます。

- InTouch HMI にはアラーム用の ActiveX コントロールが含まれます。
- ActiveFactory などその他の製品は、データの操作および分析用のコントロールを提供しています。
- サードパーティの ActiveX コントロールを使用できます。
- Visual Basic または C で独自の ActiveX コントロールを開発できます。

InTouch アプリケーションでは、任意の数の ActiveX コントロールを使用できます。以下の操作を実行できます。

- ActiveX コントロールを選択して、任意のアプリケーション ウィンドウに貼り付ける
- コントロールによってサイズ変更がサポートされている場合、コントロールのサイズを調整する
- ActiveX コントロールの複製、切り取り、コピー、貼り付け、および削除を行う
- ActiveX コントロールを整列する: 左寄せ、右寄せ、上端揃え、下端揃え、中心揃え
- ActiveX コントロールの追加
- セルを作成する場合、ActiveX コントロールを他のオブジェクトと組み合わせる
- 特定の ActiveX コントロール プロパティによってサポートされているプロパティ、メソッド、およびイベントを使用する

InTouch アプリケーションは以下のタイプの ActiveX コントロールはサポートしていません。

- ウィンドウのないコントロール
- シンプル フレーム サイトまたは領域ボックス
- コンテナ
- データ コントロール
- ディスパッチ オブジェクト

InTouch アプリケーションは、論理型（ブール）、整数型（32 ビット数）、実数型（32 ビット浮動小数点 IEEE 表記）、メッセージ型（131 文字までの文字列）のデータ型のみをサポートしています。サポートされていないデータ型には、バリアント、ポインタ、配列、構造、およびパラメータ プロパティがあります。

ActiveX コントロールは、ウィンドウ コントロールまたはグラフィック オブジェクトなど、InTouch の他のオブジェクトと重ねることはできません。1 つのウィンドウに ActiveX コントロールが多すぎると、システム パフォーマンスが低下します。

ActiveX コントロールを使用する

ActiveX コントロールを選択し、ウィンドウに貼り付け、ダブルクリックします。設定ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ActiveX コントロールを設定する際、固有のコントロール名を指定できます。コントロール名は、スクリプトで参照できるようになります。

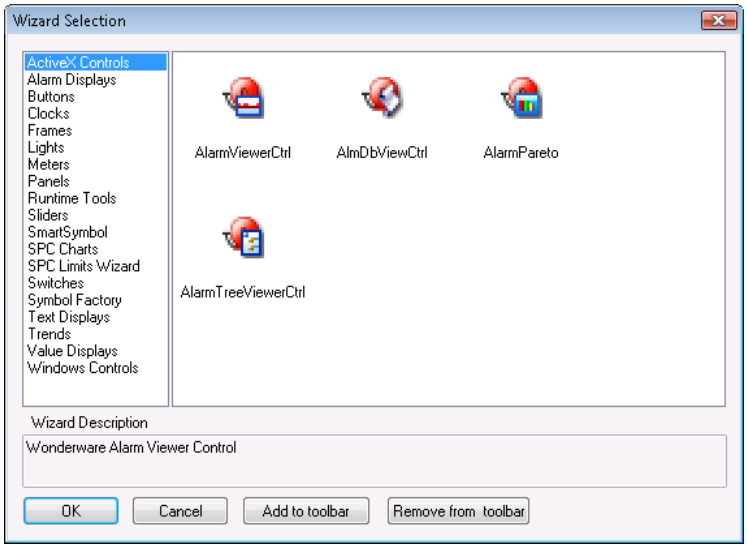
ActiveX コントロールを InTouch アプリケーションで使用する

1. ActiveX コントロールをインストールします。「ActiveX コントロールのインストールとアンインストール」を参照してください。
2. ActiveX コントロールを選択して、InTouch ウィンドウに貼り付けます。
3. ActiveX コントロール プロパティを設定して、タグにバインドします。
4. ActiveX イベントを ActiveX イベント スクリプトに関連付けます。
5. ActiveX メソッドを呼び出し、 ActiveX コントロールのプロパティを ActiveX イベント スクリプトか、他の InTouch QuickScript に設定します。

ActiveX コントロールをウィンドウに配置する

1. ［描画］メニューの［挿入］グループで［ウィザード］を選択します。

   ［ウィザードの選択］ダイアログ ボックスが表示されます。



2. ウィザード リストで、［**ActiveX Controls**］カテゴリを選択します。使用できるすべての ActiveX コントロールが、表示領域に表示されます。
3. 使用する ActiveX コントロールをダブルクリックします。ダイアログ ボックスが閉じ、カーソルがコーナー シンボルに変わります。
4. ActiveX コントロールを貼り付ける位置を選択します。

ActiveX コントロールをツールバーに追加する

1. ［描画］メニューの［挿入］グループで［ウィザード］を選択します。

   ［ウィザードの選択］ダイアログ ボックスが表示されます。

1. 追加する ActiveX コントロールを選択します。
2. ［ツールバーに追加］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ツールバーから ActiveX コントロールを削除する

1. ［描画］メニューの［挿入］グループで［ウィザード］を選択します。

   ［ウィザードの選択］ダイアログ ボックスが表示されます。

2. ［ツールバーから削除］を選択します。［削除するウィザード］ダイアログ ボックスが表示されます。
3. 削除する ActiveX コントロールを選択します。
4. ［OK］を選択します。

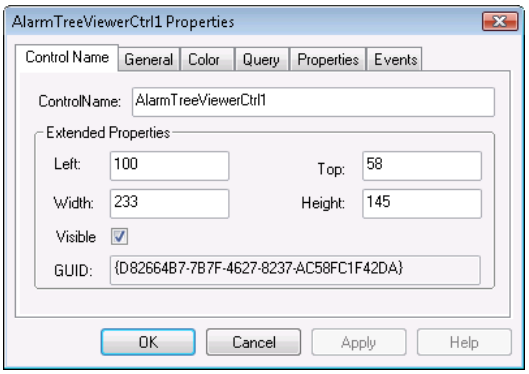
ActiveX コントロールを設定する

特定の ActiveX コントロールに設定できるプロパティは、ActiveX コントロール自体によって決定されます。すべてのプロパティにはデフォルト値があります。

プロパティを設定する前に ActiveX コントロールを InTouch ウィンドウに貼り付ける必要があります。

- ウィンドウに ActiveX コントロールを貼り付けると、Calendar1 などのデフォルトのコントロール名が生成されますコントロール名はスクリプトで参照できます。スクリプトを実行するには、開いているアプリケーション ウィンドウで ActiveX コントロールが実行されている必要があります。
- ActiveX コントロール プロパティは InTouch タグ変数に割り当てることができます。各プロパティタイプを対応する InTouch タグ変数タイプに割り当てる必要があります。

ActiveX コントロールには、［コントロール名］、［プロパティ］、［イベント］という 3 つの標準タブがあります。



［イベント］タブを使用すると、ユーザーがマウスをダブルクリックしたときなど、スクリプトを使用できるコントロール イベントに割り当てることができます。

その他のタブおよび設定内容はコントロールによって固有であり、プロパティにより異なります。たとえば、コントロールの中には色およびフォントの設定が必要であるものがありますが、他のコントロールにはこれらのプロパティが存在しない場合もあります。

ActiveX コントロールに名前を付ける

以下の操作を実行すると、コントロールの新しいインスタンスが作成され、固有の名前が与えられます。

- ［複製］を選択します。
- ［切り取り］または［コピー］を選択し、［貼り付け］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［ウィンドウに名前を付けて保存］を選択します。
- ［元に戻す］を選択してから［やり直し］を選択します。
- コントロールを含むウィンドウをインポートした場合

ActiveX コントロールには、固有の名前を付ける必要があります。コントロールにすでに存在している名前を使用しようとすると、エラー メッセージが表示されます。

ActiveX コントロールに名前を付ける

1. ActiveX コントロールを開発ウィンドウに貼り付けます。
2. コントロールをダブルクリックします。コントロールの［プロパティ］ダイアログ ボックスが表示されます。
3. ［コントロール名］タブを選択し、その ActiveX コントロールの名前を［コントロール名］ボックスに入力します。

ActiveX コントロールの標準操作

ActiveX コントロールの標準操作は、他の InTouch オブジェクトと同様に実行できます。詳細については、「すべてのオブジェクトの特殊操作」を参照してください。

ActiveX コントロールのインストールとアンインストール

コンピュータにすでに ActiveX コントロールがインストールされている場合でも、WindowMaker にインストールして InTouch HMI に通知する必要があります。

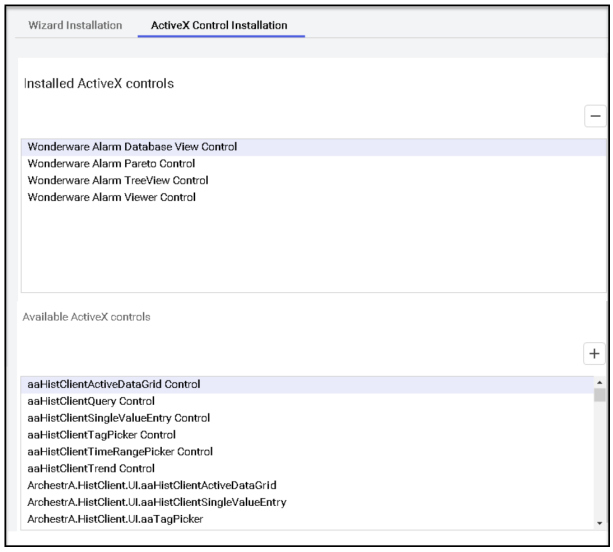
コントロールは WindowMaker のリストから消去されても、コンピュータからは削除されません。単にメモリから消去されて、機能しなくなるだけです。

ActiveX コントロールをインストールする

1. WindowMaker を開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［ウィザード/ActiveX］を選択します。

   ［ウィザード/ActiveX］画面が表示されます。
3. ［ActiveX コントロール］タブを選択します。［インストールされた ActiveX コントロール］プロパティ シートが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**インストールされた ActiveX コントロール**］リストで、［**インストール可能な ActiveX コントロール**］リストからインストールするコントロールを選択し、［+］アイコンを選択します。

   ヒント: 複数のコントロールを選択するには、Shift キーまたは Ctrl キーを使用します。

5. ［**保存**］を選択します。

ActiveX コントロールを削除する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューで、［**設定**］をポイントし、［**ウィザード/ActiveX**］を選択します。［**ウィザード/ActiveX**］画面が表示されます。
3. ［**ActiveX コントロール**］タブを選択します。［**インストールされた ActiveX コントロール**］プロパティ シートが表示されます。
4. ［**インストールされた ActiveX コントロール**］リストで、アプリケーションから削除するコントロールを選択し、［-］アイコンを選択します。

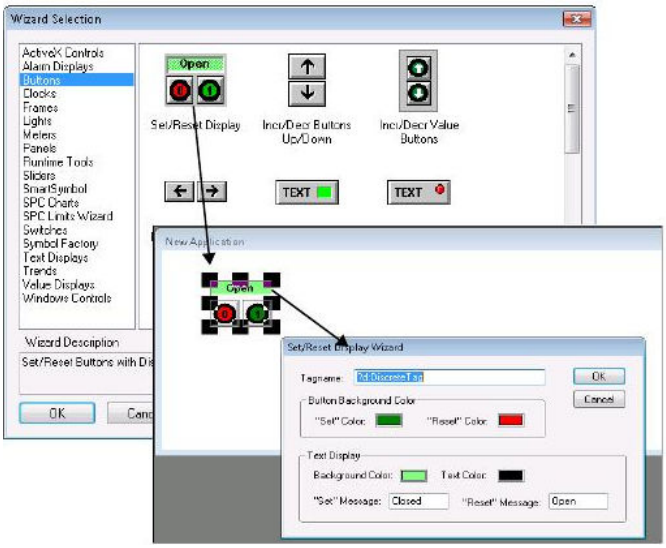
   ヒント: 複数のコントロールを選択するには、Shift キーまたは Ctrl キーを使用します。

5. コントロールを削除するには、［**はい**］を選択します。コントロールは［**インストール可能な ActiveX コントロール**］リストに移動されます。
6. ［**保存**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 13 ウィザード

ウィザードは、アプリケーション用に選択、配置、および設定するだけで使用できる事前設計、事前構築、および事前プログラムされたオブジェクトです。



ウィザードを操作する

ウィザードを使用すると、オブジェクトの個々のコンポーネントを描画したり、オブジェクトの値範囲を入力したり、オブジェクトをアニメーション化したりするために時間を費やす必要はありません。

- ［ウィザード選択］ダイアログからウィザードを選択します。
- ウィザードは設定ダイアログ ボックス内にタグおよび QuickScript を入力することによって設定します。
- 選択したウィザードをウィンドウに貼り付けて、ダブルクリックすると、［**設定**］ダイアログ ボックスが表示されます。

たとえば、スライダ ウィザードでは、有効なタグ名、スライダの移動範囲の最小値と最大値、塗りつぶし色などの項目を設定できます。必要な設定情報を入力すると、ウィザードは使用できるようになります。

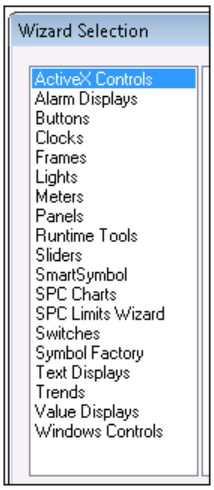
また、背後で行われるようなタイプの操作を実行する独自の複合ウィザードを開発することもできます。これらの操作には、完全な表示ウィンドウの作成、データベースの作成や変換、AutoCAD 図面のインポート、および他のアプリケーションの設定などが含まれます。

独自のウィザードを作成する前に、プログラミングが不要なウィザード機能を提供する産業用グラフィックを調べてください。

ウィザードのタイプ

ウィザードのカテゴリは、［**ウィザードの選択**］ダイアログ ボックスに表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



トレンド オブジェクトおよびウィンドウ コントロール ウィザードは、固有のパラメータを持つ特別なウィザードです。詳細については、「トレンド オブジェクト」および「ウィンドウ コントロール ウィザード」を参照してください。

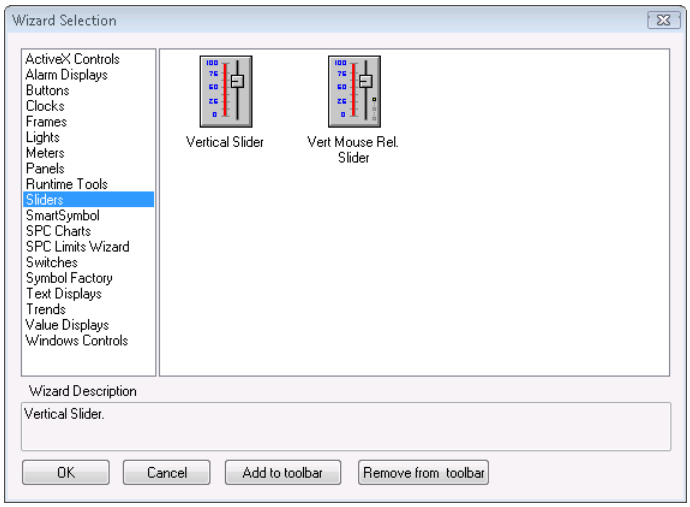
ツールバーにウィザードを追加する

頻繁に使用するウィザードをウィザード／ActiveX ツールバーに追加すると、素早く利用できるようになります。

ウィザードをウィザード ツールバーに追加する

1. ［描画］メニューの［挿入］グループで［ウィザード］を選択します。

   ［ウィザードの選択］ダイアログ ボックスが表示されます。



2. 左ペインで、［スライダ］などのウィザード カテゴリを選択します。
3. 右ペインでウィザードを選択し、［ツールバーに追加］を選択します。ウィザード ボタンがツールバーに表示されます。

ウィザードをツールバーから削除する

1. ［描画］メニューの［挿入］グループで［ウィザード］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［ウィザードの選択］ダイアログ ボックスが表示されます。

2. ［ツールバーから削除］を選択します。［削除するウィザード］ダイアログ ボックスが表示されます。
3. 削除するウィザードを選択して、［**OK**］を選択します。

ウィザード インスタンスを貼り付ける

ウィザードのインスタンスはウィンドウに配置できます。

ウィザードをウィンドウに配置する

1. ［描画］メニューの［挿入］グループで［ウィザード］を選択します。

   ［ウィザードの選択］ダイアログ ボックスが表示されます。
2. 左ペインで、ウィザード カテゴリを選択します。
3. 右ペインで、ウィザードを選択します。
4. ［**OK**］を選択します。

   ダイアログ ボックスが閉じ、カーソルがコーナー シンボルに変わります。
5. ウィザードを配置する位置を選択します。

ウィザードを設定する

ウィザードをアプリケーションに配置した後で、ウィザードをダブルクリックしてプロパティを設定します。選択したウィザードに対してカスタムであるプロパティ ダイアログ ボックスが表示されます。

特定のタイプの各ウィザードの詳細については、ウィザードのヘルプを参照してください（ヘルプが用意されている場合）。

ウィザードの標準操作を実行す r

ウィザードの切り取り、コピー、貼り付け、削除、および更新を他のオブジェクトと同じ方法で、同じ結果で実行できます。

ウィザードのインストールとアンインストール

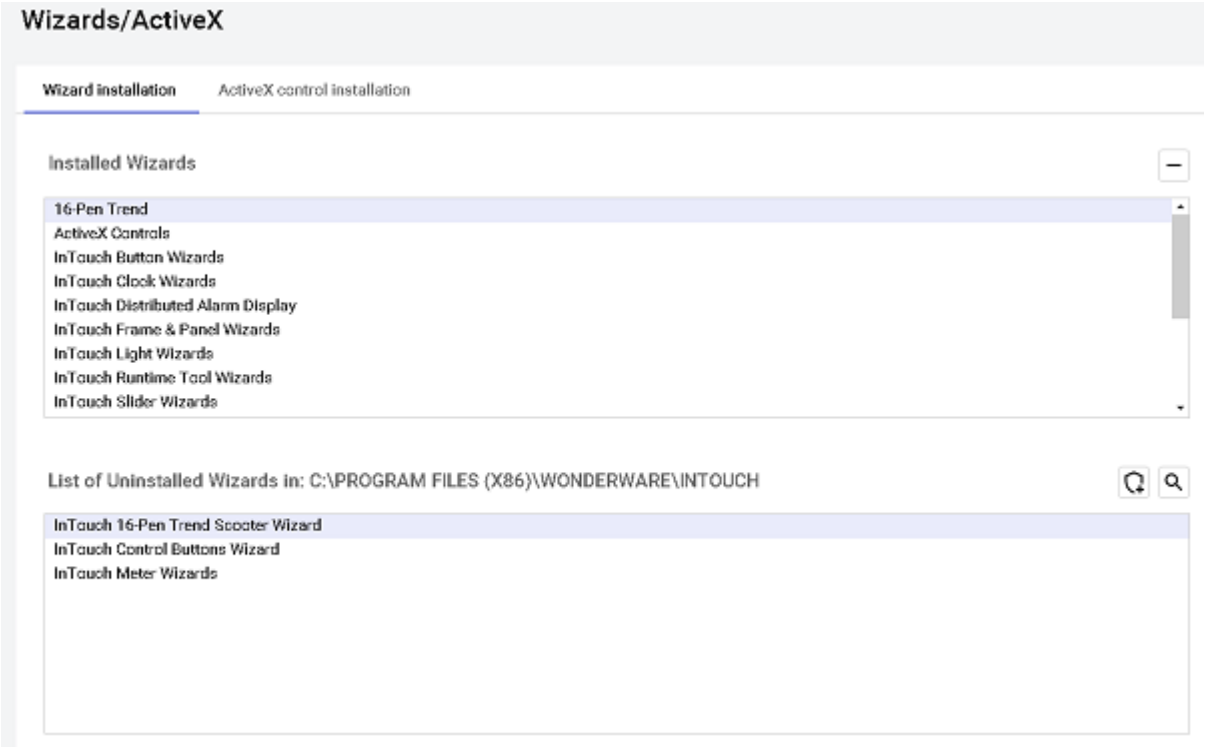
ウィザードを WindowMaker にインストールして、アプリケーションで使用できるようにする必要があります。ウィザードは WindowMaker のリストから消去されても、コンピュータからは削除されません。

ウィザードをインストールする

1. WindowMaker を開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［ウィザード/**ActiveX**］を選択します。

   ［ウィザード/**ActiveX**］設定画面が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［未インストールのウィザード リスト］から選択し、［+］アイコンを選択します。
4. ［保存］を選択します。

ウィザードを削除する

1. WindowMaker を開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［ウィザード/ActiveX］を選択します。

   ［ウィザード/ActiveX］設定画面が表示されます。
3. ［インストールされているウィザード］リストで、削除するウィザードを選択し、［-］アイコンを選択します。

   ヒント: Shift キーを押しながらクリックするか、Ctrl キーを押しながらクリックすると、複数選択ができます。

   ウィザードは、［未インストールのウィザード リスト］に表示されます。
4. ［保存］を選択します。

別のディレクトリからウィザードをインポートする

1. WindowMaker を開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［ウィザード/ActiveX］をクリックします。［ウィザード/ActiveX］画面が表示されます。
3. ［検索］アイコンをクリックします。［フォルダの参照］ダイアログ ボックスが表示されます。
4. インストールするウィザードを含むディレクトリを参照して、［OK］を選択します。［ウィザード］ダイアログ ボックスが再び表示され、インポートされたウィザードが［未インストールのウィザード リスト］に表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

トレンド オブジェクト

トレンド オブジェクトは、タグの値を時間経過でチャートにするウィザードです。

トレンド オブジェクトには 3 つの主なタイプがあります。

- リアルタイム トレンドはリアルタイムで最大 4 つまでのタグ変数をチャートにします。
- 履歴トレンドは過去の期間で最大 8 つのタグ変数をチャートにします。
- 16 ペン トレンドは最大 16 つまでのタグ変数のリアルタイムおよび履歴データをチャートにします。

ウィンドウに配置できるトレンド オブジェクト、リアルタイムまたは履歴の数に制限はありません。

以下ののアイテムを使用して、トレンド オブジェクトを設定します。

- 時間範囲
- 値範囲
- グリッド線の精度
- 時間および値の目盛り位置
- ペンと色

トレンド ウィザードを使用する前に、各タグ変数を追跡するためのログ記録を有効にして、InTouch アプリケーション内でのログ記録も有効にする必要があります。

タグ変数のログ記録を有効にする

1. **タグ変数ディクショナリ**から、タグ変数を選択して、［**データ ログ**］を選択します。
2. これを実行していない場合は、InTouch アプリケーションでログ記録を有効にします。
   a. WindowMaker を開きます。
   b. ［**ファイル**］メニューで、［**設定**］をポイントし、［**履歴ログ**］を選択します。

      ［**履歴ログ**］画面が表示されます。

   a. ［**履歴ログ実行**］チェックボックスまたは［**Historian へのストレージを有効にする**］チェックボックスをオンにして、［**OK**］を選択します。

トレンド オブジェクトの設定および使用の詳細については、「タグ データのトレンド」を参照してください。

ウィンドウ コントロール ウィザード

ウィンドウ コントロール ウィザードは、ドロップダウン リスト、コンボ ボックス、オプション選択、またはチェックボックスなどのユーザー インターフェイス オブジェクトです。

ウィンドウ コントロール ウィザードを使用して、事前定義の選択肢セットをユーザーに提供します。たとえば、プロセス、レシピ、またはオペレータ ID のドロップダウン リストを作成できます。また、指定したコントロールを有効または無効にすることもできます。ドロップダウン リストの内容をロードおよび変更することもできます。

ウィンドウ コントロール ウィザードのランタイム プロパティは、アニメーション リンクの式ではなく QuickScript 関数を通じてアクセスされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウ コントロール ウィザードには、InTouch タグ変数ドットフィールドと同様のプロパティがあり、読み取り／書き込み、または読み取り専用として設定できます。プロパティには、開発時やランタイム時にアクセスできるものもあります。これらのプロパティは「<コントロール名>.x」で識別され、x はプロパティを示します。

たとえば、ウィンドウ コントロールの［.Visible］プロパティの値が 0 の場合、コントロールはウィンドウに表示されません。InTouch タグ変数と同様に、［.Value］はウィンドウ コントロール ウィザードのデフォルト プロパティです。

注意 : 産業用グラフィックを使用している場合は、より堅牢で柔軟な .NET ベースのウィンドウ コントロール セットが利用できます。

ウィンドウ コントロールの作成と設定

ウィンドウ コントロール ウィザードを作成および設定する際は、以下の点に注意してください。

- ウィンドウ コントロール ウィザードは互いにオーバーラップできません。
- ランタイム時、ウィンドウ内のその他のオブジェクトにウィンドウ コントロール ウィザードが重なる場合、ウィンドウ コントロール ウィザードは常に一番上に表示されます。
- 各ウィンドウ コントロール ウィザードには固有のコントロール名があり、タグ変数カウントには追加されません。
- リスト ボックスやコンボ ボックスに割り当てられているタグ変数の初期値では、リスト ボックスやコンボ ボックスの値は初期化されません。SetPropertyX QuickScript 関数をスクリプトで使用して、デフォルト値と異なる初期値を割り当てる必要があります。

ヒント: その他のウィザードと同様に、ウィンドウ コントロール ウィザードはウィンドウに貼り付けられます。読みやすさを最大限に向上させるには、ウィンドウ コントロールの背景色にグレーを選択してください。背景色をグレーにできない場合、グレーのパネル ウィザードをウィンドウ コントロール ウィザードの背面に置いてください。

各ウィンドウ コントロール ウィザードに対して、英数字コントロール名を指定する必要があります。最初の文字は英字となります。アンダースコアは使用できますが、その他の特殊文字は使用できません。たとえば、「Checkbox_1」は使用できますが、「Checkbox#1」は使用できません。

ウィンドウ コントロール ウィザードは、InTouch QuickScript を使用して設定できます。

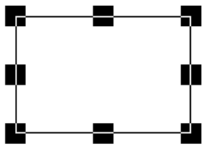
テキスト ボックスを作成する

オペレーターは、アプリケーション内でテキスト ボックスを使用してテキスト文字列を入力できます。

テキスト ボックスを作成および設定する

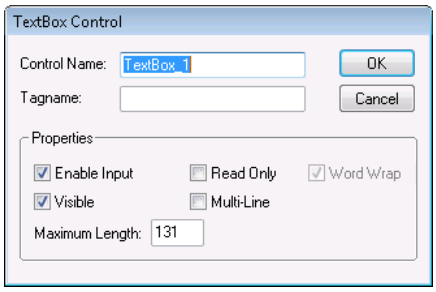
1. テキスト ボックスを作成して、配置します。以下の手順を実行します。
   a. ［**ウィザードの選択**］ダイアログ ボックスで、［**ウィンドウ コントロール ウィザード**］を選択します。コントロール ウィザード アイコンが表示されます。
   b. テキスト ボックス アイコンをダブルクリックします。アプリケーション ウィンドウが再表示され、カーソルが左コーナー シンボルに変わります。
   c. アプリケーション ウィンドウ内を選択して、ウィザードを配置します。コーナーに黒い大きなハンドルが付いたテキスト ボックス ウィザードが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



a. アプリケーションに適合するようにウィザードをドラッグして、サイズ変更します。

2. ウィザードをダブルクリックします。［**テキストボックス コントロール**］ダイアログ ボックスが表示されます。



3. ダイアログ ボックスを設定します。以下の手順を実行します。
   a. *TextBox_1* などのコントロール名を［**コントロール名**］ボックスに入力します。
   b. *New_Value* などのメモリ メッセージ型タグ変数を［**タグ変数**］ボックスに入力します。
   c. ［**プロパティ**］領域で、［**入力可能**］および［**表示可能**］を選択します。
4. ［**OK**］を選択します。［**テキストボックス コントロール**］ダイアログ ボックスが閉じます。

リスト ボックスを作成する

ユーザーがアイテムをオプション リストから選択することを可能にするアプリケーションを作成できます。

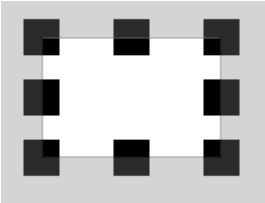
リスト ボックスをアプリケーションに追加する際、画面にコントロールを配置し、プロパティを設定してリストを設定して、必要なスクリプトを記述します。

リスト ボックスはファイルから、またはランタイムでキーボード入力からアイテムと共にロードできます。詳細については、「ウィンドウ コントロールをスクリプト化する」を参照してください。

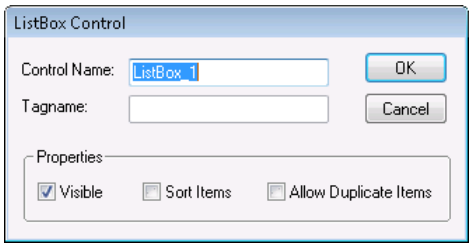
リスト ボックスを作成する

1. リスト ボックス コントロールを作成して、配置します。以下の手順を実行します。
   a. ［**ウィザードの選択**］ダイアログ ボックスで、［**ウィンドウ コントロール**］を選択します。
   b. リスト ボックス アイコンをダブルクリックします。アプリケーション ウィンドウが再表示され、カーソルが左コーナー シンボルに変わります。
   c. アプリケーション ウィンドウ内を選択して、コントロールを配置します。リスト ボックス コントロールが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. コントロールをダブルクリックします。［**リストボックス コントロール**］ダイアログ ボックスが表示されます。



3. コントロールを設定します。以下の手順を実行します。
   a. *TextBox_1* などのコントロール名を［**コントロール名**］ボックスに入力します。
   b. *LB1_Value* などのメモリ メッセージ型タグ変数を［**タグ変数**］ボックスに入力します。
   c. ［**プロパティ**］領域で、コントロールの表示方法と機能を設定します。
4. ［**OK**］を選択します。

コンボ ボックスを作成する

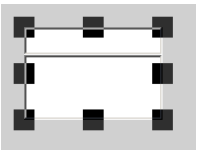
ユーザーがアイテムをコンボ ボックスを備えたオプション リストから選択することを可能にするアプリケーションを作成できます。コンボ ボックスは、テキスト ボックスとリスト ボックスを組み合わせたウィンドウ コントロールです。

コンボ ボックスをアプリケーションに追加する際、画面にコントロールを配置し、プロパティを設定してコンボ ボックスを設定して、必要なスクリプトを記述します。

コンボ ボックスはファイルから、またはランタイムでキーボード入力からアイテムと共にロードできます。詳細については、「ウィンドウ コントロールをスクリプト化する」を参照してください。

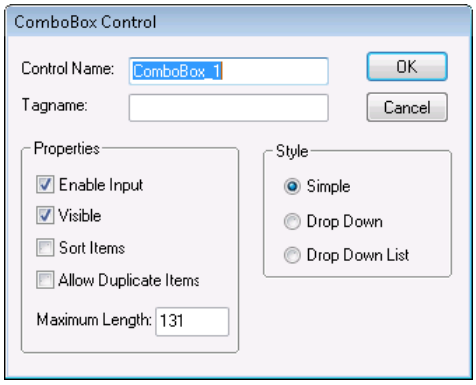
コンボ ボックスを作成する

1. コンボ ボックス コントロールを作成して、配置します。以下の手順を実行します。
   a. ［**ウィザードの選択**］ダイアログ ボックスで、［**ウィンドウ コントロール**］を選択します。
   b. コンボ ボックス アイコンをダブルクリックします。アプリケーション ウィンドウが再表示され、カーソルが左コーナー シンボルに変わります。
   c. アプリケーション ウィンドウ内を選択して、コントロールを配置します。コンボ ボックス コントロールが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. コントロールをダブルクリックします。［**コンボ ボックス コントロール**］ダイアログ ボックスが表示されます。



3. コントロールを設定します。以下の手順を実行します。
   a. *ComboBox_1* などのコントロール名を［**コントロール名**］ボックスに入力します。
   b. *CB1_Value* などのメモリ メッセージ型タグ変数を［**タグ変数**］ボックスに入力します。

      CB1_Value はメモリ タグ、IO タグ、リモート タグのいずれかであり、PLC に接続できます。PLC に接続するように I/O タグを設定する方法については、「I/O タグ変数のプロパティの設定」を参照してください。

   c. ［**プロパティ**］領域で、コントロールの表示方法と機能を設定します。
   d. ［**スタイル**］領域で、コンボ ボックスのタイプを選択します。
4. ［**OK**］ を選択します。

チェックボックスを作成する

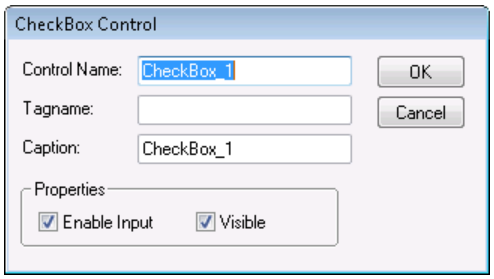
オペレータは、チェックボックス コントロールを使用してオプションを選択できます。

チェックボックス コントロールを設定するには

1. チェックボックスを作成します。以下の手順を実行します。
   a. ［**ウィザードの選択**］ダイアログ ボックスで、［**ウィンドウ コントロール ウィザード**］を選択します。コントロール ウィザード アイコンが表示されます。
   b. チェックボックス アイコンをダブルクリックします。アプリケーション ウィンドウが再表示され、カーソルが左コーナー シンボルに変わります。
   c. アプリケーション ウィンドウ内を選択して、ウィザードを配置します。チェックボックス ウィザードが表示されます。

   a. ウィザードをドラッグして、サイズ変更します。
2. ウィザードをダブルクリックします。［**チェック ボックス コントロール**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ウィザードを設定します。以下の手順を実行します。
   a. コントロール名を入力します。
   b. 論理型タグ変数.を入力するか、または空白のタグ変数ボックスをダブルクリックして、［**タグ変数を選択してください**］ダイアログ ボックスを表示して、タグ変数を選択します。
   c. ボタンの表面に表示されるキャプションを入力します。
4. ［**OK**］を選択します。

ラジオ ボタン グループを作成する

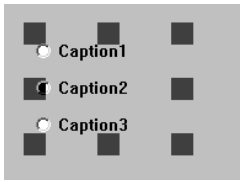
ユーザーが複数の選択肢から 1 つを選択する必要がある場合に、ラジオ ボタンを使用します。ユーザーがオプションを選択したら、以前に選択されたオプションは選択解除されます。

ユーザー用のオプションをラジオ ボタンのグループとして作成します。各ラジオ ボタンにはキャプションがあり、スクリプトに固有の値を提供します。

整数型タグ変数は、ラジオ ボタン コントロールにのみ割り当てることができます。

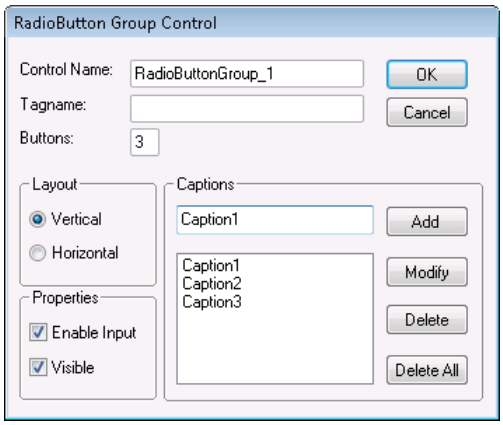
ラジオ ボタン グループを作成する

1. ラジオ ボタン コントロール ウィザードを作成します。以下の手順を実行します。
   a. ［**ウィザードの選択**］ダイアログ ボックスで、［**ウィンドウ コントロール ウィザード**］を選択します。コントロール ウィザード アイコンが表示されます。
   b. ラジオ ボタン アイコンをダブルクリックします。アプリケーション ウィンドウが再表示され、カーソルが左コーナー シンボルに変わります。
   c. アプリケーション ウィンドウ内を選択して、ウィザードを配置します。3 つのラジオ ボタンと共にラジオ グループ コントロール ウィザードが表示されます。



   a. アプリケーションに適合するようにウィザードをドラッグして、サイズ変更します。
2. ウィザードをダブルクリックします。［**ラジオ ボタン グループ コントロール**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ウィザードを設定します。以下の手順を実行します。
   a. コントロール名を入力します。
   b. 整数型タグ変数を入力して、このコントロールにリンクします。
   c. 表示するボタンの数を入力します。
   d. 各ボタンのキャプションを入力します。
   e. レイアウトおよびプロパティを設定します。
4. ［**OK**］を選択します。

ウィンドウ コントロールをスクリプト化する

スクリプトで QuickScript 関数を使用して、以下の操作を行えます。

- コントロールの値の取得または設定
- コントロールの有効化、無効化、または非表示
- コンボ ボックス、リスト ボックス、テキスト ボックス、およびチェックボックスのアイテムの操作

プロパティに基づいて、ランタイム プロパティは読み取り／書き込みまたは読み取り専用のいずれかにできます。

GetPropertyX() 関数および SetPropertyX() 関数を使用して、これらのプロパティを制御または取得します。

コントロールの値の取得または設定

.Value プロパティは、すべての InTouch ウィンドウ コントロール ウィザードのデフォルト プロパティです。

このプロパティに行われる変更は、InTouch タグ変数とウィンドウ コントロール ウィザードで同期されます。

.Value ドットフィールド

すべての InTouch ウィンドウ コントロール ウィザードのデフォルト プロパティです。このプロパティに行われる変更は、InTouch タグ変数とウィンドウ コントロール ウィザードで同期されます。

カテゴリ

ウィンドウ コントロール

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

M、I、および D は、GetProperty 関数および SetProperty 関数のメモリ型、整数型、および論理型バージョンです。

```
[ErrorNumber=]GetPropertyM("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyM("ControlName[.Value]", Value);
[ErrorNumber=]GetPropertyI("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyI("ControlName[.Value]", Value);
[ErrorNumber=]GetPropertyD("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyD("ControlName[.Value]", Value);
```

パラメータ

ControlName
ウィンドウ コントロールの名前（例：ChkBox_4）

タグ名
プロパティの値を書き込む先のタグ変数です。

[.Value]
このプロパティはオプションです。指定されていない場合、関数は .Value プロパティが使用されていると想定します。

値
関数が処理されるときに書き込まれるプロパティ値を保持する、書き込まれる実際の値または有効な InTouch タグ変数（書き込まれるプロパティと同じタイプ）。

備考

リスト ボックスまたはコンボ ボックスに割り当てられているタグ変数の初期値は、リスト ボックスまたはコンボ ボックスの値を初期化するために使用することはできません。

このドットフィールドは、開発時およびランタイム時ともに読み取り／書き込み可能です。タグ変数をリスト ボックスまたはコンボ ボックスに関連付けることによって .Value ドットフィールドにアクセスする場合は読み取り専用となります。.Value ドットフィールドがチェックボックス、ラジオ ボタン、またはテキスト ボックスに割り当てられる場合、読み取り／書き込みとなります。開発時に指定した値がランタイムでのデフォルトとなります。

データ型

テキスト ボックス、リスト ボックス、およびコンボ ボックスの場合はメッセージ型（読み取り／書き込み）

ラジオ ボタンの場合は整数型（読み取り／書き込み）

チェックボックスの場合は論理型（読み取り／書き込み）

適用対象

テキスト ボックス、リスト ボックス、コンボ ボックス、チェックボックス、ラジオ ボタン

例

以下のステートメントは、ラジオ ボタン オブジェクト RadioButton_1 の .Value ドットフィールドの値を 4 に設定します。

```
SetPropertyI( "RadioButton_1.Value", 4 );
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

GetPropertyM()、SetPropertyM()、GetPropertyI()、SetPropertyI()、GetPropertyD()、SetPropertyD()

ユーザー入力用のコントロールの有効化または無効化

.Enabled プロパティを使用して、コントロール オブジェクトをユーザー生成イベントに応答できるようにするかどうか指定します。

.Enabled ドットフィールド

コントロール オブジェクトが、ユーザーによって生成されたイベントに反応するかどうかを指定します。

カテゴリ

ウィンドウ コントロール

使用法

```
[ErrorNumber=] GetPropertyD("ControlName.Enabled",
Tagname);
[ErrorNumber=] SetPropertyD("ControlName.Enabled",
Discrete);
```

パラメータ

ControlName
ウィンドウ コントロールの名前。例：ChkBox_4

タグ名
要求されたプロパティを保持する論理型タグ変数

論理型
関数が処理されるときに書き込まれる値を保持する論理型の値または論理型のタグ変数論理値：

0 = コントロール使用不可

1 = コントロール使用可能

備考

このプロパティは開発時およびランタイム時ともに読み取り／書き込みです。

データ型

論理型（読み取り/書き込み）

適用対象

テキスト ボックス、リスト ボックス、コンボ ボックス、チェックボックス、ラジオ ボタン

例

以下のステートメントは、「"ListBox_1" という名前のリスト ボックス オブジェクトを無効にします。

```
SetPropertyD("ListBox_1.Enabled", 0);
```

参照項目

GetPropertyD()、SetPropertyD()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウ コントロールを動的に非表示にする

.Visible プロパティを使用して、ウィンドウ コントロールをウィンドウに表示するかどうかを決定します。

.Visible ドットフィールド

ウィンドウ コントロールを表示するかどうかを決定します。

カテゴリ

ウィンドウ コントロール

使用法

```
[ErrorNumber=]GetPropertyD("ControlName.Visible", Tagname);
[ErrorNumber=]SetPropertyD("ControlName.Visible", Number);
```

パラメータ

ControlName
ウィンドウ コントロールの名前。例：ListBox_1

タグ名
関数が処理されるときにプロパティ値を保持するタグ変数（返されるものと同じタイプ）

数値
関数が処理されるときに書き込まれる値を保持する論理型の値または論理型のタグ変数論理値：

0 = コントロールを非表示

1 = コントロールを表示

備考

このプロパティは開発時およびランタイム時ともに読み取り／書き込みです。

データ型

論理型（読み取り/書き込み）

有効値

適用対象

テキスト ボックス、リスト ボックス、コンボ ボックス、チェックボックス、ラジオ ボタン

例

以下のステートメントにより、"TextBox_1" という名前のテキスト ボックスが非表示になります。

```
SetPropertyD( "TextBox_1.Visible",0 );
```

参照項目

GetPropertyD()、SetPropertyD()

コンボ ボックスのアイテムの追加と削除

以下のスクリプト関数を使用して、コンボ ボックスおよびリストからアイテムを追加したり、削除したります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スクリプト関数	効果
wcAddItem()	アイテムをリスト ボックスやコンボ ボックスのリストの最後に追加します。ソートが有効になっている場合、リストはアイテムの追加後にソートされます。
wcInsertItem()	アイテムをリスト ボックスまたはコンボ ボックスのリストの指定した位置に追加します。
wcDeleteItem()	アイテムをリスト ボックスまたはコンボ ボックスのリストの指定した位置から削除します。
wcDeleteSelection()	現在選択されているアイテムをリストまたはコンボ ボックスから削除します。
wcClear()	リストまたはコンボ ボックスからすべてのアイテムを削除します。

wcAddItem() 関数

アイテムをリスト ボックスやコンボ ボックスのリストの最後に追加します。ソートが有効になっている場合、リストはアイテムの追加後にソートされます。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcAddItem("ControlName", "MessageTag");
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

MessageTag
表示されるメッセージ文字列。実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

以下のステートメントは、リスト ボックスが含まれているウィンドウ（開く時ウィンドウ QuickScript を使用）を開くと、メッセージ文字列の内容をそのリスト ボックスに追加します。

```
wcAddItem("ListBox_1", "Chocolate");
wcAddItem("ListBox_1", "Vanilla");
wcAddItem("ListBox_1", "Strawberry");
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

wcInsertItem()

wcInsertItem() 関数

指定した文字列をリスト ボックスまたはコンボ ボックスのリストの指定した位置に挿入します。wcAddItem() 関数とは異なり、ソートされたリスト ボックスまたはコンボ ボックスとして作成された場合でも、wcInsertItem() 関数はリストをソートしません。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcInsertItem("ControlName", ItemPosition, "Message");
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

ItemPosition
アイテムが追加される位置を示す数。このパラメータが -1 の場合、文字列はリストの最後に追加されます。数値または整数型タグ変数です。

Message
ItemPosition で指定した位置に挿入する文字列が含まれます。実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

アクション スクリプトが実行されると、以下のステートメントにより、"Blueberry" という新しいアイテムが、リスト ボックスの先頭から 4 番目の場所に挿入されます。

```
wcInsertItem("ListBox_1", 4, "Blueberry");
```

参照項目

wcAddItem()

wcDeleteItem() 関数

アイテムをリスト ボックスまたはコンボ ボックスのリストの指定した位置から削除します。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcDeleteItem("ControlName", ItemPosition);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

ItemPosition
アイテムの位置に対応する数。数値または整数型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

アクション スクリプトが実行されると、以下のステートメントにより、リストの 3 番目のアイテムが削除されます。

```
wcDeleteItem("ListBox_1", 3);
```

wcDeleteSelection() 関数

現在選択されているアイテムをリストから削除します。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber =]wcDeleteSelection("ControlName");
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

アクション スクリプトが実行されると、以下のステートメントにより、リストボックスで現在選択されているアイテムが削除されます。

```
wcDeleteSelection("ListBox_1");
```

wcClear() 関数

リスト ボックスまたはコンボ ボックスからすべてのアイテムを削除します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcClear("ControlName");
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

アクション スクリプトが実行されると、以下のステートメントにより、リスト ボックスのすべてのアイテムが削除されます。

```
wcClear("ListBox_1");
```

ファイルに対するリスト アイテムのロードと保存

以下のスクリプト関数を使用して、コンボ ボックスまたはリストのアイテムをファイルに対してロードしたり、保存したりします。

スクリプト関数	効果
wcLoadList()	新しいアイテムを持つリスト ボックスまたはコンボ ボックスをファイルからロードします。
wcSaveList()	リスト ボックスまたはコンボ ボックスの内容をファイルに保存します。

wcLoadList() 関数

新しいアイテムを持つリスト ボックスまたはコンボ ボックスをファイルからロードします。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcLoadList("ControlName", "Filename");
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Filename
ファイルの名前が含まれています。メッセージ パラメータの一部として完全なパス名が提供されない場合、この関数はアプリケーション ディレクトリでメッセージ ファイルを探します。実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

外部ファイルを使用して、リスト ボックスおよびコンボ ボックスの内容を入力する場合、特定の書式設定に従い、特定の情報を含む必要があります。書式：

ControlType, ListCount

ListItem, ItemIndex

ListItem, ItemIndex

: :

: :

ListItem, ItemIndex

ControlType は COMBOBOX または LISTBOX です。

たとえば、リスト ファイルをコンボ ボックスにロードする場合で、その中に選択できる 3 つのアイテムが含まれており、それらのアイテム データが割り当てられていない場合などがあります。ファイル形式には、以下の種類があります。

COMBOBOX, 3

Chocolate, 0

Vanilla, 0

Strawberry, 0

COMBOBOX は、コントロール タイプです。リスト カウントは、Chocolate、Vanilla、Strawberry の 3 つです。Chocolate は、最初のアイテムまたはポジション 1 として、Vanilla はポジション 2、Strawberry はポジション 3 として一覧されています。各アイテムはデータ値 0 を持ちます。

アイテム データの詳細については、「wcSetItemData() 関数」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

以下のステートメントは、適切に書式設定されたリスト（c:\wclist.txt）をコンボ ボックスにロードします。

```
wcLoadList("Combobox_1", "c:\wclist.txt");
```

参照項目

wcAddItem()、wcSaveList()

wcSaveList() 関数

リスト ボックスまたはコンボ ボックスのアイテムをファイルに保存します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcSaveList("ControlName","Filename");
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

Filename
ファイルの名前が含まれています。ファイルが存在しない場合、作成されます。実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

アクション スクリプトが実行されると、以下のステートメントにより、リストボックスの現在のアイテムがファイル（c:\newlist.txt）に保存されます。

```
wcSaveList("ListBox_1", "c:\newlist.txt");
```

参照項目

wcLoadList()、wcSetItemData()

コンボ ボックスまたはリストのアイテムを検索する

wcFindItem() 関数を使用して、指定したアイテムをリスト ボックスまたはコンボ ボックスで検索します。アイテムが検索されたら、この関数は対応する位置を 4 番目のパラメータとして整数型タグ変数に返します。

wcFindItem() 関数

指定されたメッセージ文字列に一致するリスト ボックスまたはコンボ ボックスの最初のアイテムの対応する位置を決定します。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcFindItem ("ControlName", "MessageTag", CaseSens, Tagname);
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

MessageTag
比較されるメッセージ文字列。実際の文字列またはメッセージ型タグ変数です。

CaseSens
文字列比較のタイプを決定します。論理値またはタグ変数を指定となります。以下値が有効です。

0 = 大文字と小文字は区別されません。

1 = 大文字と小文字は区別されます。

Tagname
一致するアイテムの位置が返される整数型タグ変数。一致するアイテムが見つからない場合は、-1 が返されます。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックス、コンボ ボックス

例

ListBox_1 が "ItemA"、"ItemB"、および "ItemC" を含むリスト ボックスであると仮定すると、この関数は以下の値を結果として返します。

```
wcFindItem("ListBox_1", "ItemB", 0, Result);
```

2 を返します

```
wcFindItem("ListBox_1", "Itemb", 1, Result);
```

-1 を返します

```
wcFindItem("ListBox_1", "itemc", 0, Result);
```

3 を返します

```
wcFindItem("ListBox_1", "XYZ", 0, Result);
```

-1 を返します

コンボ ボックスまたはリストのアイテム インデックスを操作する

以下のドットフィールドをリスト ボックスまたはコンボ ボックスのアイテム インデックスと共に使用します。

ドット フィールド	効果
.TopIndex	リスト ボックスの先頭アイテムの整数インデックス
.NewIndex	wcAddItem() 関数や wcInsertItem() 関数を使用して、リスト ボックスまたはコンボ ボックスに追加された最後のアイテムの整数インデックス（タグ変数）
.ListIndex	リストで現在選択されているアイテムのインデックス（タグ変数または数）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.TopIndex ドットフィールド

リスト ボックスの先頭アイテムの整数インデックスを設定するか読み取ります。

カテゴリ

ウィンドウ コントロール

使用法

```
[ErrorNumber=]GetPropertyI("ControlName.TopIndex", Tagname);
[ErrorNumber=]SetPropertyI("ControlName.TopIndex", Number);
```

パラメータ

ControlName
ウィンドウ コントロールの名前。例：ListBox_1

Tagname
関数が処理されるときにプロパティ値を保持する整数型タグ変数

Number
リスト ボックスの先頭アイテムを定義するインデックス番号。整数値を提供するリテラル整数値または整数型タグ変数、または式のいずれかです。

備考

このプロパティは、ランタイムでのみ使用できます。

データタイプ

整数型（読み取り／書き込み）

適用対象

リスト ボックス

例

以下のステートメントは、"ListBox_1" というリスト ボックス オブジェクトの TopIndex を 14 に設定します。

```
SetPropertyI( "ListBox_1.TopIndex",14 );
```

参照項目

GetPropertyI()、SetPropertyI()、.ListIndex、.NewIndex

.NewIndex ドットフィールド

wcAddItem() 関数または wcInsertItem() 関数を使用して、リスト ボックスまたはコンボ ボックスに追加された最後のアイテムの整数インデックス（タグ変数）を返します。

カテゴリ

ウィンドウ コントロール

使用法

```
[ErrorNumber=]GetPropertyI("ControlName.NewIndex", Tagname);
```

パラメータ

ControlName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウ コントロールの名前。例：ListBox_4

Tagname
リスト ボックスまたはコンボ ボックスに追加された最後のアイテムの整数インデックスを含むタグ変数。空のリストに対しては -1 が返されます。

備考

このプロパティは、ランタイムでのみ使用できます。

データタイプ

整数型（読み取り専用）

適用対象

リスト ボックスとコンボ ボックス

例

以下のステートメントにより、"ListBox_1" というリスト ボックスで最後に追加されたアイテムのインデックスを取得し、その値をメモリ整数型タグ変数 NewItemIndex に書き込みます。

```
GetPropertyI("ListBox_1.NewIndex", NewItemIndex);
```

参照項目

GetPropertyI()、wcAddItem()、wcInsertItem()、.ListIndex、.TopIndex

.ListIndex ドットフィールド

リストで現在選択されているアイテムのインデックス（Tagname または Number）を設定または読み取ります。

リスト ボックスを使用している場合、インデックス －1 は現在選択されているアイテムが存在しないことを示します。

コンボ ボックスを使用している場合、インデックス -1 はユーザーが新しいテキストをそのコントロールのテキスト入力フィールドに入力したことを示します。

構文

```
[ErrorNumber=]GetPropertyI("ControlName.ListIndex", Tagname);
[ErrorNumber=]SetPropertyI("ControlName.ListIndex", Number);
```

パラメータ

ControlName
ウィンドウ コントロールの名前。例：ListBox_4

Tagname
現在選択されているアイテムのインデックスを書き込む先のタグ変数です。

Number
リストで特定のアイテムを定義するインデックス番号

備考

リストで特定のアイテムを定義するインデックス番号。.ListIndex ドットフィールドを使用して、リストまたはコンボ ボックスで現在選択されているアイテムのインデックスを設定または決定します。

このプロパティは、ランタイムでのみ使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データ タイプ

整数型（読み取り／書き込み）

適用対象

リスト ボックスとコンボ ボックス

例

このステートメントは、"ListBox_1" というリスト ボックスで現在選択されているアイテムのインデックスを取得し、その値をメモリ整数型タグ変数 MyListBoxIndex に書き込みます。

```
GetPropertyI( "ListBox_1.ListIndex", MyListBoxIndex );
```

参照項目

GetPropertyI()、SetPropertyI()、.NewIndex、.TopIndex

リスト ボックスまたはコンボ ボックス アイテムをカウントする

.ListCount ドットフィールドには、リスト ボックスまたはコンボ ボックスのアイテム数が含まれています。

.ListCount ドットフィールド

リスト ボックスまたはコンボ ボックスのアイテム数を読み取ります。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]GetPropertyI("ControlName.ListCount", Tagname);
```

パラメータ

ControlName
ウィンドウ コントロールの名前。

Tagname
リストのアイテムの整数の数を含む有効なタグ変数

備考

このプロパティは、ランタイムでのみ使用できます。

データタイプ

整数型（読み取り専用）

適用対象

リスト ボックスとコンボ ボックス

例

以下のステートメントにより、*ListBox_1* というリスト ボックスからアイテム数を取得し、その値をメモリ整数型タグ変数 *MyListBoxCount* に書き込みます。

```
GetPropertyI("ListBox_1.ListCount", MyListBoxCount);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

GetPropertyI()、.ListIndex

リスト アイテムの値の取得または設定

wcGetItemData() 関数を使用して、アイテム インデックスによって識別されるリスト アイテムに関連付けられている整数値を検索します。

wcSetItemData() 関数を使用して、アイテム インデックスで指定されるリストのアイテムに整数値を割り当てます。これにより文字列に数字が割り当てられます。

wcGetItemData() 関数

ItemIndex パラメータで識別されるリスト アイテムに関連付けられている整数値を読み取ります。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcGetItemData("ControlName", ItemIndex, Tagname);
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

ItemIndex
アイテムの位置に対応する数。数値または整数型タグ変数です。

Tagname
実数型または整数型タグ変数の実際の名前。wcGetItemData() 関数は、関数から返されたアイテムに対応する数値をこのタグ変数に配置します。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

アクション スクリプトが実行されると、以下のステートメントはリスト ボックスの 5 番目のアイテムに関連付けられている数値を取得して、整数型タグ変数 ItemValue に返します。

```
wcGetItemData("ListBox_1", 5, ItemValue);
```

リスト ボックスの 5 番目のアイテムに整数値 4500 が割り当てられると、タグ変数 ItemValue には 4500 が含まれます。

参照項目

wcSetItemData()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

wcSetItemData() 関数

アイテム（Number パラメータ）の整数値を ItemIndex パラメータで指定したリストのアイテムに割り当てます。この関数では、文字列に数値を割り当てることができます。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcSetItemData("ControlName", ItemIndex, Number);
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

ItemIndex
編集するリスト アイテムを指定する整数値です。数値または整数型タグ変数です。

Number
アイテム データを表す整数値。数値または整数型タグ変数です。

備考

メモ帳などのプログラムを使用して、アイテムを含む完全なリストを作成して、1 つの関数呼び出しを使用してそれらをロードできます。wcSaveList() 関数に従ってリストを書式設定します。

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

wcGetItemData() 関数を使用して、リスト アイテムに関連付けられている値（アイテム データ）を返します。タグ変数パラメータには、返された数値が含まれます。このパラメータには、実際のデバイスに直接書き込む I/O 整数型タグ変数を使用できます。

例

小麦粉、砂糖、塩を使用するレシピがあるとします。使用する小麦粉の量は 4500 グラム、砂糖と塩はそれぞれ 1500 グラムと 325 グラムです。選択されるレシピ（タグ変数、レシピ名）によって切り替えられるデータ変化スクリプトを使用して、リスト ボックスの各アイテムに値が割り当てられます。

```
wcSetItemData("ListBox_1", 1, 4500); {リスト内の最初のアイテムを (flour)=4500 に設定します}
wcSetItemData("ListBox_1", 2, 1500); {リスト内の 2 番目のアイテムを (sugar)=1500 に設定します}
wcSetItemData("ListBox_1", 3, 325);   {リスト内の 3 番目のアイテムを (salt)=325 に設定します}
```

参照項目

wcLoadList()、wcSaveList()、wcGetItemData()

リスト アイテムの名前を取得する

wcGetItem() 関数を使用して、リスト ボックスまたはコンボ ボックスの対応するアイテム インデックスに関連付けられているアイテム文字列を返します。

wcGetItem() 関数

リスト ボックスまたはコンボ ボックスの ItemIndex に対応するアイテムの内容を含む文字列を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcGetItem("ControlName", ItemIndex, Tagname);
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

ItemIndex
アイテムの位置に対応する数。数値または整数型タグ変数です。

Tagname
メッセージ型タグ変数 wcGetItem 関数は、関数から返されたアイテム インデックスに対応するデータをこのタグ変数に配置します。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックスとコンボ ボックス

例

アクション スクリプトが実行されると、以下のステートメントは、コンボ ボックスの 10 番目のアイテムの文字列値をメッセージ型タグ変数 ListSelection に返します。

```
wcGetItem("Combobox_1", 10, ListSelection);
```

リストの 10 番目のアイテムが "Vanilla" の場合、タグ変数 ListSelection には "Vanilla" という文字列が含まれます。

テキスト ボックスの内容をロードする

wcLoadText() 関数を使用して、テキスト ボックスの内容をファイルからロードします。wcSaveText() 関数を使用して、テキスト ボックスの内容をテキスト ファイルに保存します。

注意 : タグ変数に最高文字数が定義されている場合、その文字数だけがテキスト ボックスの内容からタグ変数に割り当てられます。テキスト ボックスにタグ変数が割り当てられていない場合、テキスト ボックスには最大 65,535 文字まで入力できます。

wcLoadText() 関数

テキスト ボックスの内容をファイルの内容で置き換えます。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcLoadText("ControlName", "Filename");
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

Filename
ファイルの名前が含まれています。メッセージ パラメータの一部として完全なパス名が提供されない場合、この関数はアプリケーション ディレクトリでファイルを検索します。実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

テキスト ボックス

例

以下のステートメントは、テキスト ボックスを含むウィンドウ（開く時ウィンドウ スクリプト）が開くときにテキスト ファイル（c:\InTouch.32\readme.txt）をテキスト ボックスにロードします。

```
wcLoadText("Textbox_1", "c:\InTouch.32\readme.txt");
```

wcSaveText() 関数

テキスト ボックスに含まれる文字列を指定ファイルに保存します。ファイルが存在しない場合、作成されます。ファイルが存在する場合、読み取り／書き込みである必要があります。

カテゴリ

ウィンドウ コントロール

構文

```
[ErrorNumber=]wcSaveText("ControlName", "Filename");
```

パラメータ

ControlName
ウィンドウ コントロール オブジェクトの名前。例：ListBox_1 実際の文字列またはメッセージ型タグ変数です。

Filename
ファイルの名前が含まれています。完全なパス名が提供されていない場合、ファイルはアプリケーション ディレクトリに保存されます。ファイルが存在する場合は上書きされます。ファイルが存在しない場合、作成されます。作成されたファイルは後から wcLoadText() 関数を使用してテキスト ボックス オブジェクトにロードできます。実際の文字列またはメッセージ型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

テキスト ボックス

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

アクション スクリプトが実行されると、以下のステートメントは、テキスト ボックスに入力されている現在の情報を c:\InTouch.32\newtext.txt ファイルに保存します。

```
wcSaveText("Textbox_1", "c:\InTouch.32\newtext.txt");
```

参照項目

wcLoadText()

テキスト ボックスが読み取り専用であるかどうかを確認する

.ReadOnly ドットフィールドを使用して、テキスト ボックスの内容が読み取り専用であるか読み取り／書き込みであるかを確定できます。

.ReadOnly ドットフィールド

テキスト ボックスの内容が読み取り専用であるか読み取り／書き込みであるかを確定できます。

カテゴリ

ウィンドウ コントロール

使用法

```
[ErrorNumber=]GetPropertyD("ControlName.ReadOnly", Tagname);
```

パラメータ

ControlName
ウィンドウ コントロールの名前。例：Textbox_1

Tagname
関数が処理されるとき、プロパティ値を保持する論理型タグ変数

0 = テキスト ボックスの内容は読み取り／書き込みです。

1= テキスト ボックスの内容は読み取り専用です。

備考

このプロパティは開発時およびランタイム時ともに使用できます。

データタイプ

論理型（読み取り専用）

適用対象

テキスト ボックス

例

以下のステートメントは、"TextBox_1" という名前のテキスト ボックスの読み取り専用ステータスを取得します。

```
GetPropertyD( "TextBox_1.ReadOnly",A_Tagname );
```

参照項目

GetPropertyD()、SetPropertyD()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

チェック ボックスのラベルの取得または設定

.Caption ドットフィールドはチェックボックスのメッセージ テキストを定義します。

.Caption ドットフィールド

チェックボックスに表示するメッセージを決定します。

カテゴリ

ウィンドウ コントロール

使用法

```
[ErrorNumber=]GetPropertyM ("ControlName.Caption", Tagname);
[ErrorNumber=]SetPropertyM ("ControlName.Caption", "Message");
```

パラメータ

ControlName
ウィンドウ コントロールの名前。例：ChkBox_4

タグ名
要求されたプロパティを保持するメッセージ型タグ変数

メッセージ型
引用符で囲まれたメッセージ文字列

備考

このプロパティは開発時およびランタイム時ともに読み取り／書き込みです。

データ型

メッセージ型（読み取り／書き込み）

適用対象

チェックボックス

例

このステートメントは、チェックボックス オブジェクト "CheckBox_1" のキャプションを "Blue Paint Option." に設定します。

```
SetPropertyM( "CheckBox_1.Caption","Blue Paint Option" );
```

参照項目

GetPropertyM()、SetPropertyM()

ウィンドウ コントロール エラー メッセージを理解する

wcErrorMessage() は、指定されたエラー番号のエラーを説明するメッセージ文字列を返します。リスト ボックス、テキスト ボックス、コンボ ボックス、ラジオ ボタン、およびチェックボックスに適用されます。

ウィンドウ コントロール関数は、QuickScript 関数を処理した結果に基づいて値を返します。戻り値はエラー診断に使用されます。これらの値を整数型タグ変数に割り当てることができます。以下に例を示します。

```
ErrorNumber = wcGetItem("ControlName", Number, Tagname);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

このスクリプトでは、ErrorNumber は、返されたエラー値を含む整数型タグ変数です。この関数の戻り値は、wcErrorMessage() 関数に渡すことができます。wcErrorMessage() 関数は、そのエラーの説明文字列を返します。以下に例を示します。

```
ErrorMsg = wcErrorMessage(ErrorNumber);
```

このスクリプトでは、ErrorMsg は、返されたエラーのテキストを含むメッセージ型タグ変数です。以下の表では、エラー数値およびその定義を識別しています。

エラー メッセージ	定義
0	成功
-1	一般エラー
-2	使用できるメモリが不十分です
-3	プロパティは読み取り専用です
-4	指定したアイテムはすでに存在します
-5	オブジェクト名が不明です
-6	プロパティ名が不明です
-x	不明なエラー

wcErrorMessage() 関数

エラーを説明するメッセージ文字列を返します。

カテゴリ

ウィンドウ コントロール

構文

```
ErrorMessage=wcErrorMessage(ErrorNumber);
```

パラメータ

ErrorMessage
メッセージ型タグ変数

ErrorNumber
すべてのウィンドウ コントロール関数によって返される数。数値または整数型タグ変数です。

備考

返されるエラー番号のリストについては、「ウィンドウ コントロール エラー メッセージを理解する」を参照してください。

適用対象

リスト ボックス、テキスト ボックス、コンボ ボックス、チェックボックス、およびラジオ ボタン

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

リストのロード中にエラーが発生した場合、メッセージ型タグ変数 ErrorDescription にエラーのテキストによる説明を表示します。この例では、エラー メッセージを表示するために、値の表示のアニメーション リンクがタグ変数 ErrorDescription に割り当てられます。

"開く時" ウィンドウ QuickScript：

```
ErrorNumber=wcLoadList("ListBox_1","c:\recipe.txt");
ErrorDescription=wcErrorMessage(errornumber);
```

この関数はすべてのウィンドウ コントロール関数と使用して、エラー メッセージを表示できます。

```
ErrorNumber=wcAddItem("ListBox_1","AM_4A4356");
ErrorMsg=wcErrorMessage(ErrorNumber);
```

ウィジェット

ウィジェットは、Web ページや Web サイトの機能を拡張できる小型の Web コンポーネントです。カスタム ビルドの Web サイトでも、オープン ソース コードやフレームワークを使用してウィジェットを組み込んで全体的または部分的に特定の機能を提供できます。ウィジェットは自己包含型のコード ブロックで、その機能を変更することなく Web サイトに組み込むことができます。ウィジェットは、その他のプラットフォームおよびデータ ソースと統合するユーザー インターフェイス要素を提供するために最も頻繁に使用されます。ウィジェットは、一貫した配置およびユーザー インターフェイスで Web サイトの任意の Web ページで実行できます（ソーシャル メディア、天気予報、RSS、ポッドキャスト ウィジェットなど）。

デフォルトでは、グラフィック ツールバーのウィジェット フォルダで以下のウィジェットを使用できます。

- カルーセル
- Web ブラウザ
- QR コード スキャナ
- マップ アプリ
- DataGrid
- Teamwork

カルーセル ウィジェット

カルーセル ウィジェットでは、入力なしでカルーセルのようにさまざまな要素（画像やテキストのスライドなど）をサイクル表示できます。このウィジェットは、ダッシュボード、アラート、またはアラーム情報を工場作業場の大型モニタに表示するために使用できます。

注：カルーセル ウィジェットのクライアント コントロール（アラーム クライアント コントロールおよびトレンド クライアント コントロール）画面は、WindowViewer および Web Client でサポートされていません。

プロパティ

［**ウィジェットのプロパティ**］では、標準のグラフィック プロパティに加えて、ウィジェット固有のプロパティも設定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

名前	説明	デフォルト
自動表示	Autoplay プロパティが True に設定されている場合、起動時に カルーセル ウィジェットが自動的にロードされます。False に設定されている場合、カルーセルを開始する次の項目をユーザーが選択する必要があります。	True
背景色	ウィジェットの背景色を設定します。RGB、HTML コード（#FF0000）、または有効な HTML 色名で色の値を指定します。	白
グラフィック名	カルーセルがランタイムに実行するグラフィックのリスト（カンマ区切り）。	空
間隔	項目のサイクルを自動的に開始する際の間隔（ミリ秒）。	5000
キーボード	Keyboard プロパティが True に設定されている場合、カルーセルはキーボード入力に反応します。	True
ループ	Loop プロパティが True に設定されている場合、グラフィックが継続的にサイクルします。それ以外の場合、1 回のサイクル後に停止します。	True
休止	Pause プロパティが True に設定されている場合、カルーセル上のマウス ホバーまたはクリック イベントが検出されるとグラフィックのサイクルが一時停止します。グラフィックのサイクルは、カルーセルからマウスを離すと再開します。	True

カルーセル ウィジェットは Bootstrap 4.0 のカルーセル コンポーネントをベースにしています。Bootstrap の詳細については、以下のサイトを参照してください：https://getbootstrap.com/docs/4.0/components/carousel/

Web ブラウザ ウィジェット

Web ブラウザ ウィジェットを使用して、Web サイトを WindowViewer および Web Client に表示できます。

Web Client が HTTPS で実行している場合、読み込むことができるのは HTTPS URL ページだけです。Web Client が HTTP で実行している場合、HTTPS と HTTPS の両方のページを読み込むことができます。クロスドメイン アクセスが Web サービスのポリシーによってブロックされる場合、このウィジェットは機能しません。URL を二重引用符で囲む必要はありませんが、有効な URL である必要があります。

プロパティ

URL: Web サイトのアドレス。

制限事項

- プロトコルを指定しない場合、デフォルトで https が使用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- HTTPS を使用するように Web Client が設定されている場合、読み込むことができるのは HTTPS URL ページだけです。HTTP URL が使用されている場合、Web ブラウザ・ウィジェットは「混合コンテンツ: `'https://localhost/intouchweb'` のページは `HTTPS` で読み込まれましたが、セキュアでないフレーム `'http://**********'` に対するリクエストはブロックされました。コンテンツは `HTTPS` で提供される必要があります。」 というメッセージが表示されます。
- Web サイト ポリシーによってクロス ドメイン（クロス オリジン）アクセスがブロックされる場合、Web ブラウザ ウィジェットは機能しません。Web ページを開くリンクが別のタブに表示されます。

QR コード スキャナ

QRCode_Scanner ウィジェットはカメラに接続して QR コードをスキャンし、結果の文字列を返します。

プロパティ

プロパティ名	説明	デフォルト値
QR コード	スキャンした QR コードの結果の文字列。デフォルト値は空です。	空
自動開始	True に設定すると、カメラが自動的に起動します。	True
自動停止	True に設定すると、QR コードをスキャンした後にカメラが自動的に停止します。	True
起動	True に設定すると、カメラが起動します。	False
停止	True に設定すると、カメラが停止します。	False
背景色	ウィジェットの背景色を設定します。RGB、HTML コード（#FF0000）、または有効な HTML 色名で色の値を指定します。	黒

制限事項

- デバイスにカメラが装備されている必要があります。
- 仮想マシンではなく、物理マシンの QR コードを使用することが推奨されます。
- Web Client をリモートで使用する場合は、セキュアな URL（https://）を使用して Web Client にアクセスします。

使用法

スキャンした値に基づいて、QR コードをスキャンしてグラフィックを表示するスクリプトを設定できます。

ランタイム時、［AutoStart］、［AutoStop］、および［StartStop］ボタンを含むフローティング ツールバーと共に QR コード スキャナ ウィジェットが表示されます。

AutoStart が True に設定されている場合、ウィジェットをロードするとカメラが自動的に起動します。カメラを起動したままにするには、［**AutoStop**］を選択します。

手動でカメラを起動するには、［**StartStop**］を選択して QR コードをスキャンします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

QR コードをスキャンした後もカメラは起動したままになるので、別の QR コードをスキャンできます。カメラを停止するには、［**StartStop**］を選択します。

フローティング ツールバーには、カメラでスキャンした QR コードから派生した QRCode が表示されます。

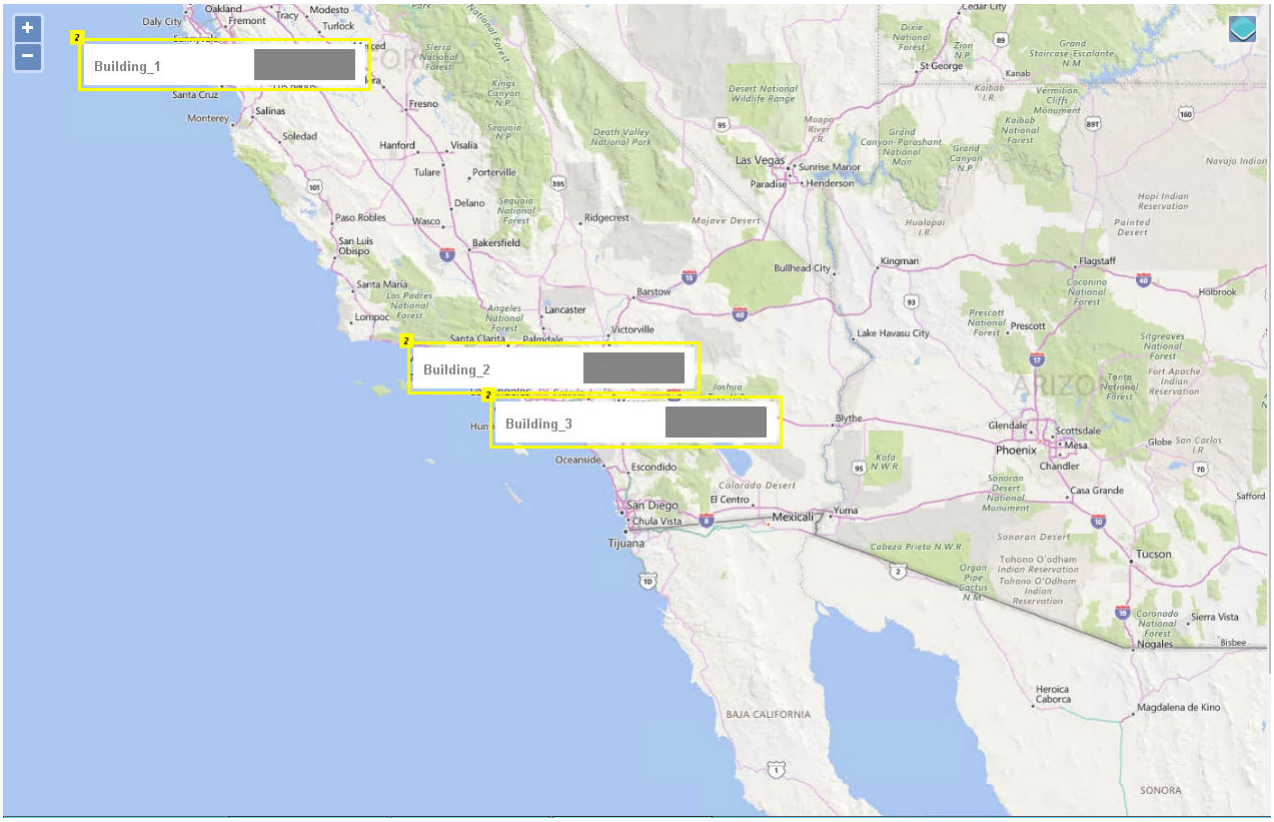
返される QRCode に基づいてアクションのスクリプトを記述できます。

MapApp ウィジェット

Map_App ウィジェットには、実行中のアプリケーション内でシンボルを含むマップを表示できます。ランタイム中、マップは、マップのさまざまな領域のパンや、ズーム インまたはズーム アウトして詳細表示のレベル調整を行うためのコントロールとタッチ サポートを提供します。一般的に、マップ内に配置されたグラフィックは、マップが示す領域内に配置されたビジネス アセットを表します。これらのグラフィックには、各ビジネスの場所のプロセスの現在の状態を示すアラームを含めることができます。

次のマップ例では、各建物の位置と現在のアラーム状態を示すアラーム インジケータが示されています。実行時、ユーザーはマップの左上隅にあるズーム ボタンを選択して、ビューのズームインまたはズームアウトを行うことができます。会社のアセット データをズーム レイヤに割り当てた方法に基づいて、異なるマップ データが表示されます。マップの右上にあるアイコンを選択すると、ユーザーが選択可能なマップが表示され、現在のマップビュー内で異なるデータが表示されます。

ViewApp がタッチ対応デバイス上で実行されている場合、MapApp はタッチ操作をサポートします。標準的なタッチ操作を使用して、マップ上に表示されるアイテムをパン、ズーム、または選択できます。



MapApp の設定: 概要

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowViewer でマップを表示するには、次の手順を実行します。

要件に基づく手順の一部を省略する場合があります。

1. 以下の準備作業を完了します。
   - MAPApp に表示するために UDT を使用する場合は、このステップを実行します。新しい InTouch アプリケーションで、ABT、インスタンス、およびメンバーを作成します。 "Longitude"（経度）や"Latitude"（緯度）などの場所属性を追加します。詳細については、マップアプリケーションのトピックを参照してください。
   - WindowsViewer に表示するデータを持つ、サポートされているマッププロバイダーのアカウントを設定し、マッププロバイダーを使用して、使用する各マップを定義します。
   - ローカル認証が必要なローカル マップ サーバーを使用する場合は、このステップを実行します。特定のマップ ベンダーが表示するデータへのアクセスを制御するためのユーザー資格情報を作成します。資格情報の作成の詳細については、「資格情報マネージャの操作」を参照してください。
   - Industrial Graphic Editor を使用して、マップに表示されるグラフィックのセットを作成します。
2. MapApp のグローバル設定を行います。
   - マップ ソースを定義します。ソース マップ定義には、マップの名前、マップ プロバイダー、マップが基本マップかオーバーレイか、およびマップ プロバイダーによって異なる追加の構成情報が含まれます。WindowViewer で使用する各ソース マップを定義する必要があります。特定のウィンドウで使用するように MapApp を設定する場合、そのウィンドウに含めるソース マップを選択します。

     名前付きズーム レイヤのセットを作成します。各レイヤは、最小ズーム割合と最大ズーム割合の範囲です。
   - 会社のアセットを表すグラフィックを、適切な地理的位置にある各ズーム レイヤに配置します。実行時、各グラフィックは、マップのズームがズーム レイヤの範囲内に設定されている場合にのみ表示されます。必要に応じて、任意のグラフィックを複数のズーム レイヤと 1 つのズーム レイヤ上の複数の場所に割り当てることができます。

   詳細と手順については、「マップ設定」を参照してください。
3. グラフィックを作成し、産業用グラフィック エディタで開きます。
   - **[産業グラフィックを埋め込む]** オプションを使用して、**マップ アプリ** ウィジェットを追加します。
   - Map_App ウィジェットのウィジェット プロパティを編集します。詳細については、「画像での MapApp の設定」を参照してください。
4. WindowMaker でウィンドウを作成し、ドラッグアンドドロップで作成したグラフィックを埋め込みます。
5. WindowViewer に切り替えます。

注:「MapApp」は、InTouch WindowViewer にマップを含めるのに使用できる InTouch MapApp を指します。MapApp を WindowViewer で使用するためにウィンドウに配置すると、"MapControl"が作成されます。MapApp の設定は、1 回のみ実行する必要があるグローバルプロセスですが、いつでも情報の追加や変更を実行できます。MapApp をウィンドウに追加するたびに、新しい MapControl が作成されます。これを設定する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マップ設定

MapApp のグローバルプロパティは、2 つのタイプの情報で構成されています。

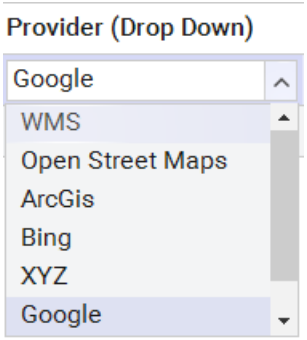
- ソース マップの定義。WindowViewer で使用するために、マッププロバイダーから取得する各マップを定義する必要があります。サポートされているマップ プロバイダーのいずれかまたはすべてのマップ プロバイダーから、1 つ以上のマップを定義できます。各ソース マップは、基本マップまたはマップ オーバーレイのいずれかです。(一部のプロバイダーは基本マップのみをサポートし、他のプロバイダーは基本マップとオーバーレイの両方をサポートします)。実行時、ベースマップは不透明な背景として表示され、1 つ以上のオーバーレイがその上に表示され、それぞれに設定された透明度の度合いに基づいて表示されます。
- ズーム レイヤと含まれるグラフィック:[ズーム レイヤ]タブで **ズーム レイヤ** を作成します。各レイヤでズーム範囲が定義されます(例:10%~20%)。次に、組織のアセット(または他の必要なグラフィック)を表すグラフィックを、[**Locations**]タブの特定の地理的な場所にある各ズーム レイヤに追加します。場所を入力するか、指定した属性の値に基づいて場所を入力できます。特定のズーム レイヤに割り当てられたグラフィックは、マップのズーム レベルがそのズーム レイヤの範囲内にある場合にのみ表示されます。ズーム レイヤと表示されるグラフィックはソース マップから独立しているため、WindowViewer で異なるソース マップを使用する実行時に、同じグラフィックが表示されることがあります。

MapControls を Windows に追加して WindowViewer で使用する前に、これらのグローバルプロパティを定義できます。既定のオプションを使用することもできます。ただし、いつでも戻ってグローバルプロパティを変更または追加できます。MapControls の追加と設定の詳細については、「画像での MapApp の設定」を参照してください。

注： MapApp グローバルプロパティを変更した場合、MapApp を使用する WindowViewer を再度開き、それらの変更を反映させる必要があります。

MapApp グローバルプロパティを設定する

1. WindowMaker を開いて 、**産業用グラフィック > ウィジェット** に移動し 、**マップ アプリ** をダブル選択して **[マップ設定]** ダイアログボックスを表示します。
2. ソース マップを定義します。
   a. [**ソース**] タブを選択します。
   b. 最初の空の **[名前]** フィールドに、このソース マップを与える名前を入力します。
   c. [**プロバイダー**] フィールドで、マッププロバイダーを選択します。

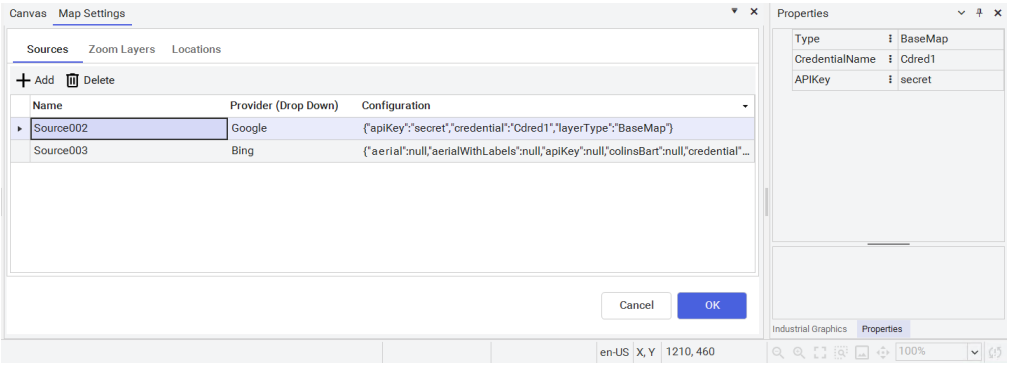


© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

[プロパティ] エリアが更新され、このプロバイダーに適用可能なフィールドが含まれます。また、**[設定]** フィールドには、このプロバイダーのパラメーターが表示されます。**[設定]** フィールドでは直接変更できません。**[プロパティ]** フィールドに入力した情報が反映されます。

d. **[産業用グラフィック)]** ウィンドウの下部で **[プロパティ]** タブを選択し、選択したマッププロバイダーの [プロパティ] を開きます。**[プロパティ]** 領域で、選択したソース マップに値を割り当てます。各マップ プロバイダーのプロパティの詳細については、「マッププロバイダープロパティ」を参照してください。

[設定] フィールドにリストされているマップ プロバイダー のプロパティが更新され、**[プロパティ]** フィールドで選択したプロパティが表示されます。



e. マップ プロバイダーが資格情報を必要とする場合は、**[CredentialName]** フィールドで適切な資格情報を選択します。

注: このステップは、認証が必要なローカル マップ サーバーを使用する場合にのみ必要です。マップ プロバイダーがユーザー資格情報を必要としない場合は、このステップをスキップします。

f. 手順 b-e を繰り返して、すべてのソース マップを定義します。

注:同じプロバイダーから異なるソース マップを使用する場合は、それぞれを個別に定義する必要があります。

3. ズーム レイヤを作成します。ズーム レイヤを使用すると、特定のズーム レベルに基づいて、マップ上に表示されるグラフィックを判断できます。詳細については、「ズーム レイヤ」を参照してください。

ズーム レベルが 0%の場合、地球全体を表示するズームは不要です。A100%ズーム レベルはサポートされている最大ズーム レベルです。道路レベルでマップビューが表示されます。

a. **[ズーム レイヤ]** タブを選択します。

b. 最初の空の **[名前]** フィールドに、このズーム レイヤを表示する名前を入力します。

c. **最小ズーム**および**最大ズーム** パーセンテージを入力します。

d. ステップ b~c を繰り返し、必要な数のズーム レイヤを作成します。

e. 1 つのレイヤを既定のズーム レイヤとして設定するには、そのレイヤを選択してから**[既定に設定]**を選択します。次のステップで画像をズーム レイヤに追加すると、別のズーム レイヤを選択しない限り、これらの画像がこのレイヤーに割り当てられます。

4. グラフィックを追加して、目的の場所にレイヤーをズームします。

a. **[場所]** タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

b. [**産業用グラフィック**] タブで目的のグラフィックに移動し、 **[Locations**] グリッドにドラッグします。

注 : これは、画像に関連付け可能な PICT インスタンスがすでに存在する場合にのみ機能します。

c. [**レイヤ**] で、グラフィックのズーム レイヤを選択します。ズーム レイヤは、実行時にグラフィックを表示する時間を決定します。詳細については、「ズーム レイヤを参照してください。

d. 必要なグラフィックをすべて追加するまでステップ b-c を繰り返します。異なるズーム レイヤや 1 つのズーム レイヤ上の複数の場所に、同じグラフィックを追加できます。**[場所]**タブには、1 つのグラフィックに関連付け可能な、2 つのインスタンスが表示されます。

マッププロバイダープロパティ

以下は、MapApp でサポートされている各マップデータ プロバイダー入力する必要があるプロパティの一覧です。

ローカル マップ サーバーを使用してマップ プロバイダーのマップを格納する場合、各マップ ソースに対して入力する URL は、マップ プロバイダーの Web サイトではなく、ローカル サーバー上のマップを指す必要があります。

重要:サーバーのデータにアクセスできるように、一部のマッププロバイダーとのアカウント設定が必要な場合があります。マップ ソース プロパティを設定する場合、これらのマップ プロバイダーには API キーまたは特定のアカウントが必要です。

ArcGIS

注 : ArcGIS マップを使用して x、y (緯度、経度)、オプションで z (標高) 座標を含める場合は、XYZ マップ プロバイダーを使用します。XYZ プロバイダーの説明を以下に示します。

タイプ	基本マップとマップオーバーレイ。
ソース レイヤ	ArcGIS マップには、複数のレイヤーを作成できます。このソース マップで使用する ArcGIS マップのレイヤを選択できます。 **[ソース レイヤ]**には、4 つのアクションを選択できます。アクションを選択し、右側のフィールドにレイヤを表示します。 • **[表示]**:リストされているレイヤのみを使用します。 • **非表示**:リスト以外のすべてのレイヤを使用します。 • **使用**: 既定のレイヤとリストに表示するレイヤを使用します。 • **除外**:既定のレイヤを使用しますが、表示するレイヤは除外します。 レイヤを表示するには、ArcGIS マップで定義されているレイヤ番号を使用します。複数レイヤーはカンマで区切ります。例えば。 `1,5` はレイヤ 1 とレイヤ 5 を指定します。
CredentialName(資格情報名)	CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。CredentialName プロパティはドロップダウンコントロールで、Credential Manager で設定されたタイプの**ユーザー名とパスワード**、**ドメイン、ユーザー名、パスワード**の名前付き資格情報をリスト表示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ソース URL	使用する ArcGIS REST サービスと定義済みの ArcGIS マップの URL。書式は次のようになります。 https://<servername>.arcgisonline.com/ArcGIS/rest/services/Specialty/<mapname>/MapServer
不透明	オーバーレイの場合のみ、オーバーレイの透明度。0 から 1 の数値で、0 は 100%透明、1 は 100%不透明を示します。

バイドゥ

タイプ	基本マップのみ。
ソース URL	ベースレイヤとして使用する Baidu マップの URL。
CredentialName(資格情報名)	CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。CredentialName プロパティはドロップダウンコントロールで、Credential Manager で設定されたタイプの**ユーザー名とパスワード**、**ドメイン**、**ユーザー名**、**パスワード**の名前付き資格情報をリスト表示します。

Bing

タイプ	基本マップのみ。
API キー	データを提供する MapApp を Bing マップサーバーに認証するための登録済みキー。 API キーの取得については、Microsoft Web サイトを参照してください。書き込み時には、これは次の URL でした。 https://www.microsoft.com/maps/create-a-bing-maps-key.aspx
CredentialName(資格情報名)	CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。CredentialName プロパティはドロップダウンコントロールで、Credential Manager で設定されたタイプの**ユーザー名とパスワード**、**ドメイン**、**ユーザー名**、**パスワード**の名前付き資格情報をリスト表示します。
基本ソース	Bing マップでサポートされる基本マップレイヤタイプ。 • **道路**:地理的領域内の道路。 • **衛星**:オルソグラフィーを利用したや衛星画像など。 • **ハイブリッド**:道路やラベルがオーバーレイされたオルソグラフィーのコードコードと衛星画像。 1 つ以上の基本マップ レイヤー タイプを選択できます。実行時、ユーザーは MapApp に表示される基本ソースを選択できます。

ググる

タイプ	基本マップのみ。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

CredentialName(資格情報名)	CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。CredentialName プロパティはドロップダウンコントロールで、Credential Manager で設定されたタイプの**ユーザー名とパスワード**、**ドメイン、ユーザー名、パスワード**の名前付き資格情報をリスト表示します。
API キー	データを提供している Google マップサーバーに対して MapApp を認証するための登録済みのキー。 API キーの取得について詳しくは、Google マッププラットフォームの Web サイトをご覧ください。書き込み時の URL は次のとおりです。 https://developers.google.com/maps/documentation/javascript/get-api-key

オープン ストリート マップ

プロパティ	説明
タイプ	基本マップのみ。
ソース URL	基本マップ タイルを提供するマップ サーバーの URL。書いている時に、全世界の街マップのために、次を使用することができます: https://www.openstreetmap.org Open Street Maps が提供するマップタイルは無料ですが、https://openstreetmaps.org の OSM サーバーでは使用制限が適用されます。OSM ベース マップ タイル サーバーを使用する場合は、使用中のサーバーが OSM ライセンス要件に準拠していることを確認します。
CredentialName(資格情報名)	CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。CredentialName プロパティはドロップダウンコントロールで、Credential Manager で設定されたタイプの**ユーザー名とパスワード**、**ドメイン、ユーザー名、パスワード**の名前付き資格情報をリスト表示します。

WMS

WMS(Web マップサービス)は、特定のマッププロバイダーを参照していません。むしろ、米国地質学調査所や ArcGIS など、複数のマップ プロバイダーによってサポートされているインターフェイス標準です。

タイプ	基本マップとマップオーバーレイ。
ソース URL	使用するマップの URL。マップ ソースが異なって、異なる URL 形式が使用されます。書式はマップ プロバイダから取得できます。
ソース レイヤ	WMS サーバーからタイルを取得するクエリを作成する場合、マップ プロバイダーサーバーが複数のレイヤをサポートするため、レイヤを指定する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	特定の WMS サーバーで使用可能なレイヤについて不明な場合は、ブラウザーに次の URL を入力して、使用可能なレイヤを照会できます。 <Source URL> ?request=GetCapabilities&service=WMS このクエリはサーバーのすべての機能を一覧表示する XML ファイルを返します。[工程能力/レイヤ]にレイヤのリストが表示されます。各レイヤは「名前」というノードを持ち、これはこのプロパティで指定されるレイヤ名です。
カスタム データ	これは WMS サーバーのカスタムデータとして送信されます。
不透明	オーバーレイの場合のみ、オーバーレイの透明度。0 から 1 の数値で、0 は 100%透明、1 は 100%不透明を示します。
CredentialName(資格情報名)	CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。CredentialName プロパティはドロップダウンコントロールで、Credential Manager で設定されたタイプの**ユーザー名とパスワード**、**ドメイン、ユーザー名、パスワード**の名前付き資格情報をリスト表示します。

XYZ

XYZ は、ArcGIS マップ プロバイダーの関数で、XY 座標という地理的な位置を含むテーブル 形式のデータをマップに追加します。テーブルに Z 座標 (標高値など) も含まれている場合、グローブまたはシーンに 3D コンテンツとしてテーブル データを追加できます。X、y 座標のテーブルを追加するには、テーブルに x 座標用と y 座標用の 2 つのフィールドを含める必要があります。フィールド内の値は、緯度、経度、メートルなどの座標系と単位を表します。3D ジオメトリを有効にする Z 座標のフィールドは任意です。

タイプ	基本マップとマップオーバーレイ。
ソース URL	XYZ サーバーの URL を次のような形式で表示します。 https://<servername>/ArcGIS/rest/services/<mapname>/MapServer/tile/{z}/{y}/{x}
不透明	オーバーレイの場合のみ、オーバーレイの透明度。0 から 1 の数値で、0 は 100%透明、1 は 100%不透明を示します。
CredentialName(資格情報名)	CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。CredentialName プロパティはドロップダウンコントロールで、Credential Manager で設定されたタイプの**ユーザー名とパスワード**、**ドメイン、ユーザー名、パスワード**の名前付き資格情報をリスト表示します。

いくつかのプロパティの詳細

マップ ソース API キー

API キーとは、マップ サーバーに関連付けられたリクエストを、使用および構築の目的で認証する一意の識別子です。アプリケーションフォルダ内に API キーなどの機密情報が保存されないようにするために、資格情報マネージャーで設定された名前付き資格情報を選択するオプションが導入されていま

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

す。マップ ソース設定の API キープロパティは、Named Credentials のシークレットをサポートしています。API キーを入力すると、編集ボックスに「シークレット」タイプの名前付き資格情報が一覧表示されます。これらの名前付き資格情報のいずれかを API キーとして選択できます。引き続き、編集ボックスで API キーを手動で指定するオプションがあります。実行時に、名前付き資格情報を使用することを選択した場合、マップ アプリケーションは安全な場所から API キー情報を読み取ります。API キーは、マップ ソースへの接続時に使用されます。

注： 名前付き資格情報を API キー編集ボックスに表示することは、スタンドアロンの InTouch アプリケーションを使用する場合にのみサポートされます。

Map Source CredentialName(ソース資格情報名をマップする)

CredentialName プロパティは、Windows ユーザーのログインを表す資格情報を指定するためのプロパティです。これは、マップ サーバーで Windows 認証が必要な場合に使用されます。**CredentialName** プロパティはドロップダウンコントロールで、「ユーザー名とパスワード」タイプまたは「ドメイン、ユーザー名、パスワード」の名前付き資格情報をリスト表示します。管理された InTouch アプリケーションの場合、リストに表示される資格情報は Galaxy リポジトリに設定された資格情報になります。スタンドアロンの InTouch アプリケーションの場合、リストされた資格情報はアプリケーションマネージャーで設定されたものになります。実行時に WindowViewer とマップアプリケーションは資格情報を取得し、それを使用してローカルマップサーバーへの認証を行います。

ローカルマップサーバー認証を設定します。

1. ローカル マップ サーバーをインストールする (例: GeoServer。詳細については、「マップ サーバーの設定」を参照してください。
2. マップ サーバー内のユーザー アカウント情報と一致するユーザー名とパスワードを持つユーザーを作成し、ローカル サーバーのマップにアクセスするために必要な役割を付与します。InTouch アプリケーションマネージャでそれに応じて認証するには、このアカウントを使用します。
3. WindowMaker でアプリケーションを開き、必要な設定を確認します。
4. WindowMaker で MapApp ソース設定ページを開き、[**産業用グラフィック**] ペインの下部で [**プロパティ**] タブを選択し、選択したマッププロバイダーのプロパティを開きます。

   対応するプロパティが表示されます。
5. **SourceURL** を指定して、ローカルマップサーバーを指し、 **SourceLayer** も選択します。
6. **CredentialName** プロパティを更新します。

   資格情報マネージャーで作成した資格情報のリストが表示されます。
7. 認証が必要なローカル マップ サーバーを使用しない場合は、 [なし] を選択することもできます。
8. WindowViewer を開き、MapApp がローカルマップサーバーに接続されており、設定された名前付き資格情報で認証されていることを確認します。

マップ サーバーの設定

次のサンプル手順では、オープン ソースの Java Runtime Environment (JRE) と GeoServer という名前のオープン ソース マップ サーバーを使用して、マップ サーバーをインストールして設定する方法を説明します。この手順は、Web ベースのマップサーバーを設定する一般的なワークフローを示す例としてのみ意図されています。MapApp で表示されるマップデータを保存するには、これらの製品または他のオープンソースまたは商用製品をローカルマップサーバーとして使用します。

ローカル マップ サーバーをセットアップし、目的のソース マップをそのソース に保存したら、MapApp マップ ソースを設定して、元のプロバイダーの Web サイトではなく、ローカル サーバー上のマップの

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

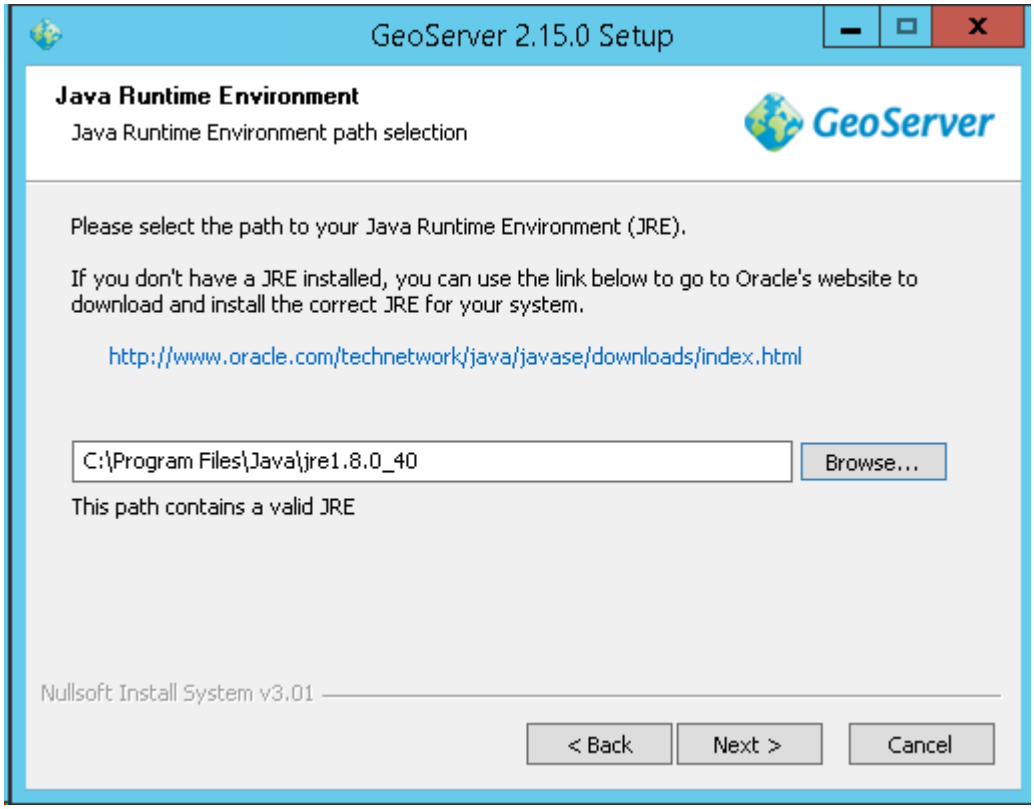
場所を参照するように設定する必要があります。マップ ソースの定義については、「マッププロバイダープロパティ」を参照してください。

注 : ユーザーが別のドメインから接続するときにマップデータを表示できるようにするには、マップサーバーでオリジン間リソース共有(CORS)を有効にする必要があります。CORS を使用すると、あるサーバー上の JavaScript が別のサーバーのリソースにアクセスできます。**これはセキュリティのリスクになりるか**ねないので、ほとんどのサーバは CORS を無効にして、サーバ上のリソースにアクセスする悪意のあるスクリプトの侵入を減らします。CORS を有効にするかどうか決める前に、セキュリティ上の影響をすべて考慮してください。

マップ サーバーを設定するには

これはサンプル **プロシージャのみ**です。お客様のニーズに合わせて、使用可能な任意のマップサーバ製品を使用できます。

1. 64 ビットバージョンの Java Runtime Environment(JRE)をインストールする。
2. GeoServer をインストールし、JRE のフォルダーパスを見つけます。



3. 既定のデータディレクトリを受け入れます。

   `C:\Program Files (x86)\GeoServer 2.15.0\data_dir`

4. GeoServer 管理者のユーザー名とパスワードを設定します。

   デフォルトの管理者ユーザー名とパスワードは `admin` および `geoserver` です。

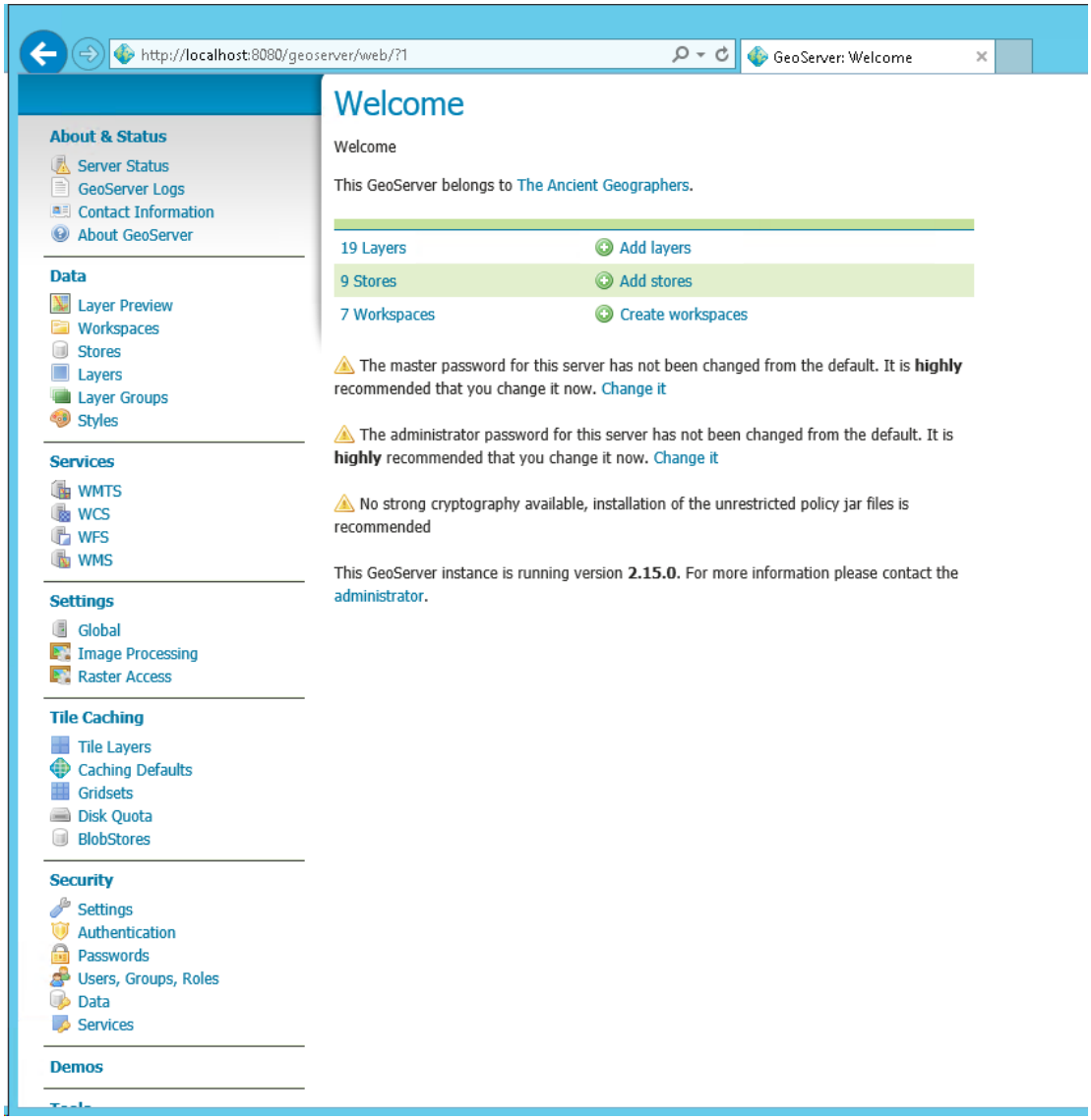
5. 既定では 8080 である GeoServer Web サーバーポートを設定します。
6. インストールの種類に応じて、 [**手動で実行**] オプションを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

7. [**インストール(Install)**] を選択して、マップ サーバーのインストールを開始します。
8. [**終了(Finish)**] を選択して、インストール手順を終了します。
9. GeoServer を起動します。

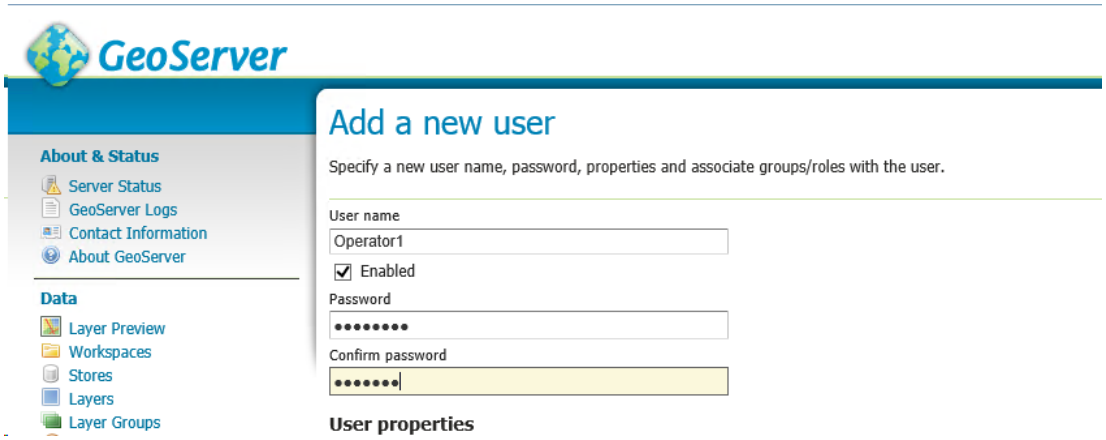
   [**GeoServer を起動する**] オプションは、Windows の [スタート] メニューから選択するか、[GeoServer] フォルダーで startup.bat 実行できます(例: C:\Program Files (x86)\GeoServer 2.15.0\bin)。
10. Web ブラウザーを開き、管理者の資格情報を使用して GeoServer 管理者ページにログオンします。

    `http://localhost:8080/geoserver/web`

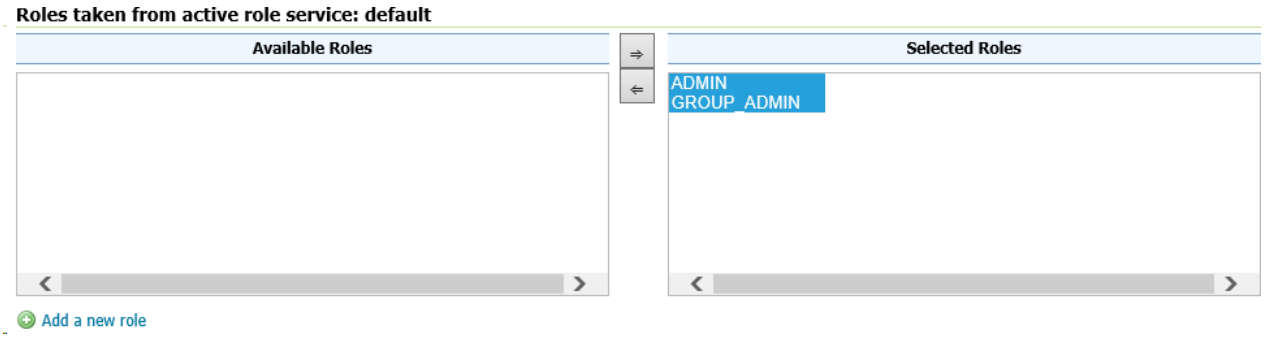


11. 管理者ページの左側の列にある**[Security]**領域から**[ユーザー、グループ、役割]**を選択します。
12. [**ユーザー、グループ、役割**] ページで [**新しいユーザーを追加**] を選択します。
13. 資格情報グループのメンバーである新しいユーザーを追加し、変更を保存します。

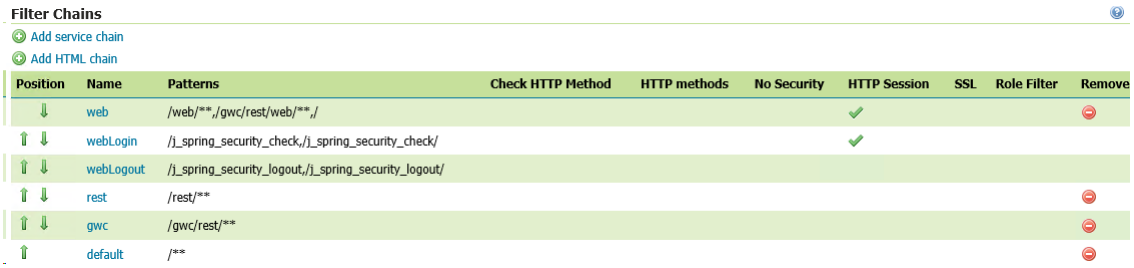
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



14. 資格情報ユーザーに割り当てられた利用可能な役割を、 [**選択された役割]** フィールドに移動します。

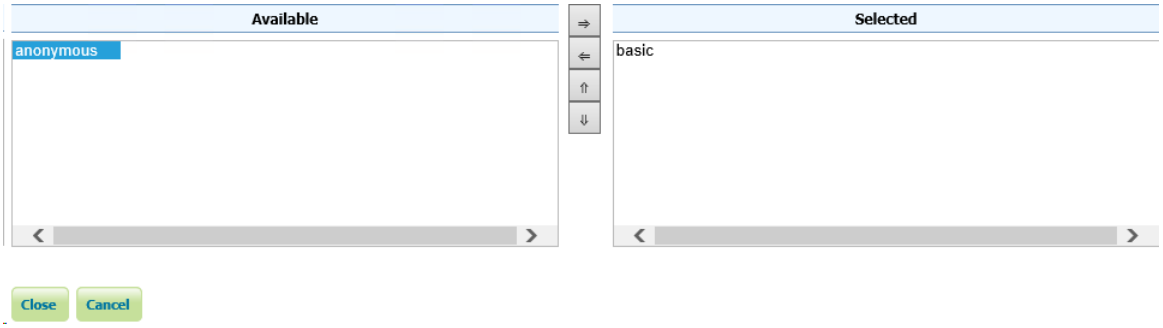


15. 管理者 Web ページの [セキュリティ] 領域から [**認証]** を選択します。
16. **認証** Web ページの **[フィルターチェーン]** 領域で、デフォルト値を選択します。



[フィルターチェーン] Web ページが開きます。

17. 「匿名を **使用可能** 」リストに移動します。



© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

ズーム レイヤ

ズーム レイヤを使用すると、特定の割合にズームしたときにマップ上に表示されるグラフィックを制御できます。ズーム レイヤの設定は、2 段階の手順で行います。

- 1 つ以上のズーム レイヤを作成する:各レイヤーは最小ズーム率と最大ズーム割合を定義します。実行時に、ズーム率がこの範囲内にある場合、このズーム レイヤに割り当てられたグラフィックがマップ上に表示されます。必要な数のズーム レイヤを定義できます。また、レイヤに割り当てるズーム範囲は重なり合うことができます。たとえば、範囲が 10% から 30% のズーム レイヤと、25% から 50% の範囲に対して 1 つのズーム レイヤを作成した場合、マップのズームが 25% から 30% の間に移動した場合、いずれかのズーム レイヤに割り当てられたグラフィックが含まれます。
- 各ズーム レイヤにグラフィックを割り当てる:[**場所**]タブで、1 つ以上のグラフィックを各ズーム レイヤに割り当てることができます。グラフィックを割り当てるときは、緯度と経度を指定して、ズーム レイヤを選択して、そのグラフィックを位置に割り当てます。

実行時に最適な外観を得るために、ズーム レイヤ間の切り替えを取得するには、いくつかの実験が必要な場合があります。状態シンボルや地域シンボルを大きくするのではなく、個々の位置やアセットをその場所に表示する前に、マップをズームインする必要がありますか? 既存のズーム レイヤを変更することも、新規作成することもできます。ズーム レイヤの境界が重なり合うと、ユーザーの表示割合が一部に及び、小さい地域や個々の位置が拡大されるなどの場合に役立ちます。

注： ズーム レイヤの定義またはグラフィックを変更した場合、WindoeMaker を再度開く必要があります。

場所アセット

[**場所**] グリッドの [産業用グラフィック] ウィンドウからグラフィックをドラッグアンドドロップすると、指定した座標のマップレイヤー上にグラフィックを配置できます。**[場所]** グリッドの各行には、実行時にマップ上に表示される画像の設定値が表示されます。

画像をマップ上に配置するには

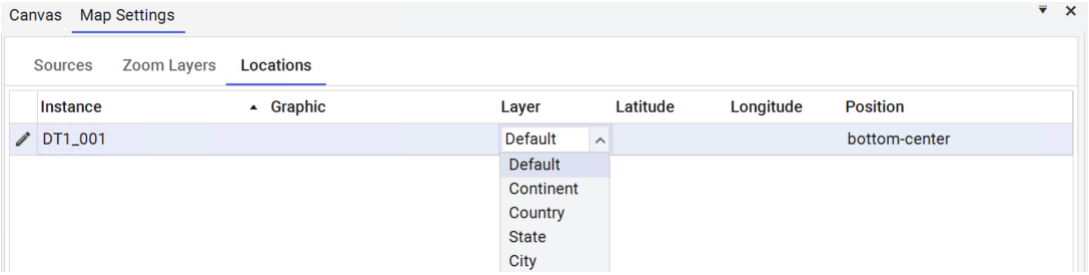
1. [**マップ設定**] から [**場所**] タブ を選択します。

   [Instance] には、使用可能なすべての TBT が表示されます。

2. [産業用グラフィック)] ウィンドウ枠から、マップ上に配置するグラフィックを探します。
3. 目的のグラフィックを **[場所]** グリッド領域にドラッグしてドロップします。
4. リスト内のグラフィックを選択します。

   暗い背景は選択した行を示します。

5. グラフィックの [**レイヤ**] 列を選択すると、定義済みのマップ ズーム レイヤのリストが表示されます。



   各グラフィックは、最初は **デフォルト** のズーム レイヤに割り当てられます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

6. グラフィックを配置するズーム レイヤを選択します。実行時、マップがズーム レイヤの範囲内のパーセンテージにズームされている場合にのみ、グラフィックが表示されます。
7. 変更を保存します。

画像での MapApp の設定

マップ アプリ ウィジェットの特定のプロパティは産業用グラフィック エディタから設定できます。

プロパティ	説明
設定名(ConfigName)	Map_App ウィジェット グローバル設定ファイルの名前。
初期緯度(InitialLatitude)	初期中心点のマップ位置の緯度（角度）。有効な緯度値は +/- 0～90 です。
初期経度(InitialLongitude)	初期中心点のマップ位置の経度（角度）。有効な経度値は +/- 0～180 です。
初期ズーム(InitialZoom)	ランタイム時に最初に表示されたマップのズーム レベル パーセンテージ。
最小ズーム(MinZoom)	ランタイム時にズーム アウトできるマップの最小ズーム パーセンテージ（0～100%）。
最大ズーム(MaxZoom)	ランタイム時にズーム インできるマップの最大ズーム パーセンテージ（0～100%）。
マックスバウンドサウス (MaxBoundsSouth)	画面の視点中心点の垂直パンの動きをマップの下の境界に制約する南側のマップ境界の緯度（+/- 0～90°）。
マックスバウンドウェスト (MaxBoundsWest)	画面の視点中心点の水平パンの動きをマップの左側の境界に制約する西側のマップ境界経度（+/- 0～180°）。
マックスバウンドノース (MaxBoundsNorth)	画面の視点中心点の垂直パン動作をマップの上の境界に制約する北側のマップ境界の緯度（+/- 0～90°）。
マックスバウンドイースト (MaxBoundsEast)	画面の視点中心点の水平パンの動きをマップの右側の境界に制約する東側のマップ境界経度（+/- 0～180°）。
アセット	表示されているマップから選択されたアセットの名前。
現在の緯度(CurrentLatitude)	マップ上で表示されているアイテムの現在の緯度。
現在の経度(CurrentLongitude)	マップ上で表示されているアイテムの現在の経度。
現在のズーム(CurrentZoom)	表示されているマップの現在のズーム レベル。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

現在の資産をフォロー (FollowCurrentAsset)	このプロパティを True に設定すると、MapApp ウィジェットが現在選択されているアセット（コンテキスト）に追随して自動的にマップをパンおよびズームして、アセットの位置および関連付けられたシンボルが表示されます（アセットは Map アプリ エディタ ページの場所タブに追加し、アセット レベルを "-1" に設定する必要があります)。 • マップを開くと、マップは選択されたアセットを中心に表示されます。アセットを使用して ViewApp 内で移動できます。たとえば、すべての地域を表示し、個々の状態のマーカーを表示できます。次に、マップ内で地域を選択することで、地域の詳細を示す別のペインに ViewApp のフォーカスを設定できます。 • マップは、選択したアセットのズーム レイヤーに 1% を加えてズームします。 • アセットが選択されていない場合、または選択したアセットがマップ上に配置されていない場合、マップは初期ズーム レイヤーとマップの中心点を表示します。 **Asset** プロパティが設定されている場合、**FollowCurrentAsset** プロパティを False に設定すると、アセットの場所および関連付けられたシンボルを使用してマップをロードできます。
ｿｰｽ	アプリのマップ設定で構成されたマップ データ ソース。(All) がデフォルト値で、Map アプリに対して指定されたすべてのマップ データ ソースが含まれます。 **注：**All は **Sources** プロパティ値として半角カッコ内に配置する必要があります。 一部のマップ ソースからのデータだけを表示するよう Map アプリを制限する場合、次に示すようにカンマ区切りの文字列を使用して複数のソースを指定します。 OSM、Bing、TemperatureOverlay
ズーム レイヤ(ZoomLayers)	マップ アプリ用に構成されたマップ ズーム レイヤ。(All) がデフォルト値で、Map アプリに対して指定されたすべてのズーム レイヤが含まれます。 **注**：All は **ZoomLayers** プロパティ値として半角カッコ内に配置する必要があります。 一部のズーム レイヤーからのデータだけを表示するよう Map アプリを制限する場合、次に示すようにカンマ区切りの文字列を使用して名前でズーム レイヤーを指定します。 国、州、市

例

1. 新しい InTouch アプリケーションを作成し、 WindowMaker を起動します。
2. UDT を作成し、名前を $Tank に変更します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. タンクの UDT を開き、次のように属性を追加します。
   - Flow: 整数として
   - Temp: 実測値として
4. **"Latitude"（緯度）"**や **"Longitude"（経度）"**などの場所属性を追加します。
5. $Tank の最初のインスタンスを作成し、Tank1 に名前を変更します。
6. 2 番目のインスタンスを作成し、Tank2 に名前を変更します。
7. [Tank1]の下にメンバーを作成し、以下のようにメンバー(属性)の値の下に設定します。
   - Tank1\Flow = 11
   - Tank1.Temp = 54.5
   - Tank1.Latitude = 33
   - Tank1.Longitude = -114
8. Tank2 の下にメンバーを作成し、以下のようにメンバー(属性)の下に設定します。
   - Tank2\Flow = 21
   - Tank2.Temp = 65.5
   - Tank2.Latitude = 36.746
   - Tank2.Longitude = -119.773
9. 新しいグラフィック「Tank1Sym」を作成し、産業グラフィックエディタで開きます。
10. 丸みを帯びた長方形を描画します。
11. 次のように、アナログエレメント用の値表示アニメーションを含む 2 つのテキストプリミティブを追加します。
    - Tank1\Flow
    - Tank1.Temp
12. Tank1 の四角形の内側に、四角形のラベルを追加します。
13. 保存して産業用グラフィック エディタを閉じます。
14. Tank2Sym に対して手順 11~15 を繰り返します。
    - Tank2\Flow
    - Tank2.Temp
15. **[産業用グラフィックス]**ウィンドウで[**ウィジェット**]フォルダーを開き、**マップ アプリ**をダブル選択してマップ アプリ設定を開きます。
16. [**場所**] タブで、各アセットのグラフィックを入力またはドラッグアンドドロップします。Tank1 アセットのグラフィックに Tank1Sym を使用します。Tank2 アセットのグラフィックに Tank2Sym を使用します。
17. Tank2 の場合、ドロップダウンから **[Layer]**を **[State]** に設定します。
18. ［**OK**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

19. 新しいグラフィックを作成し、MyMap という名前を付け、産業用グラフィック エディタで開きます。
20. グラフィックを右選択して［**産業用グラフィックの埋め込み**］を選択します
21. マップ アプリ **ウィジェット**を選択してキャンバスに埋め込み、サイズを調整します。
22. マップ アプリ ウィジェットのウィジェット プロパティを編集します。
    - ConfigName = Default
    - CurrentZoom = 5
    - InitialLatitude = 33
    - InitialLongitude = -11
    - InitialZoom = 30
23. 保存して産業用グラフィック エディタを閉じます。
24. **[ファイル]** [> **[新規作成]** の順に移動し、Frame_window を作成します。
25. ステップ 19 で作成したフレームウィンドウに MyMap グラフィックを埋め込みます。
26. ウィンドウを閉じて保存します。
27. WindowViewer への高速切り替えを行い、MyMap グラフィックを含むウィンドウを開きます。
28. 縮小するには、 **-** ボタンを選択します。

DataGrid ウィジェット

DataGrid ウィジェットは、ランタイム時に InTouch のウィンドウ内に表形式データを表示します。ウィジェットに表示されるデータは、ユーザーがフィルタリングできます。このウィジェットには、ウィジェットの外観や表示される表形式データの見た目をカスタマイズするための一連の視覚的プロパティも含まれます。

Dir...	Y	Date	We...	C...	Con
Exports	2015	2015-01-17T18:30:00.000Z	Sunday	All	All
Exports	2015	2015-01-18T18:30:00.000Z	Monday	All	All
Exports	2015	2015-01-19T18:30:00.000Z	Tuesday	All	All
Exports	2015	2015-01-20T18:30:00.000Z	Wednesday	All	All
Exports	2015	2015-01-21T18:30:00.000Z	Thursday	All	All
Exports	2015	2015-01-22T18:30:00.000Z	Friday	All	All
Exports	2015	2015-01-23T18:30:00.000Z	Saturday	All	All

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイム時のデータのフィルタリング

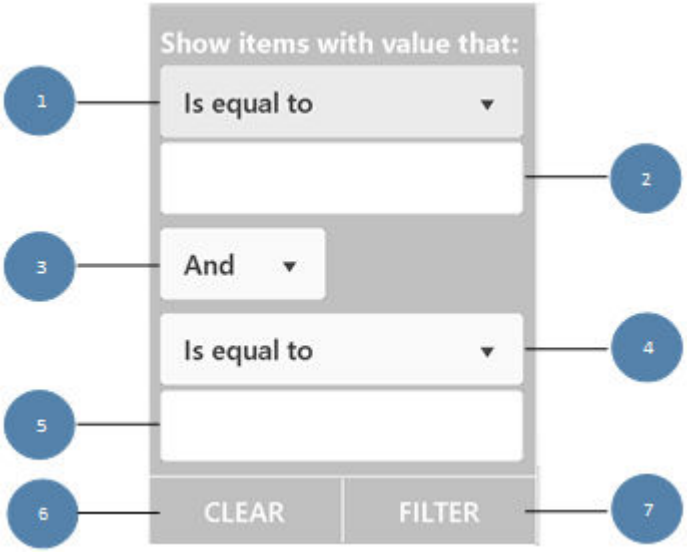
グリッドに表示される各列のデータはフィルタリングできます。フィルターが適用されると、DataGrid ウィジェットは、その列の項目がフィルター条件に一致する行のデータのみを表示します。

次の例は、［週］列にフィルターを適用して木曜日のみを表示するようにフィルタリングする前後の表形式データを示しています。

Dir...	\	Date	Week...	C...
Exports	2015	2014-12-31T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-01T18:30:00.000Z	Friday	All
Exports	2015	2015-01-02T18:30:00.000Z	Saturday	All
Exports	2015	2015-01-03T18:30:00.000Z	Sunday	All
Exports	2015	2015-01-04T18:30:00.000Z	Monday	All
Exports	2015	2015-01-05T18:30:00.000Z	Tuesday	All

Dir...	\	Date	Week...	C...
Exports	2015	2014-12-31T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-07T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-14T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-21T18:30:00.000Z	Thursday	All

各列ヘッダー内のフィルター アイコン を選択すると、フィルター条件を指定するためのドロップダウン ペインが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

1	フィルター演算子のドロップダウン リスト。選択した演算子は、ペアになっているデータ入力フィールド（2）に入力された値との比較を実行します。列データがフィルター条件に一致する場合、その行のデータがグリッドに表示されます。一致しないデータは削除され、グリッドには表示されません。 DataGrid ウィジェットは、以下のフィルター演算子をサポートしています。 • [Is equal to（次と等しい）] • [Is not equal to（次と等しくない）] • 次で開始 • 含む • 含まない • 次で終了 • Null • Null でない • 空 • 空でない • 値がありません • 値があります
2	（1）で選択したフィルター演算子によって評価されるフィルター値を入力するデータ入力フィールド。
3	フィルター条件 1 と 2 をブール演算子（AND、OR）で評価するフィルター ロジック。フィルター条件 2 が空白の場合、フィルター ロジックは適用されません。
4	1 と同じ。対応するデータ入力フィールド（5）が空白の場合、このフィルター演算子は評価されません。
5	2 と同じ。空白のままにできます。
6	［クリア］を選択すると、フィルター条件が削除され、すべてのデータがグリッドに表示されます。
7	［フィルター］を選択すると、フィルター条件が適用されます。

[プロパティ]

以下の表は、DataGrid ウィジェットのプロパティを一覧表示しています。すべてのプロパティは動的ランタイム プロパティとして公開されており、以下のいずれかとして ViewApp 名前空間の参照とバインドできます。

- 属性

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- オブジェクト属性
- MyContent 属性
- 複合式

DataGrid ウィジェットのプロパティ	説明
Type	ソース データ文字列の形式を取得または設定します。 • csv • json
AllowExport	ファイルへのデータのエクスポートを許可します。既定値は True です。
Datasource	DataGrid ウィジェットに表示される JSON または CSV データの文字列を取得または設定します。
ColumnDelimiter	選択したグリッド行のフィールド値を区切るために使用される区切り文字を取得または設定します。 • csv 文字列の場合はカンマ • json 文字列の場合は縦棒（|）
SelectedRow	選択した行フィールド値の文字列を取得または設定します。
DateFieldFormat	日付形式を取得または設定します。すべての日付形式文字列をサポートしています。
背景色	ウィジェットの背景色を設定します。
ExportFileName	AllowExport = True の場合、エクスポートするファイルの名前を指定します。
FilterMode	グリッドのフィルター モードを取得または設定します（default または checklist）。
前景色	ウィジェットのテキストの色を取得または設定します。
HeaderBackColor	列名が表示される表形式ヘッダーの背景色を設定します。
HeaderForeColor	ヘッダー列名のテキスト色を設定します。
SelectionBackColor	グリッド内で選択された行の背景色を設定します。
SelectionForeColor	グリッド内で選択された行のテキスト色を設定します。
RowHoverBackColor	マウスホバー時のグリッド内の行の背景色を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

RowHoverForeColor	マウスホバー時のグリッド内の行のテキスト色を設定します。
Font Family	ウィジェットがテキスト表示に使用するフォント ファミリー名を設定します。
ﾌｫﾝﾄ ｻｲｽﾞ	DataGrid ウィジェットに表示されるテキストのフォント サイズ（ピクセル単位）を取得または設定します。
PageSize	一度に表示するレコード数を取得または設定します。

DataGrid ウィジェットの設定

InTouch HMI ランタイムで DataGrid ウィジェットを実行するための準備には、いくつかの設定ワークフローが含まれます。

- DataGrid ウィジェットのプロパティに値を割り当てる
- DataGrid ウィジェットに表示するソースデータを準備する

この章の次のセクションでは、各設定ワークフローについて説明します。

DataGrid ウィジェットのプロパティの設定

DataGrid ウィジェットのプロパティは、産業用グラフィック エディタでグラフィックにウィジェットを配置すると表示されます。グラフィック内でウィジェットを選択すると、そのプロパティが［**プロパティ**］タブに表示され、値を割り当てることができます。

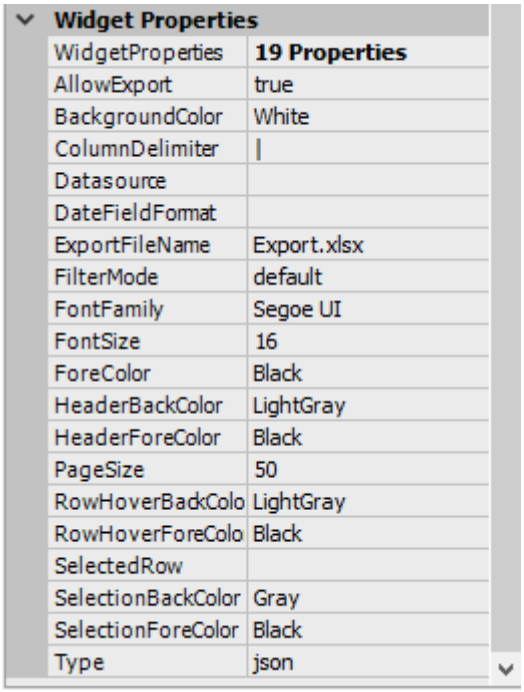
DataGrid ウィジェットのプロパティの設定

1. WindowMaker を開き、［**産業用グラフィック**］ペインを右クリックして［**新しいグラフィック**］を選択し、産業用グラフィックを作成します。
2. 新しく作成したグラフィックをダブルクリックして、産業用グラフィック エディタで開きます。
3. ツール バーの**産業用グラフィックの埋め込み**アイコン を選択します。
4. 表示されたウィンドウの［**グラフィック ツールボックス**］ペインで［**ウィジェット**］フォルダを選択し、**DataGrid** ウィジェットのサムネイルを選択して［**OK**］をクリックします。
5. DataGrid を配置する、キャンバス上の任意の場所を選択します。
6. キャンバス上の DataGrid ウィジェットのサムネイルを選択し、右側にある［**プロパティ**］タブを選択します。

   ［**プロパティ**］タブには DataGrid ウィジェットのプロパティが表示されます。次の画面キャプチャは、ウィジェットに割り当てられている既定値を示しています。

   DataGrid ウィジェットのプロパティの詳細については、「DataGrid のプロパティ」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



Widget Properties	
WidgetProperties	**19 Properties**
AllowExport	true
BackgroundColor	White
ColumnDelimiter	|
Datasource	
DateFieldFormat	
ExportFileName	Export.xlsx
FilterMode	default
FontFamily	Segoe UI
FontSize	16
ForeColor	Black
HeaderBackColor	LightGray
HeaderForeColor	Black
PageSize	50
RowHoverBackColo	LightGray
RowHoverForeColo	Black
SelectedRow	
SelectionBackColor	Gray
SelectionForeColor	Black
Type	json

7. プロパティに値を割り当て、変更を保存します。

DataGrid のプロパティ

以下の表は、DataGrid ウィジェットのプロパティを一覧表示しています。すべてのプロパティは動的ランタイム プロパティとして公開されており、以下のいずれかとして ViewApp 名前空間の参照とバインドできます。

- 属性
- オブジェクト属性
- MyContent 属性
- 複合式

DataGrid ウィジェットのプロパティ	説明
Type	ソース データ文字列の形式を取得または設定します。 • csv • json
Datasource	DataGrid ウィジェットに表示される JSON または CSV データの文字列を取得または設定します。
ColumnDelimiter	選択したグリッド行のフィールド値を区切るために使用される区切り文字を取得または設定します。 • csv 文字列の場合はカンマ • json 文字列の場合は縦棒（|）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

DataGrid ウィジェットのプロパティ	説明
SelectedRow	選択した行フィールド値の文字列を取得または設定します。
DateFieldFormat	日付形式を取得または設定します。すべての日付形式文字列をサポートしています。
背景色	ウィジェットの背景色を設定します。
前景色	ウィジェットのテキストの色を設定します。
HeaderBackColor	列名が表示される表形式ヘッダーの背景色を設定します。
HeaderForeColor	ヘッダー列名のテキスト色を設定します。
SelectionBackColor	グリッド内で選択された行の背景色を設定します。
SelectionForeColor	グリッド内で選択された行のテキスト色を設定します。
RowHoverBackColor	マウスホバー時のグリッド内の行の背景色を設定します。
RowHoverForeColor	マウスホバー時のグリッド内の行のテキスト色を設定します。
Font Family	DataGrid ウィジェットがテキスト表示に使用するフォント ファミリー名を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

DataGrid ウィジェットのプロパティ	説明
ﾌｫﾝﾄ ｻｲｽﾞ	DataGrid ウィジェットに表示されるテキストのフォント サイズ（ピクセル単位）を設定します。
Pagesize	ウィジェットに表示されるグリッド行の数を設定します。

DataGrid ウィジェット用のソース データの準備

DataGrid ウィジェットに表示するソース データは、CSV または JSON 文字列形式で指定できます。

CSV 文字列形式のソース データの場合、最初の行がヘッダー（列名）と見なされます。それ以降の行がデータです。通常、CSV 形式のファイルでは、1 行内のデータ要素間の区切り文字としてカンマが使用されます。

```
Direction,Year,Date,Weekday,Country,Commodity,Transport_Mode,Measure,Value,Cumulative
Exports,2015,01/01/2015,Thursday,All,All,All,$,104000000,104000000
Exports,2015,02/01/2015,Friday,All,All,All,$,96000000,200000000
Exports,2015,03/01/2015,Saturday,All,All,All,$,61000000,262000000
Exports,2015,04/01/2015,Sunday,All,All,All,$,74000000,336000000
```

JSON 文字列形式のソース データの場合、データは辞書オブジェクトの配列として解析されます。配列の最初の要素が、DataGrid ウィジェットのヘッダー（列名）と見なされます。

データの解析の一環として、フィールドのデータ タイプも解決されます。解決されるデータ タイプは次のいずれかです。

- 文字列
- 数値
- 日付

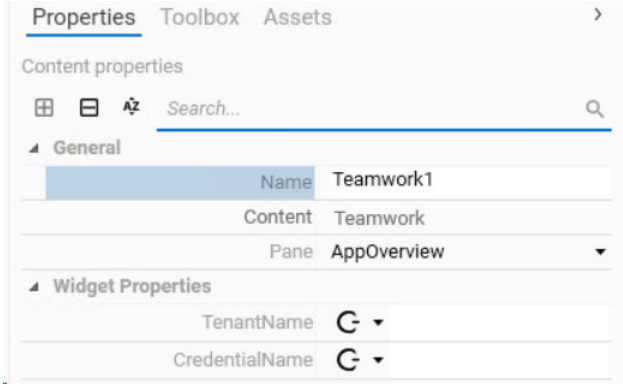
  パーサーは、値が ISO 日付形式（例：2021-08-26T07:44:36.079Z）であるか、ウィジェットの **DateFieldFormat** プロパティで指定された形式である場合にのみ、文字列を日付として認識します。

Teamwork ウィジェット

Teamwork ウィジェットには、次の 2 つのプロパティがあります。

- **TenantName**（必須）は、割り当てられたテナント名を設定する必要があります。
- **CredentialName**（任意）は、ユーザーの資格情報名を設定するか、空のままにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



[プロパティ]	説明	データ型	デフォルト値
TenantName	ランタイム時に特定の Teamwork ページを表示するためのテナント名を指定します。テナント名はテナントごとに一意です。 テナント名を取得するには、管理者にお問い合わせください。	文字列定数または参照	<空の文字列>
CredentialName(資格情報名) (任意指定)。	**CredentialName** プロパティは現在サポートされていません。Teamwork に自動ログインするには、SSO 方式を使用してください。	文字列定数または参照	<空の文字列>

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 14 XML ファイルを使用してウィンドウをインポートする

コマンド ファイルを作成して、コマンド ファイルで WindowMaker を実行し、XML ファイルからウィンドウをインポートします。ウィンドウのインポートに加えて、コマンド ファイルを使用して次を行うことができます。

- 新しいアプリケーションを作成する。
- ウィンドウを削除する。
- ウィンドウの名前を変更する。
- 印刷情報をファイルまたはプリンタに送信する。
- クロスリファレンスをファイルに送信する。

XML のインポート中は、以下の操作は参照されているシンボルでサポートされません。

- シンボルの削除
- シンボルの名前変更
- シンボル名のスワップ

しかし、これらの操作は XML インポートが完了した後に実行できます。

以下のシステム レベルの操作はサポートされません。

- コンソール セッションの起動
- 他のアプリケーションへのフォーカスの切り替え

InTouch DBDump および DBLoad ユーティリティはコマンド ファイルから実行できますしかし、マネージド InTouch アプリケーションの場合、System Platform IDE 拡張機能から DBDump および DBLoad ユーティリティを実行できます。DBDump は、InTouch アプリケーションからのタグ情報を含むエクスポート ファイルを作成します。DBLoad は、タグ情報を InTouch アプリケーションにインポートします。DBDump および DBLoad ユーティリティの詳細については、『AVEVA™ InTouch HMI アプリケーション メンテナンス』参照してください。

XML ファイルを準備する

次の手順は、XML ファイルを作成し、ウィンドウ定義を InTouch アプリケーションにインポートする方法を示します。

1. XML ファイルを作成します。こちらでは使用するツールや基本的な XML 構文を定義しません。
2. XML ファイルが XML 書式設定の標準に準拠しているかどうかを確認してください。XML 検証ツールを使用して XML ファイルの構文を検証できます。XML ファイルを表示するには、Internet Explorer で開きます。
3. アプリケーションでタグを準備します。InTouch DBDump および DBLoad ユーティリティを使用して、タグを抽出、変更および再ロードします。DBDump、DBLoad、および CSV ファイルの詳細については、『*AVEVA™ InTouch HMI アプリケーション メンテナンス*』を参照してください。
4. 必要な機能を実行するための WindowMaker コマンド ファイルを準備します。
5. InTouch アプリケーション、WindowMaker、および WindowViewer を終了します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

6. コマンド ライン プロンプトから WindowMaker コマンド ファイルを実行するか、プログラムまたは IDE 拡張機能からコマンド ファイルを実行します。
7. エラーを確認します。エラー ログ ファイルを定義した場合、エラー ログ ファイルとロガーを確認します。プログラムからコマンド ファイルを実行した場合、戻りコードを確認します。

XML インポート機能を使用してウィンドウ定義を作成およびインポートする際は、以下のサンプルが役に立ちます。

- InTouch アプリケーションにウィンドウ定義をロードするサンプル XML ファイル。
- サンプル コマンド ファイル。
- 2 つのスキーマ ファイル。インポート パーサーでは、これらのスキーマ ファイルは使用されません。スキーマ ファイルの制限は XML インポート パーサーよりも厳密です。

注意 : スキーマ ファイルは、インストールが完了した後に InTouch インストール フォルダに配置されます。

コマンド ファイルを準備する

WindowMaker コマンド ファイルはテキスト ファイルです。このファイルからコマンドが読み取られ、InTouch アプリケーション データ セットに対してアクションが実行されます。

コマンド ファイルはシングルバイトの ANSI 文字セットを使用します。マルチバイト文字セット（MBCS）および Unicode 文字は使用できません。

テキスト ファイルの構文の詳細を以下に示します。

- 8 ビット ANSI 文字セットが使用されますが、使用できるのは 127 文字までです。ほとんどの制御文字は削除されます。
- # 文字で始まる行はコマンドとして扱われます。
- 空白行は無視されます。
- 行はキャリッジ リターン ライン フィード（CRLF）シーケンスで終了します。
- スペースは、行の最初、行の最後、コマンドとコマンドの間、等号文字、およびその引数に使用できます。
- 引用符が必要な引数では、引用符とその他のテキストの間にスペースを使用することはできません。
- 行の最初および最後のホワイト スペースは、各行が処理される前に削除されます。
- 各コマンドはピリオドで開始します。
- 各コマンド ファイルは WindowMaker コマンドファイル コマンド（.WINDOMAKERCOMMANDFILE）で始まる必要があります。

1 つのコマンド ファイルに複数のコマンドを含めることができます。コマンドが失敗した場合、コマンド プロセッサはファイル内の次のコマンドの処理を続けます。

最小コマンド ファイルを作成する

最小のコマンド ファイルの例を次に示します。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ファイルに印刷情報を送信する

InTouch アプリケーション全体のリストをプリンタまたはテキスト ファイルに送信できます。InTouch ウィンドウに産業用グラフィックが含まれる場合、出力は HTML ファイルに送信されます。

次の例のコマンド ファイルは、アプリケーション情報を Printer01 というプリンタに出力します。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.PRINTAPPLICATIONINFORMATION
.OUTPUTTARGET=Printer
.OUTPUTTARGETNAME=Printer01
.GO
```

ファイルに出力するコマンド ファイルの例を以下に示します。ファイルが存在する場合は上書きされます。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.PRINTAPPLICATIONINFORMATION
.OUTPUTTARGET=TextFile
.OUTPUTTARGETNAME=C:\MyApps\AppInfo.txt
.GO
```

クロスリファレンスをファイルに送信する

クロスリファレンスを作成してファイルに送信できます。ファイルが存在する場合は上書きされます。ユーザーの操作は必要ありません。

コマンド ファイルの例を以下に示します。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.CROSSREFERENCE
.SEARCHFOR=TagName
.REFERENCETYPE=ByWindow
.OUTPUTFILE=C:\MyApps\AppCrossRef.Csv
.GO
```

ログ ファイルを作成する

情報メッセージ、警告、およびエラーをテキスト ファイルに記録できます。このコマンドを指定しない場合、処理ステータスはロガーに送信されます。ファイルが既に存在する場合は上書きされます。

エラーをファイルに記録するコマンドの例を以下に示します。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.COMMANDLOGFILE=C:\MyApps\LogFile.Txt
```

MyApps フォルダが存在することを確認してください。MyApps フォルダが存在しない場合、ログ ファイルは作成されません。

コマンドの構文

以下の表には、WindowMaker コマンド ファイルで使用できるコマンドのリストが一覧表示されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

コマンド	説明
.WINDOWMAKERCOMMANDFILE	ファイルが WindowMaker コマンド ファイルであることを識別します。このコマンドが見つからない場合、ファイルは処理されず、WindowMaker はエラー コードで終了します。
.VERSION=1	このコマンドは、読み取るファイルがあまり新しくないことを WindowMaker に伝えます。
.COMMANDLOGFILE=<フル ファイル パス>	指定したアイルに情報、警告、およびエラーが記録されます。ファイルが既に存在する場合は上書きされます。このコマンドを省略した場合、ロガーで処理ステータスに関する情報を確認します。
.GO	コマンドを実行します。1 つ以上のパラメータを含むコマンドの後に必要です。
.WINDOWCREATE	XML ファイルからウィンドウを作成します。
.XMLFILEPATH=<フル ファイル パス>	ウィンドウ仕様を含む XML ファイルへのフル ファイル パス。WINDOWCREATE コマンドを使用する場合に必要です。
.WINDOWDELETE	名前で指定した InTouch アプリケーションを削除します。ウィンドウ名が見つからない場合、エラーは生成されません。
.WINDOWNAME=<ウィンドウ名>	削除するウィンドウの名前 WINDOWDELETE コマンドを使用する場合に必要です。
.WINDOWRENAME	ウィンドウの名前を変更します。古いウィンドウ名が見つからない場合、警告が表示されます。新しい名前と同じ名前のウィンドウが存在する場合、エラー メッセージが生成されます。ウィンドウの名前を変更できない場合、エラーが発生します。古い名前と新しい名前が同じ場合、何の処理も行われません。
.OLDWINDOWNAME=<既存のウィンドウ名>	現在のウィンドウ名。WINDOWRENAME コマンドを使用する場合に必要です。
.NEWWINDOWNAME=<新しいウィンドウ名>	新しいウィンドウ名。この名前のウィンドウが存在しないことを確認してください。WINDOWRENAME コマンドを使用する場合に必要です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

コマンド	説明
.PRINTAPPLICATIONINFORMATION	InTouch アプリケーション データ セットをテキスト ファイルに出力またはダンプします。これは、WindowMaker から［ファイル］メニュー コマンドの［印刷］オプションを選択した場合と同じ動作です。
.OUTPUTTARGET=Printer|TextFile	出力をプリンタまたはテキスト ファイルに送信します。
.OUTPUTTARGETNAME=<プリンタ名>|<フルテキスト ファイル パス>	プリンタまたは出力ファイルの名前。ファイルが存在する場合は上書きされます。
.CROSSREFERENCE	クロスリファレンス情報をカンマ区切り変数（CSV）形式で生成します。
.SEARCHFOR= TagName|QuickFunctions	タグまたはクイック関数を名前で検索します。
をクリックします。REFERENCETYPE= ByTagName|ByWindow	タグまたはウィンドウ名によるクロス リファレンス。
をクリックします。OUTPUTFILE=<フル ファイル パス>	出力ファイルへのフル パス。ファイルが存在する場合は上書きされます。

アプリケーションを作成する

WindowMaker コマンド ファイルを使用して、新しいデフォルト InTouch アプリケーションを作成できます。最小のコマンド ファイルを使用して空のアプリケーションを作成します。しかし、最小のコマンド ファイルを使用して空のマネージド InTouch アプリケーションを作成することはできません。詳細については、「最小コマンド ファイルを作成する」を参照してください。

コマンド ファイルは、アプリケーションを作成するフォルダに配置する必要があります。フォルダに既存の InTouch.ini ファイルが含まれる場合、アプリケーションは生成されません。

新しいアプリケーション フォルダへのパスには、埋め込まれた "-l" および "-L" 文字シーケンスを使用することはできません。たとえば、`C:\MyApps\App-Large` というフォルダを作成することはできません。

作成された InTouch.ini ファイルには、次の例のような内容が含まれます。ウィンドウの位置は、アプリケーションを表示する画面解像度により異なります。アプリケーションのタイトルと説明は "Generated InTouch Application" です。デフォルト言語は英語です。

InTouch.ini ファイルの内容の例を以下に示します。

```
[InTouch]
AppMode=2
AppName0=Generated InTouch Application
AppName1=
AppName2=
AppName3=
AppDesc0=Generated InTouch Application
AppDesc1=
AppDesc2=
AppDesc3=
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
LanguageBase=English (United States)
LanguageBaseID=1033
InTouchView=0
SAOConverted=1
WinFullScreen=1
WinLeft=-4
WinTop=-4
WinWidth=1288
WinHeight=1004
SnapOn=1
```

新しく作成されたアプリケーションにタグを追加する

新しい InTouch アプリケーションにはシステム タグだけが含まれます。ウィンドウをインポートする前にアプリケーションにタグを追加する必要がある場合、次の手順に従います。

1. 新しいアプリケーションを作成する。
2. DBLoad を実行してタグを新しいアプリケーションにインポートします。
3. ウィンドウをインポートします。

以下に例を示します。

```
WM.Exe C:\MyApps\App001 COMMANDFILE="C:\BlankFile.Txt"
DBLoad C:\MyApps\App001,C:\TagDumps\App001.Csv,0
WM.Exe C:\MyApps\App01 COMMANDFILE="C:\Commands.Txt"
```

ウィンドウを削除します

同じ名前のウィンドウがある既存のアプリケーションにウィンドウを追加する場合、既存のウィンドウを削除する必要があります。既存のウィンドウは WindowMaker コマンド ファイルを使用して削除することができます。

重要: 削除するウィンドウが存在しない場合、エラー メッセージは表示されません。

既存のウィンドウを削除できない場合、ロガーにメッセージが表示されます。ユーザーの操作は必要ありません。

InTouch アプリケーションからウィンドウを削除するには、コマンド ファイルにコマンドのシーケンスを入力します。コマンド ファイルには、複数のウィンドウ削除コマンド シーケンスを含めることができます。

次の例は、InTouch アプリケーションからウィンドウを削除するコマンド シーケンスを示します。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.WINDOWDELETE
.WINDOWNAME=Window002
.GO
```

ウィンドウの名前を変更する

InTouch アプリケーション内のウィンドウの名前を変更できます。

InTouch アプリケーションからウィンドウの名前を変更するには、コマンド ファイルにコマンドのシーケンスを入力します。コマンド ファイルには、複数のウィンドウ名前の変更コマンド シーケンスを含めることができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

名前を変更するウィンドウが存在しない場合、エラー メッセージは表示されません。

新しい名前と同じ名前のウィンドウが既に存在する場合、何の処理も行われず、警告メッセージが記録されます。

古い名前のウィンドウが存在し、新しい名前と同じ名前のウィンドウが存在しなくても何らかの理由で名前の変更が失敗した場合、エラー メッセージが記録されます。

ユーザーの操作は必要ありません。警告とエラー メッセージがロガーに表示されます。

ウィンドウの名前を既存のウィンドウの名前に変更した場合、警告メッセージがログ ファイルまたは Logger に記録されます。以下の例は、Window005 の名前を既存のウィンドウ Window006 の名前に変更すると、警告メッセージがログ ファイルまたはロガーに記録されます。

```
.WINDOWRENAME
.OLDWINDOWNAME=Window005
.NEWWINDOWNAME=Window006
.GO
```

ウィンドウをインポートします

ウィンドウを InTouch アプリケーションにインポートできます。InTouch アプリケーションからウィンドウの名前を変更するには、コマンド ファイルにコマンドのシーケンスを入力します。新しいウィンドウ名は XML ファイル内に含まれます。

次の場合、ウィンドウはインポートされません。

- 同じ名前のウィンドウがアプリケーション内に存在する場合。
- ウィンドウ スクリプト内で解決できないエラーが発生した場合。
- オブジェクトの一部をウィンドウに追加しようとしたときに解決できないエラーが発生した場合。

次の場合、ユーザーの操作が必要です。

- スクリプトまたは式で指定した要素にエラーが含まれる場合、または要素が欠落している場合。
- スクリプトに構文エラーが含まれる場合。

ユーザーが問題を修正するためのメッセージは表示されないことがあります。詳細については、ログ ファイルまたはロガーを確認してください。

ウィンドウをインポートするには、コマンド ファイルにコマンドのシーケンスを入力します。

以下に例を示します。

```
.WINDOWCREATE
.XMLFILEPATH=C:\WMCommandTest\WMCreateFile.Xml
.GO
```

コマンド ファイルには、複数のウィンドウ インポート コマンドおよびいくつかのウィンドウ作成コマンド シーケンスを含めることができます。

エラーを処理する

XML ファイルは一般的な XML 書式設定ルールに従う必要があります。XML ファイルを Internet Explorer で開けない場合、XML アイルに XML 書式設定エラーが含まれています。WindowMaker でファイルを使用する前に、すべてのエラーを修正します。

一般的な書式設定エラーを含むファイルを使用すると、WindowMaker は「XML ファイルをロードできませんでした」というエラー メッセージがロガーに記録されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スクリプトで要素が欠落している場合、スクリプト エラーがある場合、またはその他の要素仕様エラーが発生した場合、WindowMaker は停止します。以下に例を示します。

- タグの欠落
- 外部 WindowMaker スクリプト拡張 DLL の欠落
- ActiveX コントロールの欠落
- ウィザード DLL の欠落

特定のアニメーション リンクまたはカスタム プロパティの上書きなどの場合、タグが欠落していてもメッセージ ボックスは表示されません。ロガーにメモが書き込まれ、アニメーション リンクとオブジェクトは作成されず、ウィンドウは作成されません。その他の場合、スクリプトおよび式の解析は停止します。その場合、タグを作成することができます。タグが正常に作成されると、処理が継続されます。

SmartSymbol の欠落

SmartSymbol テンプレートがターゲット アプリケーションに存在しない場合、ウィンドウはインポートされません。エラー メッセージがロガーおよび出力テキスト ファイルに表示されます。

何らかの理由で SmartSymbol インスタンスの作成に失敗した場合、ウィンドウは作成されません。SmartSymbol テンプレートが存在していても SmartSymbol のインポートが失敗することもあります。失敗に関する追加情報がロガーに記録されている可能性があります。

産業用グラフィックの欠落

産業用グラフィック参照が Galaxy リポジトリに存在しない場合、ウィンドウは作成されません。エラーメッセージがロガーおよび出力テキスト ファイルに表示されます。

産業用グラフィック参照が存在していても XML のインポート プロセスが失敗することもあります。失敗に関する追加情報がロガーに記録されます。

式、タグ名、およびスクリプト

式、タグ名、およびスクリプトが必要な場合、空のテキストやブランク テキストは使用できません。これらのテキスト項目が正しく作成されていない場合、式、タグ名、およびスクリプトは正しく解析されず、エラーが発生します。

タグ名の場合、タグ タイプはオブジェクトで予期されるタイプに一致する必要があります。ほとんどの場合、ドットフィールドを使用してタグ プロパティにアクセスできます。たとえば、整数型タグ iTag005 の 2 番目のビットには次のようにしてアクセスできます。

```
iTag005.02
```

その場合、タグとドットフィールドは論理値に解決され、論理型タグ名が必要な場所で使用できますが、iTag005 だけではエラーの結果になります。

コマンド プロンプトから WindowMaker を実行する

コマンド プロンプトから WindowMaker を実行できます。

コマンド ラインは以下のとおりです。

```
WM.EXE AppPath,Commandfile="Command.txt"
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

AppPath
InTouch アプリケーションのフル パスを定義します。WindowMaker をアプリケーションのフォルダで実行している場合、このパラメータはオプションです。

CommandFile
コマンド ファイルのフル パスを定義します。パス名は引用符で囲む必要があります。等号の周囲に余分なスペースを挿入することはできません。

InTouch アプリケーションが `C:\MyApps` フォルダにあり、コマンド ファイルが `C:\WMCommandFile.txt` である場合を考えてみます。次の例は、コマンド プロンプトから入力する WindowMaker コマンドを示します。

```
WM.EXE C:\MyApps,COMMANDFILE="C:\WMCommandFile.Txt"
```

注意 : コマンド プロンプトからマネージド InTouch アプリケーションを作成することはできません。

System Platform IDE 拡張機能

System Platform IDE 拡張機能を使用してマネージド アプリケーションに対して InTouch XML インポート機能を実行できます。IDE 拡張機能を使用して XML インポート プロセスのコマンド ファイルを選択できます。コマンド ファイルを選択した後、WindowMaker が起動し、関連付けられた XML が解析されます。

重要: XML インポートが進行中のときに新しいコンソール セッションを開始した場合、または既存のコンソール セッションを最大化した場合、WindowMaker は応答を停止します。

System Platform IDE には、XML インポートからコマンド ファイルを選択するために使用するコンテキスト メニュー項目があります。InTouchViewApp 派生テンプレートを右クリックすると、[**InTouch コマンド ファイルの処理**] メニュー項目がコンテキスト メニューに表示されます。しかし、以下のものを選択した場合、このメニュー項目は表示されません。

- 複数の InTouchViewApp テンプレート
- InTouchViewApp ベース テンプレート
- InTouchViewApp インスタンス

古いバージョンの InTouch からマネージド InTouch アプリケーションを移行した場合、コマンド ファイルを処理する前に少なくとも 1 回は WindowMaker で InTouch アプリケーションを開く必要があります。

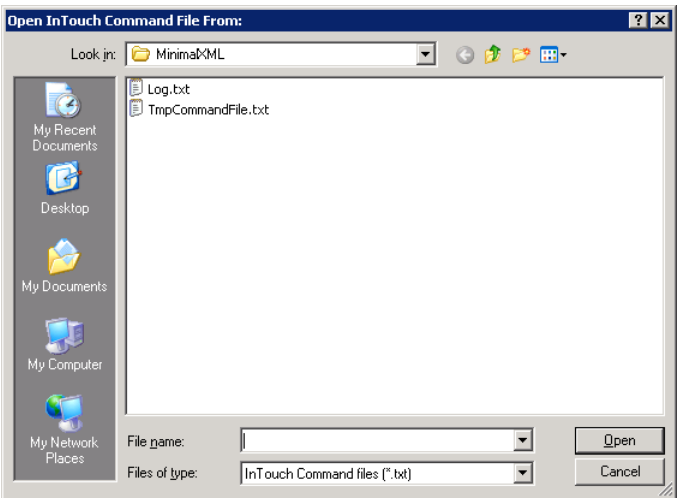
マネージド InTouch アプリケーションから WindowMaker を実行する

マネージド InTouch アプリケーションから WindowMaker 実行できます。

マネージド InTouch アプリケーションから WindowMaker を実行する

1. System Platform IDE を開きます。
2. [**テンプレート ツールボックス**] で、InTouchViewApp テンプレートから派生テンプレートを作成します。
3. 新しい InTouch アプリケーションを作成するか、既存のスタンドアロン InTouch アプリケーションをインポートすることによって InTouch アプリケーションを派生テンプレートに関連付けます。
4. 派生テンプレートを右クリックし、[**InTouch コマンド ファイルの処理**] を選択します。[**InTouch コマンド ファイルを開く:**] ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



注意：［InTouch コマンド ファイルの処理］コマンドを実行する前に WindowMaker で少なくとも 1 回 InTouchViewApp 派生テンプレートを設定する必要があります。

5. コマンド ファイルの場所を参照して［開く］を選択します。WindowMaker が起動します。

コマンド プロンプトから DBDump を実行する

DBDump を使用して InTouch アプリケーションから外部タグ情報を抽出します。コマンド プロンプトから DBDump を実行できます。DBDump を実行する前に WindowMaker を停止する必要があります。

複数の DBDump コマンド パラメータの間にはカンマが必要です。パラメータは位置に依存します。含まれるパラメータ間で 1 つのパラメータを省略する場合、省略するパラメータのカンマを含める必要があります。次の例は、コマンド プロンプトから DBDump コマンドを実行する場合の構文を示します。

```
DBDump AppPath,CsvPath,GroupTypes,OverwriteCsvFile, MessageBoxes
```

パラメータ

AppPath
InTouch アプリケーションのパスを定義します。

CsvPath
InTouch アプリケーションのタグ名ディクショナリからのタグ定義を含むエクスポート ファイルのパスを定義します。

GroupTypes
InTouch タグ タイプを基準に、タグを DBDump エクスポート ファイル内でグループ化するかどうかを指定します。1 を指定した場合、タグ データベース名がタグ グループ タイプを基準に並べ替えられます。0 を指定した場合、タグ名はタイプを基準に並べ替えられません。

OverwriteCsvFile
エクスポート ファイルを上書きするかどうかを指定します。1 を指定した場合、エクスポート ファイルは上書きされます。0 を指定した場合、エクスポート ファイルは上書きされません。

MessageBoxes
DBDump ユーティリティがアプリケーションのタグ名ディクショナリの内容をエクスポートするときにメッセージを表示するかどうかを指定します。1 を指定した場合はメッセージボックスが表示されます。0 を指定した場合はメッセージボックスが表示されません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

InTouch アプリケーションが C:\MyInTouchApps\App001 フォルダにあり、タグ データベースの内容を C:\TagDumps\App001.csv ファイルに書き込む場合を考えてみます。メッセージ ボックスを表示せず、既存のターゲット エクスポート ファイルを上書きするとします。コマンドを次に示します。

```
DBDump C:\MyInTouchApps\App001,C:\TagDumps\App001.Csv,1,1,0
```

マネージド InTouch アプリケーションに対して DBDump を実行する

マネージド InTouch アプリケーションに対して DBDump を実行できます。

マネージド InTouch アプリケーションに対して DBDump を実行する

1. System Platform IDE を開きます。
2. ［テンプレート ツールボックス］で InTouchViewApp 派生テンプレートを右クリックし、［エクスポート］をポイントして［CSV］を選択します。

   ［オートメーション オブジェクトを CSV にエクスポート］ダイアログ ボックスが表示されます。
3. CSV ファイルの名前を入力して、［保存］を選択します。アプリケーション タグ データが CSV ファイルにダンプされます。

コマンド プロンプトから DBLoad を実行する

DBLoad を使用してタグ情報を InTouch アプリケーションにインポートします。コマンド プロンプトから DBLoad を実行できます。DBLoad を実行する前に WindowMaker を停止する必要があります。

DBLoad コマンド パラメータは位置に依存します。含まれるパラメータ間で 1 つのパラメータを省略する場合、省略するパラメータのカンマを含める必要があります。次の例は、コマンド プロンプトからで DBLoad コマンドを実行する場合の構文を示します。

```
DBLoad AppPath,CsvPath,MessageBoxes
```

パラメータ

AppPath
InTouch アプリケーションのパスを指定します。

CsvPath
アプリケーションのタグ名ディクショナリにインポートするタグ定義を含むファイルへのパスを指定します。

MessageBoxes
DBLoad ユーティリティがインポート ファイルの内容をアプリケーションのタグ名ディクショナリにインポートするときにメッセージを表示するかどうかを指定します。1 を指定した場合はメッセージボックスが表示されます。0 を指定した場合はメッセージボックスが表示されません。

例

InTouch アプリケーションが C:\MyInTouchApps\App001 フォルダにあり、タグ データベース情報を C:\TagDumps\App001.Csv ファイルから読み取る場合を考えてみます。その際にメッセージ ボックスは表示しないようにします。次の例は、コマンド プロンプトに入力する DBLoad コマンドを示します。

```
DBLoad C:\MyInTouchApps\App001,C:\TagDumps\App001.Csv,0
```

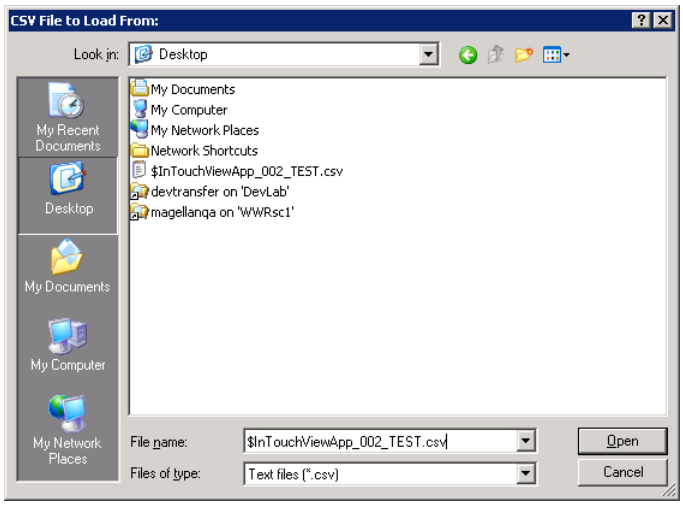
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マネージド InTouch アプリケーションに対して DBLoad を実行する

マネージド InTouch アプリケーションに対して DBLoad を実行できます。

マネージド InTouch アプリケーションに対して DBLoad を実行する

1. System Platform IDE を開きます。
2. ［**テンプレート ツールボックス**］で InTouchViewApp 派生テンプレートを右クリックし、［**インポート**］をポイントして［**DB ロード**］を選択します。［**ロード元 CSV ファイル名:**］ダイアログ ボックスが表示されます。



3. CSV ファイルの場所を参照して［**開く**］を選択します。タグ データが InTouch アプリケーションに正常にロードされます。

XML の形式

XML ファイルで定義されているウィンドウおよびほとんどのウィンドウ要素をインポートできます。タグを XML ファイルからインポートすることはできません。DBLoad ユーティリティを使用する必要があります。

一般的な XML ファイル形式

XML ファイルは一般的なすべての XML 書式設定ルールに従う必要があります。XML ファイルが Internet Explorer で開けることを確認してください。開けない場合、XML ファイルに書式設定エラーが含まれています。

スキーマを使用する

XML ファイルをスキーマ ファイルで検証できます。スキーマ ファイルで XML ファイルを検証すると、ほとんどの XML 書式設定エラーが検出されます。

スキーマの制限は XML 入力パーサーよりも厳密です。

- スキーマでは、要素の順序が表に一覧表示されている順序に一致する必要があります。
- 要素名と値では大文字と小文字が区別されます。
- スキーマでは要素を明示的に定義する必要がある場合があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スキーマ検証はウィンドウ要素から呼び出すことができます。スキーマの指定方法の詳細については、「ウィンドウの定義」を参照してください。

XML ファイル ヘッダー

ファイルの上部に XML 宣言を含める必要があります。最小の宣言を以下に示します。

```
<?xml version="1.0"?>
```

非必須の XML 要素

InTouch の XML インポート機能では、未定義の要素、属性、およびノードに関するエラーと警告は生成されません。

すべての要素、属性、およびノードを正しく入力する必要があります。<ONRIGHTDOWN> アクション スクリプトを使用する際に間違って <ONRIGHTDOWN> と入力すると、警告は生成されず、アクション スクリプトはウィンドウ オブジェクトに追加されません。

必須の要素が欠落している XML 定義に関する警告とエラーは作成されます。

ユーザー指定のテキスト

スクリプトで XML フィールド区切り文字が使用されている場合、スクリプト テキストを CDATA 要素内にカプセル化する必要があります。CDATA 要素内のテキストは XML ファイル インポート機能によって解析されません。

テキストに XML フィールド区切り文字が含まれない場合、CDATA 要素内でカプセル化せずにテキストを入力できます。

テキスト ホワイト スペースを維持する

CDATA 要素にないテキストが処理されると、先頭と末尾のスペースは削除されます。その結果、 <Title> MyWindowName</Title> のような要素が、先頭にスペースを含まない「MyWindowName」というウィンドウ名になります。

テキストの先頭と末尾のホワイト スペースを維持するには、テキストを CDATA 要素内に囲みます。たとえば、要素 <Title><![CDATA[MyWindowName]]></Title> は「MyWindowName」というウィンドウ名になります。

共通の要素定義

一部の要素または定義は多くの要素で共有できます。

色要素

色要素は、RGB 値、名前、参照値、または整数値で指定できます。色要素は、R、G、B、Name、Ref、および Value です。これらの要素は、その他の要素（FillColor、TextColor、BGCOLOR など）で使用されます。

RGB 要素

RGB 値は色を指定するために使用します。RGB 要素に割り当てられる値は 0 から 255 の範囲です。要素がない場合は、デフォルト値が使用されます。デフォルト値はウィンドウ オブジェクトに応じて異なります。

例:

```
<FillColor>
    <R>192</R>
    <G>192</G>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <B>192</B>
</FillColor>
  <TextColor> <R>0</R><G>0</G><B>0</B> </TextColor>
```

色名要素

Internet Explorer バージョン 3.0 以降でサポートされる名前を使用して色を指定できます。色名では大文字と小文字が区別され、既知の HTML 色に一致する必要があります。

色名と値のリストについては、以下の Web サイトを参照してください。

http://www.w3schools.com/html/html_colornames.asp

http://www.learningwebdesign.com/colornames.html

http://www.oreilly.com/catalog/wdnut/excerpt/color_names.html

色名のリストについては、以下の Web サイトを参照してください。

http://www.geocities.com/SiliconValley/Pines/6986/colortbl.html

```
例:
<FillColor>
    <Name>白</Name>
</FillColor>
  <TextColor> <Name>白</Name> </TextColor>
```

色参照要素

16 進数の色値を使用して色を指定できます。色値には、常に 6 桁の 16 進数が必要です。

名前とそれに対応する 16 進数値の色のリストについては、以下の Web サイトを参照してください。

http://www.w3schools.com/html/html_colornames.asp

http://www.learningwebdesign.com/colornames.html

次の Web サイトでは、ディザリング処理なしの色値のリストが公開されています。

http://www.htmlgoodies.com/tutorials/colors/article.php/3479001

次のサイトでは 4096 色の 16 進数コードが紹介されています。

http://yorktown.cbe.wwu.edu/sandvig/MIS314/Assignments/A03/ColorHexCodes.asp

例:

```
<FillColor>
    <Ref>#FF00FF</Ref>
</FillColor>
  <TextColor> <Ref>#FF00FF</Ref> </TextColor>
```

色値要素

正の整数値の色値を使用して色を指定できます。値は 0 から 16777215 まで範囲である必要があります。

例:

```
<FillColor>
    <Value>16711935</Value>
</FillColor>
  <TextColor> <Value>16711935</Value> </TextColor>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

TextInfo 要素

TextInfo 要素を使用して画面にテキストを表示する方法を指定できます。

次の要素を使用してテキストを指定できます。

要素	説明
フォント	フォント ファミリーの名前（System や Arial など）。
フォント スタイル	値は、Regular、Italic、Bold、BoldItalic です。
ﾌｫﾝﾄ ｻｲｽﾞ	フォント サイズの測定単位を指定する必要があります。通常は、ポイントを使用します。値は 0 またはそれ以上です。
下線	下線付きテキスト：{true or false}。
取り消し線	取り消し線：{true or false}。
TextColor	テキスト色の色要素。
TextJustify	両端揃え：{Left, Center, or Right}

例:

```
<TextInfo>
    <Font>Arial</Font>
    <FontStyle>標準</FontStyle>
    <FontSize>12</FontSize>
    <Underline>なし</Underline>
    <Strikeout>なし</Strikeout>
    <TextColor>
      <R>0</R>
      <G>0</G>
      <B>0</B>
    </TextColor>
    <TextJustify>左揃え</TextJustify>
  </TextInfo>
```

Point 要素

Point 要素を使用して、その他の要素の位置を定義します。Point 要素には、2 つの要素（X および Y）含まれます。X と Y の要素は、-32000 から 32000 の範囲の値を含む必要があります。

要素	説明
XX	ピクセル単位での左座標。必須。
Y	ピクセル単位での上座標。必須

例:

```
<Point>
    <X>10</X>
    <Y>25</Y>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
  </Point>
```

ペン要素

Pen 要素は、オブジェクトの境界線の特徴を指定するために使用します。

次の要素を使用して Pen 要素を指定できます。

フィールド	説明
PenColor	使用色要素：RGB 要素、名前要素、参照要素、値要素。
PenWidth	ピクセル単位のペン幅。値は、1、2、4、6、9、または 11 です。
PenStyle	値は、None、Solid、Dash、Dot、DashDot、DashDotDot です。

例:

```
<Pen>
    <PenColor>
      <R>0</R>
      <G>0</G>
      <B>0</B>
    </PenColor>
    <PenWidth>4</PenWidth>
    <PenStyle>塗りつぶし</PenStyle>
  </Pen>
```

寸法要素

Dimension 要素は、画面上のオブジェクトの左上隅の座標を指定する要素を含みます。Dimension 要素には、長方形オブジェクトの幅と高さを指定する要素も含みます。

WindowMaker では、座標の制限は、垂直と水平の両方向で -32000 から +32000 です。(-32000, -32000, 32000, 32000) 境界の外にある計算済み座標を配置する幅または高さ値で X 位置と Y 位置の組み合わせを指定すると、警告が表示され、値は最大値に設定されます。高さと幅の値は正である必要があります。

Dimension 要素内で以下の要素を使用して画面の領域を指定します。

要素	説明
左	左端の座標。必須。
最上部	上端の座標。必須。
幅	ピクセル単位の幅。
高さ	ピクセル単位の高さ。

例:

```
<Dimension>
    <Left>4</Left>
    <Top>4</Top>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <Width>632</Width>
    <Height>278</Height>
  </Dimension>
```

式

式テキストは CDATA セクション内で囲みます。そうすることにより、式で有効な XML 区切り文字で解析が失敗することを回避します。

例:

```
<EXPRESSION>
<![CDATA[
式テキストの最初の行
式テキストの 2 番目の行
式テキストの N 番目の行
]]>
</EXPRESSION>
```

仮想キー コードと仮想キー フラグ

一部の InTouch アニメーション リンクはキーボード入力をサポートします。

XML パーサーは、仮想キーの名前を仮想キー コードに変換します。また、数値ビットの組み合わせでなく、テキストを使用して修飾キーのフラグが指定されます。キー名では大文字と小文字は区別されません。

InTouch アプリケーションは、以下の表に示す仮想キー名を使用します。

表されるキー	仮想キー名
追加	追加(&A)
アルファベット キー	A ～ Z
バックスペース	バックスペース
キャンセル	Ctrl+Break
クリア	クリア
コピー	コピー
10 進数	10 進数
削除	削除
除算	除算
下矢印	下
空の文字列は割り当てられていないことを示します。	<空白>
END	終了
ENTER	戻る
ESC	エスケープ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

表されるキー	仮想キー名
実行	実行
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9
F10	F10
F11	F11
F12	F12
F13	F13
F14	F14
F15	F15
F16	F16
ヘルプ	ヘルプ
HOME	ﾎｰﾑ
挿入	挿入
左矢印	左
乗算	乗算
数値キー	1~9
数値キーパッド 0	NUMPAD0
数値キーパッド 1	NUMPAD1
数値キーパッド 2	NUMPAD2
数値キーパッド 3	NUMPAD3
数値キーパッド 4	NUMPAD4

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

表されるキー	仮想キー名
数値キーパッド 5	NUMPAD5
数値キーパッド 6	NUMPAD6
数値キーパッド 7	NUMPAD7
数値キーパッド 8	NUMPAD8
数値キーパッド 9	NUMPAD9
数値ロック	NumLock
前ページ	PageUp
次ページ	PageDown
画面印刷	印刷
右矢印	右
選択	選択
区切り記号	区切り記号
Space キー	スペース
減算	減算
TAB	タブ
上矢印	上

2 つの修飾キーがサポートされています。修飾キーは CKEYFLAGS 要素に対して個別に指定するか、組み合わせで指定することができます。修飾された名前が要素値です。修飾キーは、修飾名が属性値文字列内にある場合に適用されます。

名前	表されるキー
CTRL	通常のキーと一緒に CONTROL キーを押す必要があります。
SHIFT	通常のキーと一緒に SHIFT キーを押す必要があります。
CTRL + SHIFT	CONTROL と SHIFT の両方のキーを押す必要があります。

CTRL+A シーケンスを生成する XML を以下に示します。

```
<VirtualKeyType>
   <KeyCode>A</KeyCode>
   <KeyFlags>CTRL</KeyFlags>
</VirtualKeyType>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウ要素

XML ファイルに含めることができるウィンドウ要素は 1 つだけです。ウィンドウ要素は XML ファイルで指定する最初の要素である必要があります。

ウィンドウ名が InTouch アプリケーションの既存のウィンドウ名に一致しないことを確認してください。

ウィンドウの定義

以下の要素を使用してウィンドウを指定できます。

要素	説明
タイトル	空でない文字列としてのウィンドウ名最大 32 文字で、末尾のスペースはカウントされません。必須。
コメント	コメント テキスト。最大 59 文字。
寸法	オプションのスキーマを使用する場合に必須。スキーマを使用しない場合、Dimension 要素がなければデフォルトの寸法が適用されます。デフォルト値は 4 4 632 278 です。
WindowStyle	ウィンドウのタイプ = Replace | Overlay| Popup。
BackgroundColor	色要素で指定されたウィンドウの背景色：RGB 要素、名前要素、参照要素、値要素。
FrameStyle	Single | Double | None。
TitleBar	タイトル バーの有効化 = true | false。
CloseButton	ウィンドウの閉じるボタンの有効化 = true | false。 タイトル バーが有効になっている場合にのみ有効にすることができます。
SizeControls	サイズ コントロールの有効化 = true | false。
ScriptOnShow	スクリプト要素。
ScriptWhileShowing	スクリプト要素。
ScriptOnHide	スクリプト要素。
ObjectList	ウィンドウに含まれるオブジェクト（SmartSymbol を含む）。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

次の例では、空のウィンドウが作成されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window
 wwInTouchWindow.xsd" Version="1">
  <Title>ApplicationSecondWindow</Title>
  <Comment>基本アプリケーション ウインドウ</Comment>
  <Dimension>
    <Left>4</Left>
    <Top>4</Top>
    <Width>632</Width>
    <Height>278</Height>
  </Dimension>
  <BackgroundColor>
    <R>192</R>
    <G>192</G>
    <B>192</B>
  </BackgroundColor>
  <WindowStyle>オーバーレイ</WindowStyle>
  <FrameStyle>シングル</FrameStyle>
  <TitleBar>表示</TitleBar>
<CloseButton>表示</CloseButton>
  <SizeControls>表示</SizeControls>
</iw:InTouchWindow>
```

最小の例:

```
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window
 wwInTouchWindow.xsd" Version="1">
  <Title>ApplicationSecondWindow</Title>
  <Dimension><Left>4</Left><Top>4</Top>
   <Width>632</Width>
    <Height>278</Height></Dimension>
</iw:InTouchWindow>
```

アクティブなスキーマの検証

スキーマの検証と処理は、InTouchWindow 要素内に特定のデータを配置することによってアクティブ化されます。スキーマをアクティブ化する場合、InTouchCommon.Xsd および InTouchWindow.Xsd ファイルを XML ファイルと同じフォルダに配置する必要があります。

スキーマの使用例:

```
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window wwInTouchWindow.xsd"
Version="1">
</iw:InTouchWindow>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スキーマを使用しない場合の例:

```
<InTouchWindow Version="1">
</InTouchWindow>
```

ウィンドウ スクリプト

XML インポートファイル内には 3 タイプのウィンドウ スクリプトを作成できます：OnShow、WhileShowing、OnHide。ウィンドウ要素内にスクリプト要素を配置します。

スクリプト テキストは CDATA セクションに囲んで、スクリプト内の XML 区切り文字が XML ファイル解析に干渉することを回避できます。

OnShow ウィンドウ スクリプト要素

OnShow スクリプト要素にはスクリプト テキストが含まれます。

例:

```
<ScriptOnShow><![CDATA[
    スクリプト テキストの最初の行
    スクリプト テキストの 2 番目の行
    スクリプト テキストの N 番目の行
  ]]></ScriptOnShow>
```

最小の例:

```
<ScriptOnShow>単一行のスクリプト テキスト</ScriptOnShow>
```

WhileShowing ウィンドウスクリプト要素

WhileShowing スクリプトには 2 つの要素があります。スクリプト テキストは CDATA セクション内に配置できます。

要素	説明
テキスト	ウィンドウ スクリプトのテキスト。必須。
頻度	ミリ秒単位でのスクリプト実行頻度。オプションのスキーマを使用する場合に必須。デフォルトは 1000 です。

例:

```
<ScriptWhileShowing>
    <Text><![CDATA[
        スクリプト テキストの最初の行
        スクリプト テキストの 2 番目の行
        スクリプト テキストの N 番目の行
      ]]></Text>
    <Frequency>1000</Frequency>
  </ScriptWhileShowing>
```

最小の例:

```
<ScriptWhileShowing>
    <Text>単一行のスクリプト テキスト</Text>
    <Frequency>1000</Frequency>
  </ScriptWhileShowing>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

OnHide ウィンドウスクリプト要素

OnHide ウィンドウ スクリプトには 1 つの要素があります。スクリプト テキストは CDATA セクション内に配置できます。

例:

```
<ScriptOnHide>
      <![CDATA[
        スクリプト テキストの最初の行
        スクリプト テキストの 2 番目の行
        スクリプト テキストの N 番目の行
      ]]>
  </ScriptOnHide>
```

最小の例:

```
<ScriptOnHide>スクリプト テキストの行</ScriptOnHide>
```

ウィンドウオブジェクト

ウィンドウ オブジェクトは、ObjectList 要素によって定義されたオブジェクト リスト内に配置されます。

例:

```
<?xml version="1.0" encoding="UTF-8"?>
<iw:InTouchWindow xmlns:iw="http://www.wonderware.com/InTouch/Window" xmlns:itc="http://
www.wonderware.com/InTouch/Common" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window.xsd" Version="1">
<Title>Window0001</Title>
<Comment>シンプルウィンドウを試す</Comment>
<Dimension>
   <Left>10</Left>
   <Top>10</Top>
   <Width>400</Width>
   <Height>400</Height>
</Dimension>
<BackgroundColor>
   <R>255</R>
   <G>0</G>
   <B>255</B>
</BackgroundColor>
<WindowStyle>置換</WindowStyle>
<FrameStyle>倍</FrameStyle>
<TitleBar>表示</TitleBar>
<SizeControls>表示</SizeControls>
<ObjectList>
   <Rectangle>
      <Title>Rectangle1</Title>
      <Pen>
         <PenColor>
            <Name>黒</Name>
         </PenColor>
         <PenWidth>4</PenWidth>
         <PenStyle>塗りつぶし</PenStyle>
      </Pen>
      <Dimension>
         <Left>100</Left> <Top>50</Top>
         <Width>270</Width> <Height>80</Height>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
      </Dimension>
      <FillColor>
         <R>128</R> <G>128</G> <B>128</B>
      </FillColor>
   </Rectangle>
</ObjectList>
</iw:InTouchWindow>
```

各ウィンドウ オブジェクトにはアニメーション リンクを含めることができます。アニメーション リンクを含むウィンドウ オブジェクトには、1 つのアニメーション リンク要素が含まれます。

```
<AnimationLinks> </AnimationLinks>
```

デフォルト要素値

ウィンドウ オブジェクト要素を指定しない場合、以下の表の要素がデフォルト値として使用されます。

要素	デフォルト値
FILLCOLOR	<R>212</R><G>208</G> <B>200</B>
PENWIDTH	1
PENCOLOR	黒
PENSTYLE	実線
CORNERDIMENSION	<Width>20</Width> <Height>20</Height>
TEXTCOLOR	黒
TEXTJUSTIFY	左
ROTATION	0
FONT	システム
FONTSTYLE	標準
FONTWEIGHT	無視されます
FONTSIZE	10
UNDERLINE	False
STRIKEOUT	False
FLIP	なし
TRANSPARENT	<R>0</R><G>255</G> <B>0</B>

ペンスタイルの制限

塗りつぶし以外のペンスタイルを指定する場合、ペン幅は 1 に強制されます。ペン幅を 1 より大きく指定するには、ペンスタイルのオプションを削除するか、塗りつぶしに設定する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ペンの寸法

寸法仕様はオブジェクトの中央線に適用されます。オブジェクトのペン幅が 1 よりも大きい場合、指定された寸法内にオブジェクトが収まらないことがあります。その場合、ペン幅がオブジェクトの境界をまたぎます。一部のピクセルがオブジェクトの境界内に配置され、その他のピクセルは外側に配置されます。

長方形オブジェクト

長方形の要素に対して以下の要素を指定できます。

要素	説明
塗りつぶしの色	長方形の塗りつぶしの色。色要素で指定します。デフォルトは rgb(212, 208, 200) です。
ペン	Pen 要素。
寸法	オブジェクトの位置とサイズ。要素は、top、left、width、height です。Top、left、width、および height の値はピクセルです。結果の座標は -32000 ～ +32000 の間である必要があります。width と height は両方とも 0 以外の値を指定する必要があります。必須。 Dimension 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンクリスト。

例:

```
<Rectangle>
    <Title>Rectangle1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>4</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
    <Dimension>
      <Left>100</Left> <Top>50</Top>
      <Width>270</Width> <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks> </AnimationLinks>
  </Rectangle>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

最小の例:

```
<Rectangle>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Rectangle>
```

角丸長方形オブジェクト

角丸長方形オブジェクトに対して以下の要素を指定できます。

要素	説明
塗りつぶしの色	塗りつぶしの色。色要素で指定します。デフォルトは rgb(212, 208, 200) です。
ペン	Pen 要素。
寸法	オブジェクトの位置とサイズ。要素は、top、left、width、height です。Top、left、width、および height の値はピクセルです。結果の座標は -32000 ～ +32000 の間である必要があります。width と height は両方とも 0 以外の値を指定する必要があります。必須。 Dimension 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。
CornerDimension	width および height 要素角の幅は長方形の幅以下である必要があります。角の高さは長方形の高さ以下である必要があります。角の幅と高さは 1 以上である必要があります。デフォルトは 20,20 です。
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンク リスト。

例:

```
<RoundedRectangle>
    <Title>RoundedRectangle1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>4</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <Dimension>
      <Left>100</Left>
      <Top>50</Top>
      <Width>270</Width>
      <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks>
    </AnimationLinks>
    <CornerDimension>
      <Width>8</Width>
      <Height>8</Height>
    </CornerDimension>
  </RoundedRectangle>
```

最小の例:

```
<RoundedRectangle>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </RoundedRectangle>
```

楕円形オブジェクト

楕円形オブジェクトに対して以下の要素を指定できます。

要素	説明
塗りつぶしの色	塗りつぶしの色。色要素で指定します。デフォルトは rgb(212, 208, 200) です。
ペン	Pen 要素。
寸法	オブジェクトの位置とサイズ。要素は、top、left、width、height です。Top、left、width、および height の値はピクセルです。結果の座標は -32000 ～ +32000 の間である必要があります。width と height は両方とも 0 以外の値を指定する必要があります。必須。 Dimension 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンクリスト。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例:

```
<Ellipse>
    <Title>Ellipse1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>4</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
    <Dimension>
      <Left>100</Left>
      <Top>50</Top>
      <Width>270</Width>
      <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks>
    </AnimationLinks>
  </Ellipse>
```

最小の例:

```
<Ellipse>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Ellipse>
```

線オブジェクト

線オブジェクトに対して以下の要素を指定できます。開始ポイントと終了ポイントに同じポイントを指定することはできません。

要素	説明
ペン	Pen 要素
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンク リスト。
ポイント	Point 要素。2 つ含む必要があります。余分な Point 要素は無視されます。

例:

```
<Line>
    <Title>Line1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>160</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
  </Line>
```

最小の例:

```
<Line>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>160</Y></Point>
    </Points> </Line>
```

水平線オブジェクト

水平線オブジェクトに対して以下の要素を指定できます。その場合、WindowMaker 水平/垂直線オブジェクトが作成されます。

要素	説明
ペン	Pen 要素。
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンク リスト。
ポイント	2 つのポイントを含む必要があります。2 番目のポイントの Y 座標は無視され、最初のポイントの Y 座標に設定されます。余分な Point 要素は無視されます。必須。 Points 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。

例:

```
<HorizontalLine>
    <Title>Line1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>150</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
  </HorizontalLine>
```

最小の例:

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
<HorizontalLine>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>150</Y></Point>
    </Points>
  </HorizontalLine>
```

垂直線オブジェクト

垂直線オブジェクトに対して以下の要素を指定できます。その場合、WindowMaker 水平/垂直線オブジェクトが作成されます。

要素	説明
ペン	Pen 要素。
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンク リスト。
ポイント	2 つのポイントを含む必要があります。2 番目のポイントの X 座標は無視され、最初のポイントの X 座標に設定されます。余分な Point 要素は無視されます。必須。 Points 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。

例:

```
<VerticalLine>
    <Title>Line1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>50</X><Y>250</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
<VerticalLine>
```

最小の例:

```
<VerticalLine>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>50</X><Y>250</Y></Point>
    </Points>
<VerticalLine>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

折れ線オブジェクト

折れ線オブジェクトをロードするには、少なくとも 2 つのポイントを定義する必要があります。2 つのポイントに同じ座標を使用することは推奨されません。

折れ線オブジェクトに対して以下の要素を指定できます。

要素	説明
ポイント	Point 要素。少なくとも 2 つ必要です。 Points 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。
ペン	Pen 要素。
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンク リスト。塗りつぶしの色、テキスト色、塗りつぶしパーセント、値の表示は使用できません。

例:

```
<Polyline>
    <Title>polyline1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>60</X><Y>250</Y></Point>
      <Point><X>70</X><Y>350</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
  </Polyline>
```

最小の例:

```
<Polyline>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
  </Polyline>
```

多角形オブジェクト

多角形をロードするには、少なくとも 2 つのポイントを定義する必要があります。2 つのポイントに同じ座標を使用することは推奨されません。

多角形オブジェクトに対して以下の要素を指定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
ポイント	Point 要素を含みます。少なくとも 2 つのポイントが必要です。X および Y の値はピクセル単位です。
塗りつぶしの色	多角形オブジェクトの塗りつぶしの色。1 つの色要素を含みます。デフォルトは rgb(212, 208, 200) です。
ペン	Pen 要素。
タイトル	オブジェクトの名前。オプション。
AnimationLinks	オプションのアニメーション リンク リスト。テキスト色、値の表示のアニメーション リンクは使用できません。

例:

```
<Polygon>
    <Title>polygon1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>60</X><Y>250</Y></Point>
      <Point><X>70</X><Y>350</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks>
    </AnimationLinks>
  </Polygon>
```

最小の例:

```
<Polygon>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
  </Polygon>
```

テキスト オブジェクト

テキスト オブジェクトに対して以下の要素を指定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
タイトル	オブジェクトの名前。オプション。
TextString	表示されるテキスト文字列。必須。TextString 要素に無効な値を割り当てた場合、または値がない場合、テキスト オブジェクトは作成されません。
TextInfo	テキストの表示方法を定義します。
回転	テキストの回転を度で定義します。使用できる値は、0、90、180、270 です。デフォルト値は 0 です。
AnimationLinks	オプションのアニメーション リンク リスト。
Dimension	オブジェクトの位置とサイズ。要素は、top、left、width、height です。Top、left、width、および height の値はピクセルです。結果の座標は -32000 ～ +32000 の間である必要があります。width と height は両方とも 0 以外の値を指定する必要があります。必須。Dimension 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。

例:

```
<Text>
    <Title>text1</Title>
    <TextString>これは表示するテキストです</TextString>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TextInfo>
      <Font>Arial</Font>
      <FontStyle>標準</FontStyle>
      <FontSize>12</FontSize>
      <Underline>なし</Underline>
      <Strikeout>なし</Strikeout>
      <TextColor>
        <R>0</R>
        <G>0</G>
        <B>0</B>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
      </TextColor>
      <TextJustify>左揃え</TextJustify>
    </TextInfo>
    <Rotation>0</Rotation>
    <AnimationLinks>
    </AnimationLinks>
  </Text>
```

最小の例:

```
<Text>
    <TextString>これは表示するテキストです</TextString>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Text>
```

ビットマップ オブジェクト

指定されたソース画像が存在する必要があります。サポートされるグラフィックファイル形式：JPG、JPEG、BMP、TIF、PCX、BIF、TGA。

指定されたソース画像のないビットマップは、デフォルトの長方形要素としてウィンドウに表示されます。

透明色

ビットマップに対する InTouch の内部アプリケーション オブジェクト データ構造には、透明色を適用することを指定するブール型フィールドが含まれています。実際の透明色は別のフィールドに格納されます。

WindowMaker で新しいビットマップを作成する場合、黒のデフォルト透明色が最初に設定されますが、これは使用できません。透明色ツールを選択して、ビットマップ オブジェクトに対して恒久的な透明色を適用します。透明色ツールからは、ビットマップに透明色が割り当てられているかどうかを判断することはできません。

XML ファイルからインポートされたビットマップ オブジェクトの場合、ビットマップの透明色ノードはオプションのエントリですが、存在する場合、透明色が有効化されてビットマップ オブジェクトに割り当てられます。エンド ユーザーが WindowMaker でウィンドウを開いてビットマップ オブジェクトを選択し、透明色を無効化することはできません。

ビットマップ オブジェクト要素

ビットマップ オブジェクトに対して以下の要素を指定できます。

要素	説明
タイトル	ビットマップ オブジェクトの名前。オプション。
塗りつぶしの色	ビットマップ オブジェクトの内部 RGB 色。1 つの色要素を含みます。既定値は rgb(0, 0, 0) です。
SourceImage	ビットマップ ファイルのパス。デフォルトは Null です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
透明	透明色の RGB 色。デフォルトは rgb(0,255,0) です。
ペン	ビットマップ オブジェクトの周囲の線を定義します。
AnimationLinks	オプションのアニメーション リンク リスト。
Dimension	オブジェクトの位置とサイズ。要素は、top、left、width、height です。Top、left、width、および height の値はピクセルです。結果の座標は -32000 ～ +32000 の間である必要があります。width と height は両方とも 0 以外の値を指定する必要があります。必須。 Dimension 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。

例:

```
<Bitmap>
    <Title>bitmap1</Title>
    <Pen>
      <PenColor><Name>黒</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>塗りつぶし</PenStyle>
    </Pen>
    <Dimension>
      <Left>100</Left>
       <Top>50</Top>
      <Width>270</Width>
       <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R>
       <G>128</G> <
       B>128</B>
    </FillColor>
    <Transparent>
      <R>0</R> <G>128</G> <B>255</B>
    </Transparent>
    <SourceImage>C:\MyPictures\hello.jpg</SourceImage>
    <AnimationLinks>
    </AnimationLinks>
  </Bitmap>
```

例:

```
<Bitmap>
    <Dimension>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <SourceImage>C:\MyPictures\hello.jpg</SourceImage>
  </Bitmap>
```

空の SourceImage ノードの例:

```
<Bitmap>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <SourceImage></SourceImage>
  </Bitmap>
```

最小の例:

```
<Bitmap>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Bitmap>
```

ボタン オブジェクト

ボタン オブジェクトに対して以下の要素を指定できます。

要素	説明
タイトル	ボタン オブジェクトの名前。オプション。
TextString	表示されるキャプション。デフォルト文字列は "Text" です。
TextInfo	キャプションの表示方法を定義します。
AnimationLinks	オプションのアニメーション リンク リスト。線の色、塗りつぶしの色、テキスト色、塗りつぶしパーセント、向きのアニメーション リンクは使用できません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
Dimension	オブジェクトの位置とサイズ。要素は、top、left、width、height です。Top、left、width、および height の値はピクセルです。結果の座標は -32000 〜 +32000 の間である必要があります。width と height は両方とも 0 以外の値を指定する必要があります。必須。 Dimension 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。

例:

```
Button>
    <Title>button1</Title>
    <TextInfo>
      <Font>Arial</Font>
      <FontStyle>標準</FontStyle>
      <FontSize>12</FontSize>
      <Underline>なし</Underline>
      <Strikeout>なし</Strikeout>
      <TextColor>
        <R>0</R>
        <G>0</G>
        <B>0</B>
      </TextColor>
      <TextJustify>左揃え</TextJustify>
    </TextInfo>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TextString>すべてのロボットを停止</TextString>
    <AnimationLinks>
    </AnimationLinks>
  </Button>
```

最小の例:

```
<Button>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Button>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SmartSymbol

SmartSymbol テンプレートを使用するウィンドウをインポートする前に、アプリケーション内で SmartSymbol テンプレートを定義する必要があります。SmartSymbol テンプレートが存在しない場合、インポートは失敗し、ウィンドウは作成されません。

1 つのウィンドウに対して複数の SmartSymbol を指定できます。各 SmartSymbol は個別の <SmartSymbol> </SmartSymbol> 要素内で宣言されます。

SmartSymbol に対して以下の要素を指定できます。

要素	説明
ｼﾝﾎﾞﾙ名	SmartSymbol の名前。必須。 名前がアプリケーション内の SmartSymbol テンプレート名に一致しない場合、または名前がない場合、オブジェクトは作成されません。
Dimension	オブジェクトの位置とサイズ。要素は、top、left、width、height です。Top、left、width、および height の値はピクセルです。結果の座標は -32000 ～ +32000 の間である必要があります。width と height は両方とも 0 以外の値を指定する必要があります。必須。 Dimension 要素に無効な値を割り当てた場合、または値がない場合、オブジェクトは作成されません。
タグ置き換え	インスタンス タグの置換。
StringReplace	インスタンス文字列の置換。

例:

```
<SmartSymbol>
    <SymbolName>MyCoolSymbol</SymbolName>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TagReplace>
      <Find>Tag001</Find>
      <Replace>Tag007</Replace>
    </TagReplace>
    <StringReplace>
      <Find>このテキストを検索</Find>
      <Replace>このテキストに置換</Replace>
    </StringReplace>
  </SmartSymbol>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

最小の例:

```
<SmartSymbol>
    <SymbolName>MyCoolSymbol</SymbolName>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </SmartSymbol>
```

タグの置換

SmartSymbol インスタンス内のタグを置換できます。アプリケーション内に現在存在する SmartSymbol インスタンス内のタグのみを置換してください。

タグの置換では大文字と小文字が区別されず、タグ名文字列全体が一致する必要があります。1 つまたは複数の <TagReplace> 要素を SmartSymbol ノートに追加して複数のタグを置換できます。

以下に例を示します。

```
<TagReplace>
  <Find>タグ 1</Find>
  <Replace>D タグ A</Replace>
</TagReplace>
<TagReplace>
  <Find><![CDATA[タグ 1]]></Find>
  <Replace><![CDATA[D タグ A]]></Replace>
</TagReplace>
```

置換に指定されたタグ名がアプリケーション内に存在しない場合、SmartSymbol は作成されません。また、SmartSymbol を含むウィンドウは作成されません。

置換後のタグのタイプは、元のタグのタイプと同じである必要があります。

文字列の置換

SmartSymbol インスタンス内の文字列を置換できます。文字列の置換では大文字と小文字が区別され、ソース文字列全体が一致する必要があります。単一の SmartSymbol 要素内で複数の文字列置換要素を使用できます。

```
<StringReplace>
    <FIND>
      <![CDATA[開く]]>
    </FIND>
    <REPLACE>
      <![CDATA[ｵﾌ]]>
    </REPLACE>
  </StringReplace>
  <StringReplace>
    <FIND>開く</FIND>
    <REPLACE>オフ</REPLACE>
  </StringReplace>
```

SmartSymbol の例

タグおよび文字列の置換要素を含む SmartSymbol 要素の例を以下に示します。

```
<SmartSymbol>
    <SymbolName>MyCoolSymbol</SymbolName>
    <Dimension>
      <Left>100</Left><Top>50</Top>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TagReplace>
      <Find>D タグ 1</Find>
      <Replace>D タグ A</Replace>
    </TagReplace>
    <TagReplace>
      <Find>ATag001</Find>
      <Replace>ATag002</Replace>
    </TagReplace>
    <StringReplace>
      <Find><![CDATA[開く]]> </Find>
      <Replace><![CDATA[ｵﾝ]]> </Replace>
    </StringReplace>
    <StringReplace>
      <FIND>
        <![CDATA[閉]]>
      </FIND>
      <REPLACE>
        <![CDATA[ｵﾌ]]>
      </REPLACE>
    </StringReplace>
 </SmartSymbol>
```

産業用グラフィック

産業用グラフィック参照を使用するウィンドウをインポートする前に、Galaxy 内で産業用グラフィックを定義する必要があります。産業用グラフィック参照が存在しない場合、インポートは失敗し、ウィンドウは作成されません。

シンボル参照内のインスタンスまたはグラフィック ツールボックスからシンボルを指定できます。しかし、シンボル参照内のテンプレート シンボルを指定することはできません。

1 つのウィンドウに対して複数の産業用グラフィックを指定できます。各産業用グラフィックは個別の <Industrial Graphic> </Industrial Graphic> 要素内で宣言されます。

産業用グラフィックに対して以下の要素を指定できます。

要素	説明
SymbolReference	Galaxy 内のシンボルを検索する参照。必須。 シンボル参照によって指定された産業用グラフィックが存在しない場合、またはシンボル参照フィールドが欠落している場合や空の場合、オブジェクトは作成されません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
Dimension	オブジェクトの位置とサイズ。Dimension サブ要素は、top、left、width、および height です。必須。 width と height の指定はオプションです。width と height を指定しない場合、産業用グラフィックの元の幅と高さが使用されます。width だけを指定した場合、縦横比を使用して height が計算されます。height だけを指定した場合も同様に width が計算されます。たとえば、元の幅が 200 で、高さが 100 の場合、width に 100 を指定すると、height は 50 になります。 Dimension 要素に無効な値を割り当てた場合、または値が含まれない場合、オブジェクトは作成されません。このシンボルを含むウィンドウも作成されません。
Title	要素のテキスト名。オプション。 この名前は XML ファイル内で一意である必要があります。そうでない場合は無視されます。
Flip	要素の反転タイプ。オプション。
Rotation	要素の回転角度。オプション。
StringReplace	1 つまたは複数の文字列置換ノード。オプション。
CustomPropertyOverride	1 つまたは複数のカスタム プロパティの上書き。オプション。
AnimationLinks	アニメーション リンクのリスト。オプション。 線の色、塗りつぶしの色、テキスト色、塗りつぶしパーセント、向き、スライダー、ツールヒン

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
	ト、値の表示、点滅、およびユーザー入力は使用できません。

```
<ArchestrASymbol>
<Title>EmbedSym1</Title>
<Dimension>
      <Left>200</Left>
      <Top>200</Top>
      <Width>150</Width>
      <Height>150</Height>
   </Dimension>
   <Flip>None</Flip>
   <Rotation>0</Rotation>

   <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>
   <AnimationLinks>
   </AnimationLinks>
   <StringReplace>
      <Find>LABEL</Find>
      <Replace>OFF</Replace>
   </StringReplace>
   </ArchestrASymbol>
```

最小の例:

```
<ArchestrASymbol>
<Dimension>
      <Left>200</Left>
      <Top>200</Top>
   </Dimension>
   <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>
</ArchestrASymbol>
```

CustomPropertyOverride

上書きできるのは、産業用グラフィックに対して既に定義されているカスタム プロパティだけです。

複数のカスタム プロパティを上書きするには、1 つまたは複数の <CustomPropertyOverride> 要素を産業用グラフィック ノードに追加します。

例:

```
<ArchestrASymbol>
   <Title>EmbedSym1</Title>
   <Dimension>
   <Left>200</Left>
   <Top>200</Top>
   <Width>150</Width>
   <Height>150</Height>
```

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

```
   </Dimension>
   <Flip>None</Flip>
   <Rotation>0</Rotation>
   <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>
   <AnimationLinks>
   </AnimationLinks>
   <CustomPropertyOverride> <CustomPropertyName>cp1</CustomPropertyName>
      <OverrideValue>DTagA</OverrideValue>
      <IsConstant>false</IsConstant>
   </CustomPropertyOverride>
</ArchestrASymbol>
```

この例では、cp1 は既存のカスタム プロパティの名前です。上書きは、新しい値が DTagA に設定されたカスタム プロパティに適用されます。IsConstant はオプションのフィールドで、値を定数として解釈するかどうかを指定します。IsConstant フラグは、カスタム プロパティのタイプが文字列、時刻、または経過時間の場合にのみ適用できます。デフォルトでは、IsConstant フラグは false に設定されます。

注: OverrideValue に対して指定されたタグ名がタグ データベースに存在しない場合、産業用グラフィック参照はウィンドウに作成されず、その特定のウィンドウのインポートは失敗します。エラー メッセージがログ ファイルまたはロガーに記録されます。

産業用グラフィック文字列置換タイプ

産業用グラフィック インスタンス内の既存の文字列を置換できます。文字列の置換では大文字と小文字が区別されます。単一の産業用グラフィック ノード内で複数の文字列置換ノードを使用できます。

```
  <StringReplace>
    <FIND>
      <![CDATA[開く]]>
    </FIND>
    <REPLACE>
      <![CDATA[ｵﾌ]]>
    </REPLACE>
  </StringReplace>
<StringReplace>
    <FIND>開く</FIND>
    <REPLACE>オフ</REPLACE>
  </StringReplace>
```

サポートされないウィンドウ オブジェクト

セル、リアルタイム トレンド、および履歴トレンド オブジェクトはインポートできません。サポートされないオブジェクトの要素が XML ファイルに含まれる場合、その要素は無視されます。

XML ユーティリティを使用したウィンドウのインポート

XML ユーティリティを使用してウィンドウをインポートする際に、閉じるボタンのスクリプトを XML スクリプトに追加することができます。

注意：インポートするウィンドウは、バッチ ファイル（wm.bat）で指定します。

```
<CloseButton>表示</CloseButton>
```

XML ファイルにスクリプトが含まれていると、インポートするウィンドウのタイトル バーにウィンドウを閉じるボタンが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウ アニメーション リンク

ウィンドウ オブジェクト アニメーション リンクは <AnimationLinks> </AnimationLinks> 要素内で宣言されます。

1 つのウィンドウ要素内に複数のアニメーション リンクを設定できますが、必ずしも設定する必要はありません。

すべての要素ですべてのアニメーション リンクがサポートされるわけではありません。産業用グラフィックは、以下のアニメーション リンクをサポートします。

- ObjSize_Height
- ObjSize_Width
- Location_Vert
- Location_Hori
- TPushB_Disc
- TPushB_Action
- TPushB_ShowWin
- TPushB_HideWin
- Misc_Visib
- Misc_Disable

ウィンドウ アニメーション リンクの詳細については、InTouch HMI のマニュアルを参照してください。

アニメーション リンクの一部のタイプでは、その他のタイプのアニメーション リンクを作成できません。アニメーション リンクの処理は、XML ファイルで指定された順序で行われます。

スクリプト/式/タグ名要件のマトリックス

各アニメーション リンクには 1 つのコントロール フィールドがあります。コントロール フィールドは、スクリプト、式、またはタグ名です。一部のコントロール フィールドは、使用できるタグまたは式のタイプによって制限されています。以下の表には、各アニメーション リンクに必要なコントロール フィールドとコントロール フィールドの制限が一覧表示されています。

アニメーション リンク タイプ	コントロール フィールド	コントロール フィールドの制限
論理型ユーザー入力	タグ	論理型タグ
アナログ ユーザー入力	タグ	アナログ型タグ
文字列ユーザー入力	タグ	文字列タグ
論理線の色	式	論理型式
アナログ線の色	式	アナログ式

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アニメーション リンク タイプ	コントロール フィールド	コントロール フィールドの制限
論理型アラーム線の色	タグ	論理型タグ アラーム ステータス
アナログ アラーム線の色	タグ	アナログ型タグ アラーム ステータス
論理塗りつぶしの色	式	論理型式
アナログ塗りつぶしの色	式	アナログ式
論理型アラーム塗りつぶしの色	タグ	論理型タグ アラーム ステータス
アナログ アラーム塗りつぶしの色	タグ	アナログ型タグ アラーム ステータス
論理テキスト色	式	論理型式
アナログ テキスト色	式	アナログ式
論理型アラーム テキスト色	タグ	論理型タグ アラーム ステータス
アナログ アラーム テキスト色	タグ	アナログ型タグ アラーム ステータス
垂直スライダー	タグ	有効なアナログ型タグ
水平スライダー	タグ	有効なアナログ型タグ
オブジェクト サイズ高さ	式	アナログ値
オブジェクト サイズ幅	式	アナログ値
垂直位置	式	有効な式
水平位置	式	有効な式
垂直塗りつぶしパーセント	式	アナログ値
水平塗りつぶしパーセント	式	アナログ値

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アニメーション リンク タイプ	コントロール フィールド	コントロール フィールドの制限
論理値タッチ押しボタン	タグ	論理値
アクション タッチ押しボタン	スクリプト	有効なスクリプト
ウィンドウの表示タッチ押しボタン	ウィンドウ名	ウィンドウが存在する必要があります。
ウィンドウの非表示タッチ押しボタン	ウィンドウ名	ウィンドウが存在する必要があります。
表示オン/オフ	式	論理値
点滅	式	論理値
向き	式	アナログ値
無効化	式	論理値
ツールヒント	式	文字列タグ
論理値表示	式	論理型式
アナログ値表示	式	アナログ式
文字列値表示	式	文字列式

論理型ユーザー入力

論理型ユーザー入力アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
メッセージ型	ユーザーに表示するメッセージ テキスト。デフォルトではメッセージ テキストはありません。オプションのスキーマを使用する場合に必要です。
InputOnly	入力のみかどうか : {true, false}. デフォルトは false です。
OnMessage	ユーザーに表示するオン メッセージ テキスト。デフォルトは On です。空のテキストを指定することはできません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
OffMessage	ユーザーに表示するオフ メッセージ テキスト。デフォルトは Off です。空のテキストを指定することはできません。
SetPrompt	ユーザーに表示するセット プロンプト メッセージ テキスト。デフォルトは On です。空のテキストを指定することはできません。
ResetPrompt	ユーザーに表示するリセット プロンプト メッセージ テキスト。デフォルトは Off です。空のテキストを指定することはできません。
KeyAssignment	仮想キー要素。デフォルトでは割り当ては設定されません。空の文字列は割り当てが発生しないことを示します。
式	論理型タグ。必須。無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<UserInputDiscrete>
    <Title>UserInputDiscrete1</Title>
    <InputOnly>いいえ</InputOnly>
    <KeyAssignment>
      <KeyCode>F1</KeyCode>
      <KeyFlags>Ctrl</KeyFlags>
    </KeyAssignment>
    <Message>ポンプ弁の状態</Message>
    <Expression><![CDATA[dTag001]]></Expression>
    <OnMessage>オンメッセージ テキスト</OnMessage>
    <OffMessage>オフメッセージ テキスト</OffMessage>
    <ResetPrompt>
      <![CDATA[プロンプトテキストをリセット]]>
    </ResetPrompt>
    <SetPrompt>プロンプト設定テキスト</SetPrompt>
  </UserInputDiscrete>
```

最小の例:

```
<UserInputDiscrete>
    <Message>ポンプ弁の状態</Message>
    <Expression>dTag001</Expression>
  </UserInputDiscrete>
```

アナログ ユーザー入力

アナログ ユーザー入力アニメーション リンクに対して以下の要素を指定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
タイトル	オブジェクトの名前。オプション。
メッセージ型	ユーザーに表示するメッセージテキスト。オプションのスキーマを使用する場合に必要です。デフォルトではメッセージテキストはありません。
InputOnly	入力のみかどうかを指定します（true、false）。デフォルトは false です。
MinAnalogValue	使用できる最小の浮動小数点値。オプションのスキーマを使用する場合に必要です。デフォルトは 0.0 です
MaxAnalogValue	使用できる最大の浮動小数点値。オプションのスキーマを使用する場合に必要です。デフォルト値は 100 です。MINANALOGVALUE 要素に割り当てられた値よりも大きい値である必要があります。
KeyPadEnabled	キーパッドを表示するかどうかを指定します。使用できる値は True または False です。デフォルトは False です。
KeyAssignment	仮想キー要素、空の文字列、または存在しない。デフォルトでは割り当ては設定されません。空の文字列は割り当てが発生しないことを示します。
式	アナログ型タグ名。必須。無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<<UserInputAnalog>
    <Title>UserInputAnalog1</Title>
    <InputOnly>いいえ</InputOnly>
    <KeyAssignment>
      <KeyCode>F1</KeyCode>
      <KeyFlags>Ctrl</KeyFlags>
    </KeyAssignment>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <Message>フラッシュ ポンプ速度</Message>
    <Expression>aTag001</Expression>
    <KeyPadEnabled>無効</KeyPadEnabled>
    <MinAnalogValue>0.0</MinAnalogValue>
    <MaxAnalogValue> 100.0</MaxAnalogValue>
  </UserInputAnalog>
```

最小の例:

```
<UserInputAnalog>
    <Message>ポンプ弁の状態</Message>
    <Expression><![CDATA[aTag001]]></Expression>
    <MinAnalogValue>0.0</MinAnalogValue>
    <MaxAnalogValue>100.0</MaxAnalogValue>
  </UserInputAnalog>
```

文字列ユーザー入力

EchoEnabled または EchoMode のいずれかを使用します。両方使用することはできません。EchoMode ではパスワード モードを指定できます。EchoEnabled では、有効または無効の状態のみを指定できます。

ランタイム時にユーザー入力の表示方法を制御するには、エコー文字要素を使用します。使用できる値は、Yes、No、および Password です。

- 要素を Yes に設定すると、文字列がランタイム時に入力編集ボックスに表示されます。入力が有効になります。パスワード文字と暗号化は無効になります。
- 要素を No に設定すると、入力文字はランタイム時に入力編集ボックスに表示されません。入力が有効になります。パスワード文字と暗号化は無効になります。
- 要素を Password に設定すると、ユーザーが入力されたパスワードではなく、パスワード文字がランタイム時に表示されます。パスワード文字はオプションですが、指定した場合は空にすることはできません。デフォルトのパスワード文字はアスタリスクです。暗号化はオプションで、デフォルトで無効化されています。「入力のみ」は必須で、有効化されています。

要素が上記の条件を満たさない場合、デフォルトのオプションが使用されます。

文字列ユーザー入力アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
メッセージ型	ユーザーに表示するメッセージ テキスト。オプションのスキーマを使用する場合に必要です。デフォルトではメッセージ テキストはありません。
InputOnly	入力のみかどうか : {true, false}. デフォルトは false です。
EchoCharacters	エコー文字（no、yes、password）。オプションのスキーマを使用する場合に必要です。デフォルトは yes です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
KeyPadEnabled	キーパッドを表示するかどうかを指定します（true、false）。デフォルトは false です。
PasswordCharacter	パスワード文字。デフォルトは "*" です。
EncryptEnabled	暗号化の有効化（yes、no）。デフォルトは no です。
KeyAssignment	仮想キー要素、空の文字列、または存在しない。デフォルトでは割り当ては設定されません。空の文字列は割り当てが発生しないことを示します。
式	メッセージ型タグ。必須。無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<UserInputString>
    <Title>UserInputString1</Title>
    <InputOnly>いいえ</InputOnly>
    <KeyAssignment>
      <KeyCode>F1</KeyCode>
      <KeyFlags>Ctrl</KeyFlags>
    </KeyAssignment>
    <Message>ポンプを選択</Message>
    <Expression>mTag001</Expression>
    <EchoCharacters>有効</EchoCharacters>
    <EncryptEnabled>無効</EncryptEnabled>
    <PasswordCharacter>*</PasswordCharacter>
  </UserInputString>
```

最小の例:

```
<UserInputString>
    <Message>ポンプを選択</Message>
    <Expression>mTag001</Expression>
    <EchoCharacters>有効</EchoCharacters>
  </UserInputString>
```

論理線の色

論理線の色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
Oncolor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
Offcolor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	論理型タグまたは式。必須。Expression 要素値がない場合、または無効な場合、オブジェクトは作成されません。

例:

```
<LineColorDiscrete>
    <Title>LineColorDiscrete1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <OnColor><Name>緑</Name></OnColor>
    <OffColor><Name>赤</Name></OffColor>
  </LineColorDiscrete>
```

最小の例:

```
<LineColorDiscrete>
    <Expression>dTag001</Expression>
  </LineColorDiscrete>
```

アナログ線の色

必須のブレークポイント値には増分値が必要です。そうでない場合、警告がログに記録されます。無効な値は、以前の値に 1 を追加したもんです。

必須の値がない場合または無効な場合、アニメーション リンクを含むオブジェクトは作成されません。

アナログ線の色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
Color1<1/>Color1	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color2<1/>Color2	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color3<1/>Color3	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color4	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color5	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color6	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
Color7	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color8	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color9	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color10	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値	定数アナログ値。9 つの Value 要素を含む必要があります。必須。
式	アナログ型タグまたは式。必須。

例:

```
<LineColorAnalog>
    <Title>LineColorAnalog1</Title>
    <Expression>aTag001</Expression>
    <Colors>
      <Color1><Name>白</Name></Color1>
      <Color2><Name>赤</Name></Color2>
      <Color3><Name>オレンジ</Name></Color3>
      <Color4><Name>黄</Name></Color4>
      <Color5><Name>緑</Name></Color5>
      <Color6><Name>青</Name></Color6>
      <Color7><Name>シアン</Name></Color7>
      <Color8><Name>マゼンタ</Name></Color8>
      <Color9><Name>バイオレット</Name></Color9>
      <Color10><Name>黒</Name></Color10>
    </Colors>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </LineColorAnalog>
```

最小の例:

```
<LineColorAnalog>
    <Expression>aTag001</Expression>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
```

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

```
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </LineColorAnalog>
```

論理型アラーム線の色

論理型アラーム線の色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
AlarmColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	論理型タグ。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<LineColorDiscreteAlarm>
    <Title>LineColorDiscreteAlarm1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <NormalColor><Name>黒</Name></NormalColor>
    <AlarmColor><Name>赤</Name></AlarmColor>
  </LineColorDiscreteAlarm>
```

最小の例:

```
<LineColorDiscreteAlarm>
    <Expression>dTag001</Expression>
  </LineColorDiscreteAlarm>
```

アナログ アラーム線の色

アナログ アラーム線の色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
値：LOLOCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値：LOCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
値：NORMALCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値：HICOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値：HIHICOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差：NORMALCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差：MINORCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差：MAJORCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
ROC：NORMALCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
ROC：ROCCOLOR	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	アナログ型タグ。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

値アラームの例:

```
<LineColorAnalogAlarm>
    <Title>LineColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Value>
      <LoLoColor><Name>赤</Name><LoLoColor>
      <LoColor><Name>暗い赤</Name><LoColor>
      <NormalColor><Name>黒</Name></NormalColor>
      <HiColor><Name>暗い緑</Name></HiColor>
      <HiHiColor><Name>緑</Name></HiHiColor>
    </Value>
  </LineColorAnalogAlarm>
```

偏差アラームの例:

```
<LineColorAnalogAlarm>
    <Title>LineColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
   <Deviation>
      <NormalColor><Name>黒</Name></NormalColor>
      <MinorColor><Name>緑</Name></MinorColor>
```

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

```
      <MajorColor><Name>赤</Name></MajorColor>
    </Deviation>
  </LineColorAnalogAlarm>
```

ROC アラームの例:

```
<LineColorAnalogAlarm>
    <Title>LineColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <ROC>
      <NormalColor><Name>黒</Name></NormalColor>
      <ROCColor><Name>赤</Name></ROCColor>
    </ROC>
  </LineColorAnalogAlarm>
```

論理塗りつぶしの色

論理塗りつぶしの色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
OnColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
OffColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	論理型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<FillColorDiscrete>
    <Title>FillColorDiscrete1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <OnColor><Name>緑</Name></OnColor>
    <OffColor><Name>赤</Name></OffColor>
  </FillColorDiscrete>
```

最小の例:

```
<FillColorDiscrete>
    <Expression>dTag001</Expression>
  </FillColorDiscrete>
```

アナログ塗りつぶしの色

必須の値がない場合または無効な場合、アニメーション リンクを含むオブジェクトは作成されません。

論理型塗りつぶしの色アニメーション リンクに対して以下の要素を指定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
タイトル	オブジェクトの名前。オプション。
Color1<1/>Color1	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color2<1/>Color2	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color3<1/>Color3	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color4	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color5	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color6	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color7	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color8	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color9	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color10	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値	定数アナログ値。9 つの Value 要素を含む必要があります。必須。
式	アナログ型タグまたは式。必須。

例:

```
<FillColorAnalog>
    <Title>FillColorAnalog1</Title>
    <Expression>aTag001</Expression>
    <Colors>
      <Color1><Name>白</Name></Color1>
      <Color2><Name>赤</Name></Color2>
      <Color3><Name>オレンジ</Name></Color3>
      <Color4><Name>黄</Name></Color4>
      <Color5><Name>緑</Name></Color5>
      <Color6><Name>青</Name></Color6>
      <Color7><Name>シアン</Name></Color7>
      <Color8><Name>マゼンタ</Name></Color8>
      <Color9><Name>バイオレット</Name></Color9>
      <Color10><Name>黒</Name></Color10>
    </Colors>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </FillColorAnalog>
```

最小の例:

```
<FillColorAnalog>
    <Expression>aTag001</Expression>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </FillColorAnalog>
```

論理型アラーム塗りつぶしの色

論理型アラーム塗りつぶしの色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
AlarmColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	論理型タグ。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<FillColorDiscreteAlarm>
    <Title>FillColorDiscreteAlarm1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <NormalColor><Name>黒</Name></NormalColor>
    <AlarmColor><Name>赤</Name></AlarmColor>
```

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

```
  </FillColorDiscreteAlarm>
```

最小の例:

```
<FillColorDiscreteAlarm>
    <Expression>dTag001</Expression>
  </FillColorDiscreteAlarm>
```

アナログ アラーム塗りつぶしの色

アナログ アラーム塗りつぶしの色アニメーション リンクに対して以下の要素を指定できます。必須の値がない場合または無効な場合、アニメーション リンクを含むオブジェクトは作成されません。

要素	説明
タイトル	オブジェクトの名前。オプション。
値：LoLoColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値：LoColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値：NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値：HiColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値：HiHiColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差：NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差：MinorColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差：MajorColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
ROC：NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
ROC：ROCColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	アナログ型タグ。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

値アラームの例:

```
<FillColorAnalogAlarm>
    <Title>FillColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <Value>
      <LoLoColor><Name>赤</Name><LoLoColor>
      <LoColor><Name>暗い赤</Name><LoColor>
      <NormalColor><Name>黒</Name></NormalColor>
      <HiColor><Name>暗い緑</Name></HiColor>
      <HiHiColor><Name>緑</Name></HiHiColor>
    </Value>
  </FillColorAnalogAlarm>
```

偏差アラームの例:

```
<FillColorAnalogAlarm>
    <Title>FillColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Deviation>
      <NormalColor><Name>黒</Name></NormalColor>
      <MinorColor><Name>緑</Name></MinorColor>
      <MajorColor><Name>赤</Name></MajorColor>
    </Deviation>
  </FillColorAnalogAlarm>
```

ROC アラームの例:

```
<FillColorAnalogAlarm>
    <Title>FillColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <ROC>
      <NormalColor><Name>黒</Name></NormalColor>
      <ROCColor><Name>赤</Name></ROCColor>
    </ROC>
  </FillColorAnalogAlarm>
```

論理テキスト色

論理テキスト色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
OnColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
OffColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	文字列タグまたは式。必須。

例:

```
<TextColorDiscrete>
    <Title>TextColorDiscrete1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <OnColor><Name>緑</Name></OnColor>
    <OffColor><Name>赤</Name></OffColor>
  <TextColorDiscrete>
```

最小の例:

```
<TextColorDiscrete>
    <Expression>dTag001</Expression>
  <TextColorDiscrete>
```

アナログ テキスト色

アナログ テキスト色アニメーション リンクに対して以下の要素を指定できます。必須の値がない場合または無効な場合、アニメーション リンクを含むオブジェクトは作成されません。

要素	説明
タイトル	オブジェクトの名前。オプション。
Color1<1/>Color1	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color2<1/>Color2	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Color3<1/>Color3	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
色 4	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
色 5	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
色 6	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
色 7	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
色 8	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
色 9	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
色 10	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値	定数アナログ値。9 つの Value 要素を含む必要があります。必須。
式	アナログ型タグまたは式。必須。

例:

```
<TextColorAnalog>
    <Title>TextColorAnalog1</Title>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
    <Expression>dTag001</Expression>
    <Colors>
      <Color1><Name>白</Name></Color1>
      <Color2><Name>赤</Name></Color2>
      <Color3><Name>オレンジ</Name></Color3>
      <Color4><Name>黄</Name></Color4>
      <Color5><Name>緑</Name></Color5>
      <Color6><Name>青</Name></Color6>
      <Color7><Name>シアン</Name></Color7>
      <Color8><Name>マゼンタ</Name></Color8>
      <Color9><Name>バイオレット</Name></Color9>
      <Color10><Name>黒</Name></Color10>
    </Colors>
    <Values>
      <Value>10.0</Value> <Value>20.0</Value>
      <Value>30.0</Value> <Value>40.0</Value>
      <Value>50.0</Value> <Value>60.0</Value>
      <Value>70.0</Value> <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </TextColorAnalog>
```

最小の例:

```
<TextColorAnalog>
    <Expression>dTag001</Expression>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </TextColorAnalog>
```

論理型アラーム テキスト色

論理型アラーム テキスト色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
通常色	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
警告色	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	論理型タグ。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例:

```
<TextColorDiscreteAlarm>
    <Title>TextColorDiscreteAlarm1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <NormalColor><Name>黒</Name></NormalColor>
    <AlarmColor><Name>赤</Name></AlarmColor>
  </TextColorDiscreteAlarm>
```

最小の例:

```
</TextColorDiscreteAlarm>
    <Expression>dTag001</Expression>
  </TextColorDiscreteAlarm>
```

アナログ アラーム テキスト色

アナログ アラーム テキスト色アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
値: LoLoColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値: LoColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値: NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値:HiColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
値: HiHiColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
変化幅: NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差: MinorColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
偏差: MajorColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
ROC: NormalColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
ROC:ROCColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
式	アナログ型タグ。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

例:

```
<TextColorAnalogAlarm>
    <Title>TextColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Value>
      <LoLoColor><Name>赤</Name><LoLoColor>
      <LoColor><Name>暗い赤</Name><LoColor>
      <NormalColor><Name>黒</Name></NormalColor>
      <HiColor><Name>暗い緑</Name></HiColor>
      <HiHiColor><Name>緑</Name></HiHiColor>
    </Value>
  </TextColorAnalogAlarm>
```

偏差型アラームの例:

```
<TextColorAnalogAlarm>
    <Title>TextColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Deviation>
      <NormalColor><Name>黒</Name></NormalColor>
      <MinorColor><Name>緑</Name></MinorColor>
      <MajorColor><Name>赤</Name></MajorColor>
    </Deviation>
  </TextColorAnalogAlarm>
```

ROC 型アラームの例:

```
<TextColorAnalogAlarm>
    <Title>TextColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <ROC>
      <NormalColor><Name>黒</Name></NormalColor>
      <ROCColor><Name>赤</Name></ROCColor>
    </ROC>
  </TextColorAnalogAlarm>
```

垂直スライダー

垂直スライダー アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
ReferenceLocation(参照位置情報)	垂直参照: top、middle、bottom。デフォルトは bottom です。
上値	上の値。BottomValue と同じ値にすることはできません。デフォルト値は 0 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
下値	下の値。既定値は 100 です。
上向き移動	上への移動。値の範囲は 0 から 32767 です。範囲外の値は固定され、エラー メッセージが記録されます。既定値は 50 です。
下向き移動	下への移動。値の範囲は 0 から 32767 です。範囲外の値は固定され、エラー メッセージが記録されます。既定値は 50 です。
式	アナログ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<SliderVertical>
    <Title>SliderVertical1</Title>
    <ReferenceLocation>ボトム</ReferenceLocation>
    <TopValue>10.0</TopValue>
    <BottomValue> 110.0</BottomValue>
    <UpwardMovement> 20.0</UpwardMovement>
    <DownwardMovement> 120.0</DownwardMovement>
    <Expression> <![CDATA[aTag001]]> </Expression>
  </SliderVertical>
```

最小の例:

```
<SliderVertical>
    <Expression>dTag001</Expression>
  </SliderVertical>
```

水平スライダー

水平スライダー アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
ReferenceLocation(参照位置情報)	参照位置: left、center、right。デフォルトは left です。
左値	スライダーの左の値。デフォルト値は 0 です。
右値	スライダーの右の値。既定値は 100 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
LeftMovement	左の水平移動。値の範囲は 0 から 32767 です。範囲外の値は固定され、エラー メッセージが記録されます。既定値は 50 です。
RightMovement	右の水平移動。値の範囲は 0 から 32767 です。範囲外の値は固定され、エラー メッセージが記録されます。既定値は 50 です。
式	アナログ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<SliderHorizontal>
    <Title>SliderHorizontal1</Title>
    <ReferenceLocation>左</ReferenceLocation>
    <LeftValue>10.0</左値>
    <RightValue>120.0</右値>
    <LeftMovement>20.0</LeftMovement>
    <RightMovement>150.0</RightMovement>
    <Expression>
      <![CDATA[aTag001]]>
    </Expression>
  </SliderHorizontal>
```

最小の例:

```
<SliderHorizontal>
    <Expression><![CDATA[aTag001]]><Expression>
  </SliderHorizontal>
```

オブジェクト高さ

オブジェクト高さアニメーション リンクに対して以下の要素を指定できます。オブジェクト高さアニメーション リンクは向きアニメーション リンクと一緒に使用できません。

要素	説明
タイトル	オブジェクトの名前。オプション。
SizeAnchor(サイズアンカー)	オブジェクト上でアンカーを配置する位置を定義します。値は、bottom、middle、top です。デフォルトは bottom です。
SizeMin(サイズ最小)	最小高さの値。デフォルト値は 0 です。
SizeMax(サイズ最大)	最大高さの値。既定値は 100 です。SizeMin 要素に割り当てられた値よりも大きい値である必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
MinPercent(最小パーセント)	高さの最小パーセンテージ。デフォルトは 100 です。範囲は 0 から 100 です。
MaxPercent(最大パーセント)	高さの最大パーセンテージ。既定値は 100 です。最小パーセントよりも大きい値である必要があります。範囲は 0 から 100 です。
式	アナログ型タグ名または式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<ObjectSizeHeight>
    <Title>ObjectSizeHeight1</Title>
    <Expression> <![CDATA[aTag001]]> <Expression>
    <SizeMin>0.0</SizeMin>
    <SizeMax>100.0</SizeMax>
    <MinPercent>0.0</MinPercent>
    <MaxPercent>100.0</MaxPercent>
    <SizeAnchor>Top</SizeAnchor>
  </ObjectSizeHeight>
```

最小の例:

```
<ObjectSizeHeight>
    <Expression>dTag001</Expression>
  </ObjectSizeHeight>
```

オブジェクト幅

オブジェクト サイズ幅アニメーション リンクに対して以下の要素を指定できます。オブジェクト幅アニメーション リンクは向きアニメーション リンクと一緒に使用できません。

要素	説明
タイトル	オブジェクトの名前。オプション。
SizeAnchor(サイズアンカー)	オブジェクト上でアンカーを配置する位置を定義します。値は{Left, center, right}になります。デフォルトは left です。
SizeMin(サイズ最小)	最小幅の値。デフォルト値は 0 です。
SizeMax(サイズ最大)	最大幅の値。既定値は 100 です。SizeMin よりも大きい値である必要があります。
MinPercent(最小パーセント)	幅の最小パーセンテージ。デフォルトは 100 です。範囲は 0 から 100 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
MaxPercent(最大パーセント)	幅の最大パーセンテージ。既定値は 100 です。MinPercent よりも大きい値である必要があります。範囲は 0 から 100 です。
式	アナログ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
< ObjectSizeWidth>
    <Title>ObjectSizeWidth1</Title>
    <Expression> <![CDATA[aTag001]]> <Expression>
    <SizeMin>0.0</SizeMin>
    <SizeMax>100.0</SizeMax>
    <MinPercent>0.0</MinPercent>
    <MaxPercent>100.0</MaxPercent>
    <SizeAnchor>中央</SizeAnchor>
  </ObjectSizeWidth>
```

最小の例:

```
<ObjectSizeWidth>
    <Expression>dTag001</Expression>
  <ObjectSizeWidth>
```

垂直位置

垂直位置アニメーション リンクに対して以下の要素を指定できます。オブジェクト垂直位置 アニメーション リンクは向きアニメーション リンクと一緒に使用できません。

要素	説明
タイトル	オブジェクトの名前。オプション。
最小値	上の値。既定値は 0 です。下と同じ値にすることはできません。
最大値	下の値。既定値は 100 です。上と同じ値にすることはできません。
DecreaseMovement	上への垂直移動。既定値は 0 です。0 より小さくすることはできません。
IncreaseMovement	下への垂直移動。既定値は 100 です。100 より小さい値である必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
式	アナログ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<LocationVertical>
    <Title>LocationVertical1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <MinValue>0.0</MinValue>
    <MaxValue>100.0</MaxValue>
    <DecreaseMovement>0</DecreaseMovement>
    <IncreaseMovement>100</IncreaseMovement>
  <LocationVertical>
```

最小の例:

```
<LocationVertical>
    <Expression>dTag001</Expression>
  <LocationVertical>
```

水平位置

水平位置アニメーション リンクに対して以下の要素を指定できます。オブジェクト水平位置 アニメーション リンクは向きアニメーション リンクと一緒に使用できません。

要素	説明
タイトル	オブジェクトの名前。オプション。
最小値	左の値。既定値は 0 です。右端と同じ値にすることはできません。
最大値	右の値。既定値は 100 です。左端と同じ値にすることはできません。
DecreaseMovement	左への水平移動。既定値は 0 です。0 より小さくてはいけません。
IncreaseMovement	右への水平移動。既定値は 100 です。100 より小さい値である必要があります。
式	アナログ型タグまたは式。必須。式が無効な場合、または

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
	欠落している場合、オブジェクトは作成されません。

例:

```
<LocationHorizontal>
    <Title>LocationHorizontal1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    <Expression>
    <MinValue>0.0</MinValue>
    <MaxValue>100.0</MaxValue>
    <DecreaseMovement>0</DecreaseMovement>
    <IncreaseMovement>100</IncreaseMovement>
  </LocationHorizontal>
```

最小の例:

```
<LocationHorizontal>
    <Expression>dTag001</Expression>
  </LocationHorizontal>
```

垂直塗りつぶしパーセント

垂直塗りつぶしパーセント アニメーション リンクに対して以下の要素を指定できます。垂直塗りつぶしパーセント アニメーション リンクは向きアニメーション リンクと一緒に使用できません。

要素	説明
タイトル	オブジェクトの名前。オプション。
FillDirection	移動方向を定義します。可能な値は {Up、down} です。デフォルトは up です。必須。
塗りつぶしの色	純色の塗りつぶしの色要素。既定値は rgb(0, 0, 0) です。
FillMin	最小値を定義します。デフォルト値は 0 です。
FillMax	最大値を定義します。既定値は 100 です。最小塗りつぶしよりも大きい値である必要があります。
FillMinPercent	最小値をパーセンテージとして定義します。デフォルトは 100 です。範囲は 0 から 100 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
FillMaxPercent	最大値をパーセンテージとして定義します。既定値は 100 です。範囲は 0 から 100 です。最小パーセンテージよりも大きい値である必要があります。
式	アナログ型タグまたは式。必須。

例:

```
<PercentFillVertical>
    <Title>PercentFillVertical1</Title>
    <Expression><![CDATA[aTag001]]></Expression>
    <FillMin>0.0</FillMin>
    <FillMax>100.0</FillMax>
    <FillMinPercent>0</FillMinPercent>
    <FillMaxPercent>100</FillMaxPercent>
    <FillColor><Name>紫</Name></FillColor>
    <FillDirection>上</FillDirection>
   </PercentFillVertical>
```

最小の例:

```
<PercentFillVertical>
    <Expression>dTag001</Expression>
    <FillDirection>上</FillDirection>
   </PercentFillVertical>
```

水平塗りつぶしパーセント

水平塗りつぶしパーセント アニメーション リンクに対して以下の要素を指定できます。水平塗りつぶしパーセント アニメーション リンクは向きアニメーション リンクと一緒に使用できません。

要素	説明
タイトル	オブジェクトの名前。オプション。
FillDirection	left、right。デフォルトは right です。
塗りつぶしの色	1 つの色要素を含みます。既定値は rgb(0, 0, 0) です。
FillMin	最小塗りつぶしの値。デフォルト値は 0 です。
FillMax	最大塗りつぶしの値。既定値は 100 です。最小塗りつぶしよりも大きい値である必要があります。
FillMinPercent	オブジェクトを塗りつぶす最小パーセント。デフォルトは 100 です。範囲は 0 から 100 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
FillMaxPercent	オブジェクトを塗りつぶす最大パーセント。デフォルトは 100 です。範囲は 0 から 100 です。最小塗りつぶしパーセントよりも大きい値である必要があります。
式	アナログ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<PercentFillHorizontal>
    <Title>PercentFillHorizontal1</Title>
    <Expression><![CDATA[aTag001]]></Expression>
    <FillMin>0.0</FillMin>
    <FillMax>100.0</FillMax>
    <FillMinPercent>0</FillMinPercent>
    <FillMaxPercent>100</FillMaxPercent>
    <FillColor><Name>紫</Name></FillColor>
    <FillDirection>右</FillDirection>
   </PercentFillHorizontal>
```

最小の例:

```
<PercentFillHorizontal>
    <Expression>dTag001</Expression>
    <FillDirection>左</FillDirection>
   </PercentFillHorizontal>
```

論理値押しボタン

論理値押しボタン アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
ボタンタイプ	論理ボタンのタイプ: {direct, reverse, toggle, reset, or set}。必須。無効な場合、または欠落している場合、オブジェクトは作成されません。
式	論理型タグまたは式。必須。Expression 要素が無効な場合、または欠落している場合、オブジェクトは作成されません。
キー割り当て	仮想キー要素、空の文字列、または存在しない。デフォルトでは割り当ては設定されません。空の文

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
	字列は割り当てが発生しないことを示します。

例:

```
<ButtonDiscreteValue>
    <Title>ButtonDiscreteValue1</Title>
    <ButtonType>リバース</ButtonType>
    <KeyAssignment>
      <KeyCode>F2</KeyCode>
      <KeyFlags>シフト</KeyFlags>
    </KeyAssignment>
    <Expression>
        <![CDATA[dTag001]]>
    <Expression>
  </ButtonDiscreteValue>
```

最小の例:

```
<ButtonDiscreteValue>
    <ButtonType>リバース</ButtonType>
    <Expression><![CDATA[dTag001]]></Expression>
  </ButtonDiscreteValue>
```

ウィンドウの表示押しボタン

リンクを生成したときに名前付きウィンドウが存在しない場合、警告がログに記録されます。ウィンドウに対するアクションなしでリンクが生成されます。少なくとも 1 つの名前付きウィンドウが存在する必要があります。そうでない場合、ShowWindow アニメーション リンクはインポートされません。

ウィンドウの表示押しボタン アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
WindowName	表示するウィンドウの名前。少なくとも 1 つのウィンドウ名を指定する必要があります。そうでない場合、このアニメーション リンクはオブジェクトに追加されません。

例:

```
<ButtonShowWindow>
    <Title>ButtonShowWindow1</Title>
    <WindowName>
      <![CDATA[Window006]]>
    </WindowName>
    <WindowName>Window007</WindowName>
    <WindowName><![CDATA[SliderWindow]]></WindowName>
  </ButtonShowWindow>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウの非表示押しボタン

リンクを生成したときに名前付きウィンドウが存在しない場合、警告がログに記録されます。ウィンドウに対するアクションなしでリンクが生成されます。少なくとも 1 つの名前付きウィンドウが存在する必要があります。そうでない場合、HideWindow アニメーション リンクはインポートされません。

ウィンドウの非表示押しボタン アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
WindowName	非表示にするウィンドウの名前。少なくとも 1 つのウィンドウ名を指定する必要があります。そうでない場合、このアニメーション リンクはオブジェクトに追加されません。

例:

```
<ButtonHideWindow>
    <Title>ButtonHideWindow1</Title>
    <WindowName>
      <![CDATA[Window001]]>
    </WindowName>
    <WindowName>Window002</WindowName>
    <WindowName> <![CDATA[SliderWindow]]> WindowName：
  </ButtonHideWindow>
```

表示オン/オフ

表示オン/オフ アニメーション リンクに対して以下の要素を指定できます。

要素	説明
Title	オブジェクトの名前。オプション。
State	オブジェクトを表示するかどうかを定義します。可能な値は以下のとおりです。on、off。デフォルトは On です。オプションのスキーマを使用する場合に必要です。
Expression	論理型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<Visibility>
<Title>Visibility1</Title>
<Expression>dTag001</Expression>
<State>On</State>
</Visibility>
```

最小の例:

```
<Visibility>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
<Expression>dTag001</Expression>
<State>On</State>
</Visibility>
```

点滅

点滅アニメーション リンクに対して以下の要素を指定できます。式が真の場合、アニメーション リンクが点滅します。

要素	説明
Title	オブジェクトの名前。オプション。
BlinkAttribute	Invisible、Visible。デフォルトは Visible です。
BlinkSpeed	オブジェクトの点滅レートを定義します。可能な値は以下のとおりです。Slow、Medium、Fast。デフォルトは Medium です。
TextColor	点滅するテキストの色を定義します。1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
LineColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
FillColor	1 つの色要素を含みます。デフォルトは rgb(0,0,0) です。
Expression	論理型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<Blink>
<Title>Blink1</Title>
<Expression>dTag001</Expression>
<TextColor><R>0</R><G>0</G><B>0</B></TextColor>
<LineColor><R>0</R><G>0</G><B>0</B></LineColor>
<FillColor><R>0</R><G>255</G><B>0</B></FillColor>
<BlinkAttribute>Invisible</BlinkAttribute>
<BlinkSpeed>Slow</BlinkSpeed>
</Blink>
```

最小の例:

```
<Blink>
<Expression>dTag001</Expression>
<BlinkAttribute>Visible</BlinkAttribute>
<BlinkSpeed>Fast</BlinkSpeed>
</Blink>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

向き

向きアニメーション リンクに対して以下の要素を指定できます。向きアニメーション リンクは、スライダー、サイズ、位置、および塗りつぶしパーセント アニメーション リンクと一緒に使用することはできません。

要素	説明
Title	オブジェクトの名前。オプション。
X	オブジェクトの中心点からの水平オフセット。オプション。デフォルトは 0 です。
Y	オブジェクトの中心点からの垂直オフセット。オプション。デフォルトは 0 です。
CWMax	最大時計回りの回転の値。デフォルトは 100 です。
CWRotation	時計回りの回転。デフォルトは 360 です。値の範囲は 0 から 360 です。CWROTATION+CCWROTATION は 360 以下である必要があります。
CCWMax	最大反時計回りの回転の値。デフォルトは 0 です。
CCWRotation	反時計回りの回転。デフォルトは 0 です。値の範囲は 0 から 360 です。CWRotation+CCWRotation は 360 以下である必要があります。
Expression	アナログ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<Orientation>
<Title>Orientation1</Title>
<Expression> </Expression>
<X>0</X> <Y>0</Y>
<CWMax>100.0</CWMax>
<CWRotation>360.0</CWRotation>
<CCWMax>0.0</CCWMax>
<CCWRotation>0.0</CCWRotation>
</Orientation>
```

最小の例:

```
<Orientation>
<Expression>aTag001</Expression>
</Orientation>
```

無効化

無効化アニメーション リンクに対して以下の要素を指定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
Title	オブジェクトの名前。オプション。
State	on、off。デフォルトは On です。オプションのスキーマを使用する場合に必要です。
Expression	論理型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<Disable>
<Title>Disable1</Title>
<Expression>dTag001</Expression>
<State>On</State>
</Disable>
```

最小の例:

```
<Disable>
<Expression>dTag001</Expression>
<State>On</State>
</Disable>
```

静的ツールヒント

静的ツールヒント アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
メッセージ型	静的テキスト メッセージ。必須。無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<TooltipStatic>
    <Title>TooltipStatic1</Title>
    <Message>
      <![CDATA[ 100 万ドルを獲得するにはここを選択してください!
      ]]>
    </Message>
  </TooltipStatic>
```

動的ツールヒント

動的ツールヒント アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

要素	説明
式	Expression 要素。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<TooltipTag>
    <Title>TooltipTag1</Title>
    <Expression>
      <![CDATA[sTag001]]>
    <Expression>
  </TooltipTag>
```

論理値表示

論理値アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
オンメッセージ	アニメーション リンクの値が真のときに表示する文字列。デフォルト テキストは On です。
オフメッセージ	アニメーション リンクの値が偽のときに表示する文字列。デフォルト テキストは Off です。
式	論理型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<ValueDisplayDiscrete>
    <Title>ValueDisplayDiscrete1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    <Expression>
    <OnMessage>
      <![CDATA[ポンプがオン]]>
    </OnMessage>
    <OnMessage>
      <![CDATA[ポンプがオフ]]>
    </OffMessage>
  <ValueDisplayDiscrete>
```

最小の例:

```
<ValueDisplayDiscrete>
    <Expression>dTag001</Expression>
  </ValueDisplayDiscrete>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アナログ値表示

アナログ値アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
式	アナログ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<ValueDisplayAnalog>
    <Title>ValueDisplayAnalog1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    <Expression>
  </ValueDisplayAnalog>
```

文字列値表示

文字列値アニメーション リンクに対して以下の要素を指定できます。

要素	説明
タイトル	オブジェクトの名前。オプション。
式	メッセージ型タグまたは式。必須。式が無効な場合、または欠落している場合、オブジェクトは作成されません。

例:

```
<ValueDisplayString>
    <Title>ValueDisplayString1</Title>
    <Expression>
        <![CDATA[sTag001]]>
    <Expression>
  </ValueDisplayString>
```

押しボタン アクション スクリプト

押しボタン アクション スクリプト要素は、アニメーション リンク要素内で 0 宣言されたボタン アクション スクリプト要素内で宣言されます。

例:

```
<AnimationLinks>
  <ButtonScripts>
    <OnLeftDown>
        <![CDATA[
          スクリプト テキストの最初の行
          スクリプト テキストの 2 番目の行
          スクリプト テキストの N 番目の行
        ]]>
    </OnLeftDown>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
  </ButtonActionScripts>
</AnimationLinks>
```

左のクリック時/押した時

同じオブジェクトで OnLeftDown と OnKeyDown アニメーション リンクの両方を使用しないでください。

例:

```
<OnLeftDown>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnLeftDown>
```

左クリック/左を押している間

WhileLeftDown または WhileKeyDown アニメーション リンクに対して以下の要素を指定できます。

要素	説明
Text	スクリプト テキスト。必須。
頻度	ミリ秒単位でのスクリプト実行頻度。範囲は 1 から 360000 です。オプションのスキーマを使用する場合に必要です。デフォルトは 1000 です。

同じオブジェクトで WhileLeftDown と WhileKeyDown アニメーション リンクの両方を使用しないでください。

例:

```
<WhileLeftDown>
<Text>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</Text>
<Frequency>1000</Frequency>
</WhileLeftDown>
```

左のクリック時/離した時

OnLeftUp 要素内にスクリプトのテキストを配置します。同じオブジェクトで OnLeftUp と OnKeyUp アニメーション リンクを使用しないでください。

例:

```
<OnLeftUp>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnLeftUp>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

左のダブルクリック時

OnLeftDoubleClick アニメーション リンク内にスクリプトのテキストを配置します。

例:

```
<OnLeftDoubleClick>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnLeftDoubleClick>
```

右のクリック時/押した時

OnRightDown 要素内にスクリプトのテキストを配置します。同じオブジェクトで OnRightDown と OnRightKeyDown アニメーション リンクを使用しないでください。

例:

```
<OnRightDown>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnRightDown>
```

右クリック/右を押している間

WhileRightKeyDown または WhileRightDown アニメーション リンクに対して以下の要素を指定できます。

要素	説明
Text	スクリプト テキスト。必須
頻度	ミリ秒単位でのスクリプト実行頻度。1 から 360000 の数値。デフォルトは 1000 です。

同じオブジェクトで WhileRightDown と WhileRightKeyDown アニメーション リンクの両方を使用しないでください。

例:

```
<WhileRightDown>
<Text>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</Text>
<Frequency>1000</Frequency>
</WhileRightDown>
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

右クリック/右を離した時

OnRightUp および OnRightKeyUp 要素内にスクリプトのテキストを配置します。同じオブジェクトで OnRightUp と OnRightKeyUp アニメーション リンクの両方を使用しないでください。

例:

```
<OnRightUp>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnRightUp>
```

右のダブルクリック時

OnRightDoubleClick 要素内にスクリプトのテキストを配置します。

例:

```
<OnRightDoubleClick>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnRightDoubleClick>
```

中央のクリック時/押した時

OnMiddleKeyDown 要素内にスクリプトのテキストを配置します。同じオブジェクトで OnMiddleDown と OnMiddleKeyDown アニメーション リンクを使用しないでください。

例:

```
<OnMiddleDown>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnMiddleDown>
```

中央クリック/中央を押している間

WhileMiddleKeyDown または WhileMiddleDown アニメーション リンクに対して以下の要素を指定できます。

要素	説明
Text	スクリプト テキスト。必須
頻度	ミリ秒単位でのスクリプト実行頻度。1 から 360000 の数値。オプションのスキーマを使用する場合に必要です。デフォルトは 1000 です。

同じオブジェクトで WhileMiddleDown と WhileMiddleKeyDown アニメーション リンクを使用しないでください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例:

```
<WhileMiddleDown>
<Text>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</Text>
<Frequency>1000</Frequency>
</WhileMiddleDown>
```

中央のクリック時/離した時

OnMiddleKeyUp および OnMiddleUp アニメーション リンク内にスクリプトのテキストを配置します。同じオブジェクトで OnMiddleUp と OnMiddleKeyUp アニメーション リンクを使用しないでください。

例:

```
<OnMiddleUp>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnMiddleUp>
```

中央のダブルクリック時

OnMiddleDoubleClick 要素内にスクリプトのテキストを配置します。

例:

```
<OnMiddleDoubleClick>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
スクリプト テキストの N 番目の行
]]>
</OnMiddleDoubleClick>
```

マウス ホバー時

OnMouseOver アニメーション リンクに対して以下の要素を指定できます。

要素	説明
Text	スクリプト テキスト。必須。
タイムアウト	ミリ秒単位のタイムアウト。オプションのスキーマを使用する場合に必要です。デフォルトは 250 です。

例:

```
<OnMouseOver>
<Text>
<![CDATA[
スクリプト テキストの最初の行
スクリプト テキストの 2 番目の行
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
スクリプト テキストの N 番目の行
]]>
</Text>
<Timeout>250</Timeout>
</OnMouseOver>
```

左キー アクションの割り当て

左キー アクションの割り当てアニメーション リンクに対して以下の要素を指定できます。

要素	説明
キーコード	仮想キー コード名または空の文字列。必須。空の文字列は割り当てが発生しないことを示します。
キーフラグ	仮想キー フラグ組み合わせまたは空の文字列。必須。空の文字列は割り当てが発生しないことを示します。KeyCode を指定しない場合、無効化されます。

"CTRL+B" の例:

```
<LeftKey>
    <KeyCode>B</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </LeftKey>
```

"L" に対するキー フラグ修飾子なしの例:

```
<LeftKey>
    <KeyCode>L</KeyCode>
  </LeftKey>
```

CTRL+SHIFT+F7 に対する両方のキー フラグ修飾子の例:

```
<LeftKey>
    <KeyCode>F7</KeyCode>
    <KeyFlags>CtrlShift</KeyFlags>
  <LeftKey>
```

右キー アクションの割り当て - サポートされていません

右キー アクションの割り当ては、InTouch XML インポート機能でサポートされていません。

右キー アクションの割り当てアニメーション リンクに対して以下の要素を指定できます。

要素	説明
キーコード	仮想キー コード名または空の文字列。必須。空の文字列は割り当てが発生しないことを示します。
キーフラグ	仮想キー フラグ組み合わせまたは空の文字列。必須。空の文字列は割り当てが発生しないことを示します。KeyCode を指定しない場合、無効化されます。

例:

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
<右キー>
    <KeyCode>B</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </右キー>
```

中央キー アクションの割り当て - サポートされていません

中央キー アクションの割り当ては、InTouch XML インポート機能でサポートされていません。

中央キー アクションの割り当てアニメーション リンクに対して以下の要素を指定できます。

要素	説明
キーコード	仮想キー コード名または空の文字列。必須。空の文字列は割り当てが発生しないことを示します。
キーフラグ	仮想キー フラグ組み合わせまたは空の文字列。必須。空の文字列は割り当てが発生しないことを示します。KeyCode を指定しない場合、無効化されます。

```
<中央キー>
    <KeyCode>B</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </中央キー>
```

サポートされていない InTouch 機能

ウィザードと ActiveX コントロールは、XML インポート機能を使用してインポートできません。

タグまたはタグ データベースは、XML インポート機能を使用してインポートできません。タグ定義を含むファイルを作成し、DBLoad ユーティリティを使用してアプリケーションのタグ名ディクショナリにインポートすることができます。DBLoad の使用の詳細については、『*AVEVA™ InTouch HMI アプリケーション メンテナンス*』参照してください。コマンド プロンプトからの DBLoad の実行の詳細については、「コマンド プロンプトから DBLoad を実行する」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 15 CONNECT での産業用グラフィックの操作

CONNECT は、産業用グラフィック、コントロール、ウィジェットなどの共通コンテンツをクラウドで管理できる共通クラウド リポジトリです。

グラフィックは、オンデマンドでダウンロードおよびアップロードできます。グラフィックは CONNECT の「ストア」に格納されます。グローバル、テナント、およびユーザー固有の 3 種類のストアがあります。各ストア内で複数のドライブを設定できます。クラウドでグラフィックを管理するには CONNECT ユーザー アカウントが必要です。各ドライブには、さまざまなユーザーに別のアクセス レベルを設定できます。

管理者が付与したアクセスに応じてユーザーに読み取り専用または読み取り/書き込みのアクセスが設定されます。

- 読み取り専用アクセスを持つユーザーは、クラウドでグラフィックを表示できます。このユーザーは、クラウドからグラフィックをダウンロードできますが、そのグラフィックは読み取り専用になります。グラフィックを変更することやアップロードすることはできません。
- 読み取り/書き込みアクセスを持つユーザーは、グラフィックの表示、ダウンロード、変更、およびアップロードを行うことができます。

複数のユーザーが同じドライブ内のグラフィックにアクセスできますが、1 つのグラフィックを複数のユーザーが同時に編集することはできません。

CONNECT にログインする

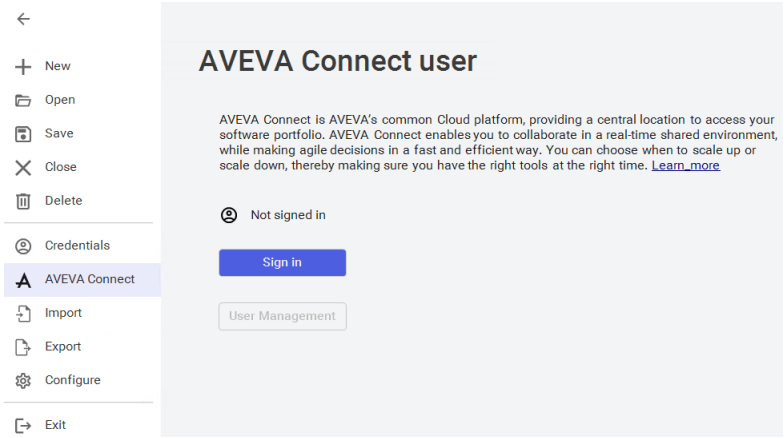
CONNECT にアクセスするには、AVEVA Connect アカウントに加えて管理者からの承認が必要です。

1. InTouch を使用している場合、WindowMaker を起動し、［**ファイル**］メニューから［**CONNECT** ］を選択します。

   System Platform IDE を使用している場合、［**Galaxy**］メニューを選択し、［**CONNECT** ］を選択します。

   注：コンフィグレータの［**ライセンス モード**］タブで［**接続エクスペリエンス**］モードが選択されている場合、**CONNECT** オプションは使用できません。

   CONNECT ユーザーの設定画面が表示されます。



2. **[サインイン]**を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. CONNECT 電子メール アドレスを入力します。
4. パスワードを入力します。

   資格情報が検証されてサインイン処理が完了します。

   注：CONNECT 産業用グラフィックドライブにアクセスするには、コンテンツ投稿者の役割が必要です。これは、Operations Control 接続エクスペリエンスと非接続エクスペリエンスの両方で使用する場合に必要です。

5. **AVEVA Cloud Integration Studio ソリューション**を選択します例:Pym Technologies。
6. AVEVA Drive が［視覚化］フォルダ内の個別のリポジトリとして表示されます。

注：選択した Integration Studio ソリューションはブラウザ メモリにキャッシュされます。Integration Studio ソリューションを切り替えるには、ブラウザ キャッシュをクリアする必要があります。

共有ドライブ間のナビゲーション

特定のストアにログインした後、ドライブ間を移動してグラフィックのアップロードとダウンロードを行うことができます。

1. ［**表示**］メニューの［**クラウド ドライブ**］グループで［**共有ドライブ**］を選択します。
2. 使用可能なオプションからドライブを選択します。

   可視化フォルダが更新され、選択したドライブが表示されます。

注：別のユーザーによってロックされているグラフィックを移動することはできません。

グラフィックのアップロード/ダウンロード

AVEVA Drive 機能は、ローカルの可視化フォルダに似ています。AVEVA Drive にフォルダをアップロードまたは作成してグラフィックを整理できます。読み取り/書き込みアクセスがあるユーザーは、グラフィックをローカルの可視化フォルダにダウンロードしてアプリケーションで使用できます。グラフィックをダウンロードまたはアップロードするとき、コンテンツは選択した上書きオプションに従って上書きされます。

- ターゲット ドライブにコンテンツが既に存在する場合、コンテンツは同じツールセットの場所で保持されます。そうでない場合、コンテンツは同じツールセット パスにソースとして追加されます。

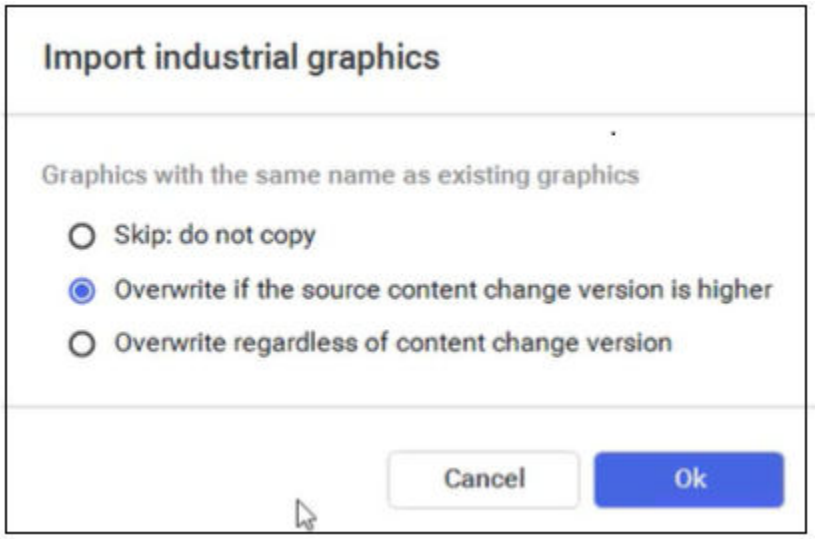
選択した埋め込みコンテンツに対して次の操作を行います。

- ターゲット ドライブに埋め込みコンテンツが既に存在する場合、コンテンツはターゲットの同じツールセットの場所で保持されます。
- 埋め込みコンテンツが親コンテンツと同じツールセットの場所または同じフォルダ階層内に存在する場合、埋め込みコンテンツは親として同じツールセット パスに追加されます。そうでない場合、埋め込みコンテンツは同じツールセット パスにソースとして追加されます。

CONNECT へのグラフィックのアップロード

1. ローカルの Visualization フォルダーからグラフィックを AVEVA Drive にドラッグ アンド ドロップします。
2. ドライブにグラフィックが既に存在する場合、上書きするかアップロードをスキップするかを尋ねるダイアログが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 適切なオプションを選択し、「**OK**」を選択します。
4. ［**コンテンツをアップロード**］ダイアログが表示されて処理の進捗状況が表示されます。［**詳細の表示**］を選択すると進捗状況が表示されます。
5. **[閉じる]**を選択します。

複数のグラフィックを含むフォルダをアップロードすることもできます。

ローカルの可視化フォルダーにグラフィックをダウンロードする

読み取り/書き込みアクセスのあるユーザーは、グラフィックを AVEVA Drive からローカルの可視化フォルダーにダウンロードできます。

1. グラフィックまたはフォルダーを AVEVA ドライブからドラッグし、ローカルの可視化フォルダーにドロップします。

   コンテンツのダウンロードの確認を要求する［**コンテンツのダウンロード**］ダイアログが表示されます。

- ダウンロードを続行するには、［**はい**］を選択します。
- 操作をキャンセルするには、［**いいえ**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 使用されているコントロールとネームスペースの一覧を表示するには、［**詳細の表示**］を選択します。ダウンロード中に発生したエラーも表示されます。

2. グラフィックが既に存在する場合、［産業用グラフィックのインポート］ダイアログ ボックスが表示されます。
3. 適切なオプションを選択します。
4. グラフィックはローカルの可視化フォルダーにダウンロードされます。埋め込み画像もコピーされます。

さまざまなクラウド コンテンツのアップロード/ダウンロードのサポート

さまざまなクラウド コンテンツのアップロード/ダウンロードのサポートを以下の表に示します。

クラウド コンテンツ	アップロード サポート	ダウンロード サポート
埋め込まれたクライアント コントロールおよび HTML5 ウィジェット	アップロードでは、埋め込まれたグラフィック コンテンツ（クライアント コントロールや HTML5 ウィジェットなど）が aaPKG 形式でパッケージ化され、クラウドにコピー/上書きされます。	ダウンロードでは、埋め込まれたグラフィック コンテンツ（クライアント コントロールや HTML5 ウィジェットなど）が抽出され、コントロールがローカル アプリケーションにインポート/上書きされます。
埋め込まれたアプリケーションのオブジェクトグラフィック	サポートされていません	サポートされていません
カスタム スクリプト ライブラリ	サポートされていません	サポートされていません
アプリケーション スタイル ライブラリ	サポートされていません	サポートされていません

CONNECT でグラフィックを管理する

ローカル グラフィックの管理と同様に、AVEVA Drive 上でグラフィックの作成、名前の変更、更新、複製、および削除を直接行うことができます。

- AVEVA Drive または任意のフォルダを右クリックして新しいグラフィックまたはツールセットを作成します。
- グラフィックをダブルクリックし、グラフィック エディタを使用して編集します。
- 右クリックして目的のオプションを選択し、グラフィックの名前の変更、複製、または削除、およびサムネイルの更新を行います。

  クラウドでサポートされないコンテキスト メニュー オプションはグレー表示されます。（［Web Client ホーム シンボルの作成］、［Web Client ルート フォルダの設定］、［シンボルのエクスポート］、［ローカリゼーションのエクスポート］など）。

  注：埋め込まれたグラフィックのサムネイルを AVEVA Drive で直接更新すると、多くのクラウド トラフィックが発生します。したがって、サムネイルはローカルで更新することをお勧めします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

複数のユーザーによるグラフィックの管理

AVEVA Drive で使用可能なグラフィックは複数のユーザーが同時に使用できます。しかし、グラフィックを編集できるのは一度に 1 人のユーザーだけです。クラウド レポジトリ内のグラフィックを 1 人のユーザーが編集している場合、グラフィックはチェックアウトおよびロックされるので、他のユーザーは編集を行うことができません。他のすべてのユーザーは、チェックアウトされた変更なしの読み取り専用グラフィックのみを表示できます。最新の変更は、変更を行ったユーザーがチェックインした後に反映されます。

- フォルダー内のグラフィック アイコンに表示される鍵のアイコンは、グラフィックがログイン ユーザーによってチェックアウトされていることを示します。
- フォルダー内のグラフィック アイコンに対する編集アイコンは、別のユーザーがグラフィックを編集していることを示します。

最初のユーザーがグラフィックを保存してチェックインすると、2 番目のユーザーがチェックアウトして変更を加えることができます。グラフィックを保存してチェックインすると、すべてのユーザーが変更されたグラフィックを表示できます。

クラウドグラフィックバージョンについて

AVEVA Drive グラフィックは、さまざまなバージョンのグラフィック サブシステムの多様な製品からパブリッシュされたクラウド上の産業用グラフィック、クライアント コントロール、およびウィジェットをサポートします。

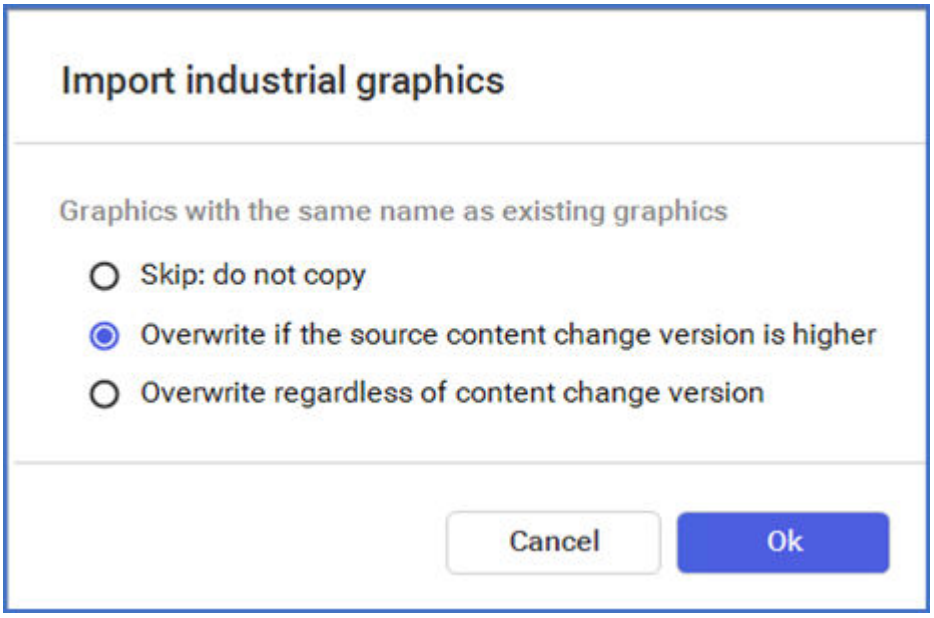
クラウド グラフィック バージョンのサポートの主な概念

- 産業用グラフィックは、グラフィック サブシステムで使用できるすべての機能をサポートする特定の形式でローカルおよびクラウドで保存されます。
- グラフィック サブシステムに新機能を追加した場合、グラフィックを保存して、その他のユーザーが使用できるようにチェックインする必要があります。
- 産業用グラフィック サブシステムのバージョンは、一意の内部バージョン番号によって識別されます。ユーザーが視覚的に識別できるものではありません。各製品はリリース時に産業用グラフィック サブシステムの最新バージョンを使用します。
- 産業用グラフィック サブシステムのバージョン番号は、保存された各産業用グラフィックの一部として含まれ、一致するバージョンまたはより高いバージョンの産業用グラフィック サブシステムに接続された製品のみがその産業用グラフィックの編集、埋め込み、またはダウンロードを行うことが許可されるように使用されます。
- 同じクラウド レポジトリに接続されたすべてのユーザーには、同じセットのグラフィックが表示されます。クラウド グラフィックの編集、埋め込み、またはダウンロードできる機能は、ユーザーが接続されている製品内の産業用グラフィック サブシステムのバージョンに依存します。
- 新しいバージョンの産業用グラフィック サブシステムに接続された製品によってクラウド リポジトリに保存された産業用グラフィックは、同じクラウドにアクセスする古いバージョンの製品の可視化フォルダでは無効化された状態で表示されます。古い製品バージョンのユーザーは、グラフィック レポジトリの編集、埋め込み、およびダウンロードを行うことができません。
- ローカルの可視化フォルダと同様に、グラフィック名は 1 つのクラウド レポジトリ内で一意である必要があります。これは、グラフィックが産業用グラフィック サブシステムの別のバージョンに接続された製品によって保存された場合でも同じです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クラウドでのグラフィック コンテンツの上書き

アップロードまたはダウンロードしたグラフィック コンテンツが既に存在する場合、グラフィックを上書きするかどうかを尋ねるメッセージが表示されます。



上書きのオプションは、ソースとターゲットのグラフィック サブシステムのバージョン、およびグラフィックの変更ログのバージョンに応じて異なります。

- オプション 1 では、グラフィック サブシステムのバージョンやソースとターゲットの変更ログのバージョンに関係なく上書きがスキップされます。
- グラフィック サブシステムのバージョンがソースとターゲットで異なる場合、オプション 2 と 3 の両方では、変更ログのバージョンに関係なくグラフィックが上書きされます。
- ソースとターゲットのグラフィック サブシステムのバージョンが同じである場合、次の処理が行われます。
  - オプション 2 では、ソースの変更ログのバージョンが高い場合にのみ上書きされます。
  - オプション 3 では、変更ログのバージョンにかかわらず上書きされます。

クラウドでのカスタム クライアント コントロールとウィジェットの上書き

クラウド グラフィックのバージョン管理のガイドラインは、このリリースの時点では、カスタム クライアント コントロールとウィジェットには適用されません。クライアント コントロールとウィジェットは、ダウンロードおよびアップロード操作の両方でサポートされます。しかし、上書きはサポートされません。カスタム クライアント コントロールまたはウィジェットがターゲットの場所に既に存在する場合、上書きはクライアント コントロールの内部バージョンに関係なくスキップされます。ソース リポジトリのクライアント コントロールを上書きするには、最初にソース リポジトリのクライアント コントロールを削除する必要があります。その後、クライアント コントロールのアップロードまたはダウンロードが正常に行われます。

注：カスタム クライアント コントロールまたはウィジェットをクラウドにアップロードした場合、最新のカスタム クライアント コントロールまたはウィジェットを使用するには、そのクラウド リポジトリに接続された他のすべてのユーザーは WindowMaker を再起動する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クラウド グラフィックのバージョン管理 - シナリオと例

このセクションでは、クラウド グラフィックのバージョン処理に関する高レベルのシナリオと予想される内容の非排他的なリストを紹介します。

バージョンが異なる 2 つの製品 P1_Old と P2_New について考えてみます。これらの製品は AVEVA Cloud の同じテナントに接続されています。

- P1_Old では古いバージョンのグラフィック サブシステムが使用され、P2_New では新しいバージョンが使用されています。
- 開始するにあたって、クラウド リポジトリには P1_Old によってアップロードされたグラフィック S1 と S2 があります。

シナリオ / ステップ	例
シナリオ 1: P2_New が S3 を作成します。 P1_Old が S3 を編集しようとします。	S3 は新しいグラフィック バージョンで保存されているので、古いバージョンでは編集、埋め込み、およびダウンロードできません。 P1_Old は、S3 が新しいバージョンで保存されていて開くことができないというメッセージを受け取ります。
シナリオ 2: P2_New が S2 を開きます。 P2_New が S2 をクラウドに保存します。 その後、P1_Old が S2 を開こうとします。	古いグラフィック バージョンで作成された S2 は、新しいグラフィック バージョンで開いて保存することができます。 S2 は新しいグラフィック バージョンで保存されているので、古いバージョンでは編集、埋め込み、およびダウンロードできません。 P1_Old は、S3 が新しいバージョンで保存されていて開くことができないというメッセージを受け取ります。
シナリオ 3: P1_Old が S1 と S2 を保存します。ここで、S2 は S1 に埋め込まれています。 P2_New が S2 だけを保存して更新します。 P1_Old が S1 を編集しようとします。	S1 は P1_Old と同じバージョンなので開くことができます。 埋め込まれた S2 はロードできません。
シナリオ 4: P2_New が S4 をクラウドに作成します。 P2_New が S1、S2、および S3 を S4 に埋め込みます。	S1 は古いグラフィック バージョンですが、S4（新しいバージョン）でロードできます。 S2 と S3 は S4 と同じグラフィック バージョンなので埋め込むことができます。
シナリオ 5:	S1 は同じグラフィック バージョンなので埋め込むことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

P1_Old がクラウドに新しい S5 を作成します。 P1_Old が S1、S2、および S3 を S5 に埋め込みます。	S2 と S3 は新しいバージョンなので埋め込むことができません。
シナリオ 6: P1_Old が S3 をダウンロードします。	S3 は新しいグラフィック バージョンなのでダウンロードできません。
シナリオ 7: 新しいグラフィック バージョンのグラフィック S3 がクラウド リポジトリにあります。 P1_Old が同じ名前（S3）のグラフィックをアップロードしようとします。	オプション 3 を選択することにより、グラフィックを上書きすることができます。
シナリオ 8: P2_New にグラフィック S8 があります。 P1_Old が S8 と S9 をローカルに作成し、S8 を S9 に埋め込みます。 P1_Old が S9 をクラウドにアップロードします。	S9 が クラウドにアップロードされます。 クラウドに新しいバージョンがあるので S8 はアップロードできません。オプション 3 を選択することにより、コンテンツを上書きすることができます。

ショートカットキー

[ファイル]メニュー

キーボード ショートカット	説明
Ctrl + N	新規ウィンドウ
Ctrl + O	ウィンドウを開く
Ctrl + S	ウィンドウを保存する

ホーム メニュー

グループ	キーボード ショートカット	説明
クリップボード		
	Ctrl + X	切り取り
	Ctrl+C	コピー

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	Ctrl + D	複製
	Ctrl+V	貼り付け
Alignment (整列)		
	Ctrl + F4	上揃え
	Ctrl + F6	上下中央揃え
	Ctrl + F8	下揃え
	Ctrl+F3	左揃え
	Ctrl+F5	水平中央揃え
	Ctrl+F7	右揃え
配置		
	Shift + F9	最前面へ移動
	F9	最背面へ移動
	F6	時計回りに回転
	Shift + F6	反時計回りに回転
	F7	左右反転
	Shift + F7	上下反転
間隔		
	Ctrl + I	水平方向
	Ctrl + K	垂直方向
	Ctrl + F9	中心点
グラフィック		
	F5	作成
	Shift + F5	中断
セル		
	F3	作成
	Shift + F3	中断
アラーム グループ		
	Shift + A	アラーム グループ リストを開く

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	Alt + A	アラーム グループを追加
タグ ディクショナリ		
	Ctrl+T	タグ名ディクショナリを開く
	Ctrl + P	アクセス名ダイアログを開く
	Ctrl + R	クロス リファレンス画面を開く
	Ctrl + U	使用ｶｳﾝﾄを更新
	Ctrl + G	未使用タグの削除

描画メニュー

グループ	キーボード ショートカット	説明
Mode		
	Ctrl + A	アニメーション リンク
	Shift+W	要素を選択
形式		
	A+ または Ctrl+ プラス記号	フォントを拡大
	A- または Ctrl+ マイナス記号	フォントを縮小
Alignment (整列)		
	Ctrl + F4	上揃え
	Ctrl + F6	上下中央揃え
	Ctrl + F8	下揃え
	Ctrl+F3	左揃え
	Ctrl+F5	水平中央揃え
	Ctrl+F7	右揃え
配置		

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	Shift + F9	最前面へ移動
	F9	最背面へ移動
	F6	時計回りに回転
	Shift + F6	反時計回りに回転
	F7	左右反転
	Shift + F7	上下反転
間隔		
	Ctrl + I	水平方向
	Ctrl + K	垂直方向
	Ctrl + F9	中心点
SmartSymbol		
	Shift + G	生成
	Shift+S	編集の開始
	Shift+M	管理
	Shift + E	編集の終了
	Shift + R	復元

アニメーション メニュー

グループ	キーボードショートカット	説明
Mode		
	Shift+W	選択
	Ctrl + A	アニメーション リンク
ﾘﾝｸ		
	Shift + X	切り取り
	Shift + C	コピー
	Shift + L	クリア
グラフィック		
	Shift + B	作成
	Shift + F5	中断

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

セル		
	F3	作成
	Shift + F3	中断
オブジェクト		
	Alt + F5	再形成
	F2	すべて選択
ビットマップ		
	Alt + F7	[Import]
	Alt + F8	元のサイズ
	Alt + F9	編集
	Ctrl+V	貼り付け
変更		
	Ctrl + E	タグ変数の変更
	Ctrl + L	文字列の変更

プロパティメニュー

グループ	キーボードショートカット	説明
グラフィック		
	Shift + O	元のサイズ
	Alt + B	編集
	Shift + D	ダイナミック サイズの反映
Symbol FactorySymbol Factory		
	Shift + V	クリップボードから貼り付け
	Shift + P	ﾌｧｲﾙからｲﾝﾎﾟｰﾄ
	Shift+Y	新規ｶﾃｺﾞﾘの作成
	Alt + J	選択した InTouch オブジェクトの追加
ｸﾗｲｱﾝﾄ		

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	Shift + N	今すぐ通知
	Shift + F	終了時に通知

表示メニュー

グループ	キーボードショートカット	説明
表示		
	Alt+Y	リストア
	Alt + W	ウィンドウ
	Alt + A	スーパータグ
	Alt + O	ツール
	Alt + I	スクリプト
	Alt + G	ｸﾞﾗﾌｨｯｸ
	Alt + E	[プロパティ]
	Alt + D	グリッドにスナップ
	Alt + R	ルーラー
	Alt+T	ステータス バー
更新		
	Alt + M	ｳｨﾝﾄﾞｳ ｻﾑﾈｲﾙ
アプリケーション		
	Alt + N	ﾌｫﾙﾀﾞ を開く
テーマ		
	Alt + L	ダーク モードとライト モードを切り替え

他

キーボードショートカット	説明
Shift+H	ヘルプ トピックを開く
Alt + U	ランタイムを開く
Ctrl + W	プロパティ グリッドを開く

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Shift + プラス記号	半径を拡大する（角丸長方形）
Shift + マイナス記号	半径を縮小する（角丸長方形）
Ctrl + H	水平方向のスペース

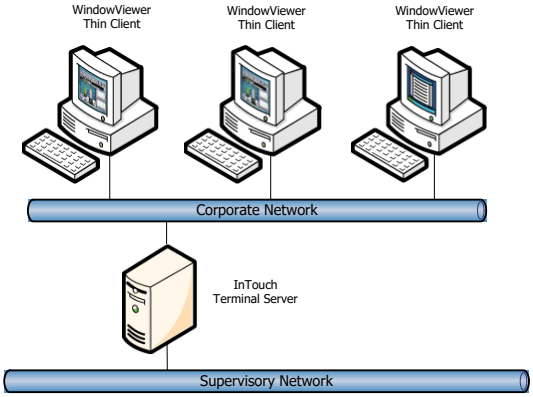
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Deploy

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 16 ターミナル サービスとリモート デスクトップ サービスの配置と操作

ターミナル サービスとは、Microsoft Windows Server オペレーティング システムに含まれている設定可能なサービスであり、サーバーを中心として Windows ベースのアプリケーションを実行できます。ターミナル サービスでは、InTouch ソフトウェア アプリケーションの複数のインスタンスが同時に実行されるサーバー ノードにクライアント コンピュータがアクセスします。



ターミナル サービスの環境には、次のような主な 3 つの部分があります。

- **ターミナル サービス サーバー**。このサーバーは、各クライアント セッションのコンピュータ処理用リソースを管理し、クライアント ユーザーに対し、各ユーザー自身の環境を提供します。サーバーは、リモート クライアントで実行されるキー入力、およびマウス操作をすべて処理し、オペレーティング システムとアプリケーションの両方の表示出力を適切なクライアントに送信します。ターミナル サービス アプリケーションのすべての処理がサーバーで発生します。
- **リモート デスクトップ プロトコル（RDP）**。リモート デスクトップ プロトコル（RDP）クライアント アプリケーションにより、キー入力、マウスの動きなどの入力データがサーバーに渡されます。
- **クライアント**。ターミナル サービス クライアントでは、ローカル アプリケーションの処理は実行されません。アプリケーション出力が表示されるだけです。Windows の［プログラム］メニューから［**ターミナル サービス クライアント**］コマンドを実行し、クライアントからターミナル サービスにアクセスします。ターミナル サーバーに接続すると、クライアント環境が Windows サーバーと同じように見えます。ただし、アプリケーション自体がローカルで実行されているわけではないということが分かります。

ターミナル サービスの機能および利点を含む詳細については、Microsoft のマニュアルを参照してください。

ターミナル サーバー アプリケーションの計画

重要 : アプリケーションを運用環境に配置する前に、テスト サーバーにインストールすることをお勧めします。

ターミナル サービスをインストールする前に、以下を行います。

- サーバーにインストールするクライアント アプリケーション（InTouch など）を決定する
- クライアントに対するハードウェアの必要条件を確認する

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- クライアントのサポートに必要なサーバー設定を決定する
- ターミナル サービスやその他のアプリケーションの実行に必要なライセンスを確認する
- ターミナル サービスの下で実行される InTouch アプリケーションのアラーム、セキュリティ、I/O、およびスクリプトなどを理解する

ターミナル サービス環境への InTouch アプリケーションの配置

InTouch アプリケーションをターミナル サービス環境に配置する場合、各ノードに個別の InTouch アプリケーションを配置してください。

ターミナル サービス環境のアラーム

分散アラーム システムを Terminal Services for InTouch と併用することで、異なるターミナル セッションで実行しているアラーム クライアント上でどのアラームをどのように表示するかを選択できます。

アラーム プロバイダは、アプリケーションに固有な名前とアプリケーションのインスタンスで識別できます。この情報は、アラーム プロバイダまたはアラーム コンシューマが分散アラーム システムに登録するときに提供されます。

アラーム プロバイダを実行しているノードは、システム内のコンピュータ ノードを識別する固有な名前で識別されます。この情報は、そのコンピュータ ノードでそのインスタンスが起動したときに分散アラーム システムに提供されます。

アラーム イベントがログ記録されると、ノードと完全なアラーム プロバイダの名前によってアラームのソースが識別されます。

ターミナル サービス環境でアラームが確認されると、記録されるオペレータ ノードは、オペレータによって使用されるターミナル サービス セッションが実行されているクライアント コンピュータの名前になります。コンピュータのノード名を取得できない場合、代わりにその IP アドレスが使用されます。

注意：アラーム プロバイダはターミナル セッションではサポートされていません。これらは、ターミナル コンソールでしかサポートされていませんので注意してください。

ターミナル サービス環境のセキュリティ

InTouch アプリケーション、IndustrialSQL Server、およびその他のシステムに関する機密情報を保護するには、アプリケーション セキュリティを使用します。

- アプリケーションを保護するには、$Operator システム タグ変数を使用します。これで、内部タグ変数に特定の関数をリンクすることにより、オペレータを制御してこれらの特定の関数にアクセスできるようになります。

  $Operator システム タグ変数の使用の詳細については、「InTouch のセキュリティ保護の概要」を参照してください。

- クライアント コンピュータを識別するには、GetNodeName() 関数を新しい TseGetClientId() 関数に置き換えます。ターミナル サービスを使用すると GetNodeName() 関数は、クライアント コンピュータ名ではなく、ターミナル サーバー名を返します。

侵入の試行を監視するには、セキュリティの監査を使用します。システム侵入のような攻撃を受けている疑いがある場合は、一連の監査可能なイベントをログできます。デフォルトでは、セキュリティ ログや監査は、通常は余分な処理能力を必要とするため、無効にされています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

警告：セキュリティの監査には大量のリソースを必要とします。InTouch アプリケーションのハードウェアの必要条件を正確に見積もるには、パイロット サーバーを評価するときに監査を有効にします。

ターミナル サービス環境の I/O

ターミナル サービス環境では、InTouch HMI は I/O Server を起動できません。表示セッションが開始される順序によっては、IOReinitialize() 関数を使用する必要があります。すべてのサーバー（I/O デバイスまたは表示アプリケーション）を実行してから、アプリケーションを起動してください。アプリケーションは、これらのサーバーから値を読み込みます。

WindowViewer の起動時に「I/O の初期化」のエラー メッセージが表示されないようにする

1. WindowMaker を開きます。
1. ［**ファイル**］メニューの［**設定**］を選択し、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
1. ［**環境設定**］タブの［**I/O**］セクションで［**ローカルのサーバーを起動**］チェック ボックスをオフにします。

ターミナル サービス環境でのスクリプトの実行

ターミナル サービスで実行されるすべてのアプリケーションでは、単一の基準タイミング（サーバー クロック）が使用されるため、CPU 負荷が異常に高い場合はスクリプトが実行されない場合があります。異常な CPU 負荷は、過度なビデオ処理を行う場合や複数のアプリケーションに同じスクリプトのトリガ（就労時間の終了イベントなど）が定義されている場合に発生する可能性があります。サーバーが多数のクライアントからのスクリプト処理で忙しくなり、タイマーで一定時間ごとに起動される別のクライアントのスクリプトが起動できなくなることがあります。これにより、スクリプトがクライアントで実行されなくなります。

スクリプトを適切に実行するには、複数のスクリプトを共通のトリガに結合し、それらをタグ変数サーバーのような単一のアプリケーションに移します。このような理由から、パイロット導入を行います。パイロットを配置すると、十分なハードウェアを選択しているかどうかを決定するための耐久テストを実行できます。

InTouch を実行するためのターミナル セッションへの適切なログオン

各セッションは一意のアカウントでログオンする必要があります。これは、手動で行うか、一意のログオンを強制するようターミナル サービスを設定することによって行うことができます。

注意：複数のセッションで同じログオン アカウントを使用すると、破損やその他の予期しない結果が生じることがあります。

ターミナル サービス環境のアラーム クエリの構文

セッションのアラームのアラーム クエリーの構文を以下に示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
\\ServerNodename\InTouch!$System
```

コンソール アラームのアラーム クエリーの構文の場合、ノード名の後にコロン（:）を含めます。次に例を示します。

```
\\ServerNodename\InTouch!$System
```

ターミナル サービス環境のその他の制限

以下の表では、ターミナル サーバーでアプリケーションを実行する際の制限および提案するソリューションを説明します。

機能	サポート	コメント
WindowViewerWindowViewer	可	WindowViewer は、ターミナル サービスでのサービスとしての実行はサポートされていません。
I/O デバイスまたは MS Office（例：Excel）への DDE	不可	タグ変数サーバーを使用します（コンソールまたは別のコンピュータ）。これには DDE QuickScript（WWExecute()、WWpoke()、WWRequest()）が含まれます
MS Office からの DDE（例：Excel で設定したホットリンク）	可	Excel と InTouch HMI は、同一のセッションで実行する必要があります。
履歴トレンド	可	値をログするには、タグ変数サーバーまたは NAD を使用します。複数のセッションが同じ履歴ファイルを読み取りますが、コンソールのみが履歴ファイルに書き込むことができます。
InTouch Alarm Logger	可	--
MEM OLE オートメーション	可	--
アラームの印刷	不可	--
変数値保持タグ変数	可	NAD を使用する必要があります。
SQL アクセス（ODBC）	可	データベースは、別のコンピュータ上に格納されている必要があります。
I/O デバイスまたは別の InTouch アプリケーションへの SuiteLink	可	別の表示セッションと通信する場合は、ターミナル サーバー ノード名を含め、必要なセッションの IP アドレスをアプリケーション名に追加する必要があります（たとえば、view10.103.25.6. I/O Servers はクライアント セッションではサポートされていません。

スクリプトを使用した InTouch クライアント セッションの情報の取得

ターミナル サービスでは、次の InTouch QuickScript 関数を使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- TseGetClientId() 関数
- TseGetClientNodeName() 関数
- TseQueryRunningOnConsole() 関数
- TseQueryRunningOnClient() 関数

TseGetClientId() 関数

WindowViewer アプリケーションがターミナル サーバー クライアントで稼動している場合は、クライアント ID（クライアントの TC/IP アドレス）の文字列を返します。この ID は内部で使用され、SuiteLink サーバー名およびロガー ファイル名が生成されます。そうでない場合、TseGetClientId() 関数は空の文字列を返します。

構文

```
MessageResult=TseGetClientId();
```

例

MsgTag タグ変数に、クライアント IP アドレス 10.103.202.1 が保存されます。

```
MsgTag=TseGetClientID();
```

TseGetClientNodeName() 関数

Windows によって識別可能な名前が割り当てられたターミナル サーバー クライアントで表示アプリケーションが実行されている場合、クライアントのノード名を返します。そうでない場合、TseGetClientNodeName() 関数は空の文字列を返します。

構文

```
MessageResult=TseGetClientNodeName();
```

例

MsgTag タグ変数に割り当てられた値として、クライアントのノード名が返されます。

```
MsgTag=TseGetClientNodeName();
```

TseQueryRunningOnConsole() 関数

表示アプリケーションがターミナル サービス コンソールで実行されているかどうかを示すために、スクリプトから TseQueryRunningOnConsole() 関数を実行できます。

構文

```
Result=TseQueryRunningOnConsole();
```

戻り値

WindowViewer アプリケーションがターミナル サービス コンソールで稼動している場合は、ゼロ（0）以外の整数値を返します。そうでない場合、TseQueryRunningOnConsole() 関数はゼロを返します。

例

WindowViewer がターミナル サービス コンソールで実行されている場合、IntTag は 1 に設定されます。

```
IntTag=TseQueryRunningOnConsole();
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

TseQueryRunningOnClient() 関数

WindowViewer アプリケーションがターミナル サービス クライアントで実行されている場合は、ゼロ（0）以外の整数値を返します。そうでない場合、ゼロを返します。

構文

```
Result=TseQueryRunningOnClient();
```

戻り値

WindowViewer がターミナル サーバー クライアントで実行されていない場合は、0 を返します。

例

WindowViewer がターミナル サービス クライアントで実行されている場合、IntTag は 1 に設定されます。

```
IntTag=TseQueryRunningOnClient;
```

リモート デスクトップ サービスの概要

リモート デスクトップ サービス（以前の名称はターミナル サービス）は、Windows Server® 2008 R2 以降のバージョンにおいて、リモート デスクトップ セッション ホスト（RD セッション ホスト）サーバー上にインストールされた Windows ベースのプログラム、または完全な Windows デスクトップにアクセスすることのできるサーバーの役割です。リモート デスクトップ サービスでは、ユーザーは、 企業ネットワーク内またはインターネットから RD セッション ホスト サーバーにアクセスできます。

ユーザーが RD セッション ホスト サーバー上のプログラムにアクセスすると、プログラムはサーバー上で実行します。各ユーザーには、自身のセッションだけが表示されます。セッションは、他のクライアント セッションから独立していて、サーバー オペレーティング システムによって等価的に管理されます。さらに、Hyper-V™ を使用するようにリモート デスクトップ サービスを構成して、仮装マシンをユーザーに割り当てることや、接続時に使用可能な仮想マシンをリモート デスクトップ サービスでユーザーに動的に割り当てることができます。

リモート デスクトップ サービスの詳細については、Windows Server 2008 R2 TechCenter（http://go.microsoft.com/fwlink/?LinkId=138055）のリモート デスクトップ サービスのページを参照してください。

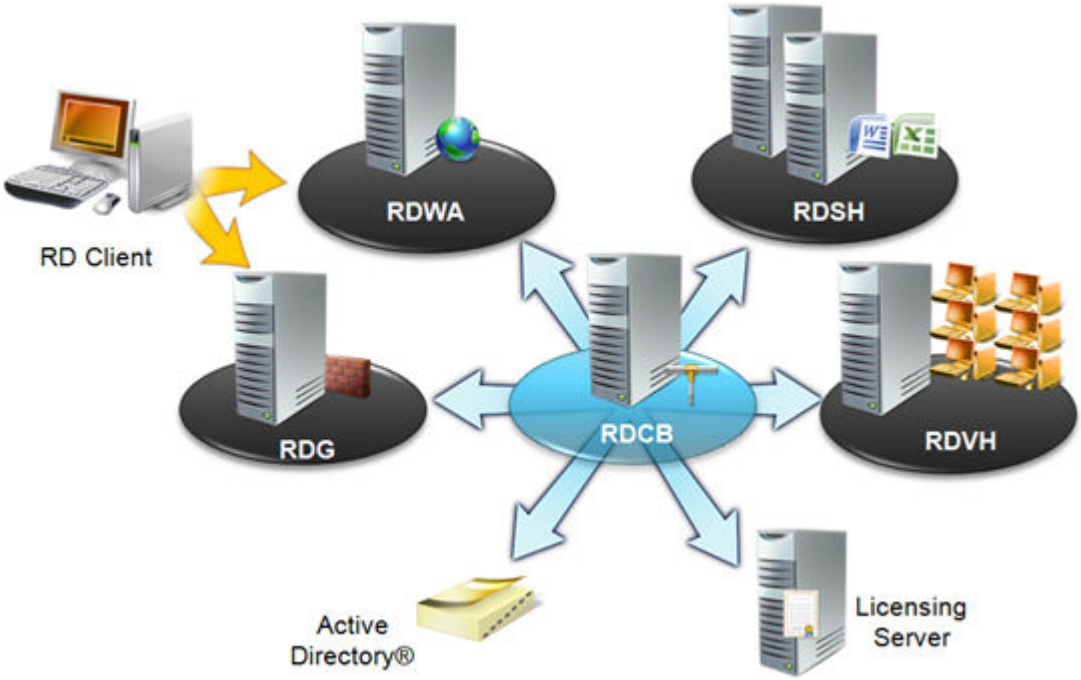
リモート デスクトップ サービスの役割サービス

リモート デスクトップ サービスは、いくつかの役割サービスで構成されるサーバーの役割です。Windows Server 2008 R2 以降のバージョンでは、リモート デスクトップ サービスは以下の役割サービスで構成されます。

- **RD セッション ホスト：** リモート デスクトップ セッション ホスト（RD セッション ホスト）（以前の名称はターミナル サーバー）では、サーバーで Windows ベースのプログラムまたは完全な Windows デスクトップをホストできます。ユーザーは、RD セッション ホスト サーバーに接続して、プログラムを実行することに加えて、ファイルの保存およびそのサーバー上のネットワーク リソースの使用を行うことができます。
- **RD Web アクセス**： リモート デスクトップ Web アクセス（RD Web アクセス）（以前の名称は TS Web アクセス）では、Windows 7 以降を実行するコンピュータの［スタート］メニューまたは Web ブラウザから RemoteApp およびデスクトップ接続にアクセスできます。RemoteApp およびデスクトップ接続は、RemoteApp プログラムおよび仮装デスクトップのカスタム ビューをユーザーに提供します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **RD ライセンス：** リモート デスクトップ ライセンス（RD ライセンス）（以前の名称は TS ライセンス）は、RD セッション ホスト サーバーに 接続する各デバイスまたはユーザーに必要なリモート デスクトップ サービス クライアント アクセス ライセンス（RDS（CAL）を管理します。RD ライセンスを使用して、 リモート デスクトップ ライセンス サーバーで RDS CAL のインストール、発行、および可用性の追跡を行います。
- **RD ゲートウェイ**： リモート デスクトップ ゲートウェイ（RD ゲートウェイ）（以前の名称は TS ゲートウェイ）では、承認済みのリモート ユーザーがインターネットに接続されたデバイスから社内ネットワーク上のリソースに接続できます。
- **RD 接続ブローカー：** リモート デスクトップ接続ブローカー（RD 接続ブローカー）（以前の名称は TS セッション ブローカー）は、負荷分散された RD セッション ホスト サーバー ファームでのセッション負荷分散とセッション再接続をサポートします。RD 接続ブローカーは、RemoteApp およびデスクトップ接続からの RemoteApp プログラムおよび仮装デスクトップへのユーザー アクセスも提供します。
- **RD 仮装化ホスト：** リモート デスクトップ仮想化ホスト（RD 仮想化ホスト）は Hyper-V と統合し、仮装マシンをホストして仮装デスクトップとしてユーザーに提供します。一意の仮装デスクトップを組織内の各ユーザーに割り当てることや、仮装デスクトップのプールへの共有アクセスを提供することができます。



リモート デスクトップ サービス（RDS）接続の保護

攻撃から保護するために、以下のセキュリティ対策を実施してください。

1. 強力なパスワードを使用する

   リモート デスクトップにアクセスするすべてのアカウントで強力なパスワードを使用します。

2. ソフトウェアを更新する

   Microsoft Update の自動実行を有効にして監査することによって、クライアント ソフトウェアとサーバー ソフトウェアの両方の最新バージョンを実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. アカウント ロックアウト ポリシーを設定する

   特定の回数のログイン試行が失敗した場合に一定期間アカウントをロックすることによって、"ブルート フォース" 攻撃を防止できます。

4. 2 要素認証を使用する

   RD ゲートウェイは、スマートカードを使用したした 2 要素認証をサポートします。

5. リモート デスクトップのライセンス ポートを変更する

   デフォルトのリモート デスクトップ ポート（TCP 3389）へのアクセスを試みるネットワーク攻撃およびワームを防止します。

6. RD ゲートウェイを使用する

   RD ゲートウェイは、リモート デスクトップ ポートへのアクセスを制限する一方で、単一の "ゲートウェイ" サーバーを介したリモート デスクトップ接続をサポートします。RD ゲートウェイ サーバーを使用する場合、デスクトップおよびワークステーション上のすべてのリモート デスクトップ サービスは RD ゲートウェイを介してルーティングされます。RD ゲートウェイ サーバーは HTTPS（ポート 443）でリモート デスクトップ リクエストをリスンし、クライアントをターゲット マシン上のリモート デスクトップ サービスに接続します。手順については、http://technet.microsoft.com/ja-jp/library/cc770601.aspx を参照してください。

7. リモート デスクトップ サービス接続にネットワーク レベルの認証を設定する

   ネットワーク レベルの認証では、ユーザーはセッションを作成する前に RD セッション ホスト サーバーに対して認証される必要があります。ネットワーク レベルの認証を使用すると、RD セッション ホスト サーバーの可用性が向上します（RD セッション ホスト サーバーの差ビス拒否攻撃のリスクが削減されます）。詳細については、https://technet.microsoft.com/ja-jp/library/hh831778.aspx を参照してください。

8. サーバー認証および暗号化レベルを設定する

   デフォルトでは、ターミナル サービス セッションでは、ネイティブのリモート デスクトップ プロトコル（RDP）暗号化が使用されます。しかし、RDP では、ターミナル サーバーの ID を検証する認証は提供されません。サーバー認証にトランスポート層セキュリティ（TLS）1.0 を使用してターミナル サービス セッションのセキュリティを強化して、ターミナル サーバー通信を暗号化できます。セキュリティを強化するには、RDS およびクライアント コンピュータを TLS に対して正しく構成する必要があります。デフォルトでは、クライアントとサーバーの間の RDS 接続が最高のセキュリティ レベル（128 ビット）で暗号化され、転送されるデータの整合性と機密性が保証されます。

Windows Server 2016 のリモート デスクトップ サービスのベスト プラクティス

Windows Server 2016 環境では、リモート デスクトップ サービスに対して以下のベスト プラクティスを実装できます。

- 多要素認証を使用する

  多要素認証で Active Directory のパワーを活用して高いセキュリティ保護を適用します。詳細については、Microsoft のドキュメント（https://docs.microsoft.com/ja-jp/windows-server/remote/remote-desktop-services/rds-plan-mfa）を参照してください。

- ユーザー プロファイル ディスク（UPD）でデータ ストレージを保護する

  ユーザー プロファイル ディスク（UPD）では、ユーザーは単一のコレクション内のユーザー データ、カスタマイズ、およびアプリケーション設定を使用できます。UPD はユーザーおよびコレクシ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ョン単位の VHD ファイルです。UPD は、ユーザーがサインインするときにユーザーのセッションにマウントされる一元的な共有に保存され、セッション中はローカルディスクとして扱われます。詳細については、Microsoft のドキュメント（https://docs.microsoft.com/ja-jp/windows-server/remote/remote-desktop-services/rds-plan-secure-data-storage）を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 17 アプリケーションの分散

通常、分散アプリケーションには中央開発ステーション、データの中央記憶装置、およびクライアント ステーションがあります。分散アプリケーションを構築および保持するには、InTouch ネットワーク アプリケーション開発（NAD）を使用します。NAD を使用すると、アプリケーションの開発に支障をきたすことなく、多くのクライアント ステーション上にそのアプリケーションのコピーを保持できます。アプリケーションの個々のコピーを使用すると、ビューアが冗長になります。アプリケーションが変更されると、クライアント ステーションに自動的に通知されます。単一のコンピュータ、クライアント ベースおよびサーバー ベースの InTouch アプリケーションを作成できます。

System Platform IDE を使用して InTouch アプリケーションを管理および配置することもできます。

サポートされている InTouch アーキテクチャ

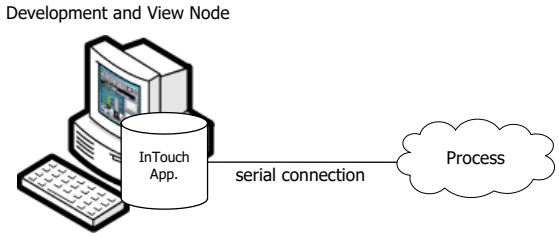
次の InTouch ネットワーク アーキテクチャがサポートされています。

- 単一のコンピュータ
- クライアント ベース
- サーバー ベース
- NADNAD

単一のコンピュータのアーキテクチャ

通常、単一のコンピュータ アプリケーションは、ネットワークに接続されていない、メインのオペレータ インターフェイスとして機能する 1 つのコンピュータから構成されています。このコンピュータは、シリアル ケーブルなどにより、生産プロセスに直接接続されています。

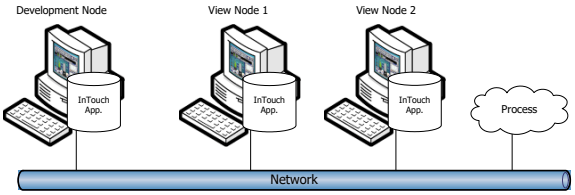
このアーキテクチャでは、単一のコンピュータで InTouch アプリケーションを開発します。アプリケーションを他のコンピュータにコピーしてから変更し、後で元のコンピュータにコピーを戻すことができます。



クライアント ベースのアーキテクチャ

クライアント ベースのアーキテクチャでは、WindowViewer が実行されている各コンピュータに対して（表示ノード）、またはネットワーク サーバー上の一意の場所に、1 つの InTouch アプリケーションの一意のコピーがあります。次の例では、アプリケーションが開発ノードで開発およびテストされ、次に各表示ノードにコピーされています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



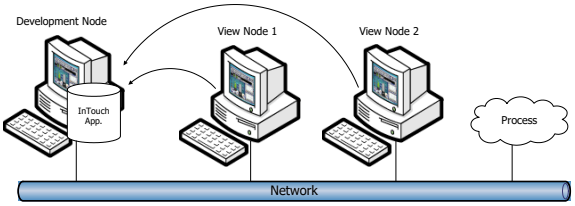
各ノード上にアプリケーションを重複して保持しているため本質的に冗長であり、使用できる表示ノードの数に制限はありません。

各表示ノードは、同じアプリケーションのコピーを持ち、I/O Servers や IndustrialSQL Server などのネットワーク データ ソースに対して同じアクセス権を持っている必要があります。ただし、各表示ノードが共有サーバーと個別に通信することになるので、ネットワークの負荷が増大します。

実行中のプロセスに影響を与えずに、開発ノードでアプリケーションを変更およびテストできます。ただし、変更されたアプリケーションを表示ノードに分散する必要があります。各表示ノードをローカルでシャットダウンし、新規アプリケーションをコピーし、次に再起動する必要があります。

サーバー ベースのアーキテクチャ

サーバー ベースのアーキテクチャにより、共通の InTouch アプリケーションが複数の表示ノードに分散されます。次の図では、2 つの表示ノードが、開発ノードから同じアプリケーションにアクセスしています。



各表示ノードの場合：

- 開発ノードの共有ネットワーク ドライブに論理ドライブをマップする必要があります。
- InTouch プログラムで、共有アプリケーションを登録する必要があります。
- コンピュータは、アプリケーションによって参照されるどのデータ ソースにも同じアクセス権を持っている必要があります。スクリプトの組み合わせを使用してノード名を識別し、その名前に基づいて各データの場所を変更することによって、データ ソースの場所を定義する方法もあります。

このアーキテクチャでは、単一のアプリケーションを保持します。アプリケーションが変更されると表示ノードが自動的に更新され、WindowViewer が再起動します。

このアーキテクチャの短所は次のとおりです。

- アプリケーションの開発が制限される
- 開発ノードがダウンした場合、冗長性がない
- すべてのノードを同じ画面解像度にする必要がある

ネットワーク アプリケーション開発（NAD）

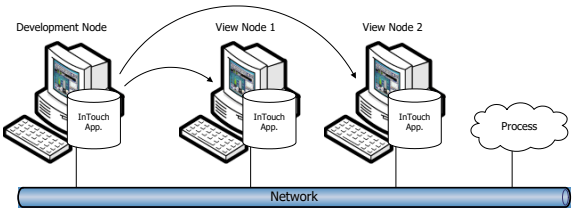
ネットワーク アプリケーション開発（NAD）アーキテクチャでは、中央のネットワーク ノードにアプリケーションの NAD リポジトリのコピーを保持します（通常は開発ノード）。各表示ノードによって、ユーザーが定義した場所にアプリケーションがコピーされ、実行されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

（WindowMaker の［システム］メニューの［クライアントに通知］コマンドを使用して）アプリケーションの変更がクライアントに通知されると、アプリケーション ディレクトリにフラグが設定され、表示ノードによって読み取られます。

アプリケーションの変更が表示ノードでどのように処理されるのかを設定できます。「変更を無視する」、「表示ノードを自動的に終了して再起動する（NAD リポジトリを再ロードする）」などを定義できます。

次の図では、2 つの表示ノードには開発ノードから登録した NAD リポジトリがありますが、実際には各コンピュータ上でローカルで実行されています。



注意：NAD リポジトリ ノードのアプリケーション ディレクトリに履歴データを書き込むようアプリケーションを設定した場合、すべての NAD ノードが履歴データを NAD リポジトリ アプリケーションに書き込もうとします。この問題を避けるには、各 NAD ノードで、NAD リポジトリ アプリケーション ノードではなくローカル ディレクトリに書き込まれるように履歴データを設定します。

大規模で複雑なアプリケーションを多くのノードに分散すると、最初のダウンロード時にシステムの応答時間が明らかに長くなります。ただし、更新は最適化されます。アプリケーションの転送は、低速ネットワークやシリアル接続では問題となる可能性があります。

また、特定のタイプのネットワーク トラフィックおよびアドレスを除去するルーターのユーザーなど、ネットワークに関するその他の制限に注意してください。

ネットワーク アプリケーションの計画

InTouch アプリケーションを構築する際に選択したアーキテクチャにかかわらず、次のことを考慮してください。

- I/O データ ソースへのアクセス
- 共有ファイルへのアクセス
- データがログ記録される場所
- 特殊なネットワークの必要条件

ネットワーク アプリケーションの I/O データ アクセス

InTouch HMI では、リアルタイムの I/O データのリファレンスとして、アクセス名を使用します。各アクセス名は、ノード名、アプリケーション、およびトピックから構成されている I/O アドレスと同等です。分散アプリケーションでは、I/O リファレンスをネットワーク I/O Server のグローバル アドレスとして、またはローカル I/O Server のローカル アドレスとして設定できます。

注意：InTouchView では、単一の Galaxy アクセス名のみ使用できます。InTouchView では他のアクセス名を作成できません。InTouchView の制限の詳細については、「ランタイム時のアプリケーションの表示の概要」を参照してください。

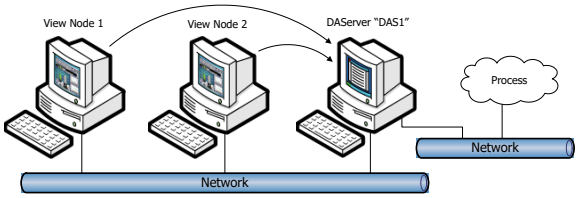
表示ノードは、データ ソースに対して開発ノードと同じアクセス権を持っている必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グローバル I/O アドレスの使用

I/O データのグローバル アドレスを使用すると、すべての表示ノードで、ネットワーク ベースの I/O サーバーを共有できます。これにより複数の I/O サーバーを持つ必要がなくなりますが、耐故障性が低下し、結果的にはパフォーマンスが低下します。

以下の図では、2 つの表示ノードで同じアプリケーションのコピーが実行されています。両方の表示ノードが同じ I/O データ ソースを参照しています。このアプリケーションは、完全にマッピングされた I/O アドレスを使用してデータ ソースを参照するので、すべてのリファレンスが同じ I/O サーバーを指すことになります。



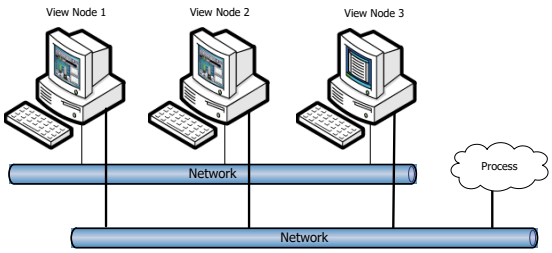
アクセス名の 3 つの部分からなるアドレス形式を使用して、別のノードに保存されているデータ要素を識別するように、InTouch アプリケーションを設定できます。アクセス名のアドレス形式には、リモート データがあるノード名、アプリケーション名、およびトピック名が含まれています。InTouch アプリケーションは、アクセス名とアイテム名を組み合わせて使用してリモート データを取得します。リモート I/O サーバーのアクセス名の定義の詳細については、『*AVEVA™ InTouch HMI アプリケーション開発*』の「I/O を使用したデータ アクセス」を参照してください。

注意：WindowMaker でアクセス名を作成するとき、アクセス名で SuiteLink プロトコルが使用されている場合、アクセス名は同じノード、アプリケーション、およびトピックにアクセスできません。ランタイム時にアクセス名を複製する際に、IOSetAccessName() 関数を使用してアクセス名にリダイレクトしないでください。リダイレクトされたアクセス名が機能しなくなります。

ローカル I/O アドレスの使用

各表示ノードに独自の I/O サーバーがある場合、I/O データに対するローカル アドレスが使用されます。このアーキテクチャでは、ネットワークで障害が発生しても各表示ノードが独立して動作し続けるため、耐故障性の高い操作を実現できます。

以下の図では、2 つの表示ノードで、独自の I/O データ ソースを参照する同じアプリケーションのコピーが実行されています。各アプリケーションはローカルの I/O アドレスを使用してデータ ソースを参照するため、各参照はローカルの I/O サーバーを指すことになります。



ローカルの I/O サーバーを使用すると、プロセス接続ネットワークの負荷が大幅に増加します。たとえば、3 つのノードの場合、各ノードのリクエストを別々に処理しなければならないため、トラフィック量は、1 つのノードの場合の 3 倍になります。

ローカル I/O サーバーのアクセス名の定義の詳細については、『*AVEVA™ InTouch HMI アプリケーション開発*』の「I/O を使用したデータ アクセス」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SuiteLink

SuiteLink 通信プロトコルは、TCP/IP プロトコルをベースにしています。高速の産業用アプリケーションでは、次のような機能を持つ SuiteLink を使用してください。

- タイムスタンプと品質インジケータがすべてのデータ値に関連付けられている VTQ（Value Time Quality）が、VTQ 対応のクライアントに送信されます。InTouch HMI は、VTQ インジケータと共にタグ変数データが送信される VTQ 対応のクライアントです。
- データのスループット、サーバーのロード、コンピュータ リソースの消費、およびネットワークの転送に対する包括的な診断を、Microsoft Windows オペレーティング システムのパフォーマンス モニタから行うことができます。
- アプリケーションが 1 つのノード上にある場合や、数多くのノードに分散されている場合にかかわらず、アプリケーション間で一貫した大容量のデータを維持できます。

SuiteLink は DDE、FastDDE、または NetDDE に取って代わるものではありません。クライアントとサーバー間の接続はそれぞれ、ご使用のネットワーク要件によって異なります。

共有ファイルへのアクセス

分散アプリケーションでは、ファイル参照を次のように設定できます。

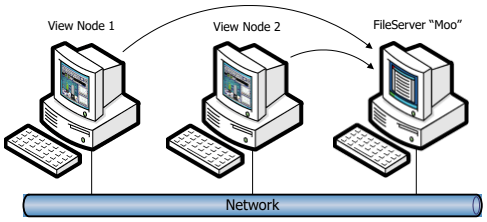
- ネットワーク ファイル サーバーに対するグローバル アドレス
- ローカル ファイルに対するローカル アドレス

表示ノードは、データ ソースに対して開発ノードと同じアクセス権を持っている必要があります。

ファイル データに対するグローバル アドレスの使用

ファイル データに対するグローバル アドレスを設定すると、共通のネットワーク ベースの一連のファイルをすべての表示ノードで共有できます。これによりファイルを集中管理できますが、ローカル コピーを持つ場合に比べると耐故障性が低下します。

次の図では、同じアプリケーションを実行している 2 つの表示ノードが同じレシピ ファイルを参照しています。各アプリケーションは、完全に規定どおりのネットワーク パスにマッピングされたドライブ文字を使用するため、すべての参照が同じファイルを指すことになります。



共有ファイルを設定する

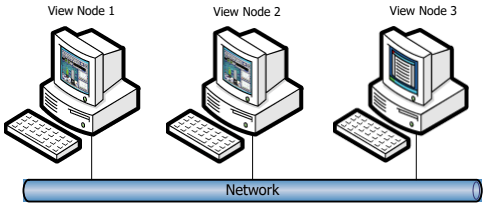
1. 参照ファイルを含む共有パスに、ネットワーク ドライブをマッピングします。たとえば、G:\Directory\Recipe.csv では、「G:\」は \\Moo\Share を参照するようにマッピングされたドライブ文字です。すべての表示ノードで、この同じドライブをマッピングする必要があります。
2. スクリプトでは、共有パスを参照します。以下に例を示します。

```
RecipeSelectRecipe("G:\Directory\Recipe.csv", "review", "RecipeName");
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ファイル データに対するローカル アドレスの使用

各表示ノードにファイルの独自のコピーがある場合、ファイル データに対するローカル アドレスを使用できます。次の図では、3 つの各表示ノードで同じアプリケーションのコピーが実行されており、レシピ ファイルのローカル コピーを参照しています。



この例では、ローカル アドレスは次のとおりです。

```
C:\Directory\Recipe.csv
```

「C:\」はローカル ドライブです。

スクリプトでは、ローカル パスを参照します。以下に例を示します。

```
RecipeSelectRecipe("C:\Directory\Recipe.csv", "review", "RecipeName");
```

このアーキテクチャは高い耐故障性を持っています。ただし、ファイルの変更をすべての表示ノードにコピーする必要があります。

すべてのファイル アクセスを「読み取り専用」に設定し、ローカル ファイルへの変更を禁止する必要があります。

UNC を経由した共有ファイルへのアクセス

通常、アプリケーション ディレクトリのエントリ、アイテムの設定、および分散アラームなどのためにファイル パスを入力する場合に、UNC（Universal Naming Convention）アドレスを使用できます。UNC 名を使用した場合、マッピングされたドライブを作成する必要はありません。

UNC アドレスの形式は「\\Node\Share\Path」です。各要素の意味を以下に示します。

- 「Node」は、ファイル共有が含まれているコンピュータの名前です。
- 「Share」は、そのコンピュータ上の共有フォルダに割り当てられている論理名です。
- 「Path」は共有に対する通常のパスです。

注意：SuiteLink を使用する場合、ノード名は 15 文字に制限されています。

UNC を使用してファイルにアクセスするには、アクセス先のコンピュータ上にファイル共有を作成する必要があります。詳細については、Windows のマニュアルを参照してください。

たとえば、「EngineRm」というネットワーク名のコンピュータがあり、共有名「Root」とルート ドライブ「C:\」を共有しているとします。「C:\IT\Apps\Boiler」アプリケーションの UNC パスを設定するには、次の UNC を使用する必要があります。

\\EngineRm\Root\IT\Apps\Boiler

「Boiler」ディレクトリ自体が「Boiler」として共有されている場合、UNC は以下のように省略できます。

\\EnginerRm\Boiler

PATH 環境変数で指定したパスに共有がある場合、パスは必要ありません。

注意：UNC アドレスで参照されるファイルに書き込む必要がある場合、ローカル ノードのときでもその共有は読み書き共有でなければなりません。UNC を使用してパスワード保護された共有にアクセスす

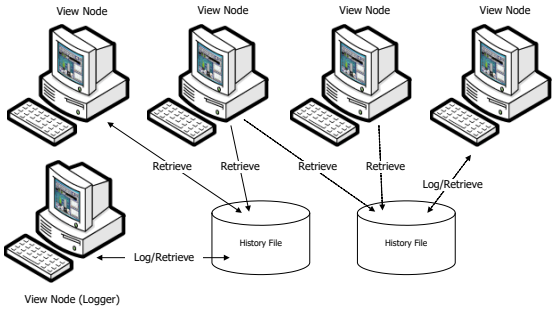
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

るには、まずネットワーク ドライブのマッピングを設定しなければなりません。Windows エクスプローラを使用すれば、リモート ノードからドライブのマッピングを設定できます。

分散環境でのデータのログ

InTouch 分散履歴システムを使用すると、ネットワーク上のどの InTouch アプリケーションからでも履歴データを取得できます。また、このシステムでは、複数の履歴データベースから同時にリモートでデータを取得することもできます。これらのデータベースは、履歴プロバイダと呼ばれます。

分散履歴ファイルには 1 つの InTouch ノードのみがロギングできます。ただし、ファイルのコンテンツを表示できる InTouch ノードの数には制限はありません。



履歴ファイルのデータを検索するリモート ノードから、最新時刻のデータ（ロギング ノードの時間）を参照できないことがあります。リモート トレンド グラフはロギング ノード ディスクに書き込まれたデータしか表示できません。

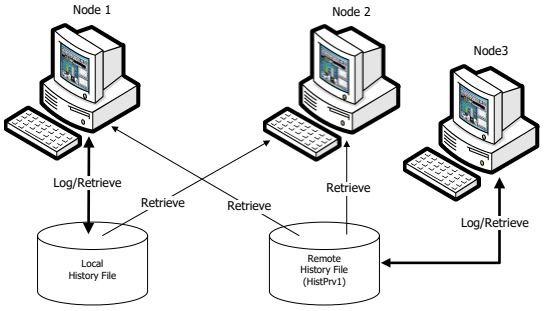
「データログ」用に指定された各タグ変数のデータは、そのタグのサンプルが 22 個集められたときに自動的にディスクに書き込まれます。HTUpdateToCurrentTime() 関数が呼び出された場合は、集められたサンプルの数に関係なく、データがディスクに書き込まれます。デフォルトでは、1 時間に 1 回データがディスクに書き込まれます。この間隔を変更するには、INTOUCH.ini ファイルに次の行を追加します。

```
ForceLogging=X
```

X は分単位の時間で、5 ～ 120 の範囲で設定することができます。

注意：分散履歴システムを使用するときは、NetDDE Helper サービスが実行されていなければなりません。

以下の図は、ネットワーク アプリケーション開発（NAD）を使用してアプリケーションを分散する一般的な分散履歴システムの設定を示しています。



ノード 1 とノード 2 には、同じ InTouch アプリケーションのコピーが存在します。どちらのノードもローカルまたはリモートの履歴データ ファイルから検索できますが、ノード 1 のアプリケーションだけがローカルの履歴データ ファイルにログを出力するよう設定されています。ノード 3 も、同じリモートの履歴データ ファイルに対してログの出力と検索ができます。履歴プロバイダであるノード 3 には、

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

HistPrv1 という名前が割り当てられています。ノード 1 は開発とランタイムの両ステーションを兼ねていますが、ノード 2 はランタイム専用ステーションです。

このタイプのアプリケーションを作成するには、以下を実行します。

1. 履歴プロバイダ リストを作成します。「リモート履歴プロバイダの設定」を参照してください。
2. 履歴トレンド オブジェクトを作成および設定します。詳細については、『*AVEVA™ InTouch HMI アプリケーション開発*』の「タグ データのトレンド」を参照してください。
3. 分散ログ出力用アプリケーションを設定します。「分散履歴ログの設定」を参照してください。
4. アプリケーションを配布します。「NAD 用 InTouch アプリケーションの設定」を参照してください。

アプリケーションは、手作業または NAD（Network Application Development）を使用して配布できます。アプリケーションを配布すると、アプリケーションの一部である履歴プロバイダ リスト ファイルも自動的に配布されます。

アプリケーションをいったん配布すると、表示ノードからローカルのタグ変数とリモートの履歴プロバイダのタグ変数を検索できます。アプリケーションはすべての表示ノード上で動作しますが、履歴ログ ファイルにログを出力できるのはロギング ノードだけで、他のノードは履歴ログ ファイルの読み取りしかできません。

リモート履歴プロバイダの設定

InTouch HMI と共に使用する各リモート履歴プロバイダの名前およびネットワーク ノードを指定する必要があります。リモートの InTouch 履歴プロバイダまたはリモートの IndustrialSQL Server 履歴プロバイダのどちらかを使用できます。

注意：InTouchView アプリケーションでは、リモート履歴プロバイダを設定できません。InTouchView アプリケーションの制限事項の詳細については、「InTouchView アプリケーション」を参照してください。

ローカルの InTouch アプリケーションは履歴プロバイダと見なされるため、アプリケーションで定義する必要はありません。

アプリケーションで未定義の履歴プロバイダを参照する場合、参照が WindowViewer によって無視され、エラー メッセージがロガーに書き込まれます。

HistData ユーティリティは、Historian プロバイダから履歴情報を取得できません。

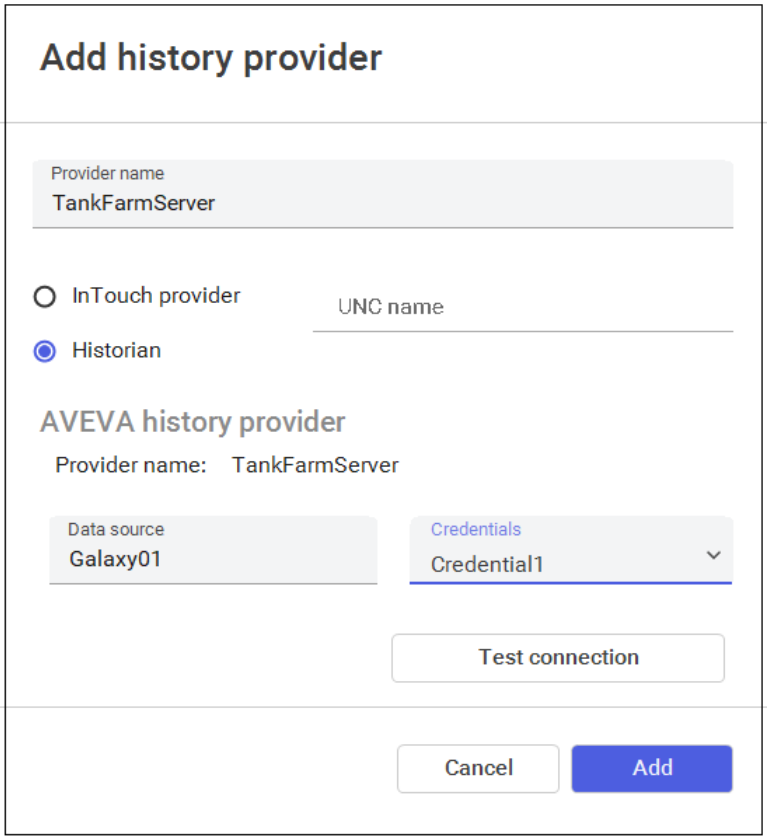
履歴プロバイダを設定する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］を選択し、［**分散名前マネージャ**］を選択します。

   ［**分散アラーム**］のタブおよび［**分散履歴**］のタブを含む［分散名前マネージャ］設定画面が表示されます。
3. ［**分散履歴**］タブで［**+**］アイコンを選択して履歴プロバイダを追加します。または、Alt+A を押します。

   ［**履歴プロバイダの追加**］ダイアログが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**プロバイダ名**］ボックスに、新しい履歴プロバイダの名前を入力します。

   プロバイダ名には、16 文字以下の英数字を使用できます。

5. InTouch 履歴プロバイダを設定します。

   a. ［**InTouch プロバイダ**］を選択します。

   b. ［**UNC 名**］ボックスに InTouch アプリケーション ディレクトリへの UNC パスを入力して、［**追加**］を選択します。

      UNC パスの形式は以下のとおりです。

      ```
      \\node_name\volume_name\directory\
      ```

      UNC の場所はパスワードによって保護されているため、Windows エクスプローラを使用して、まずノード接続を確立する必要があります。

6. AVEVA 履歴プロバイダを設定します。

   a. ［**Historian**］を選択します。

   b. ［**データ ソース**］ボックスに、Historian Server がインストールされているサーバーのノード名を入力します。

   c. ［**資格情報**］ドロップダウン リストから、認証に使用する資格情報を選択します。

注：スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「資格情報マネージャの操作」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

7. ［接続テスト］を選択して、Historian Server データベースへの接続を確認します。データベースへの接続が成功したかどうかを通知するメッセージが表示されます。
8. ［追加］を選択して、ダイアログ ボックスを閉じます。

   ［履歴プロバイダ］リストに Historian Server ノードが表示されます。

リモート履歴プロバイダの動的な設定

ランタイム中に履歴トレンド グラフのリモート履歴プロバイダを動的に設定することもできます。これはスクリプトで HTSetPenName 関数を使用して、リモート履歴プロバイダのタグ変数の参照を指定します。以下に例を示します。

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

ここで、「1」は指定したリモート履歴プロバイダ タグ変数を表示するトレンド ペンの番号を表します。

ランタイム中の［**履歴トレンドの設定**］ダイアログ ボックスと Pen ドットフィールドは、リモート履歴プロバイダに対してはサポートされていません。

分散履歴ログの設定

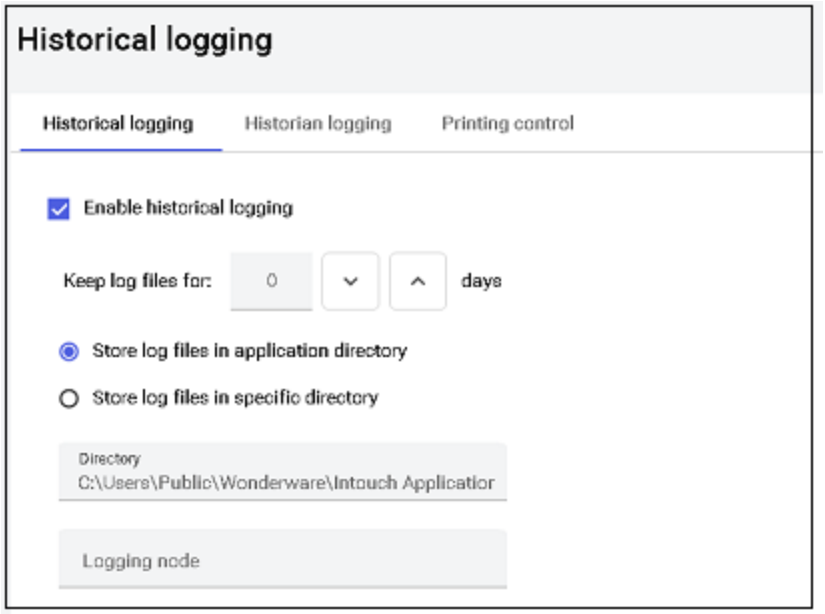
履歴ファイルには 1 つの InTouch ノードのみをログ記録できます。ただし、複数の InTouch ノードにファイルを表示できます。

注意：InTouchView アプリケーションでは、履歴ログを設定できません。InTouchView アプリケーションの制限事項の詳細については、「InTouchView アプリケーション」を参照してください。

分散履歴ログの設定

1. ［**ファイル**］メニューの［**設定**］をクリックし、［**履歴ログ**］をクリックします。

   ［**履歴ログ**］設定画面が表示されます。



2. ［**履歴ログ実行**］チェック ボックスをオンにし、グローバル タグのログ機能を有効にします。
3. ［**他のディレクトリにログ ファイルを保存**］をオンにし、この入力ボックスにログ ファイルの保存場所のパスを入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ここでは有効な UNC（汎用名前付け規則）パスを入力する必要があります。例: `\\Node\Share\Path`

NAD を使用している場合、アプリケーション フォルダ以外のフォルダをパスがポイントするようにしてください。

4. ［**ロギング ノード名**］ボックスに、履歴ログ ファイルに記録を行うノード名を入力します。

   ここで指定したノードだけが、ログ ファイルに記録されます。

5. ［**OK**］ を選択します。

注意：［**履歴ログ実行**］オプションを選択したアプリケーションを WindowViewer ノードに分散すると、ノードはこのオプションをチェックしてログ ファイルへの記録を行うかどうかを判断します。［**履歴ログ実行**］を選択した場合、2 つの設定（［**フィールドがノードの名前に等しい場合 – 履歴ログを実行する**］、［**フィールドがノードの名前に等しくない場合 – 履歴ログを実行しない**］）が使用可能です。

特殊なネットワークの考慮事項

低速なネットワークで作業をしていて、InTouch HMI の起動または情報の保存に長時間かかる場合、NAD クライアントの win.ini の設定を次のように変更します。

```
ViewNadClearNADCopyDirectory=0
ViewNADCopyApplicationOnStartup=1
ViewNADOnApplicationChanged=3（または 4）
ViewNADThreadPriority=2
```

ViewNADOnApplicationChanged パラメータでは、3 の設定は、InTouch アプリケーション マネージャの［**ノードの設定**］ダイアログ ボックスの［**WindowViewer に変更をロード**］オプションに対応します。4 の設定は、［**WindowViewer へ変更ロード時にメッセージ表示**］オプションに対応します。これらの設定により、NAD のダウンロードが並行して発生している間でも、アプリケーションは個別の実行スレッドで実行を続行できます。

NAD がアプリケーションへの更新を実行している場合は、NAD リポジトリから変更されたファイルのみがコピーされます。NAD は、ランタイム言語の切り替えのために、開発時の SmartSymbol のディクショナリ ファイルをコピーすることはしません。

NAD 用 InTouch アプリケーションの設定

ネットワーク アプリケーション開発（NAD）は、クライアント ベースとサーバー ベースのアーキテクチャの利点を組み合わせたものです。NAD を使用すれば、アプリケーションの変更を自動的に通知し、更新されたアプリケーションを表示ノードに自動的に分散できます。

NAD 用にアプリケーションを設定する場合、WindowViewer による NAD リポジトリ アプリケーションのコピー先のフォルダを指定する必要があります。

- これが開発ノードの場合は、ローカル フォルダ パス（c:\InTouch\NAD など）を入力することができます。また、ネットワーク上のリモート UNC パス（\\node\share\path など）を入力することもできます。これは、ファイルの大半が中央に保存されるファイル サーバー ベースのネットワークの場合に便利です。
- これがクライアント ノード（ランタイムのみ）である場合、通常ローカル フォルダ パスを使用します。

WindowViewer の動作がネットワークの遅延や障害の影響を受けないように、できるだけローカル フォルダを指定することが推奨されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

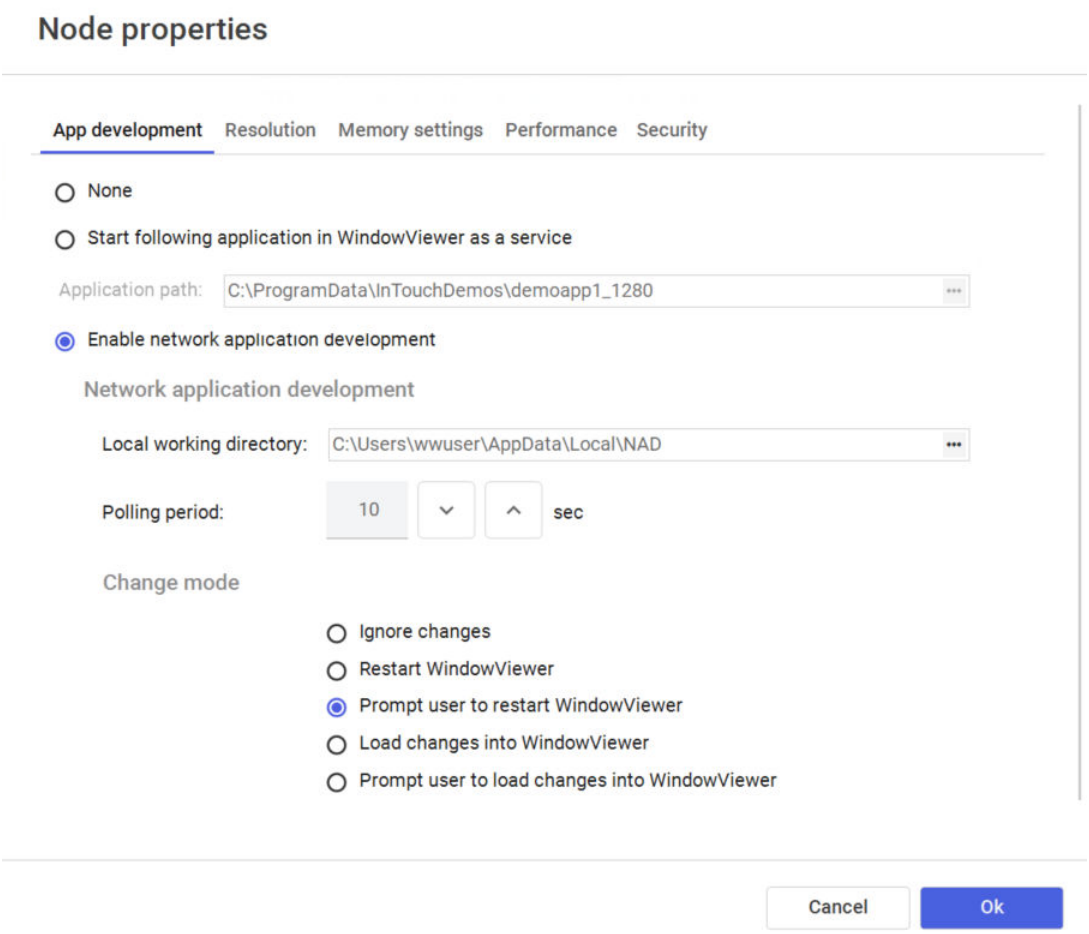
警告：ルート フォルダまたはルート フォルダを指す UNC パス名は指定しないでください。表示ノードは、NAD リポジトリ アプリケーション ディレクトリをコピーする前に、指定されたコピー先アプリケーション フォルダの下にあるファイルとサブフォルダをすべて削除してしまいます。そのため、NAD リポジトリ アプリケーション フォルダのパス、または NAD リポジトリ アプリケーション フォルダへの UNC は決して使用しないでください。

フォルダを指定しない場合、WindowViewer が起動するフォルダ内に、NAD という名前のローカル サブフォルダが WindowViewer によって自動的に作成されます。NAD フォルダは、一時フォルダと見なされます。NAD 自体によってコピーされたもの以外のフィルはこのディレクトリに保存しないでください。

NAD 用アプリケーションを設定する

1. アプリケーション マネージャを起動します。
2. ［ツール］タブで、［ノードのプロパティ］を選択します。

   ［ノードのプロパティ］画面が表示されます。



3. ［ネットワーク アプリケーション開発をオン］ラジオ ボタンを選択します。
4. ［ローカル ディレクトリ］ボックスに、WindowViewer が NAD リポジトリ アプリケーションをコピーするフォルダのパスを入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［**ポーリング周期（秒）**］ボックスに、表示ノードが開発ノードの更新をチェックする間隔を秒単位で入力します。
   - ここでは、小さすぎる値を設定しないようにしてください。WindowViewer が NAD リポジトリ アプリケーションの変更を必要以上に頻繁にチェックすると、実行中のアプリケーションへのサービスが妨げられる可能性があります。
6. ［**モードの変更**］領域で、NAD リポジトリ アプリケーションが変更された場合の WindowViewer での動作を決定するオプションを選択します。
   - 開発ノードで行われた変更を WindowViewer ノードが無視するように設定するには、［**変更を無視**］を選択します。
   - 更新された NAD リポジトリ アプリケーションを WindowViewer ノードにコピーし（そのように設定されている場合）、次に再起動するように設定するには、［**WindowViewer の再起動**］を選択します。
   - アプリケーションが変更されたというメッセージをオペレータに表示するには、［**WindowViewer の再起動時にメッセージ表示**］を選択します。オペレータは、更新されたアプリケーションで WindowViewer を再起動するのか、または現在のアプリケーションの使用を続行するのかを選択できます。
   - 開発ノードで行われた変更を WindowViewer に動的にロードするには、［**WindowViewer に変更をロード**］を選択します。大量の更新が行われた場合には、ロードする速度が遅くなることがあります。

   注意：アプリケーションの変更がマイナーで少ない場合のみ、［**WindowViewer に変更をロード**］オプションを使用することが推奨されます。マイナーな変更の例には、既存のウィンドウ内の変更、グラフィック ツールバー要素のサイズ変更、新しいグラフィック ツールバー要素の追加、および参照の変更などがあります。WindowViewer の再起動が必要な変更（新しいタグの追加、新しいウィンドウの追加、設定の変更など）を行った場合、または再起動が必要かどうかがわからない場合は、いずれかの再起動オプションを使用してください。

   - アプリケーションが変更されたというメッセージをオペレータに表示するには、［**WindowViewer へ変更ロード時にメッセージ表示**］を選択します。変更をロードするよう、オペレータにメッセージが表示されます。
7. ［**OK**］ を選択します。

自動 NAD 更新の実行

アプリケーション開発中に、自動 NAD 更新を起動できます。

［**クライアントに通知**］コマンドを実行すると、フラグが NAD リポジトリ アプリケーションで変更したものをすべてのリモート表示ノードに通知するように設定されます。クライアントは、各ノードに対して定義された［**モードの変更**］オプションに基づいて、更新プロセスを自動的に開始できます。

スタンドアロン アプリケーション（産業用グラフィック埋め込み済み）を表示ノードで初めて開くとき、グラフィックが表示されず、エラーがロガーに記録されることがあります。この問題を回避するには、NAD リポジトリ ノードから［**クライアントに通知**］コマンドを実行します。この操作により、産業用グラフィックが［**変更モード**］オプションに基づいて表示ノードに読み込まれます。

自動更新を実行する

1. WindowMaker でアプリケーションを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［プロパティ］メニューの［クライアント］グループで次の操作を行います。
   a. クライアントに直ちに通知を行うには、［今すぐ通知］を選択します。
   b. WindowMaker を終了するときに NAD クライアントに通知を送信するには、［終了時に通知］を選択します。

注意：［終了時に通知］オプションを選択すると、WindowMaker が閉じるたびに、最後の通知以降に行われた変更がないかが検証されます。変更が検出された場合、NAD クライアントに通知するかどうかを確認するメッセージを含むダイアログ ボックスが表示されます。クライアントに通知する場合は、［はい］を選択します。変更を無視する場合は、［いいえ］を選択します。

手動による NAD 更新の実行

稼働中の表示ノード上でオペレータが手動で NAD 更新を開始できるようにするスクリプトを記述できます。

NAD を持つアプリケーションを手動で更新するには、［ノードのプロパティ］ダイアログ ボックスで［モードの変更］オプションを［変更を無視］に設定する必要があります。詳細については、「NAD 用 InTouch アプリケーションの設定」を参照してください。

NAD 更新を手動で実行するには、スクリプトで次のシステム タグ変数および関数を使用します。

- $ApplicationChanged システム タグ変数
- $ApplicationVersion システム タグ変数
- RestartWindowViewer() 関数
- ReloadWindowViewer() 関数

$ApplicationChanged システム タグ変数

ネットワーク アプリケーション開発（NAD）アーキテクチャで、NAD リポジトリ アプリケーションに変化があったことを通知します。

カテゴリ

アプリケーションに変換するには

使用法

```
$ApplicationChanged
```

備考

このシステム タグ変数は、WindowMaker の［システム］メニューにある［クライアントに通知］コマンドの選択により、更新シグナルが発生するたびに 1 に変化します。アプリケーションが更新されると、$ApplicationChanged が 0 にリセットされます。NAD リポジトリ アプリケーションが変更されたことをオペレータに通知するメッセージを生成するために、このタグ変数を使用できます。

また、$ApplicationChanged システム タグ変数をデータ変更スクリプト内で使用すれば、ノード更新通知スクリプトを作成できます。このスクリプトにより、独自のダイアログ ボックスを起動したり、実行中のプロセスを停止できます。その後、ReloadWindowViewer() 関数を使用して更新プロセスを開始できます。

データ型

論理型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

データ変化スクリプトのタグ変数ボックスで次のステートメントを使用すると、スクリプトの本文が実行されます。変化を有効にするには WindowViewer の再起動が必要ですが、スクリプトの本文を使って、この通知をユーザーへウィンドウで表示することもできます。

```
$ApplicationChanged
```

参照項目

$ApplicationVersion

$ApplicationVersion システム タグ変数

アプリケーションの現在のバージョン番号です。この番号は、保存したり元に戻すことができるすべての変更により変化します。

カテゴリ

アプリケーションに変換するには

使用法

```
$ApplicationVersion
```

備考

$ApplicationVersion システム タグ変数に関連する値は、InTouch アプリケーションの現在のバージョンに設定されます。保存したり元に戻すことができるアプリケーションに対するすべての変更により、バージョンが変化します。NAD リポジトリ アプリケーションが変更されたことをオペレータに通知するメッセージを生成するために、このタグ変数を使用できます。

データ型

実数型（読み取り専用）

例

アナログ データ表示リンクに使用すると、このシステム タグ変数により、WindowViewer で実行中のアプリケーションの現在のバージョン番号が表示されます。

```
$ApplicationVersion
```

参照項目

$ApplicationChanged

RestartWindowViewer() 関数

WindowViewer をシャットダウンし、更新された NAD リポジトリ アプリケーションをコピーし（そのように設定されている場合）、次に WindowViewer を再起動します。

カテゴリ

システム

構文

```
RestartWindowViewer();
```

備考

自動更新のネットワーク アプリケーション開発（NAD）機能がオフの場合、アプリケーションの更新にこの関数を使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

NAD 更新がいつ発生するのかを決定するには、$ApplicationChanged システム タグ変数を使用します。

NAD 更新を開始するには、［クライアントに通知］コマンドを使用します。ただし、オペレータが更新を遅らせ、後で実行したい場合もあります。ボタンの動作スクリプトと共にこの関数を使用すると、オペレータの都合がよいときに WindowViewer を再起動するようにできます。

WindowViewer をシャットダウンせずに表示ノードを更新するには、ReloadWindowViewer() 関数を使用できます。

参照項目

$ApplicationChanged、ReloadWindowViewer()

ReloadWindowViewer() 関数

更新された NAD リポジトリ アプリケーションを使用して、サービスを中断せずに動的に WindowViewer を更新します。

カテゴリ

システム

構文

```
ReloadWindowViewer();
```

この関数は、ユーザーが WindowViewer の再ロードを管理することを可能にします。

備考

自動更新のネットワーク アプリケーション開発（NAD）機能がオフの場合、アプリケーションの更新にこの関数を使用します。

NAD 更新がいつ発生するのかを決定するには、$ApplicationChanged システム タグ変数を使用します。

NAD 更新を開始するには、［クライアントに通知］コマンドを使用します。ただし、オペレータが更新を遅らせ、後で実行したい場合もあります。ボタンの動作スクリプトと共にこの関数を使用すると、オペレータの都合がよいときに WindowViewer でアプリケーションを再ロードするようにできます。

参照項目

$ApplicationChanged

アプリケーション編集のロック

複数の開発者が 1 つのアプリケーションを同時に編集しないように、WindowMaker は編集作業中にアプリケーションをロックします。ロックされたアプリケーションを開こうとすると、エラー メッセージが表示されます。メッセージには、アプリケーションを編集しているノード名が含まれています。

アプリケーションがロードされている状態で WindowMaker が異常終了すると、appedit.lok ファイルが削除されない場合があります。appedit.lok ファイルをアプリケーション ディレクトリから削除することにより、ロックを手動解除することができます。

NAD 更新中のアプリケーションの変更

WindowViewer ノードはアプリケーションを更新するとき、NAD リポジトリ アプリケーションの属性（読み取り専用、システム、隠しファイルなど）を維持しようとします。

また、WindowViewer は、すべての NAD リポジトリ アプリケーションのファイルおよびサブフォルダをコピーします。ただし、*.WVW、*.DAT、*.LGH、*.IDX、*.LOG、*.LOK、*.FSM、*.STG、*.DBK、*.CBK、

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.HBK、.KBK、*.LBK、*.NBK、*.OBK、*.TBK、*.WBK、*.XBK、*.$$$、RETENTIV.X、RETENTIV.D、RETENTIV.A、RETENTIV.S、RETENTIV.H、RETENTIV.T、SSD_、WM.INI、DB.INI、LINKDEFS.INI、TBOX.INI、GROUP.DEF、および ITOCX.CFG は除きます。

注意：WindowViewer は、ランタイム言語の切り替えに必要なものを除く、コピー先のアプリケーション内のすべてのファイルおよびサブフォルダを削除します。このフォルダは、一時フォルダと見なされます。その他のファイルはその場所に配置することはできません。

NAD クライアントは、クライアント アプリケーション ディレクトリに表示される、ファイルおよびサブディレクトリのローカル リストを作成することにより、更新を開始します。NAD リポジトリ ファイルのリストの更新を探すとき、NAD クライアントは、ローカル リストの各 NAD リポジトリ ファイルに対応するクライアント ファイルを削除します。ローカル リストの残りのエントリは、アプリケーションから削除される必要がある、使われていないファイルおよびサブディレクトリです。

すべてのダウンロードされたファイルは、NAD_Temp と呼ばれる一時サブディレクトリにコピーされます。新しい、更新されたすべてのファイルが再試行制限内で正常にコピーされた場合にのみ、ファイルは、NAD_Temp からアプリケーション ディレクトリにコピーされます。NAD クライアントが更新を中止する必要がある場合、新しい、更新されたファイルを部分的にコピーしたことによって、実行中のアプリケーションが壊れることはありません。

すべての新しい、更新されたファイルがダウンロードされた後、NAD リポジトリとの連携が失敗した場合でも、NAD_Temp の更新をコピーし、使われていないファイルを削除することにより更新を完了できます。これにより、接続が失われたことによって、NAD リポジトリ アプリケーション上でファイルの存在を確認できないという理由で、ファイルが消去されるのを防ぐことができます。

NAD は、アプリケーションのダウンロード中に、NAD リポジトリ アプリケーションに追加の変更が行われたかどうかを検出できます。そのような状況が発生した場合、NAD はアプリケーションのダウンロードを中止します。最後の更新後に［**クライアントに通知**］コマンドを実行すると、NAD は、次のポーリング周期に最新のアプリケーション ファイルのダウンロードを自動的に開始します。そうでない場合、アプリケーションのダウンロードが発生する前に、次の［**クライアントに通知**］コマンドが発行されるまで待ちます。

ランタイム時のアプリケーションの解像度のスケーリング

ダイナミック レゾリューション機能（DRC）を使用すると、作成した分散アプリケーションを異なる画面解像度で実行できます。

各表示ノードでは、解像度のカスタマイズを含め、アプリケーションを適切にスケーリングできます。このスケーリングは、WindowViewer がアプリケーションをコンパイルするときに行われるため、WindowMaker は必要ありません。各表示ノードはそれぞれ異なる DRC 設定を使用できるため、それぞれのノードで独自の設定を行う必要があります。

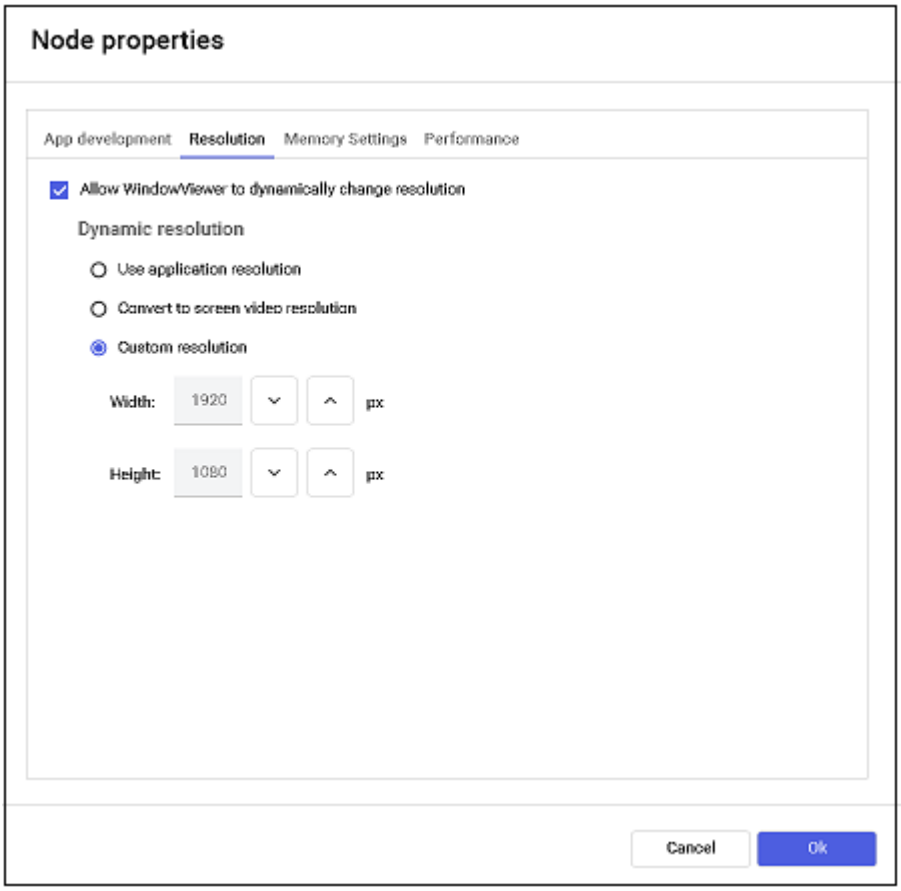
警告：DRC を使用してアプリケーションをスケーリングしない場合、WindowViewer は、ノードの画面解像度がアプリケーション開発ノードの画面解像度と同じ場合のみ、アプリケーションを実行します。2 つの画面解像度が異なる場合、WindowViewer はオペレータに WindowMaker を実行してノードの解像度に合わせてアプリケーションを変換するよう指示します。UNC パスを NAD リポジトリ アプリケーション ディレクトリに設定している場合、元のアプリケーションのみが変更されるため、上記の操作には注意が必要です。

DRC 用アプリケーションを設定する

1. アプリケーション マネージャを起動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**ツール**］メニューで、［**ノードのプロパティ**］を選択します。［ノードのプロパティ］ダイアログ ボックスが表示されます。
3. ［**解像度**］タブを選択します。



4. WindowViewer で NAD リポジトリ アプリケーションをローカルでスケーリングするには、［**WindowViewer で動的に解像度を変更**］チェック ボックスをオンにします。
5. ［**解像度**］領域で、以下のいずれかを選択します。
   - WindowViewer がアプリケーションを開発されたときの解像度で実行し、ノードの解像度を無視するようにするには、［**アプリケーションの解像度を使用**］を選択します。たとえば、アプリケーションが 800 x 600 の解像度で開発され、ノードの解像度は 1024 x 768 であるという場合でも、WindowViewer はアプリケーションを動的にスケーリングしません。アプリケーションの解像度は 800x600 のままになります。
   - WindowViewer が、ノードの解像度でアプリケーションを実行し、アプリケーションが開発されたときの解像度を無視するようにするには、［**スクリーン ビデオの解像度に変換**］を選択します。たとえば、ノードの解像度が 800 x 600 で、アプリケーションが 1280 x 1024 で開発された場合、WindowViewer はアプリケーションをノードの 800 x 600 の解像度に合わせて自動的にスケーリングします。
     - アプリケーションが作成されたときに画面解像度とターゲット解像度が異なる場合、元のアプリケーションの解像度から現在の画面解像度にスケールされます。元のアプリケーション解像度は、ターゲット解像度の設定に関係なく、アプリケーションが作成されたときの画面解像度です。たとえば、1280x1024 のターゲット解像度でアプリケーションが 1920x1080

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

で開発され、表示ノードが 800x600 の解像度でアプリケーションを実行している場合、WindowViewer では、1920x1080 の元のアプリケーション解像度を使用するようアプリケーションが動的にスケールされます。詳細については、「元のアプリケーション解像度」を参照してください。

- ［**幅（X）**］および［**高さ（Y）**］（整数値）ボックスに指定された特定の解像度で WindowViewer でアプリケーションを実行するには、［**解像度のカスタマイズ**］を選択します。アプリケーションの解像度とノードの解像度はどちらも無視されます。たとえば、［**幅（X）**］が 512 に設定され、［**高さ（Y）**］が 384 に設定されている場合、アプリケーションはノードの画面上の 512 x 384 のピクセル領域に合わせて動的にスケーリングされます。
  - アプリケーションが作成されたときに画面解像度とターゲット解像度が異なる場合、元のアプリケーションの解像度から現在の画面解像度にスケールされます。元のアプリケーション解像度は、ターゲット解像度の設定に関係なく、アプリケーションが作成されたときの画面解像度です。

6. ［**OK**］ を選択します。

アプリケーションの解像度のロック

WindowMaker プロパティを設定することで、InTouch アプリケーション ウィンドウのサイズをロックすることができます。この機能は、ウィンドウとグラフィックのスケーリングをせずに、アプリケーションの解像度を変換できます。

このオプションを選択した場合、異なる解像度のコンピュータでアプリケーションを開くと、ウィンドウとグラフィックのスケーリングを行わずにアプリケーションを新しい解像度に変換するかどうかが尋ねられます。

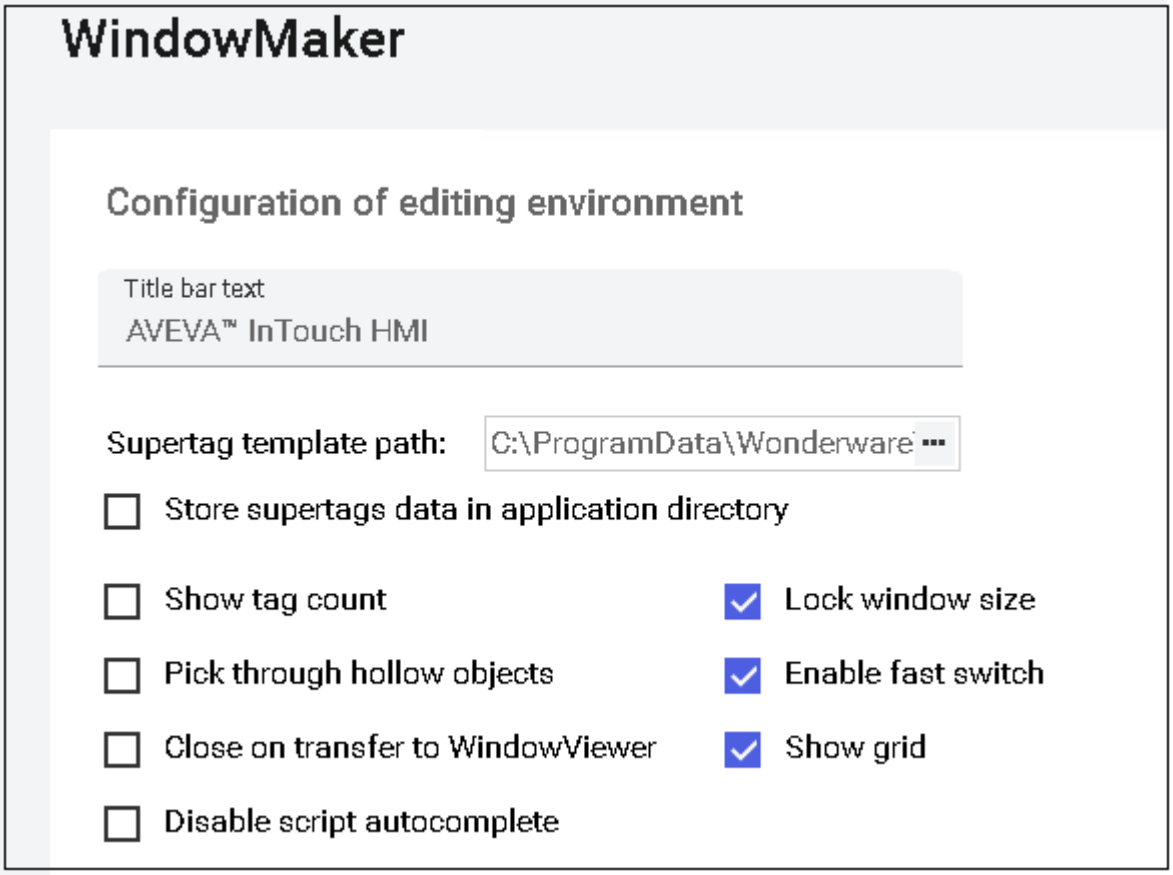
アプリケーションの解像度は、WindowMaker 内から、またはアプリケーション マネージャからロックできます。

アプリケーションの解像度を WindowMaker からロックする

1. WindowMaker を開きます。
2. ［**ファイル**］メニューで、［**構成**］をポイントし、［**WindowMaker**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowMaker の**設定**画面が表示されます。



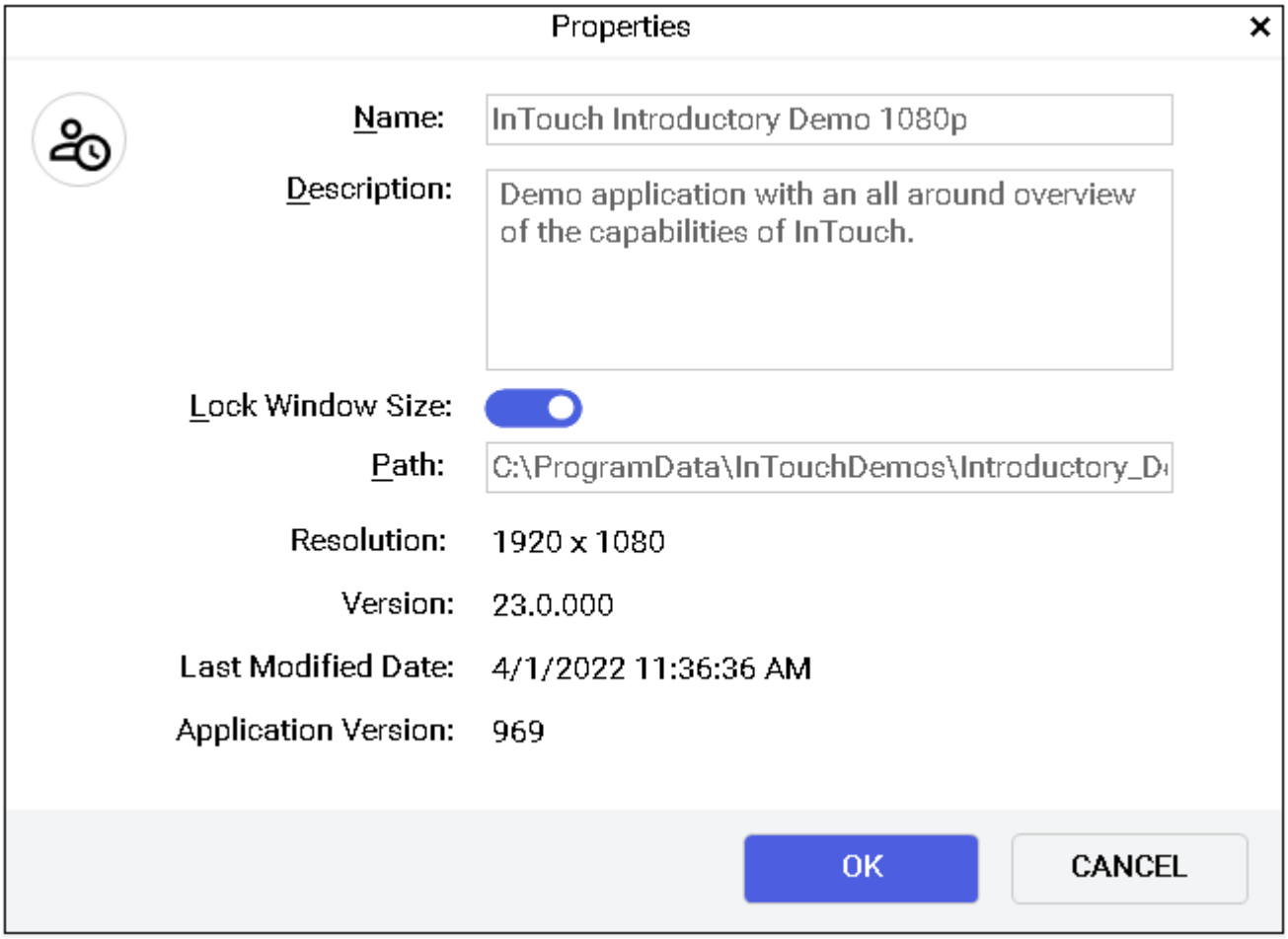
3. ［**ウィンドウ サイズをロック**］チェック ボックスをオンにします。デフォルトでは、このチェック ボックスはオンになっていません。
4. **保存**を選択します。

アプリケーションの解像度をアプリケーション マネージャからロックする

1. アプリケーション マネージャを開きます。
2. 設定するアプリケーションを選択します。
3. メニュー バーの［**ファイル**］を選択し、［**プロパティ**］を選択します。

   ［**プロパティ**］ダイアログ ボックスが表示されます。
4. ［**ウィンドウ サイズをロック**］スイッチを選択します。デフォルトでは、このチェック ボックスはオンになっていません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. ［**OK**］ を選択します。

リモート ノードへのアプリケーションのパブリッシュ

アプリケーション パブリッシャを使用すると、別のコンピュータ上に InTouch アプリケーションをインストールするために必要な関連するすべてのファイルおよび設定手順が含まれている、圧縮された自己解凍形式のパッケージ ファイルを作成できます。アプリケーション パブリッシャを使用すると、スタンドアロンの InTouch アプリケーションをパブリッシュできます。マネージド InTouch アプリケーションは System Platform IDE を使用してパブリッシュします。

アプリケーションをパブリッシュするには、次の 2 つのオプションがあります。

- **ランタイムのみ。**ランタイムのみのパッケージには、アプリケーションを実行するために必要なファイルが含まれていますが、アプリケーションを編集するためのファイルは含まれていません。
- **開発時およびランタイム。**開発時およびランタイムのパッケージには、アプリケーションを編集および実行するために必要なすべてのファイルが含まれています。コンパイル済みの *.wvw ファイルなどのいくつかのランタイム ファイルは、開発時のファイルから再作成できるため除外されています。

パブリッシュ済みアプリケーションは、ダウンロードおよびインストール可能な Web Server にポストできます。ポストされたアプリケーションに対する、次のパッケージ情報が表示されます。

- パッケージの説明

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- パブリッシャ名
- パブリッシュ済みファイル名（実行可能ファイル）
- アプリケーションの解像度

以下に例を示します。

ディスクリプション	乳製品処理アプリケーション
発行者	Navin Johnson
ファイル名	Dairy.exe / ビデオ解像度...（1024x768）

ディスクリプション	乳製品処理アプリケーション
発行者	Navin Johnson
ファイル名	Dairy_2.exe / Video Resolution...(800x600)

発行したファイルのコンテンツ

以下の表は、すべての発行済みスタンドアロン InTouch アプリケーションの含まれているフォルダとファイル、および除外されたファイルを示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

含まれているフォルダ	含まれているファイル	除外されたファイル
メイン アプリケーション フォルダ	すべて	バックアップ ファイル。これらのファイルのファイル名には .?bk という拡張子が付いています。
	これらの拡張子を持つファイル： .win、.dat,、.lgh、.idx、.log、.fsm、.stg、.$$$	特殊ディレクトリ リストにないサブフォルダ
	retentiv.x retentiv.d retentiv.a retentiv..s（ピリオドが 2 つ） retentiv.h wm.ini db.ini linkdefs.ini tbox.ini group.def itocx.cfg	WindowMaker でアプリケーションが開かれていることを示す appedit.lok ファイル
	SSD_*.xml という形式の名前を持つすべてのファイル	.wvw という拡張子が付いたファイル名を持つコンパイルされたウィンドウ ファイル
ランタイム言語の切り替えのためのディクショナリ サブフォルダ	.xml という拡張子を持つすべてのファイル	
Symbol サブフォルダ	すべてのファイルおよびサブフォルダ	
	ウィザードがインストールされている場合は、wiz.ini ファイル	
	ウィザードの実行可能ファイルのコピー	
	.dll ファイル、 .wdo ファイル .wdf ファイル	

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイムのみのアプリケーションでは、SSD_*.xml というファイル名を持つすべてのファイルが除外されます。

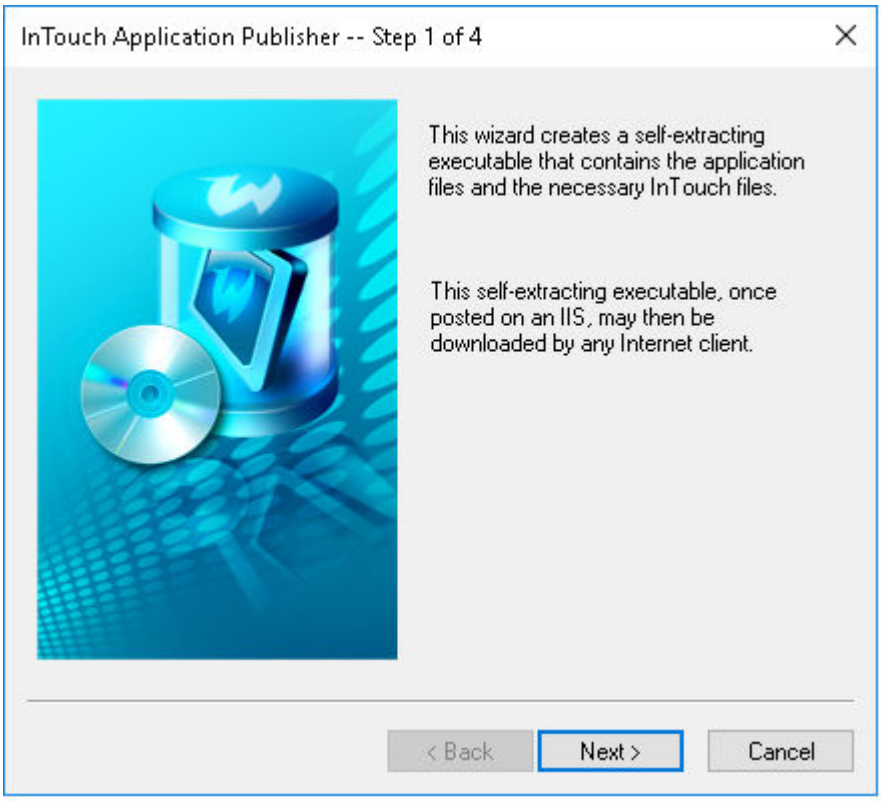
スタンドアロン InTouch アプリケーションのパブリッシュ

アプリケーション パブリッシャを使用すると、スタンドアロン InTouch アプリケーションをパブリッシュできます。パブリッシュ済みアプリケーションを特定の画面解像度で実行する場合、元のアプリケーションをパブリッシュする前に解像度を設定します。マネージド InTouch アプリケーションをパブリッシュするには、System Platform IDE 使用します。

スタンドアロン InTouch アプリケーションのパブリッシュ

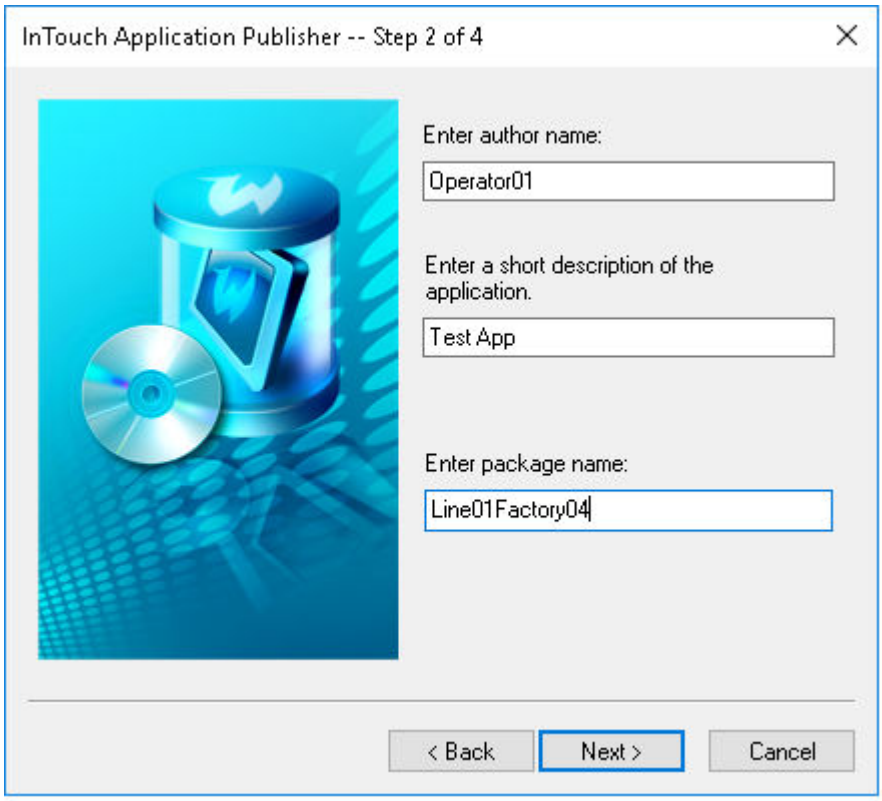
1. アプリケーション パブリッシャを起動します。
   a. WindowMaker を開きます。
   b. ［ツール］パネルを展開し、［アプリケーション］を展開します。
   c. ［**Application Publisher**］をダブルクリックします。

   ［**InTouch アプリケーション パブリッシャ – ステップ 1/4**］ダイアログ ボックスが表示されます。



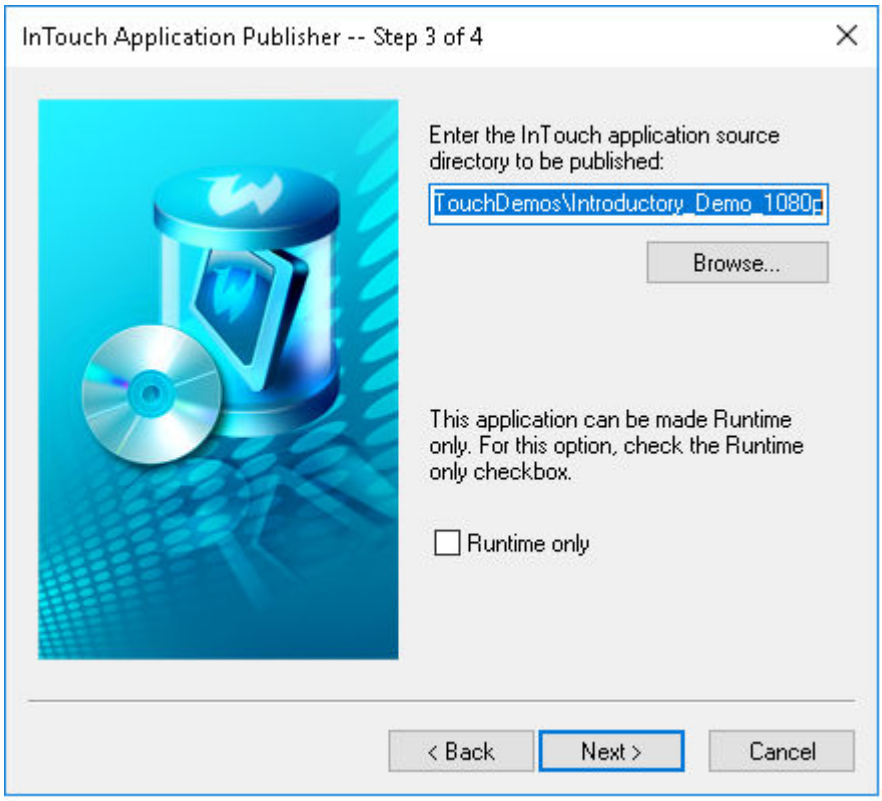
2. **[次へ]** を選択します。［**InTouch アプリケーション パブリッシャ – ステップ 2/4**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



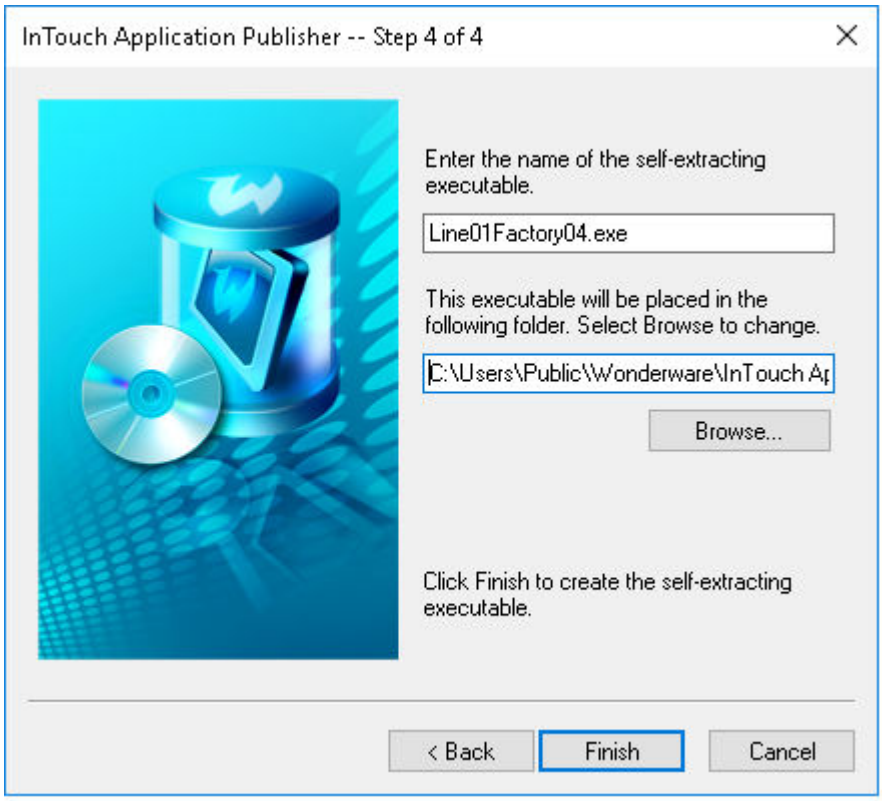
1. パッケージの詳細を設定します。
   - ［**作成者名を入力してください**］ボックスで、アプリケーションに関する連絡先の名前を入力します。入力できる名前は 256 文字までです。
   - ［**このアプリケーションの説明を入力してください**］ボックスに、アプリケーションの説明を入力します。入力できる文字数は 256 文字までです。
   - ［**パッケージ名を入力してください**］ボックスに、パブリッシュ済みアプリケーション パッケージの一意の名前を入力します。入力できる文字数は 32 文字までです。既存のパッケージの名前を使用している場合、既存のパッケージが上書きされます。
2. **[次へ]** を選択します。［**InTouch アプリケーション パブリッシャ – ステップ 3/4**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. パブリッシュの詳細を設定します。
   - ボックスに、InTouch アプリケーション フォルダへのパスを入力します。デフォルトのパスは WindowMaker のアプリケーション フォルダです。
   - パブリッシュ済みファイルで開発 WindowMaker ファイルを除外するには、[**ランタイム専用**] チェック ボックスをオンにします。
4. **[次へ]** を選択します。[**InTouch アプリケーション パブリッシャ – ステップ 4/4**] ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. 実行可能ファイルのアプリケーション パッケージの詳細を設定します。
   - 最初のボックスで、最初のボックスの実行可能ファイル名が正しいことを確認します。デフォルトでは、実行可能ファイル名はパッケージ名と同じです。
   - 2 番目のボックスで、実行可能ファイルを保存するフォルダへのパスを入力するか、［参照］を選択して別のフォルダを選択します。デフォルトでは、実行可能ファイルは現在の一時フォルダに保存されます。
6. **[完了]** を選択します。

マネージド InTouch アプリケーションのパブリッシュ

マネージド InTouch アプリケーションをパブリッシュできます。パブリッシュ済み InTouch アプリケーションは、InTouchViewApp テンプレートには関連付けられていません。

パブリッシュ済みアプリケーションを IDE 内で編集したり、他の InTouchViewApp テンプレートにインポートすることはできません。つまり、IDE で管理したり再パブリッシュすることはできません。

パブリッシュ済み InTouch アプリケーションは、埋め込まれた産業用グラフィックを使用して、引き続き Galaxy と通信できます。たとえば、Galaxy にデータを書き込んだり、Galaxy データを表示することができます。

コピー、切り取り、貼り付け、複製、移動、サイズ変更、反転、回転、および InTouch アニメーションリンクの設定などの基本的な InTouch の操作を使用して、産業用グラフィックを編集できます。

ただし、産業用グラフィックは変更できません。また、InTouch アプリケーションに新しい産業用グラフィックを埋め込むことはできません。これらの操作は、マネージド InTouch アプリケーションを使用した場合のみ可能です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

これは、ArchestrA の処理要件をサポートしていない環境で行うことができます。たとえば、リモートでの工場操業または小規模ネットワークなどです。

マネージド InTouch アプリケーションのパブリッシュ

関連付けられている InTouchViewApp オブジェクトからマネージド InTouch アプリケーションをパブリッシュできます。

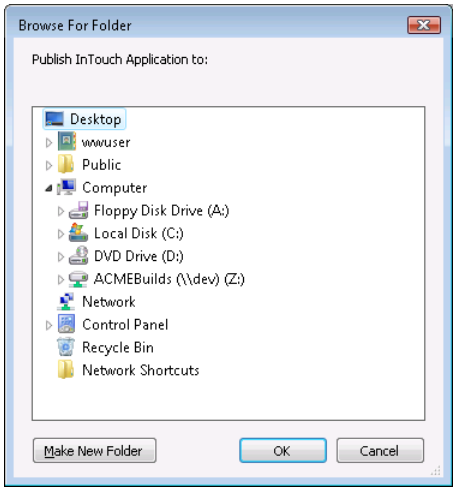
エクスポートは、オブジェクトおよびマネージド InTouch アプリケーションに関する情報が含まれているフォルダで構成されています。

これは、InTouchViewApp オブジェクト自体のエクスポートとは異なります。詳細については、「InTouchViewApp オブジェクトのインポートとエクスポート」を参照して下さい。

パブリッシュ済みの InTouch アプリケーションは、InTouchViewApp オブジェクトに再インポートできません。パブリッシュ済みの InTouch アプリケーションは編集できません。

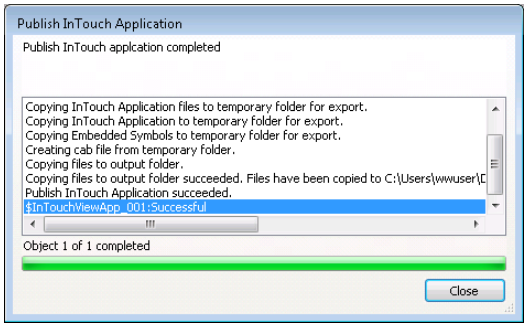
マネージド InTouch アプリケーションのパブリッシュ

1. System Platform IDE を開きます。
2. パブリッシュするマネージド InTouch アプリケーションが含まれている InTouchViewApp オブジェクトを検索します。
3. オブジェクトを右クリックし、［**InTouch アプリケーションの発行**］を選択します。［**フォルダの参照**］ダイアログ ボックスが表示されます。



4. InTouch アプリケーションのパブリッシュ先のフォルダを指定します。以下のいずれかを実行します。
   - 既存のフォルダを参照します。
   - 新しいフォルダまたはフォルダ構造を作成するには、［**新規フォルダの作成**］を選択します。
5. ［**OK**］ を選択します。［**InTouch アプリケーションのパブリッシュ**］進捗ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. パブリッシュが完了したら、［**閉じる**］を選択します。新しいパブリッシュ済み InTouch アプリケーションが含まれているディレクトリが、選択したフォルダに作成されます。任意のランタイム ノードにコピーできるようになりました。

Insight にアプリケーションをパブリッシュする

Insight Publisher を使用して、アプリケーションを Insight Web サイトにパブリッシュできます。アプリケーション マネージャまたは WindowMaker を使用できます。

Insight にアプリケーションをパブリッシュする

1. アプリケーション マネージャを開きます。
2. ［**ツール**］メニューの［**ツール**］グループで［**Insight Manager**］を選択します。

   ［**Insight Publisher**］ウィンドウが表示されます。

InTouch WindowMaker では、［**アプリケーション**］の下にある［**ツール**］ペインから［**AVEVA Insight Publisher**］を選択できます。

次のいずれかのオプションを選択して、画面に表示される手順に従います。

- ［**パブリッシュ**］ - 既存の InTouch アプリケーションから新しい Insight データ ソースを作成します。
- ［**インポート**］ - OPC、MQTT、または OI Server から Excel スプレッドシート リスト項目をインポートします。
- ［**権限付与**］ - データ ソースを作成します。

詳細については、Historian のマニュアルを参照してください。

注意：アプリケーションをパブリッシュするには、Insight アカウントが必要です。

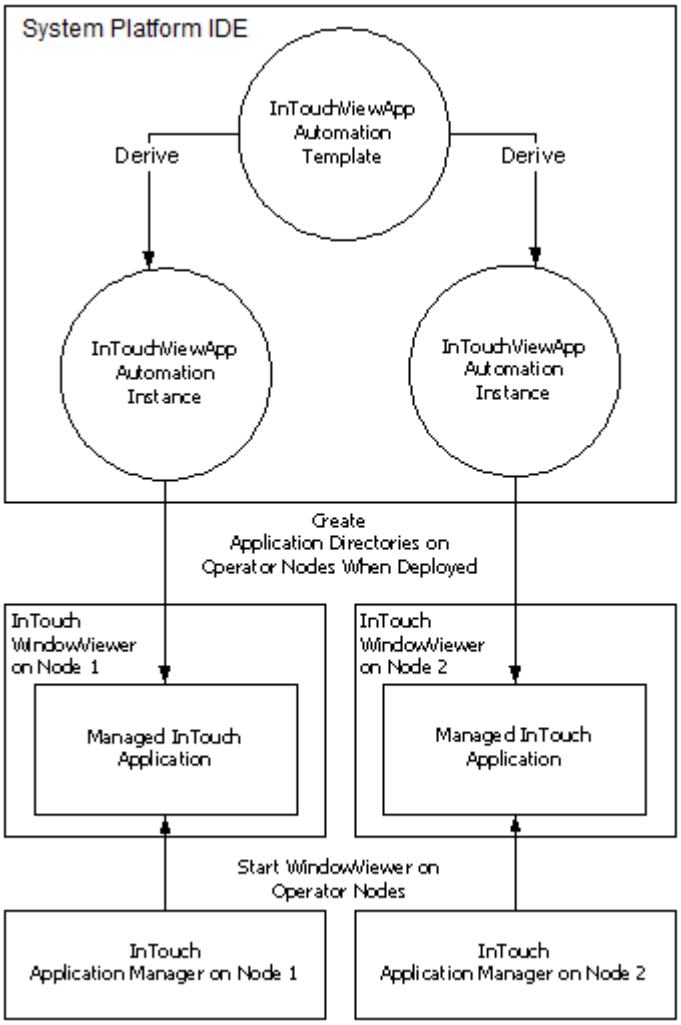
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 18 ランタイム時のマネージド InTouch アプリケーションの使用

InTouchViewApp インスタンスをリモート ノードに配置すると、マネージド InTouch アプリケーションをリモート ノードで実行できます。

また、これらのノードに InTouch アプリケーションや含まれている産業用グラフィックを配置することや、各ノードの変更を受け入れるのか拒否するのかを選択することもできます。

以下の図は、マネージド InTouch アプリケーションがランタイム ノードにどのように配置されるのかを示しています。



マネージド InTouch アプリケーションを配置する

System Platform IDE からローカル ノードまたはリモート ノードに、マネージド InTouch アプリケーションを配置することができます。アプリケーションを配置した後、リモート ノード上の WindowViewer で実行できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：WindowViewer で一度に開くことのできるアプリケーションは 1 つのみです。プラットフォームがローカル ノードに配置されている場合、他のスタンドアロン アプリケーションまたはマネージド アプリケーションで設定されているスタイルよりも Galaxy で設定したスタイルが優先されます。

初めての InTouchViewApp オブジェクトの配置

InTouchViewApp オブジェクトを初めて配置する場合、関連付けられている InTouch アプリケーションが、オブジェクトをホストするプラットフォームのノードにコピーされます。これはオペレータ ノードと呼ばれます。

マネージド InTouch アプリケーションを配置する

1. System Platform IDE を開きます。
2. マネージド InTouch アプリケーションを配置する InTouchViewApp のインスタンスを選択します。
3. ［**オブジェクト**］メニューで、［**配置**］を選択します。［**配置**］ダイアログ ボックスが表示されます。
4. ［**OK**］ を選択します。完全な InTouch アプリケーションがオペレータ ノードにコピーされます。

マネージド InTouch アプリケーションへの変更の配置

注意：［**WindowViewer のデフォルト設定を上書き**］オプションは、英語以外のオペレーティング システムではテストされていません。Web Client では、このオプションがサポートされません。

マネージド InTouch アプリケーションは以下のように変更できます。

- マネージド InTouch アプリケーションで使用されている産業用グラフィックの参照、コンテンツ、またはサイズを変更する。
- InTouchViewApp テンプレートから WindowMaker を開いて、マネージド InTouch アプリケーション自体を変更する。

どちらの場合でも、変更を保存するときに、更新されたテンプレートから派生したインスタンスに変更が反映されます。変更は［変更の保留］アイコンによって識別されます。

変更は、実行中の WindowViewer セッションにすぐには反映されません。各ノードのオペレータは、変更を受け入れるのかまたは拒否するのかを選択できます。詳細については、「オペレータ ノードでの新しいアプリケーション バージョンの受け入れ」を参照してください。

マネージド InTouch アプリケーションへの変更の配置

1. System Platform IDE を開きます。
2. マネージド InTouch アプリケーションの変更を配置する InTouchViewApp のインスタンスを選択します。
3. ［**オブジェクト**］メニューで、［**配置**］を選択します。［**配置**］ダイアログ ボックスが表示されます。
4. ［**OK**］ を選択します。変更がオペレータ ノードにコピーされます。

マネージド InTouch アプリケーションの起動

ユーザー ノードから、アプリケーション マネージャを起動し、実行するマネージド InTouch アプリケーションを選択できます。

また、異なる解像度でマネージド InTouch アプリケーションを実行するために、アプリケーション マネージャの解像度を設定することもできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マネージド InTouch アプリケーションを起動する

1. InTouchViewApp オブジェクトが配置されるノードで、InTouch アプリケーション マネージャを起動します。
2. アプリケーション リストで、WindowViewer で実行するマネージド InTouch アプリケーションを選択します。
3. ［WindowViewer］アイコンを選択します。しばらくすると、WindowViewer でアプリケーションが起動します。

ダイナミック レゾリューション機能を設定する

1. InTouch アプリケーション マネージャを開きます。
2. ［**ツール**］メニューで、［**ノードのプロパティ**］を選択します。［**ノードの設定**］ダイアログ ボックスが表示されます。
3. ［**解像度**］タブを選択します。
4. ［**WindowViewer で動的に解像度を変更**］チェック ボックスをオンにします。

   ［**WindowViewer で動的に解像度を変更**］がオフになっている場合、開発された解像度でマネージド アプリケーションが実行されます。
5. アプリケーションを実行する方法を設定します。以下のいずれかを実行します。
   - 開発されたときの解像度と同じ解像度でマネージド アプリケーションを実行するには、［**アプリケーションの解像度を使用**］を選択します。
   - 画面の解像度を使用して実行できるようにマネージド アプリケーションを変換するには、［**スクリーン ビデオの解像度に変換**］を選択します。
   - マネージド アプリケーションを指定した解像度に変換するには、［**解像度のカスタマイズ**］を選択します。
6. ［**OK**］ を選択します。

マネージド アプリケーションを配置するときの WindowViewer の再起動待機時間の制御

再起動が必要なマネージド アプリケーションを配置するときに WindowViewer が再起動するまでの遅延時間を制御できます。遅延時間を実装すると、アラーム サブシステムの適切な動作を保証できます。

WindowViewer の再起動遅延は、C:\Windows folder にある win.ini ファイルの以下のパラメータによって制御できます。すべてのパラメータの単位は秒です。

- `ViewManagedRestartWaitPeriod=0`

  このパラメータは、WindowViewer の再起動待機時間を制御します。デフォルト値は 0 で、待機時間はありません。
- `ViewNADShutdownWaitPeriod=30`

  ネットワーク アプリケーション開発（NAD）を使用して管理されたアプリケーションの場合、このパラメータは NAD がシャットダウンするまでの遅延時間を制御します。デフォルトは 30 秒です。
- `ViewNADRestartWaitPeriod=90`

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

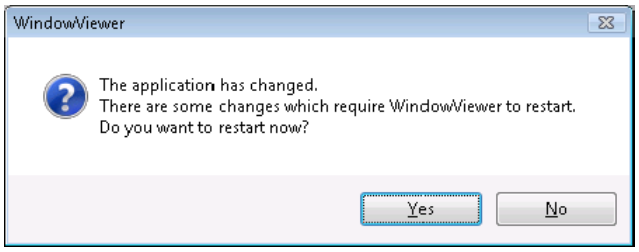
NAD を使用して管理されたアプリケーションの場合、このパラメータは NAD が再起動するまでの遅延時間を制御します。デフォルトは 90 秒です。

オペレータ ノードでの新しいアプリケーション バージョンの受け入れ

マネージド InTouch アプリケーションが変更され、その関連付けられている InTouchViewApp インスタンスが配置された場合、変更を受け入れるかまたは拒否するかをランタイム ノードで選択できます。

以下の場合に、マネージド InTouch アプリケーションへの変更を受け入れるかどうか確認するメッセージが表示されます。

- WindowViewer を InTouch アプリケーション マネージャから起動するとき。
- WindowViewer が実行しているとき。



変更の性質に応じて、WindowViewer アプリケーションを再起動するか、または再ロードのみを行うのかが確認されます。また、アプリケーションの変更に対して、WindowViewer の動作を以下のように設定することもできます。

- これらの変更を WindowViewer が受け入れたり拒否する方法。
- 変更が保留中である間に、WindowViewer を再ロードするのか再起動するのかを WindowViewer が確認する頻度。

オペレータ ノードでの新しいアプリケーション バージョンの受け入れ

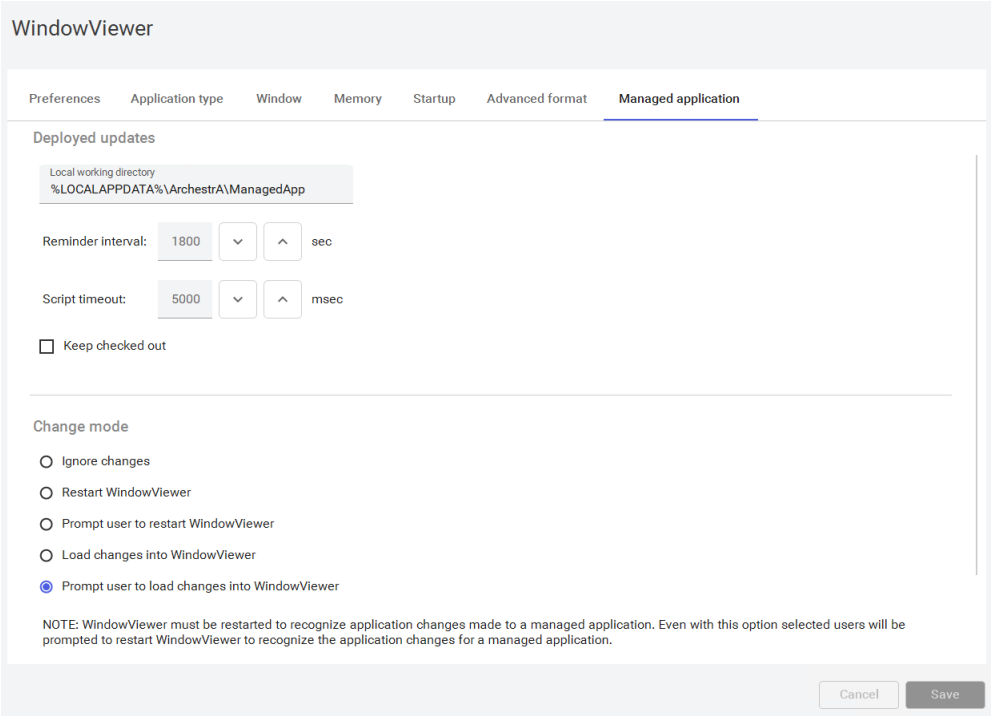
- ［はい］を選択します。マネージド InTouch アプリケーションへの変更がオペレータ ノードにコピーされ、WindowViewer が再起動または再ロードされます。

アプリケーションの変更に対する WindowViewer の動作を設定する

1. マネージド InTouch アプリケーションを WindowMaker で開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［マネージド アプリケーション］タブをクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**モードの変更**］領域で、変更が配置されたときに WindowViewer がどのように応答するのかを設定します。以下のいずれかを実行します。
   - 配置された変更を WindowViewer が無視するように設定するには、［**変更を無視**］を選択します。RestartWindowViewer() スクリプト関数および ReloadWindowViewer() スクリプト関数を手動で設定して、$ApplicationChanged システム タグに応じて変更を受け入れることができます。
   - WindowViewer を自動的に再起動するように設定するには、［**WindowViewer の再起動**］を選択します。
   - WindowViewer を再起動するように WindowViewer でプロンプトが表示されるように設定するには、［**WindowViewer の再起動時にメッセージ表示**］を選択します。
   - WindowViewer が変更を自動的にロードするように設定するには、［**WindowViewer に変更をロード**］を選択します。
   - WindowViewer への変更をロードするように WindowViewer でプロンプトが表示されるように設定するには、［**WindowViewer へ変更ロード時にメッセージ表示**］を選択します。

注意：［WindowViewer に変更をロード］または［WindowViewer へ変更ロード時にメッセージ表示］オプションを選択した場合、WindowViewer を再起動して、マネージド アプリケーションに対する変更を認識させる必要があります。これらのオプションが選択されている場合でも、WindowViewer を再起動するよう求めるメッセージが表示されます。

1. ［**リマインダ間隔（秒）**］ボックスに、WindowViewer への変更のロードまたは WindowViewer の再起動をユーザーに確認する頻度を秒で入力します。このオプションは、該当する変更モードが設定された場合のみ使用できます。ユーザーに再度確認しない場合、間隔を 0 に設定します。
2. ［**OK**］ を選択します。

WindowViewer のデフォルトの動作を設定する

1. マネージド InTouch アプリケーションを WindowMaker で開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［ファイル］メニューで、［設定］をポイントし、［**WindowViewer**］を選択します。**WindowViewer** の設定画面が表示されます。
3. ［マネージド アプリケーション］タブをクリックします。
4. ［既定に戻す**(&S)**］を選択します。設定がデフォルトの値にリセットされます。
5. ［**OK**］ を選択します。

埋め込まれた産業用グラフィックでの ArchestrA スクリプトの実行

マネージド InTouch アプリケーションを WindowViewer で実行すると、要素またはシンボル スクリプト自体に関連付けられているすべての ArchestrA スクリプトが予期したとおりに実行されます。

ただし、シンボルに含まれているいくつかのスクリプトは長期に渡って実行され、その他の InTouch 要素の操作を妨げる場合があります。

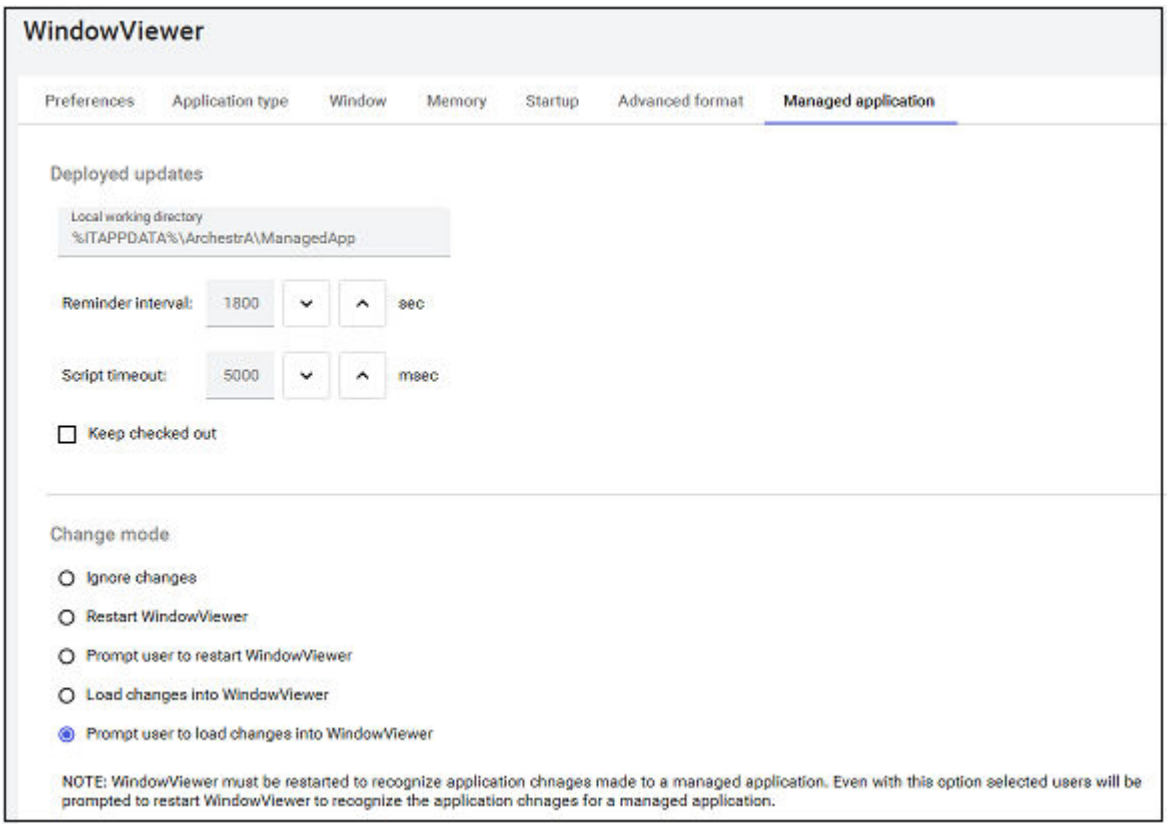
これを回避するには、マネージド InTouch アプリケーション内のすべてのスクリプトに適用可能なスクリプト タイムアウトを設定します。スクリプト タイムアウトによりスクリプトの実行が停止され、オペレータに制御が戻ります。

デフォルトでは、スクリプトは 5 秒後にタイムアウトします。

スクリプト タイムアウトを設定する

1. マネージド InTouch アプリケーションを WindowMaker で開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［**WindowViewer**］を選択します。**WindowViewer** の設定画面が表示されます。
3. ［マネージド アプリケーション］タブをクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［スクリプト タイムアウト（ミリ秒)］ボックスに、値をミリ秒で入力します。
5. ［**OK**］ を選択します。

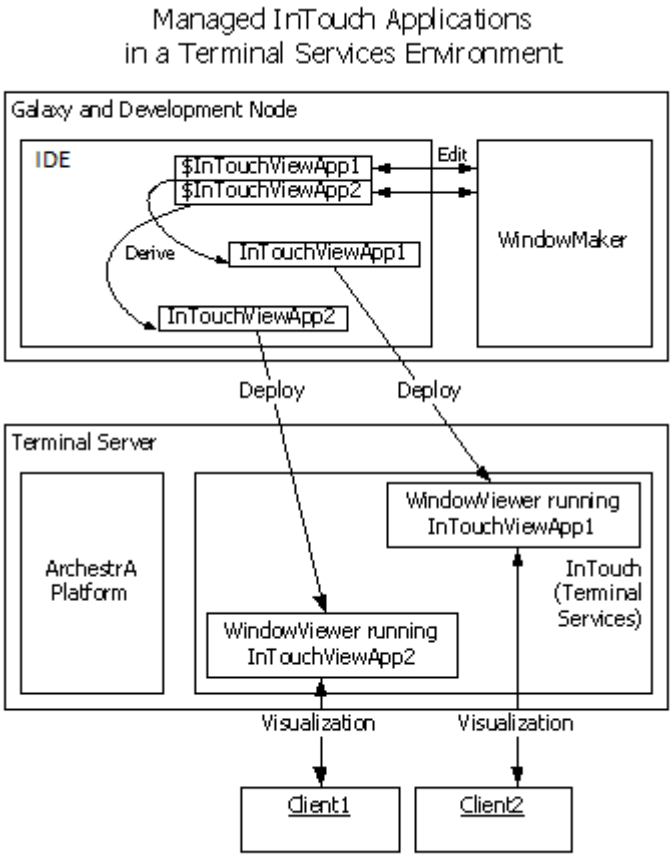
ターミナル サービス環境での InTouchViewApp オブジェクトの配置

ターミナル サービス環境でマネージド InTouch アプリケーションを実行できます。このアーキテクチャのおもな利点は、複数の InTouch アプリケーションを 1 つのコンピュータ上で同時に実行できる点です。

ターミナル サービス環境のマネージド InTouch アプリケーションを実行する

- ターミナル サービス エディションの InTouch を使用する。
- それぞれの ViewEngine ホストと共に InTouchViewApp の各インスタンスを配置した場合（ターミナル サーバー上のいくつかの InTouchViewApplication インスタンスを選択した場合)。
- 各マネージド InTouch アプリケーションを、独自のターミナル サービス クライアント セッションで実行する。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



注意：アプリケーションを複数のターミナル セッション クライアントで使用可能にする場合、ターミナル サーバーに配置する必要がある InTouchViewApp オブジェクトは 1 つのみです。アプリケーションを使用する各クライアントに対して個別の InTouchViewApp オブジェクトを配置する必要はありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 19 AVEVA Edge での InTouch HMI アプリケーションの使用

AVEVA Edge には、InTouch HMI アプリケーションを配置および再使用するためのオプションが搭載されています。アプリケーション マネージャを使用して、AVEVA Edge からアプリケーションをダウンロードすることや、アプリケーションを CONNECT にアップロードすることができます。

ダウンロードされた zip ファイルは、エッジデバイスおよびコンテナへの配置に適したサイズの小さい圧縮バージョンの InTouch HMI プリケーションです。同じ zip を CONNECT にアップロードして、そこから他のデバイスにダウンロードすることもできます。

注意：AVEVA Edge 関連のオプションが有効化されていない場合は、WindowMaker を起動します。

AVEVA Edge zip ファイルのダウンロード

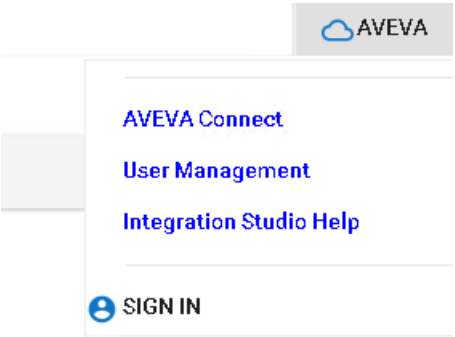
1. アプリケーションを選択します。
2. アイコンを選択します。

   ダウンロード用の圧縮（.zip）ファイルが準備されます。
3. ファイルを保存します。
4. AVEVA Edge Management ポータルを使用して、**新しいデバイスを追加**します。［**プロジェクト設定**］オプションを使用して zip ファイルをアップロードします。

   その他の展開およびペアリングの手順については、AVEVA Edge ヘルプを参照してください。
5. アプリケーションが正しく実行しているかどうかは、InTouch HMI Web Client を使用して確認できます。Web Client の詳細については、「*Web ブラウザでのアプリケーション グラフィックの表示*」を参照してください。

AVEVA Edge zip ファイルのアップロード

1. アプリケーション マネージャ ウィンドウの右端にある［**AVEVA**］ボタンを選択します。



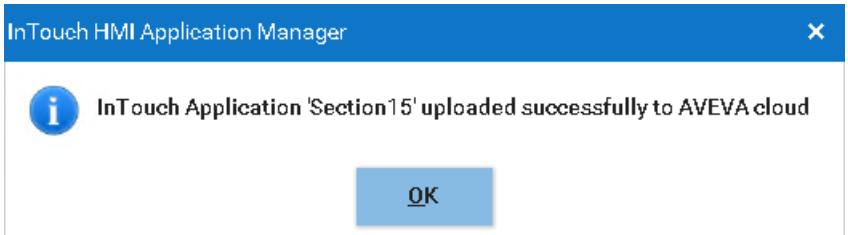
2. ［**サインイン**］を選択します。
3. CONNECT の資格情報を入力します。
4. ログインした後、アップロードするアプリケーションを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドライブを変更することができますが、ドライブ間でアプリケーションを移動することはできません。

5. を選択します。

アプリケーションが CONNECT にアップロードされると、アップロード処理が成功したことを示すメッセージが表示されます。

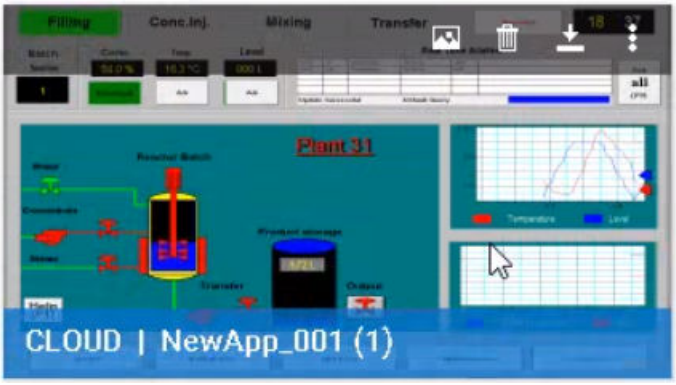


アップロードされたアプリケーションは、"*Cloud | <アプリケーション名>*" という接頭辞が付けられてアプリケーション リストに表示されます。

クラウド アプリケーションには以下のオプションがあります。

- サムネイル
- 削除
- ダウンロード

その他のすべてのオプションは無効化されています。



共有ドライブにアップロードされたアプリケーションは、すべてのユーザーがアクセスできます。ユーザーは、クラウド ストアからアプリケーションをダウンロードすることや削除することができます。

エッジ デバイスのユーザーの設定

エッジ デバイスから InTouch Web Client にアクセスする場合、アプリケーション マネージャでユーザーを設定する必要があります。アプリケーション マネージャで設定するユーザーは、Web Client の適切なユーザー グループに存在する必要があります。ユーザーのリストは、csv ファイルとしてエクスポートまたはインポートすることができます。ファイルのテンプレートは次のようになります。

```
UserName,Password,IsWrite
TestUser,TestPass#1,1
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザーを追加する

1. を選択します。

   ［ユーザー情報］ダイアログ ボックスが表示されます。
2. ユーザー名およびパスワードを入力します。
3. パスワードは以下の規則に準拠する必要があります。

   6 文字以上

   1 つ以上の小文字

   1 つ以上の大文字

   1 つ以上の数字

   1 つ以上の特殊文字
4. アクセス タイプを選択します。
5. **保存**を選択します。
6. を選択して新規ユーザーを追加します。
7. を選択してユーザーを削除します。
8. すべてのユーザーを設定したら、 を選択して設定済みリストをエクスポートします。
9. AVEVA Edge Management の「デバイスの追加」の手順で［ユーザー設定］オプションを使用してリストをアップロードします。

   エッジ デバイスの設定とペアリングが完了すると、エッジ デバイス上にユーザーのリストが作成されます。これらのユーザーは、Web Client にアクセスしてアプリケーション グラフィックを表示できます。

既存のユーザー リストのインポート

インポートする既存のユーザーのリストがある場合、必要な形式で .csv ファイルを準備します。

1. を選択して、.csv ファイルのパスを指定します。
2. ファイル内のエントリがユーザーのリストに追加されます。
3. AVEVA Edge Management ポータルを使用して、**新しいデバイスを追加**します。［**ユーザー設定**］オプションを使用して csv ファイルをアップロードします。

   その他の展開およびペアリングの手順については、AVEVA Edge ヘルプを参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Operate

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 20 ランタイム時のアプリケーションの表示の概要

WindowViewer を使用して、InTouch アプリケーションを実行できます。Application Server 環境専用に設計されているアプリケーションは InTouchView アプリケーションと呼ばれます。InTouch Web Client を使用して、HTML5 対応の任意の Web ブラウザで産業用グラフィックを表示できます。このようなアプリケーションは WindowViewer で動作しますが、HMI のほとんどの機能は Application Server によって提供されます。

ランタイム時のアプリケーションの表示の概要

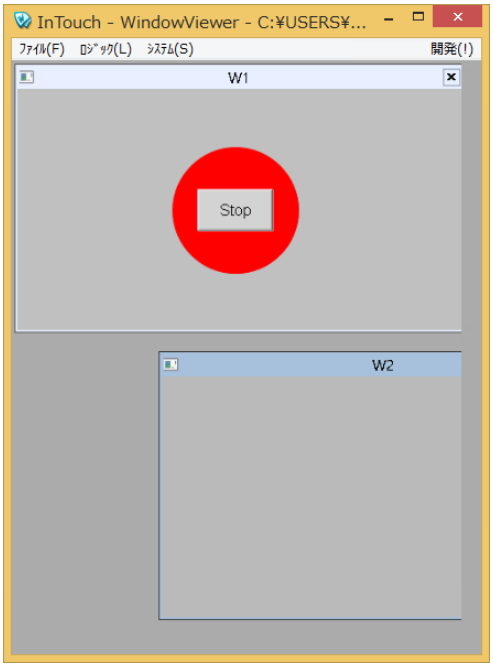
WindowViewer を使用して、InTouch アプリケーションを実行できます。Application Server 環境専用に設計されているアプリケーションは InTouchView アプリケーションと呼ばれます。InTouch Web Client を使用して、HTML5 対応の任意の Web ブラウザで産業用グラフィックを表示できます。このようなアプリケーションは WindowViewer で動作しますが、HMI のほとんどの機能は Application Server によって提供されます。

別のターゲット解像度サイズでのラインタイムでアプリケーションを表示

画面解像度と異なるアプリケーション ターゲット解像度を指定した場合、WindowViewer では、アプリケーションは指定されたターゲット解像度で表示されます。ランタイム時に表示されるのは、ターゲット解像度に収まるキャンバス境界内で開発されたウィンドウ、ウィンドウ コントロール、およびグラフィックだけです。ターゲット解像度サイズは、WindowViewer のメニューおよびタイトル バー コントロールに応じてランタイム時に自動的に調整されます。

埋め込まれたコントロールのアスペクト比はランタイム時に維持されます。

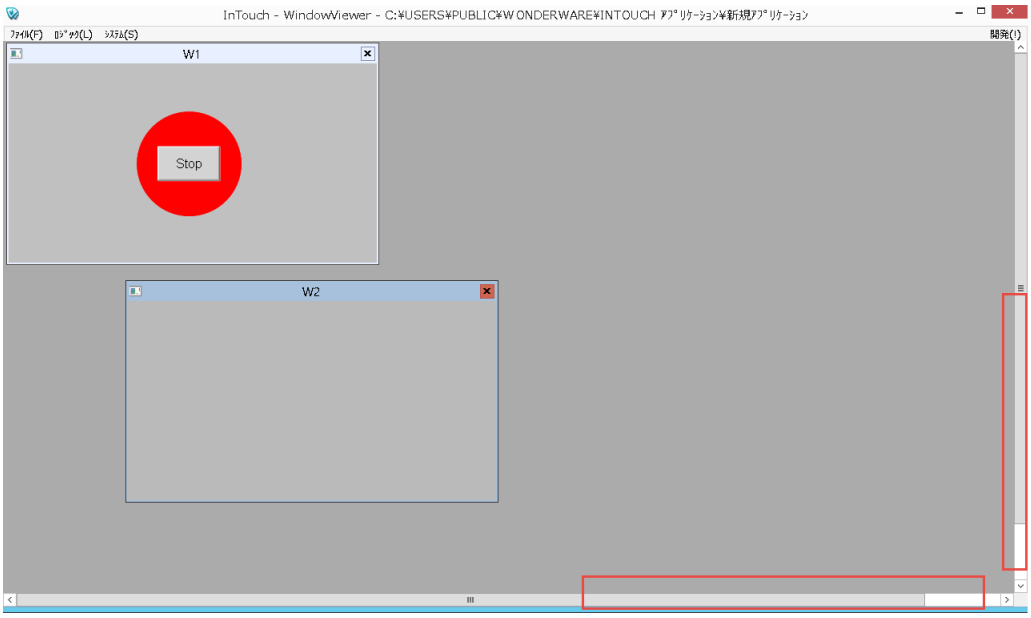
以下に例を示します。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：指定したターゲット解像度が画面解像度よりも低い場合、ランタイム ウィンドウの幅と高さは、指定したターゲット解像度以上に調整することはできません。WindowViewer ウィンドウを最大化した場合、ターゲット解像度の最大解像度までしか拡大されません。

指定したターゲット解像度が画面解像度よりも高い場合、ランタイム時に垂直および水平スクロール バーが表示されます。ウィンドウはスクロール バーの操作に合わせてスクロールします。シンボルの表示アニメーションおよび ShowGraphic スクリプトのポップアップ ウィンドウとポップアップ グラフィックはスクロールしません。



エラー メッセージとポップアップ ダイアログは、画面の中央ではなく、ターゲット解像度でのアプリケーションの中央に表示されます。キーボードとキーパッドの表示も同様です。

InTouch Web Client について

マウス ジェスチャでズームするには: InTouch Web Client を使用すると、InTouch HMI アプリケーション内で使用されている産業用グラフィックを HTML5 対応 Web ブラウザで表示できます。組み込みの Web Werver によって、リモート デスクトップ プロトコル (RDP) または Microsoft Windows® Server 用の Internet Information Services (IIS) を使用することなく、任意の Microsoft Windows クライアントまたはサーバー オペレーティング システムからアプリケーション グラフィックへのアクセスが提供されます。スタンドアロン アプリケーションとマネージド アプリケーションの両方のアプリケーション グラフィックを Web ブラウザで表示できます。スタンドアロン アプリケーション ウィンドウを Web Client で表示するには、産業用グラフィックに変換する必要があります。

InTouch Web Client の詳細については、「*Web ブラウザでのアプリケーション グラフィックの表示*」を参照してください。Web Client は、System Platform のインストールの一部としてインストールされます。Web Client の設定の詳細については、『System Platform インストール ガイド』を参照してください。

元のアプリケーション解像度

元のアプリケーション解像度は、ターゲット解像度の設定に関係なく、アプリケーションが作成されたときの画面解像度です。

元のアプリケーション解像度は、以下の条件の下でのみ更新されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アプリケーションが作成されたとき
- アプリケーションに変換が適用されたとき

後でアプリケーションが画面解像度に戻された場合、現在の画面解像度が元のアプリケーション解像度と異なっているとアプリケーション変換が発生します。それ以外の場合は、変換は発生しません。アプリケーションを作成するときにターゲット解像度を使用する場合、動的な解像度変換の動作に影響します。

ランタイム時のキーボード、マウス、およびタッチジェスチャの操作によるパンとズーム

枠ウィンドウを使用すると、ランタイム時に産業用グラフィックのパンとズームを行うことができます。この機能は、WindowMaker の **InteractionMode** プロパティで有効にします。

ランタイムのズーム

ランタイム時にフレーム コンテンツをズーム インまたはズーム アウトすることができます。枠で正しいパンおよびズーム プロパティが有効化されていることを確認してください。ズーム インの上限は 5000% で、下限は 100% です。

シンボルの **ZoomPercent** プロパティを編集して、ランタイム時のシンボルまたは要素の可視性を変更できます。たとえば、次に示すプロパティを表示アニメーションに追加すると、ズーム パーセント レベルに応じてシンボルを動的に変更できます。

```
ZoomPercent => 200
```

注意：このプロパティへはランタイム時に書き込むことができます。

シンボルに対して **ZoomPercent** が設定されている場合、シンボルは設定されたパーセントで表示可能な領域の中央にズームされます。

シンボルの要素に対して **ZoomPercent** が設定されている場合、シンボルは設定されたパーセントでズームされますが、中央に表示されるのは要素になります。

ZoomPercent を要素に対して設定するスクリプトの例を以下に示します。

```
TextBox1.ZoomPercent = 500
```

マウスでズーム

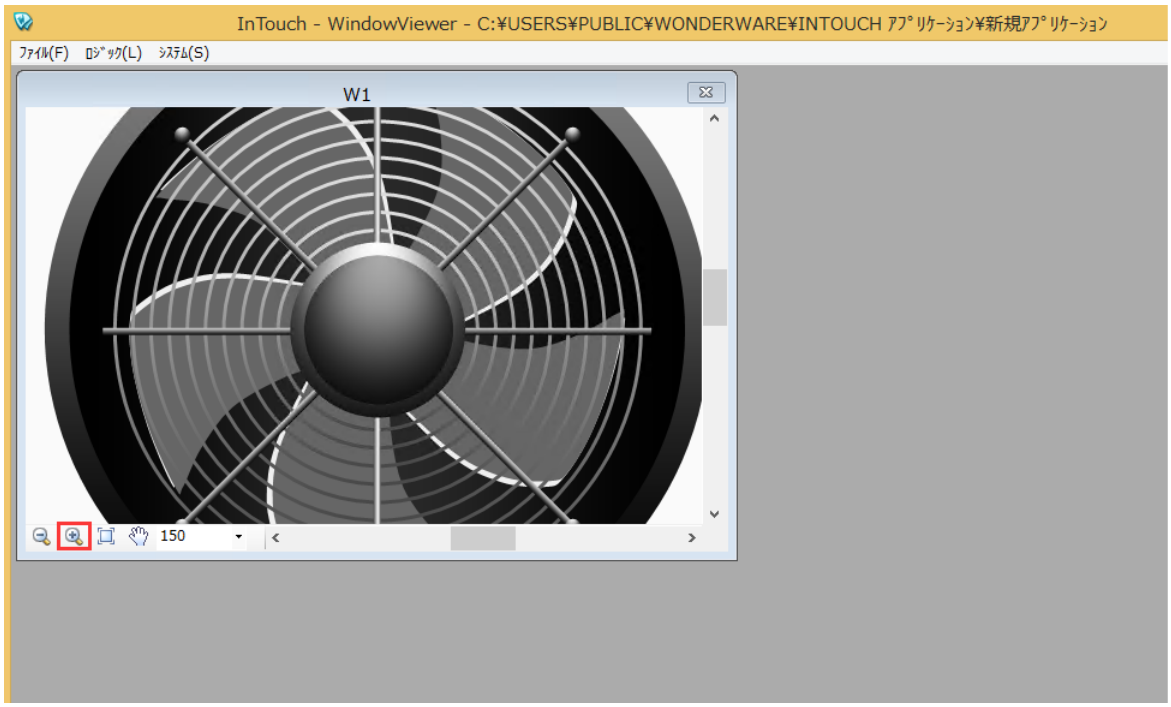
以下の手順を実行します。

- 枠コンテンツをズーム インするには、"Ctrl" キーを押してマウス ホイールを上にスクロールします。コンテンツがマウス ポインタの現在の位置からズーム インします。

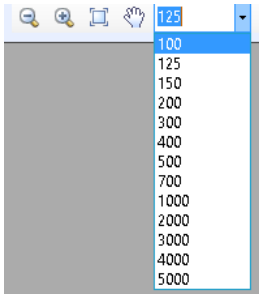
  注意：マウスポインタが枠外にある場合、枠をズームインできません。

- 枠コンテンツをズーム アウトするには、マウス ホイールを下にスクロールします。
- ［パンとズーム］コントロール ツールバーのズーム インおよびズーム アウト アイコンを選択します。その後、ズームインする枠のコンテンツをクリックする必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



- 枠コンテンツをダブルクリックすると、100% のズーム レベルに戻ります。
- ［ズーム レベル］コンボボックスを使用して、事前定義済みのズーム レベルを選択します。



- ［パンとズーム］コントロール ツールバーで［ラバー バンド ズーム］アイコンを選択し、ズーム インする特定の領域を選択します。

キーボードでズーム

以下の手順を実行します。

- ズーム インするには、Ctrl と + キーを同時に押します。
- ズーム アウトするには、Ctrl と - キーを同時に押します。

タッチジェスチャでズーム

以下の手順を実行します。

- ズーム インするには、2 本の指を画面に置いて指の間の幅を広げます。
- ズーム アウトするには、2 本の指を画面に置いて指の間の幅を狭めます。
- ダブルタップすると、100% のズーム レベルに戻ります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ズームの制限

ランタイム時のズーム機能には、以下の制限が適用されます。

- Windows の共通コントロールおよびクライアント コントロールのズームは 500% までです。このようなコントロールには以下が含まれます。
  - カスタム コントロール: ラジオ ボタン グループ、チェックボックス、編集ボックス、コンボ ボックス、カレンダー、日付時刻ピッカー、およびリストボックス
  - 埋め込まれたアラーム クライアントおよびトレンド クライアント コントロール
  - サードパーティ製クライアント コントロール
- Windows コントロールは、マウス、キーボード、およびタッチによる入力を上書きすることがあります。
- InTouch は Windows コントロールのカスタム フォントを上書きしません。
- マルチペントレンドでは、シンボルがズームされている間、カーソル表示はズームされません。

ランタイムに画面を移動

該当するシンボル プロパティを設定して、ランタイム時の枠内のパン機能を有効化します。マウスおよびタッチ ジェスチャをしようしてランタイム時にパンできます。キーボードでのパン操作はサポートされていません。ランタイム時にパンするには、グラフィックのズーム レベルは 100% 以上である必要があります。

マウスジェスチャでパン

以下のいずれかを実行します。

- マウスの中ボタンを押します。

  パン ハンドが表示されます。
- ［パンとズーム］コントロール ツールバーのパン ハンドを選択します。マウスの左ボタンを押したまま、パンハンドを移動して画面をパンします。

  画面は、マウス ボタンを離すまでパンされます。

タッチジェスチャでパン

1. 枠コンテンツ上に 1 本の指を置きます。
2. 指を画面上で移動してパンします。

注意：どちらのパン方法でも、水平および垂直スクロール バーはパン方向に従って調整されます。

タッチジェスチャを使用してパンとズームを同時に行うには

1. 画面上に 2 本の指を置き、指を下方向および左右に移動して、ズームしたコンテンツの中心を調整します。
2. 枠コンテンツ上でパンするには 1 本の指を使用します。
3. 同時にズームするには、2 本目の指を枠コンテンツ上に置きます。

パンの制限

以下の制限が適用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- キーボード ジェスチャを使用したパンはサポートされていません。
- Windows コントロールは、マウス、キーボード、およびタッチによる入力を上書きすることがあります。その結果、Windows コントロールを含む領域でのパンが無効化されることがあります。

タッチジェスチャのアニメーションサポート

タッチ サポートのすべてのアクション スクリプトは、ズーム レベルに関係なく同様に機能する必要があります。枠コンテンツをズーム インしたとき、すべてのインタラクションおよび表示アニメーションは、枠コンテンツが標準表示されている場合と同様に動作します。インタラクション アニメーションは、タッチで適切に機能します。

注意： アニメーションまたは **ShowSymbol** スクリプトによって表示される産業用グラフィックポップアップでは、デフォルトでパンとズームが有効化されています。しかし、この設定を無効にすることはできません。

以下の表には、タッチ サポート向けに一般的に設定されるアクション スクリプトが一覧表示されています。

アクション スクリプト	トリガされるタッチ
左クリック時	ダッチ下げ
押している間	タッチしてスライド
左を離した時	タッチ上げ
左ダブルクリック時	ダブルタップ
右クリック時	タッチして保持
右クリックを離した時	タッチしてホールドし、離したときに実行
右ダブルクリック時	サポートされていません。
右クリックを押している間	タッチしてホールドし、押したまま少しスライド
中クリック時	サポートされていません。
中クリックを押している間	サポートされていません。
中クリックを離した時	サポートされていません。
中ダブルクリック時	サポートされていません。

注意： アニメーションを含む領域でタッチインタラクションを使用すると、アニメーションがパン操作よりも優先され、パン操作は無視されます。1 本の指でのタッチ インタラクションを維持すると、その後のタッチ ポイントは無視されます。

このシナリオでパンを有効にするには、［**パンおよびズーム**］ツールバー.の［**パン**］アイコンを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タッチジェスチャの制限

ランタイム時のパンとズームと同じ制限がタッチ ジェスチャに適用されます。以下の機能はサポートされていません。

- Windows 共通コントロールのフォントのスケール

注意：さらに、埋め込まれたウィンドウコントロールを含むシンボルのスケール メカニズムは、コントロールを含まないシンボルのスケールメカニズムと異なります。埋め込まれたコントロールを含むシンボルの最大ズーム制限は 500% ですが、コントロールを含まないシンボルは最大 5000% までズームできます。コントロールを含まないシンボルは、スムーズにスケールすることができます。ズームを行う際、埋め込まれたコントロールを含むシンボルを使用する場合のその他の制限は、コントロールが明滅することです。これは、タッチ ジェスチャでズームする場合に特に目に付きます。

枠ウィンドウと ShowGraphic() 関数を使用

ShowGraphic() スクリプト関数を実行して、枠ウィンドウに関連付けられたシンボルをランタイム時に変更できます。次の例に示すように、枠ウィンドウ名および関連付けられたシンボルを指定します。

```
暗いグラフィック InfoasaaGraphic.GraphicInfo;
graphicInfo.Identity = "InTouch:FrameWindow01";
graphicInfo.GraphicName="symbol002";
ShowGraphic( graphicInfo );
```

枠ウィンドウのデザイン時の設定に関係なく、「Symbol_002」に「FrameWindow01」がランタイム時に表示されます。ShowGraphic スクリプトを使用して、同じウィンドウで別のシンボルをホストできます。

注意： InTouch HMI 2020 R2 SP1 P01 より前のバージョンでは、Y 値が 200 を超えた場合にのみ、顕著な位置の変化が発生します。Y 値が 200 を下回る場合、グラフィックの位置に変化がないか、または最小限のシフトしか行われないため、観察および検証が困難です。InTouch HMI 2023 より後のバージョンでは、Y 値が 300 を超えた場合にのみ、顕著な位置の変化が発生します。Y 値が 300 を下回ると、グラフィックの位置に変化がないか、またはごくわずかなシフトしかないため、観察や検証が困難になります。

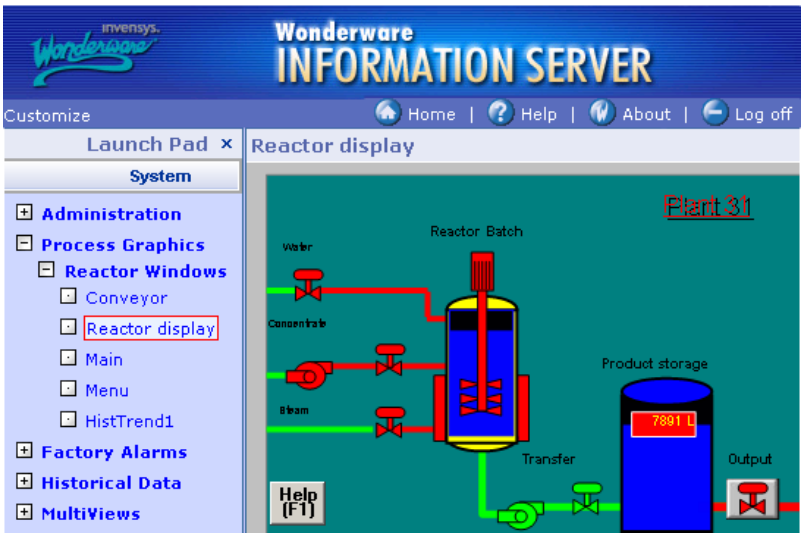
制限事項:

グラフィックのキャッシュ処理は、ウィンドウが閉じてキャッシュされるときに枠ウィンドウに現在表示されているシンボルに対してのみ行われます。置換されたシンボルはキャッシュされません。

インターネット上での InTouch ウィンドウの実行

Information Server は、Web または社内のイントラネット上の工場での製造データを集計および表示することができる Web ポータル アプリケーションです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



以下の方法で Information Server で InTouch を使用できます。

- プロセスのビジュアル化

  InTouch アプリケーションを Information Server ポータルにパブリッシュして、製造プロセスを表示することや、Web ブラウザを使用して制御することができます。

- データの操作

  Information Server ポータルを使用して、InTouch タグ変数から値を読み取ったり、値を書き込むことができます。これにより、InTouch クライアントを使用しなくても工場のプロセスを操作できます。

- アラーム表示

  Information Server ポータルを使用して、InTouch のリアル タイム データおよび履歴アラーム データを表示できます。

- 履歴データ表示

  Information Server ポータルを使用して、Historian データベースに保存された InTouch の履歴データを表示できます。

- Table Weaver 表示

  InTouch を使用して、Table Weaver コンテンツ単位の表示を作成できます。

詳細については、Information Server のマニュアルを参照してください。

実行時に表示されるウィンドウを設定する

「起動時」ウィンドウは、ユーザーがアイコンまたはメニュー コマンドから WindowViewer を直接起動したときに WindowViewer に表示されるウィンドウです。

WindowViewer を起動するために［実行］高速切り替えを使用した場合、ホーム ウィンドウは表示されません。

スクリプトで ShowHome() 関数を使用すると、ランタイム時にいつでもホーム ウィンドウを表示できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

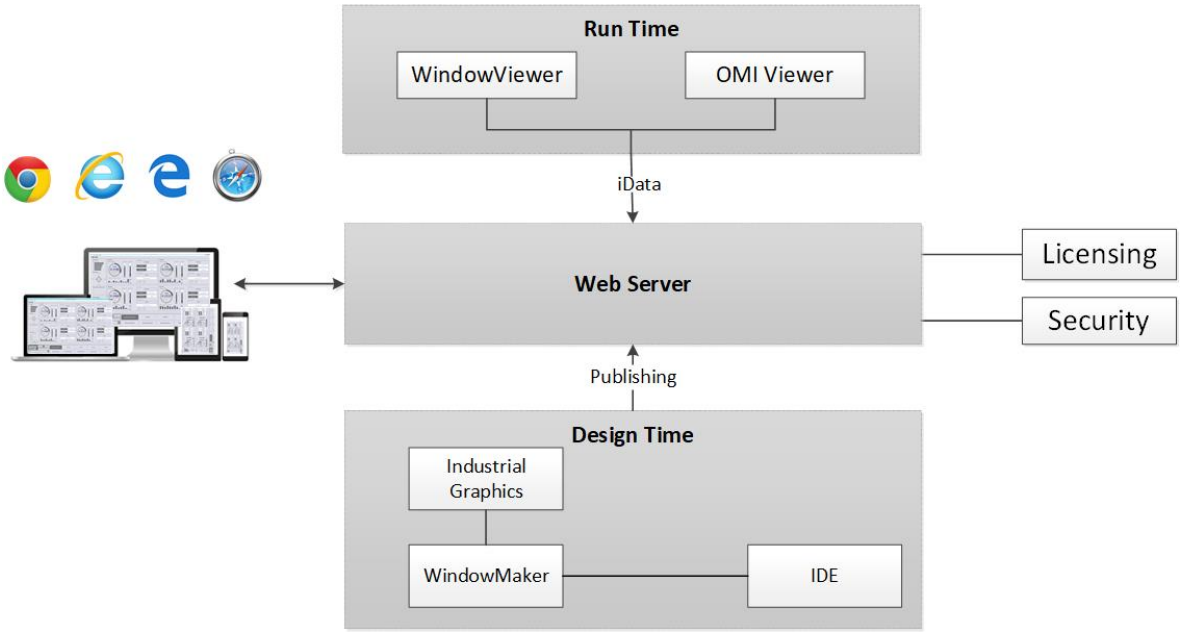
ホームウィンドウを設定

1. WindowMaker を開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［**WindowViewer**］を選択します。**WindowViewer** の設定画面が表示されます。
3. ［起動］タブをクリックします。
4. WindowViewer の起動時に開くウィンドウを選択します（複数可）。
5. ［**OK**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 21 Web Client でのアプリケーショングラフィックを表示

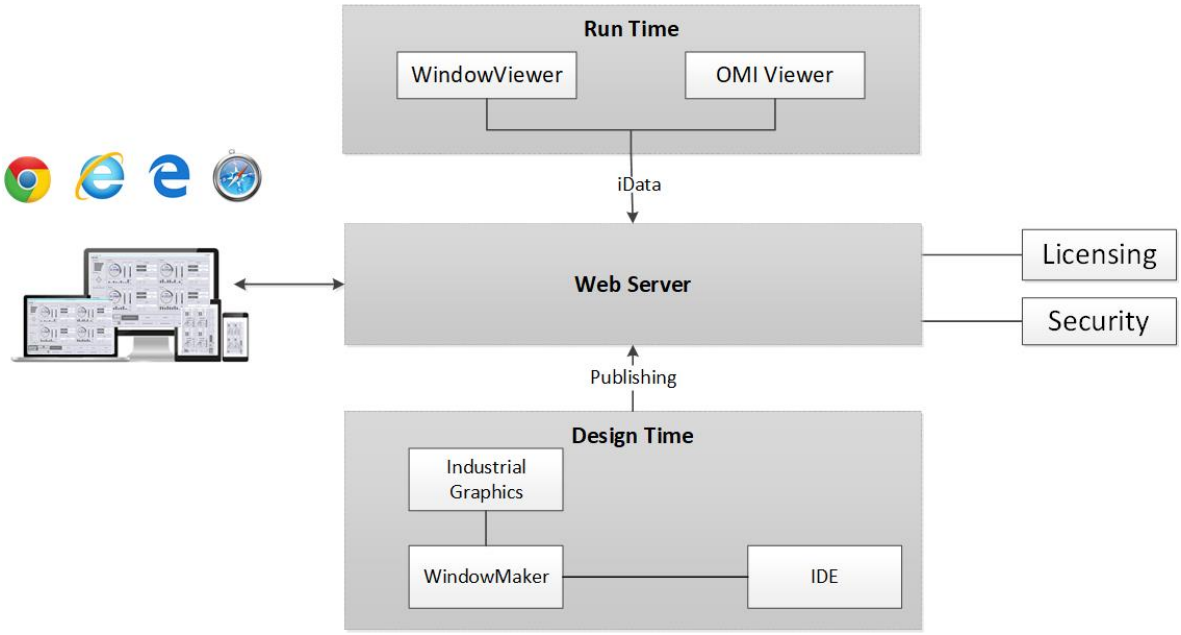
Web Client では、HTML5 をサポートする任意の Web ブラウザで AVEVA InTouch HMI アプリケーションおよび AVEVA OMI ViewApp を表示できます。Web Client は、モバイル プラットフォームでも使用できます。組み込みの Web Server により、Microsoft Windows® Server のリモート デスクトップ プロトコル（RDP）またはインターネット インフォメーション サービス（IIS）を使用することなく、任意の Microsoft Windows クライアントまたはサーバー オペレーティング システム内の Web ブラウザからアプリケーションにアクセスできるようになります。



Web Client を使用

Web Client では、HTML5 をサポートする任意の Web ブラウザで AVEVA InTouch HMI アプリケーションおよび AVEVA OMI ViewApp を表示できます。Web Client は、モバイル プラットフォームでも使用できます。組み込みの Web Server により、Microsoft Windows® Server のリモート デスクトップ プロトコル（RDP）またはインターネット インフォメーション サービス（IIS）を使用することなく、任意の Microsoft Windows クライアントまたはサーバー オペレーティング システム内の Web ブラウザからアプリケーションにアクセスできるようになります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



InTouch Web Client アプリケーションを設計

InTouch Web Client アプリケーションの設計には以下のステージがあります。

設定（オプション）

1. セキュリティ プロトコルを設定します。
   a. System Platform コンフィグレータを使用して System Management Server に接続します（リモート サーバーまたはローカル サーバーを選択します）。
2. AVEVA Identity Manager を設定します。
3. リバース プロキシ サーバーを設定します。

設計

1. アプリケーションを作成または編集します。
2. Web Client に表示するすべてのグラフィックを含むルート フォルダを識別します。
3. Web Client ルート フォルダを識別して設定します。
4. Web Client ホームのシンボルを識別して設定します。
5. InTouch WindowViewer への高速切り替えを行います。
   - InTouch タグにアクセスするには WindowViewer が実行している必要がありますが、アプリケーション サーバー オブジェクト属性参照にアクセスする場合は必要がありません。
6. Web Client の高速切り替えボタンを使用するか、ブラウザを起動して http://localhost/InTouch にアクセスします。

反復的開発

1. ホームのシンボルが Web ページに表示されます。
2. ナビゲーション オーバーレイを使用してシンボル/ウィンドウに移動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. WindowMaker でグラフィック関連の変更を行うと、その変更はブラウザに自動的に反映されます。

   クォリティとステータス スタイル、要素スタイル、および書式設定スタイルの変更は、グラフィックに対して以下の変更を行った場合または WindowViewer を再起動した後にのみ適用されます。

   - シンボルの内容の更新および保存
   - シンボルの作成、インポート、または削除
   - 別のツールセット フォルダへのシンボルの移動
   - ルート フォルダまたはホームのシンボルの割り当ての更新
4. アプリケーションの構築を続行して、出力をブラウザでテストできます。

配置

1. ユーザーを Web Client 関連のユーザー グループ（ローカルまたはドメイン）に割り当てるか、匿名アクセスを設定します。
2. アプリケーションをパブリッシュまたは配置します。
3. アプリケーションを InTouch WindowViewer で実行します。InTouch タグからライブ データを受信するには、WindowViewer が実行している必要があります。

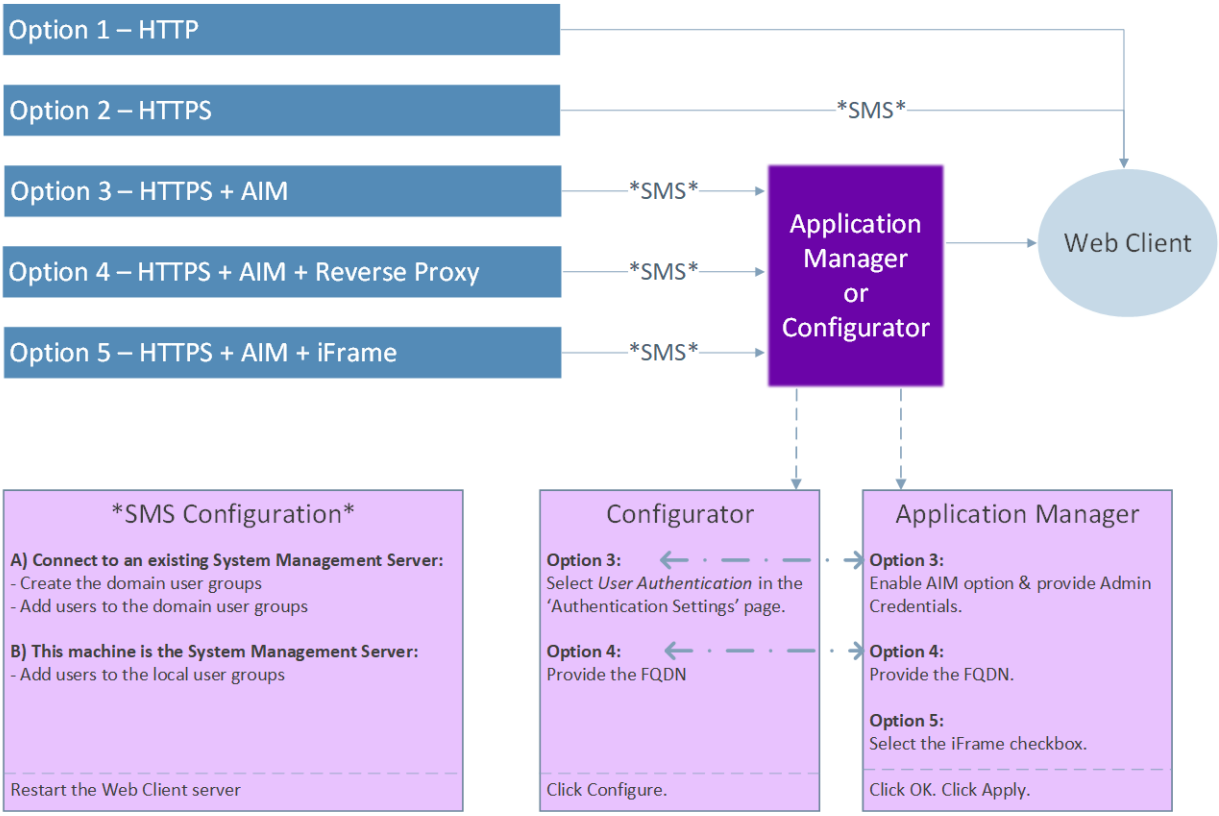
   パブリッシュまたは配置したアプリケーションには、イントラネット上の任意のコンピュータからアクセスできます。

   http://<IPAddress>/InTouch または http://<NodeName>/InTouch を指定できます。<IPAddress> または <NodeName> は、アプリケーションがパブリッシュまたはデプロイされるマシンに対応します。

セキュアな InTouch ウェブクライアントアクセス

System Management Server（SMS）を設定すると、HTTPS プロトコルで InTouch Web Client を安全に使用できます。さらに、SMS を設定した後、 InTouch Web Client を AVEVA Identity Manager（AIM）と共に使用して、Windows ベースでないセキュリティおよびシングル サインオンをサポートできます。管理者は、ユーザー認証およびセキュリティの複数の設定オプションを使用できます。セキュリティおよびユーザー管理は連動させる必要があります。詳細については、「Web Client でのユーザー アクセスの設定」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



オプション 1：デフォルトでは、InTouch Web Client には http プロトコルおよび Windows ベースの認証を使用してアクセスできます。

オプション 2：コンフィグレータを使用してローカルまたはリモートの System Management Server に接続します。ここで、InTouch Web Client は https プロトコルと Windows ベースのセキュリティを使用します（デフォルト）。リモート サーバーに接続する場合、Web Client にアクセスするときにドメイン ユーザーの資格情報を入力する必要があります。

オプション 3：System Management Server を設定した後、AVEVA Identity Manager（AIM）を有効化できます（オプション）。AIM は、OpenID Connect エンドポイントを公開するスタンドアロン認証サーバーです。AIM を使用するには、クライアントを Identity Server に登録する必要があります。AIM は、System Management Server コンフィグレータまたは InTouch HMI アプリケーション マネージャを使用して設定できます。詳細については、「AVEVA Identity Manager での登録」を参照してください。

ランタイム時に InTouch Web Client ページがロードされると、有効なセキュリティ トークンがない場合、以下の処理が行われます。

- Web Client が AIM のログイン ページにリダイレクトされます。
- AIM が Active Directory のユーザー資格情報をチェックします。
- 資格情報が有効な場合、Active Directory からセキュリティ トークンが提供され、Web Client に返されます。
- Web Client がトークンのあるユーザーにアクセス権を付与します。

セキュリティ トークンが既に存在する場合、ユーザーにアクセス権が付与されます。AIM では、Active Directory で検証されたユーザーのみがサポートされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

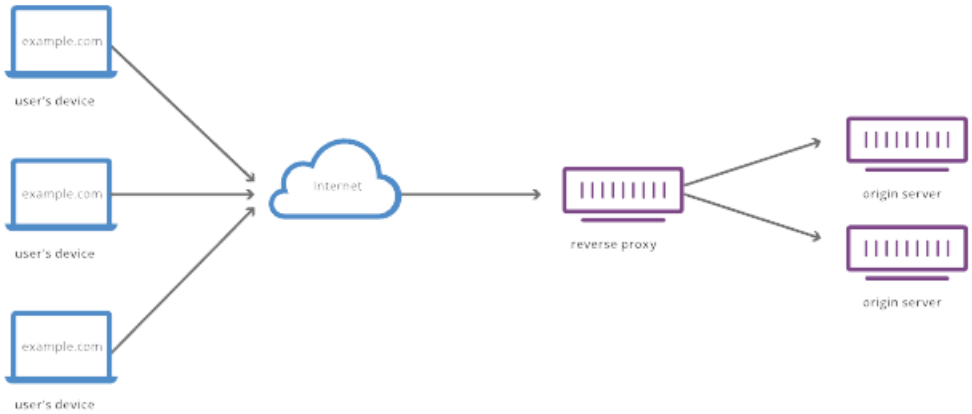
オプション 4：［リバースプロキシ］オプションを使用すると、OT ネットワーク外のユーザーが InTouch Web Client にアクセスできます。コンフィグレータのセキュア ゲートウェイ フィールドまたは InTouch HMI アプリケーション マネージャの［Web Client］タブにリバース プロキシ サーバーの FQDN を入力します。

オプション 5：ユーザー認証に AIM を使用する場合、グラフィックは iFrame に表示されません。［**Web サイトへの Web Client の埋め込みを許可する**］チェック ボックスをオンにすると、ランタイム時に iFrame 内にグラフィックを表示できます。ランタイム時に［共有］アイコンを使用して、iFrame に挿入するコード スニペットを選択します。

OT ネットワーク外からの Web Client へのアクセス

リバース プロキシは、クライアントからのリクエストを傍受するために Web Server の前に配置されているサーバーです。リバース プロキシを使用すると、クライアントから Web サイトのオリジン サーバーに送信されたリクエストはリバース プロキシ サーバーによってネットワーク エッジで傍受されます。その後、リバース プロキシ サーバーは、リクエストをオリジン サーバーに送信し、オリジン サーバーから応答を受け取ります。いずれかの Web Server がダウンしている場合、または Web Server のフェイルオーバーが行われている場合、リバース プロキシはクライアントからのリクエストを冗長 Web Server に送信します。

フォワード プロキシはオリジン サーバーがその特定のクライアントと直接通信することのないようにクライアントの前に配置されます。一方、リバース プロキシはオリジン サーバーの前に配置され、クライアントがオリジン サーバーと直接通信することを防止します。



OT ネットワーク外のデバイスで InTouch Web Client にアクセスするには、DMZ サーバー上でリバース プロキシを設定する必要があります。詳細は、「AVEVA DMZ Secure Link」を参照してください。リバース プロキシが機能するには、AVEVA Identity Manager を有効にする必要があります。多くのプロバイダがリバース プロキシ ソリューションを提供しています。該当するリバース プロキシ ソリューションのマニュアル、冗長サポートの情報、およびインフラストラクチャでリバース プロキシ ソリューションを設定する手順を参照してください。

AVEVA Identity Manager での登録

AVEVA Identity Manager（AIM）を使用すると、デフォルトの Windows OS ベースの認証ではなく、シングル サインオンを使用するように Web Client を設定できます。

注意：Web Client は、AIM 設定を含む ArchestrA ベースのセキュリティのみをサポートします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Identity Server を設定する

1. System Platform コンフィグレータで［**共通プラットフォーム**］>［**System Management Server**］を設定します。
2. AVEVA アプリケーション マネージャでユーザー資格情報を使用して AIM サーバーを登録します。

［AIM 登録］ダイアログ ボックスを使用して、リバース プロキシ サーバーを設定することもできます。

1. リバース プロキシ サーバーをセットアップします。
2. System Platform コンフィグレータで［**共通プラットフォーム**］>［**System Management Server**］を設定します。
3. ［AIM 登録］ダイアログ ボックスにセキュア ゲートウェイのアドレスを入力します。

リモートまたはローカルの System Management Server を選択できます。System Platform Server の設定の詳細については、『*System Platform インストール ガイド*』を参照してください。

Identity Server に登録する

1. アプリケーション マネージャで［**Web Client**］タブをクリックします。
2. ［**ツール**］、［**ID マネージャ**］の順にクリックします。

   ［**Identity Server 設定**］画面が表示されます。
3. ［**AIM サーバーを認証サーバーとして使用する**］チェック ボックスをクリックして、AVEVA Identity Manager の使用を有効にします。

   ［**Identity Server**］フィールドにコンフィグレータで設定した Identity Server が表示されます。
4. 以下の設定を更新します。
   a. **[ユーザー名]**：Identity Server に接続するユーザー名を入力します。ユーザーは Administrators ユーザー グループに属する必要があります。
   b. **[パスワード]**：入力したユーザー名のパスワードを入力します。
5. リバース プロキシのセットアップを完了するには、リバース プロキシまたは DMZ サーバーの URL を［**セキュア ゲートウェイ**］フィールドに入力します。

   これはオプションの設定で、Web Client がリバース プロキシ サーバーの背後でホストされる場合にのみ設定する必要があります。詳細は、「AVEVA DMZ Secure Link」を参照してください。
6. ［**Web サイトへの産業用グラフィックの埋め込みを許可する**］チェック ボックスをクリックして、ランタイム時に HTML iframe 内に Web Client を表示することもできます（オプション）。
7. **[OK]** を選択します。
8. **[適用]** を選択します。

注意：Web サイトの名前またはアドレスが変更された場合、AVEVA Identity Manager を設定して新しい Web サイトを登録する必要があります。

シングルサインオンを使用した InTouch Web Client へのアクセス

コンフィグレータで［System Management Server］オプションが有効化されている場合、シングル サインオンを使用して InTouch Web Client にアクセスできます。これは、デフォルトの Windows ベースの認証とは異なります。この方法では HTTPS プロトコルが使用され、Microsoft Active Directory から検証でき

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

るドメイン ユーザーだけがサポートされます(リモート SMS を使用する場合)。HTTP プロトコルを使用する場合、Web Client は Windows ベースの認証のみを使用するようにフォールバックします。

System Platform Server の設定の詳細については、『*AVEVA System Platform インストール ガイド*』を参照してください。

1. Web Client を起動します。

   ログイン ページが表示されます。

2. ユーザー名およびパスワードを入力します。
3. **[ログイン] を選択します。**

   ［**Windows 統合**］を選択すると、現在ログインしているユーザーの資格情報を使用できます（オプション)。

   ユーザーは、Identity Manager の資格情報でログに記録されます。

4. セッションをログアウトするには、［**サイン アウト**］を使用します。

AIM でのローカルタグを使用

Web Client のローカルタグの動作は、WindowViewer に依存しません。AIM が有効化されている場合、ローカル タグの動作は異なります。AIM が有効化されている場合、各ユーザーの各セッションで一意のトークンが使用されます。

	ユーザー 1 – セッション 1	**ユーザー 1 – セッション 2**	**ユーザー 2 – セッション 1**
AIM が無効化されている場合	同じタグ値	同じタグ値	別のタグ値
AIM が有効化されている場合	別のタグ値	別のタグ値	別のタグ値

InTouch Web Client でのユーザーアクセスの設定

匿名での InTouch Web Client へのアクセス

InTouch Web Client では、匿名ユーザー アクセスおよび認証ユーザー アクセスの両方を使用できます。デフォルトでは、匿名アクセスは無効化されています。このオプションを有効にするには、InTouch HMI アプリケーション マネージャの［Web Client］タブの［**匿名アクセスを許可**］チェック ボックスをオンにします。ランタイム時にユーザー資格情報を入力する必要はありません。ユーザーは「ゲスト」ユーザーとしてログインされます。ゲスト ユーザーは読み取り専用で Web Client にアクセスできます。

InTouch Web Client の認証ユーザーの設定

認証およびライセンスの両方のワークフローでは、ユーザー グループを使用してアクセス権を付与するユーザーおよび Web セッション用に取得するライセンスのタイプを決定します。ユーザー グループは、SMS の設定に応じてローカル ノードまたはリモート ノードで設定できます。アクセスが必要なすべてのユーザーは、アクセス レベルに応じていずれかのグループに含める必要があります。

リモート System Management Server を使用するように Web Client が設定されている場合、リモート サーバーに aaInTouchUsers および aaInTouchRWUsers というドメイン ユーザー グループを作成します。アプリケーションを配置する前に、これらのグループに該当するすべてのユーザーを追加します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch Web Client の仮想アカウントの使用

Web Client 仮想サービス アカウントの特権は、セキュリティを向上させる目的で削減されています。InTouch アプリケーションに対して、Web Client を継続して使用するには、InTouch Web Client 仮想サービス アカウントに InTouch アプリケーション フォルダへの読み取り/書き込みアクセス権限が必要です。InTouch Web サービスの仮想サービス アカウントは、"NT SERVICE\AIGWebServer" と呼ばれます。

InTouch Web Client の仮想サービス アカウントには、次のアプリケーションに対する読み取り/書き込みアクセス権限が付与されます。

- 既存のすべての InTouch アプリケーション
- アプリケーション マネージャから作成/インポートされたすべての InTouch アプリケーション
- ［検索］機能で見つかったすべての InTouch アプリケーション（見つかったアプリケーションに対する読み取り/書き込みアクセス権限を Web Client 仮想アカウントに付与するかどうかを確認するダイアログで［はい］を選択する必要があります）

InTouch アプリケーションへのアクセス権限を InTouch Web Client 仮想アカウントに手動で付与するには

1. ファイル エクスプローラを使用して InTouch アプリケーション フォルダに移動します。
2. フォルダを右選択して［**プロパティ**］を選択します。
3. ［**プロパティ**］ダイアログの［**セキュリティ**］タブで［**編集**］を選択します。
4. ［**グループ名またはユーザー名**］セクションで［**追加**］を選択します。

   ［**ユーザー、コンピュータ、サービス アカウント、またはグループの選択**］ダイアログが表示されます。
5. **[場所]** を選択します。

   ［**場所**］ダイアログが表示されます。
6. コンピュータ名を選択し、［**OK**］を選択します。
7. ［**選択するオブジェクト名を入力してください**］フィールドに「NT SERVICE\AIGWebServer」と入力します。
8. ［**OK**］ を選択します。
9. **Permissions for Authenticated Users** セクションの **Allow** 列で、チェックボックスを選択する： Read & execute（読み取りと実行）、List folder contents（フォルダの内容のリスト）、Read（読み取り）、Write（書き込み）。
10. ［**OK**］ を選択します。

注意：アプリケーションのパスが "\\" で始まる場合、ファイル共有のパスを示します。InTouch Web Client 仮想サービス アカウントには、共有場所にあるアプリケーション フォルダへのアクセス権限は付与されません。

InTouch Web Client 機能の有効化

デフォルトでは、インストール時に InTouch Web Client 機能は無効化されます。InTouch Web Client 機能を使用するには、この機能を有効にする必要があります。コンフィグレータで以下のオプションのいずれかを設定すると、InTouch Web Client の機能は自動的に有効化されます。

- ［**共通プラットフォーム**］>［**System Management Server**］

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［アプリケーション マネージャ］>［**Web Client**］

コンフィグレータの詳細については、『*System Platform インストール*』を参照してください。Web Client の機能を手動で有効化または無効化するには、管理特権が必要です。Web Client の機能は、アプリケーション マネージャまたは InTouch WindowMaker から有効化できます。

アプリケーションマネージャからの InTouch Web Client 機能の有効化

1. アプリケーション マネージャを起動します。
2. ［Web Client］タブを選択します。
3. ［ファイル］メニューの［メイン］グループで［**Web Client**］を選択します。
4. 無効化するには、［**Web Client**］を再選択します。

WindowMaker からの InTouch Web Client 機能の有効化

管理特権のあるユーザーは、WindowMaker から Web Client 機能を有効にすることができます。

1. WindowMaker でアプリケーションを起動します。
2. **Web Client** の高速切り替えボタンを選択します。

   次のダイアログボックスが表示されます :「このシステムでは、InTouch Web Client の機能は現在、無効化されています。この機能を有効にしますか?」というメッセージを含むダイアログ ボックスが表示されます。
3. ［はい］を選択します。

   InTouch Web Client ページが表示されます。

無効化されている Web クライアントを起動しようとすると、「**HTTP Error 404.0 – Not Found**」というエラー メッセージが表示されます。

管理者以外のユーザーが WindowMaker から Web Client 機能を有効にしようとすると、アプリケーション マネージャから機能を有効にするようリダイレクトされます。

InTouch Web Client のカスタマイズ

AVEVA アプリケーション マネージャの［Web Client］タブには、Web Client 関連の設定を構成するオプションがあります。

InTouch Web Client の設定を更新

1. アプリケーション マネージャを起動し、［**Web Client**］タブをクリックします。

   ［**Web Client の設定**］画面が表示されます。
2. 以下の設定を構成します。
   - *[現在のアプリケーション]* : Web Client の設定を変更するアプリケーションをリストから選択します。
   - *[グラフィック更新レート（ms）]* : Web ブラウザが Web サーバー にグラフィック データをクエリーする頻度を設定します。デフォルトは 1 秒です。詳細については、『System Platform インストール ガイド』を参照してください。
   - *アラーム更新レート（ms）* : Web ブラウザが Web サーバー にアラーム データをクエリーする頻度を設定します。デフォルトは 1 秒です。詳細については、『System Platform インストール ガイド』を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- *[ヘッダーを表示]*：タイトル バーを表示するには、チェック ボックスをオンにします。
- *[ナビゲーションバーを有効化]*：このチェック ボックスをオンにするとナビゲーション バーが表示されます。

  ［**ヘッダーを表示**］設定が選択されていない場合、この設定は無効化されます。
- *[匿名アクセスを許可]*:ユーザーが認証なしで Web Client にアクセスすることを許可する場合は、このチェック ボックスをオンにします。
- *[作業領域を有効化]*：作業領域を有効にするには、このチェック ボックスをオンにします。

3. ［詳細設定］セクションで［**カスタマイズ**］をクリックして以下の設定を構成します：
   - *[ウェブサイト名]*：標準の URL を置き換える文字列を入力します。
   - *[アプリケーションのタイトル]*:アプリ バーのアプリケーションのタイトルとして表示する文字列を入力します。
   - *[Web サイトのタイトル]*: ブラウザのタイトルバーにタイトルとして表示する文字列を入力します。
   - *[Web サイトのアイコン]*:ブラウザのタイトル バーのアイコンを置換する画像ファイルを入力します。
     i. ［**ファイルを選択...**］をクリックしてアイコン画像を選択します。
     ii. ファイルを選択します。
     iii. **[開く]**を選択します。
4. ［**保存**］をクリックして詳細設定を保存します。
5. 設定を変更したら、**[適用]**を選択 します。
6. Web Client を表示するには、［**起動**］をクリックします。

注意：Web サイトの名前またはアドレスが変更された場合、AVEVA Identity Manager を設定して新しい Web サイトを登録してください。詳細については、「AVEVA Identity Manager での登録, AVEVA Identity Manager での登録」を参照してください。

これらの設定の詳細については、「*Web ブラウザでのアプリケーション グラフィックの表示*」を参照してください。

InTouch Web Client で表示するためのグラフィックを設定する

グラフィック ツールボックスを使用して、InTouch Web Client に表示されるアプリケーション グラフィックを含むフォルダを設定できます。Web ブラウザに表示されるのは、Web Client のルート フォルダとして設定したフォルダ内に格納されているグラフィックだけです。

注意:インストール時に **aaInTouchUsers** および **aaInTouchRWUsers** グループが InTouch ランタイム ノードに作成されます。Web ブラウザでアプリケーションのグラフィックにアクセスできるのは、aaInTouchUsers または aaInTouchRWUsers ユーザー グループに含まれるユーザーだけです。アプリケーションを構成する前に、関連するユーザーをグループに追加します。デフォルトでは、インストールを行ったユーザーがグループに追加されます。

---

1. InTouch WindowMaker を起動します。
2. アプリケーションを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

産業用グラフィック ツールボックスで、Web ブラウザに表示するフォルダの階層およびグラフィックを含めるフォルダを作成または選択します。

3. フォルダーを右選択して、 **[Set The Web Client Root Folder]** を選択します。

   アイコンのサムネイルが変更され、設定が反映されます。

アイコン	説明
	ルート フォルダが Web Client のルート フォルダとして設定された場合
	ルート フォルダ以外のフォルダが Web Client のルート フォルダとして設定された場合

   デフォルトでは、グラフィック ツールボックスのルートが Web Client のルート フォルダとして設定されます。

InTouch Web Client のホームシンボルを設定

Web クライアント ホームのシンボルを設定して、起動時に InTouch Web クライアント ページに表示するデフォルトのシンボルを設定できます。

1. Web Client のルート フォルダを設定します。「Web ブラウザで表示するための InTouch アプリケーション グラフィックの設定」を参照してください。

   ホームのシンボルは、Web Client ルート フォルダの子である必要があります。

2. Web Client ルート フォルダの下でホームのシンボルを見つけます。
3. シンボルを右クリックして、［**Web Client ホームのシンボルの作成**］を選択します。

アイコンのサムネイル（ ）が変更され、設定が反映されます。ホームのシンボルが設定されていない場合、起動時の Web Client ページには空白のページが表示されます。ブラウザで Web Client を表示しているときにアプリバーの［ホーム］アイコンを選択してホームのシンボルに移動します。

InTouch Web Client ホームアイコンの動作の理解

スタンドアロン アプリケーションは Web Client 上で表示できます。スタンドアロン アプリケーション ウィンドウを Web Client で表示するには、産業用グラフィックに変換する必要があります。Web Client で表示できるのは、枠ウィンドウと産業用グラフィックだけです。

InTouch アプリケーションの場合、WindowMaker で［ホーム ウィンドウ］タブを使用して、WindowViewer でデフォルトで表示されるウィンドウを設定できます。詳細については、「*ランタイム時にウィンドウを表示する設定*」を参照してください。同様に、Web Client を起動したときに表示されるデフォルト シンボルとしてグラフィックを設定できます。「InTouch Web Client のホームシンボルを設定」を参照してください。

InTouch ShowHome スクリプト関数を使用した場合、ウィンドウをグラフィックに変換すると、スクリプトは ShowHome() 関数に変換され、対応するグラフィックがホーム シンボルとして WindowViewer に表示されます。

これらの設定は、InTouch Web Client の起動時にデフォルトで表示されるウィンドウまたはグラフィックに影響します。InTouch HMI アプリケーションの場合、2 つの可能性があります。

1. ホーム ウィンドウが設定されている場合

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザーがホーム シンボルを設定している場合、すべての状況で、デフォルトでホーム ウィンドウだけが表示されます。［ホーム］アイコンを選択すると、ホームウィンドウが表示されます。

2. ホーム ウィンドウが設定されていない場合

   ホーム シンボルが設定されている場合、デフォルトでホーム シンボルが表示されます。

   ホーム シンボルが設定されていない場合、デフォルトではシンボルやホーム ウィンドウは表示されません。ホームシンボルを選択してもグラフィックやウィンドウは表示されません。

Web ブラウザでのアプリケーションの表示

InTouch Web Client アプリケーションを設定した後、Web ブラウザでアプリケーションを表示できます。テスト済みのサポートされるブラウザのリストについては、「ブラウザとモバイルのサポート」セクションを参照してください。

注意：InTouch Web Client が正しく機能するためには、常に最新バージョンのブラウザーをインストールするか、使用できる最新バージョンにブラウザーをアップグレードすることをお勧めします。

1. HTML5 対応のブラウザを起動し、URL を入力します： `http://<hostname>/InTouch` または `http://<IPAddress>/InTouch.`

   資格情報が認証され、有効な Web Server ライセンスが取得されると、InTouch Web Client ページが表示されます。デフォルトで、ホームのシンボルが表示されます。

   URL: https://<hostname>/InTouch o または r https://<IPAddress>/InTouch を使用して Web Client にアクセスできます。サーバーとクライアント間の暗号化された通信用に共通プラットフォームの System Management Server が証明書（ローカルまたはリモート サーバー）で設定されている場合、Web Client では HTTPS プロトコルが自動的に使用されます。Web Client ページが表示されない場合は、「Web ブラウザでのグラフィックの表示に関するトラブルシューティング」を参照してください。

   Web Client の URL のカスタマイズについては、「InTouch Web Client のカスタマイズ」を参照してください。

2. ハンバーガー アイコンを選択します。

   左側のオーバーレイで、実行中のアプリケーション内の選択した Web Client のルート フォルダの下にグラフィックの階層構造が表示されます。

3. フォルダ名を選択します。

   フォルダ内のグラフィックが表示されます。

4. 画像を選択します。

   選択したグラフィックが InTouch Web Client に表示されます。

Web ブラウザに表示されるのは、Web Client のルート フォルダとして設定したフォルダ内に格納されているグラフィックだけです。匿名アクセスが無効な場合、グラフィックにアクセスするユーザー アカウントは認証および承認が必要です。詳細については、「InTouch アプリケーショングラフィックスのライセンスを取得する」を参照してください。

注意：向上したグラフィック レンダリングと全体的なパフォーマンス改善を活用するために、GPU ハードウェアアクセラレーションを有効にすることが推奨されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Web Client 内のシンボルで InTouch タグからのライブ データを受信するには、読み取り/書き込み InTouch ライセンスで InTouch WindowViewer がホスト マシン上で実行している必要があります。WindowViewer をサービスとして実行することもできます。アプリケーション オブジェクトからライブ データを受信するには Application Server が実行している必要があります。

注意: 読み取り専用 InTouch ライセンスはリモート データ接続をサポートしないので、InTouch Web Client のデータ ソースとして機能するための十分な機能を提供しません。InTouch Web Client にデータをフィードするすべての WindowViewer アプリケーションは、読み取り/書き込み InTouch ライセンスで実行する必要があります。

InTouch Web Client の高速切り替え

アプリケーションの開発時に Web Client の高速切り替え機能を使用して、WindowMaker と InTouch Web Client を切り替えることができます。

- Web Client の高速切り替え機能を使用するには、InTouch WindowMaker の［**Web Client**］ボタンを選択します。

同じアプリケーションを WindowMaker で編集し、Web Client で表示している場合、WindowMaker で行った変更が Web Client に動的に反映されます。自動更新が機能するのは、一度に 1 つのアプリケーションを編集および表示している場合だけです。InTouch Web ページは以前に表示されていたシンボルに更新されます。別のアプリケーションに対して WindowViewer を実行している場合、Web Client ページは、そのアプリケーションに自動的に切り替わります。

- ［**グラフィックのリフレッシュ レート**］は、**コンフィグレータ**から指定できます。デフォルトでは、ポーリング レートは 1000 ミリ秒です。
- ［**アラームのリフレッシュ レート**］は、**コンフィグレータ**から指定できます。デフォルトでは、ポーリング レートは 1000 ミリ秒です。

アラーム リフレッシュ レートにはグラフィック リフレッシュ レートよりも高い値を設定してください。また、グラフィック リフレッシュ レートの値は、180 秒未満に設定してください。

注意 : リモート IDE からマネージド InTouch アプリケーションを編集する際に InTouch Web クライアントを高速切り替えモードで実行するには、GR マシンとリモート IDE マシンの両方の **aaConfigTools** ユーザーグループのメンバーであるユーザーアカウントで **InTouch Web Windows** サービスを実行するように設定します。詳細については、『アプリケーションサーバーのユーザーマニュアル』を参照してください。

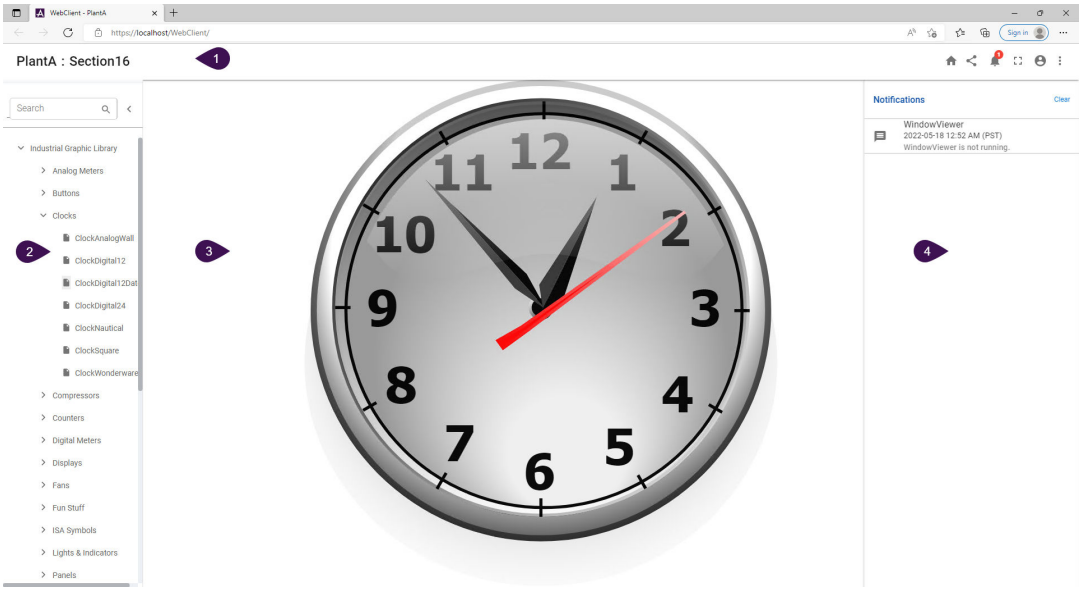
InTouch アプリケーションに対して、WindowMaker の Web Client の高速切り替えボタンはデフォルトで無効化されています。 アプリケーションを WindowViewer で実行すると、このボタンは自動的に有効になります。

注意 : Web Client の高速切り替え機能は、次の場合に使用できます。
- 開発時のみ
- InTouch WindowMaker および InTouch WindowViewer がインストールされているコンピュータ上のみ
- URL で *http://localhost/InTouch* をポイントして Web Client にアクセスする Web ブラウザ上のみ

InTouch Web クライアントページの理解

InTouch Web Client ページは、ページ上のアイコンを使用してユーザー エクスペリエンスに関する情報の提供、およびユーザー エクスペリエンスの編成を行うことのできるさまざまな機能があります。これらのオプションの説明を次の表に示します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



要素	説明
1	アプリ バーには、InTouch アプリケーションの名前、シンボル名、ホーム アイコン、通知アイコン、全画面アイコン、およびプロファイル アイコンが表示されます。
アプリケーション名:シンボル名	アプリ バーにはアプリケーションとシンボルの名前が表示されます。 例 – *PlantA:セクション 16* - この例では、PlantA がアプリケーションの名前で、Section16 がシンボルの名前です。
	選択すると Web Client ホームのシンボルとして設定されているシンボルが表示されます。
	選択するとグラフィックをクリックすると、iFrame コードスニペットが表示されます。
	ライセンスのステータスに関連する通知アイコンメッセージを選択すると、WindowViewer のステータスの変化に関するメッセージが表示されます。通知の数がアイコンの上に表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

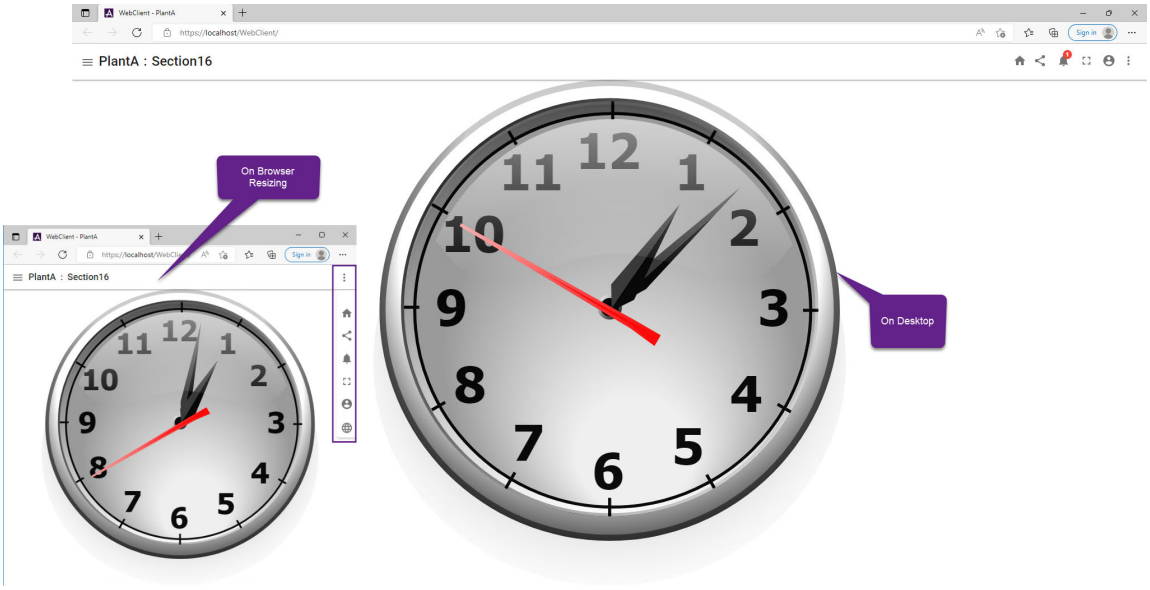
	選択するとキャンバス領域が最大化され、Web ページが全画面モードで表示されます。F11 キーを押しても同じ操作を行うことができます。全画面モードではアプリ バーとナビゲーション オーバーレイは表示されません。 **タイトル バーを表示するには** • カーソルをページの上部に移動するとアプリ バーが下にスライドして表示されます。 • 下向き矢印を選択すると、アプリバーがページにピン留めされます。 • 上向き矢印を選択すると、アプリバーが非表示になります。
	選択するとキャンバス領域が最小化され、全画面モードが終了します。
	プロファイル アイコンには、ログインしているユーザーが表示されます。
Sign Out	プロファイルアイコンを選択して、［**サインアウト**］アイコンをクリックします。［サイン アウト］は、System Management Server オプションが設定されている場合にのみ使用可能です。詳細については、「シングル サインオンを使用した InTouch Web Client へのアクセス」を参照してください。
	アプリバーから ⋮ を選択して［**言語**］アイコンをクリックします。言語を選択します。選択した言語でツールヒント、翻訳された静的テキスト、およびカスタム プロパティが表示されます。表示される言語は WindowMaker で設定されている言語だけです。 **注意：**言語を変更すると、新しいセッションが作成され、ローカルタグのデータは失われます。
2	アイコンを選択して、フォルダ ナビゲーション オーバーレイを表示します。選択した Web Client ルート フォルダの下にあるすべてのフォルダとシンボルが表示されます。 オーバーレイ上部の検索ボックスを使用して、シンボルを検索できます。検索ボックスではオートコンプリート機能が有効になり、検索用語を入力すると入力した検索用語に一致するシンボル名が一覧表示されます。シンボル名を選択すると、シンボルがキャンバス領域にシンボルが表示されます。
3	キャンバス領域には、ナビゲーション オーバーレイで選択したシンボルが表示されます。シンボルのサイズは、アスペクト比を維持した状態でブラウザのビューポート サイズに合わせて調整されます。ナビゲーションおよび通知オーバーレイはキャンバスのスペースを使用しません。また、グラフィックのサイズにも影響しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4	通知アイコンを選択して。接続とライセンスの問題に関する通知を表示します。
⚠	警告アイコンを選択すると、InTouch Web Client でサポートされないアニメーションまたはプロパティのリストが表示あれます。このアイコンは、アプリケーション開発時およびブラウザが http://localhost/InTouch という URL をポイントしている場合にのみ使用できます。

アプリケーション グラフィックおよびブラウザのサイズ調整

InTouch Web Client ページは、さまざまな画面サイズで表示できます。ブラウザを調整すると、新しいブラウザのビューポート寸法に合わせてグラフィックのサイズが変更されます。グラフィックのサイズは、モバイル デバイスを縦向きまたは横向きに回転させたときにも変更されます。



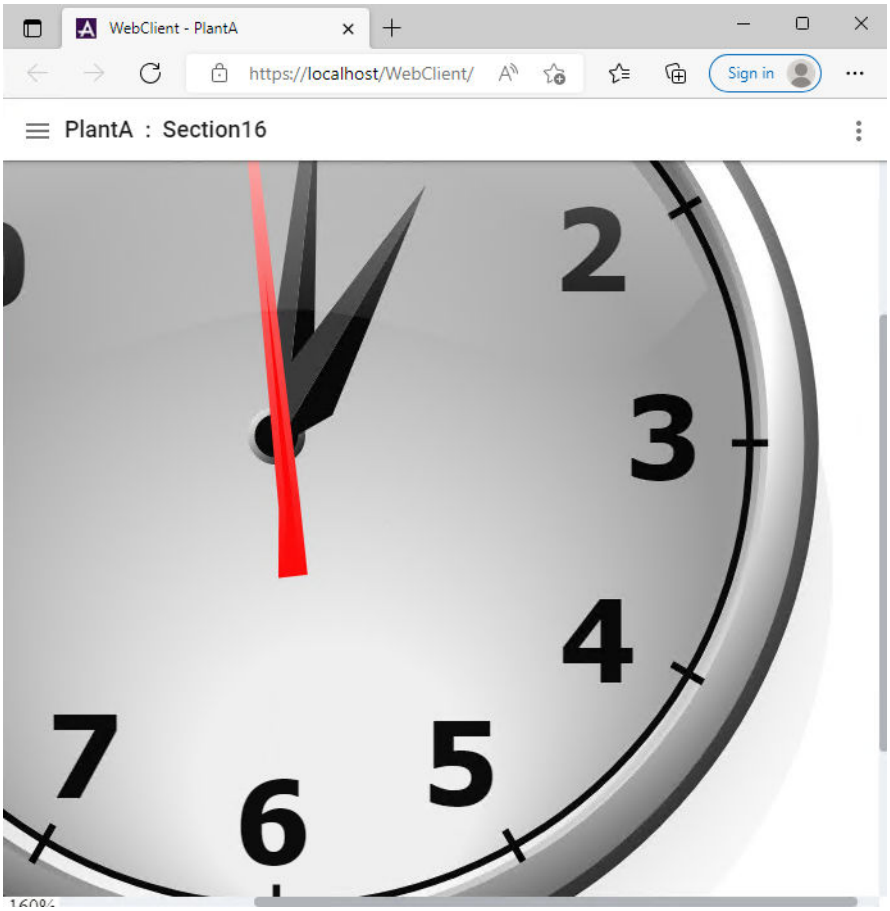
アプリ バーと小さい画面

画面が非常に小さい場合、右側のアイコンは、垂直バーに格納されます。画面のスペースを節約するためにアイコンは非表示になります。産業用グラフィックは任意のモバイル デバイスで表示して、パンとズームのジェスチャを使用してグラフィックを拡大表示することが可能です。

パンとズームのサポート

InTouch Web Client は、サポートされるすべてのブラウザおよびデバイスでのパンとズームのジェスチャをサポートします。モバイル デバイス以外のデバイスで Web ブラウザに表示されているグラフィックをズーム インすると、水平スクロールバーの左下にズーム率が表示されます。ズーム率は最大 500% に制限されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



タッチ操作によるパンとズーム

タッチ対応ディスプレイでは、2 本の指を使用してズーム インおよびズーム アウトを行うことができます。

- ズーム インするには、2 本の指を画面に置いて指の間の幅を広げます。ズーム アウトするには、2 本の指の間を狭くします。
- ダブルタップすると 100% ズームの表示に戻ります。

表示をパンするには、1 本の指で画面を押したまま指を移動します。

マウス操作によるパンとズーム

マウスを使用してズーム インおよびズーム アウトを行うことができます。

- ズーム インまたはズーム アウトするには、**Ctrl** キーを押したままホイールを上または下にスクロールします。マウス ポインタがペインの外にある場合、ズームは行われません。
- マウスの左ボタンをダブルクリックすると、100% ズームの表示に戻ります。

ディスプレイをパンするには、マウス ホイールを押した状態でマウスを目的の領域に移動してホイールを離します。

キーボード操作によるズーム

キーボードを使用してズーム インおよびズーム アウトを行うことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ズーム インするには、**Ctrl** キーと **+** キーを同時に押します。ズーム アウトするには、**Ctrl** キーと **-** キーを同時に押します。

場所によるズーム

選択した場所からグラフィックをズーム インすることもできます。

パンおよびズーム ツールバーはサポートされていません。パンおよびズームのジェスチャは、ShowSymbol アニメーションまたは ShowGraphic スクリプト、ボタン、選択、およびスライダーを使用して表示されるグラフィックに対してサポートされません。

外部 Web サイトでの InTouch アプリケーション グラフィックのホスティング

InTouch WindowMaker でグラフィックを設定し、グラフィックへのダイレクト URL を使用して外部 Web サイトでグラフィックをホストすることができます。

1. 産業用グラフィック ツールボックスを使用してシンボルを作成します。
2. Web Client でシンボルをテストします。
3. 外部 Web サイトで HTML **<iframe>** タグを作成します。
4. URL 属性には http://<hostname>/InTouch/?iframe=true&symbol=<graphicname> または http://<IPAddress>/InTouch/?iframe=true&symbol=<graphicname> へのリンクを含めます。ここで、<hostname> または <IPAddress> は、アプリケーションがパブリッシュまたは配置されているマシンを表し、<graphicname> は表示するグラフィックの名前を表します。
5. 英語以外の言語でシンボルを表示する場合、URL の形式は異なります。たとえば、フランス語の URL は以下のようになります。
   http://<hostname>/InTouch/?localeid=1036&iframe=true&symbol=<graphicname>

   ランタイム時の言語変更が可能な新しいバージョンの Web Client では URL の形式が変更されます。最新バージョンの Web Client にアップグレードすると、以前の URL 形式では英語以外の言語向けに設定されたシンボルは表示されません。
6. アプリケーションをパブリッシュまたは配置します。
7. アプリケーションを InTouch WindowViewer で実行します。InTouch タグからライブ データを受信するには、WindowViewer が実行している必要があります。

構文:

```
<iframe src="http://localhost/InTouch/?iframe=true&symbol=YourGraphicName"></iframe>
または
<iframe src="http://NodeName/InTouch/?iframe=true&symbol=YourGraphicName"></iframe>
```

例: ローカル マシンの Web Client アプリケーションに格納されているシンボル ClockAnalogWall を例に説明します。

```
<iframe src="http://localhost/InTouch/?iframe=true&symbol=ClockAnalogWall"></iframe>
<iframe src="http://localhost/InTouch/?localeid=1036&iframe=true&symbol=ClockAnalogWall"></
iframe>
```

URL を使用してシンボルに直接アクセスする場合、同じ InTouch Web Client のセキュリティとライセンスが適用されます。InTouch Web Client のエラー メッセージと同様に、該当するエラー メッセージが <iframe> セクションに返されます。

注記: Google Chrome 80 以降のバージョンでのポリシーの変更により、http プロトコルを使用する場合、<iframe> タグ内に配置されたグラフィックは表示されません。http プロトコルを使用してグラフィックを表示するには、**chrome://flags** にアクセスして以下の cookie 設定を無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- SameSite by default cookies
- Enable removing SameSite=None cookies
- Cookies without SameSite must be secure
詳細については、https://www.chromestatus.com/feature/5633521622188032 を参照してください。セキュリティの問題を防止するために、Web Client では https プロトコルを使用することが推奨されます。

iFrame コードブロックを生成

シンボルの <iframe>ブロックを Web ブラウザから直接生成できます。その後、このコード ブロックを外部の Web サイトで使用して、ランタイム時に産業用グラフィックを表示できます。

1. グラフィックに移動します。
2. [**共有**] アイコン 選択します。
3. ［**所有オブジェクト**］フィールドで、このグラフィックの所有オブジェクトを指定します。
4. サポートされるカスタム プロパティのリストが表示されます。カスタム プロパティを選択して値を指定します。

   指定したすべての値は、データ型に対して検証されます。たとえば、カスタム プロパティで整数値が要求される場合、文字値を入力することはできません。
5. Web Client（ローカルまたはリモート マシン）上のスクリプトを使用する場合、［**クロス オリジンをサポート**］を選択します。

   埋め込まれた HTML コードは、選択したオプションに基づいて変更されます。
6. [**コピー] を選択します**。
7. 生成されたコードを Web サイトの .html ファイルに貼り付けることができます。

```
<iFrame src="http://<hostname>/InTouch/?iframe=true&symbol=Sym1?crossOrigin=1"
id="symbol" width="640px" height="480px" customProperty='[{"n":"CP1","v":"True","t":1},
{"n":"CP2","v":"24.5","t":6},{"n":"CP3","v":"Orange","t":7}]' onload="postMsg()"></
iFrame>
<script type="text/javascript">
関数 postMsg(){
    var iframe=document.getElementById('symbol');
    var ajax={};
    for (var i=0;i<iframe.attributes.length;i++) {
        ajax[iframe.attributes[i].nodeName]=iframe.attributes[i].nodeValue;
    }
    var win=iframe.contentWindow;
    win.postMessage(dev,dev['src']);
};
</script>
```

言語がフランス語（France）の場合

```
<iFrame src="http://<hostname>/InTouch/?localeid=1036&iframe=true&symbol=Sym1"
id="symbol" width="640px" height="480px" onload="postMsg()"></iFrame>
<script type="text/javascript">
関数 postMsg(){
    var iframe=document.getElementById('symbol');
    var ajax={};
    for (var i=0;i<iframe.attributes.length;i++) {
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
        ajax[iframe.attributes[i].nodeName]=iframe.attributes[i].nodeValue;
    }
    var win=iframe.contentWindow;
    win.postMessage(dev,'*');
};
</script>
```

サポートされているグラフィック要素と既知の制限事項

このセクションでは、サポートされているグラフィック要素と既知の問題について説明します。

既知の制限事項

InTouch Web Client は、製品リリースごとに強化される予定です。以下のリストに現在のリリースの主な制限事項を示します。

- Web Server とのデータ通信では、RDS セッションで実行している InTouch WindowViewer はサポートされません。
- System Platform 2017 以前のバージョンからパブリッシュ済みアプリケーションは Web Client でサポートされません。Web Client で表示するには、元のアプリケーションを現在のバージョンに移行して、アプリケーションを再度パブリッシュする必要があります。
- ウィンドウ名に特殊文字（サポートされていない文字）を含む枠ウィンドウはサポートされません。
- 間接型タグはサポートされていません。
- PlaySound() スクリプト関数はサポートされません。
- InTouch HMI マネージド アプリケーションでは、Galaxy ベースのセキュリティはサポートされません。
- システム タグは、ユーザー名および各ユーザーの関連詳細を返しません。
- スクリプトを使用したグラフィック・プロパティの設定：スクリプトを使用したグラフィック・プロパティの設定：スクリプトを使用してグラフィック要素のプロパティを読み書きできるのは、以下のプロパティのみです：
  - 位置 (X, Y)
  - サイズ(高さ、幅)
  - 角度
  - 開始角度
  - ｽｲｰﾌﾟ 角度
- シンボル ライブラリのいくつかのシンボルには、現在のリリースの Web Client でサポートされていないグラフィック要素または色設定が組み込まれています。
- Windows 共通コントロール（WCC）とは、ラジオボタン グループ、チェック ボックス、編集ボックス、コンボ ボックス、カレンダー、日付時刻ピッカー、およびリスト ボックスを指します。

  注意：コンボボックス コントロールは、文字列を入力している間、参照オブジェクトに値を渡すことができません。選択したタイプに関係なく、コンボボックスの事前定義された値からのみ選択できます。

- マルチ ペン トレンド グラフィック要素はサポートされません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ビット状態アニメーションはサポートされません。
- [無効]、[表示]、[点滅]、[値の表示]の各真理値テーブルのアニメーションはサポートされていません。
- 履歴サマリ データ型を含むカスタム プロパティはサポートされません。

以下のセクションでは、これらの制限事項について詳細に説明します。

重要：Web Client にグラフィックをロードすると、サポートされていない機能は、ロガーに警告として記録されます。生成されるレポートには警告パスが含まれます。

すべてのサポート対象グラフィック要素でサポートされるプロパティ

ブラウザで表示する場合、以下のプロパティは、シンボル内のサポートされるすべてのグラフィック要素でサポートされます。

- 角度
- XX
- Y
- 幅
- 高さ
- 有効化
- 表示
- 絶対アンカー
- 相対アンチョル
- 透明度
- 要素ｽﾀｲﾙ
- 所有オブジェクト
- 起動
- 終了
- 半径
- 張力
- カスタム プロパティの上書き

注意：ElementStyle プロパティでは、定義済みの要素スタイルを選択して、グラフィック要素に適用できます。要素のスタイルには、色プロパティ（FillColor、LineColor、TextColor、OutlineColor など）を定義するプロパティが含まれています。

すべてのグラフィック要素でサポートされるプロパティ（いくつかの制限あり）

ブラウザで表示する際、以下のグラフィック要素にはレンダリング制限があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グラフィック要素プロパティ	プロパティの制限
塗りつぶしの色	テクスチャはサポートされていません。FillColor 上の一部の制限が個々の要素に適用されることがあります。
UnFilledColorUnFilledColor	FillColor プロパティと同じ制限が UnFilledColor プロパティに適用されます。
線の色	テクスチャはサポートされていません。
フォント	Font プロパティのフォント サイズ、フォント タイプ、およびフォント スタイル（太字または標準）オプションのみがサポートされます。
TextColor(テキスト色)	TextColor プロパティの純色オプションのみがサポートされます。
塗りつぶしの向き	RelativeToScreen は、色が 'Solid'、'Fill'、または 'UnFill' に設定されている場合にのみサポートされます。
塗りつぶしの動作	常に Both に設定されます。
ランタイム動作	TabOrder、TabStop、ZoomPercent は、いずれのグラフィック要素でもサポートされていません。
相対参照	MyPlatform、MyEngine、MyHost、MyArea、MyContainer、および Me がサポートされています。MyViewApp はサポートされていません。
[自動スケール]	サポートされません。
グラフィックの表示	サポートされます。
線	シンボルの縁が 1px よりも大きい LineWeight プロパティ値で設定される線グラフィックは切り詰められます。
画像のグループ化/埋め込み	表示オン/オフ、点滅、無効化、押しボタン、ユーザー入力/値の表示、ShowSymbol、およびアクション スクリプト アニメーションがサポートされます。 スクリプトによる 'Treat as Icon' および 'Enabled' プロパティの変更はサポートされません。 %FillHorizontal および %FillVertical アニメーション: - グループにのみ適用されます。サブ要素ではサポートされません。 - ボタンと画像はサポートされません。 - グループとサブ要素の［画面を基準］はサポートされません。 - サブ要素の角度が変更された場合はサポートされません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

サポートされているグラフィック要素と追加のプロパティ

ブラウザベースのクライアントで表示されるグラフィック要素を以下の一覧に示します。**この表に含まれないグラフィック要素はブラウザベースのクライアントでサポートされません。**

表には、各グラフィック要素でサポートされるプロパティ、およびそれらのプロパティの制限が示されています。色プロパティを組み込むグラフィックを使用する場合、色には、FillColor、LineColor およびその他の関連するグラフィック プロパティと同じ制限が適用されます。

注意：ここに示すプロパティに加えて、「すべてのグラフィック要素でサポートされるプロパティ」および「ブラウザ:制限のあるグラフィックプロパティ」に一覧表示されているすべてのプロパティは、サポートされているグラフィック要素で使用できます。

グラフィック要素	サポートされるプロパティ	制限事項
線 折れ線 曲線 2 ポイントの円弧型 3 ポイントの円弧型 コネクタ	線スタイル 線の太さ 線パターン StartCap： 終了ｷｬｯﾌﾟ	サポートされますが、レンダリングが若干異なることがあります。
ボタン	ボタンのスタイル テキスト WordWrap 塗りつぶしの向き 塗りつぶしの色 整列	標準のボタンスタイルのみがサポートされます。 ボタンの幅を超えるキャプションは切り詰められます。 FillOrientation: 'RelativeToScreen' は、色が設定されている場合にのみサポートされます。 純色のみがサポートされます。複数の色のグラデーションを選択した場合、最初の色を使用して単一の色に変換されます。 中央揃えのみがサポートされます。
テキスト	整列	左上のみがサポートされます。
画像	HasTransparentColor TransparentColor 画像ｽﾀｲﾙ	FillColor、FillPercent、および UnFilledColor はサポートされません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グラフィック要素	サポートされるプロパティ	制限事項
テキスト ボックス	テキスト ﾃｷｽﾄ形式 折り返す 線の太さ 線パターン 整列 フォント	Flip はサポートされません。
ステータス	ｸﾞﾗﾌｨｯｸ 式	デフォルト設定のみがサポートされます。
Web アラーム クライアント（EAC） （この要素は一部の HMI/SCADA ソフトウェアでのみサポートされます。）	アラーム モード 色 列の詳細 クエリーとフィルタ TimeSettings： 実行時の動作	**アラーム モード** • ［現在のアラーム］、および［最近のアラームとイベント］のみがサポートされます。 • AlarmQuery: InTouch および ApplicationServer アラームがサポートされます。 • アラームとイベントを示すコメントのみがサポートされます。 **アラームとイベント:** ［シェルブ］および［点滅未確認アラーム］はサポートされません。 ソートはサポートされません。 クエリーとフィルタ : “デフォルト” クエリーのみをサポートします。カスタム クエリーおよびフィルタはサポートされません。 データバインド:サポートされていません。 イベント:サポートされていません。 時間設定:サポートされていません。 ［見出しの表示］および［グリッドの表示］のみがサポートされ、カラムのサイズが変更できます。選択したレコードのユーザー ACK またはすべてのアラーム レコードの ACK のみがサポートされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グラフィック要素	サポートされるプロパティ	制限事項
Web トレンド クライアント （この要素は一部の HMI/SCADA ソフトウェアでのみサポートされます。）	ペン 外観 オプション 履歴ソース	**ペン** 表示、ペン名（説明）、および式または参照のみがサポートされます。 ペンの詳細:最小、最大 カラープロットタイプ（すべて線として処理されます）。 **外観** 単一タグモード、グリッド（垂直グリッドを表示、水平グリッドを表示、色）。 X time axis: カーソルを表示（Cursor1: 色）、値の数。 Y value axis: 値の数、値軸ラベル（複数のスケール、単一のスケール）。 **Options** ［取得: トレンド期間］のみがサポートされます。 データバインド:設定なし イベント:設定なし 接続タイムアウト（秒）、クエリータイムアウト（秒）、および **HTTP** の使用はサポートされません。 トレンドクライアントのメソッドおよびプロパティはサポートされません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グラフィック要素	サポートされるプロパティ	制限事項
Web SQL データ グリッド （この要素は一部の HMI/SCADA ソフトウェアでのみサポートされます。）		データ バインディング アニメーション イベント アニメーション カスタム プロパティ • CmdCopy_dg • CmdPaste_dg • CmdWrite_dg • RowsChanged_dg • WriteButtonHide_dg] • ColumnAggregateEnable_dg • SupportThemes_dg - SQLDataGrid コントロールはサポートされません。 - selectedRowNumber プロパティ値はカスタム プロパティを使用して取得できません。 - データ バインディングはサポートされないので、プロパティがカスタム プロパティで参照されていて、デフォルト値が空の場合、プロパティのデフォルト値は取得できません。 - kendo グリッドの制限により RowCount プロパティではグループ化のカウントを表示できません。レコードの総数が常に表示されます。 - ローカル データベースで Windows 認証を使用する場合、仮想ユーザー NT Service\AIGWebServer を追加してデータベースへのアクセスを許可します。'Windows' 認証でのリモート データベースへのアクセス、および 'SQLServer' モードはサポートされません。 - カルーセル ウィジェットで使用する場合、および WindowViewer で表示する場合、SQLGrid はサポートされません。 - 2000 以上のレコードと 5 つ以上の列を含む SQLGrid にグループを適用する場合、行のスクロールと更新でパフォーマンスの問題が発生することがあります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

グラフィック要素	サポートされるプロパティ	制限事項
トレンド ペン (この要素は一部の HMI/SCADA ソフトウェアでのみサポートされます。)	データ ソース	トレンド期間:固定タイプはサポートしていません。
チェック ボックス		ユーザー設定に関係なく、直ちに送信された値のみがサポートされます。 Checked プロパティはサポートされません。
ラジオ ボタン グループ	静的値およびｷｬﾌﾟｼｮﾝ 状態:	ユーザー設定に関係なく、直ちに送信された値のみがサポートされます。
編集ボックス		ユーザー設定に関係なく、直ちに送信された値のみがサポートされます。
コンボ ボックス		ユーザー設定に関係なく、直ちに送信された値のみがサポートされます。 MaximumLength はサポートされません。
カレンダー		ユーザー設定に関係なく、直ちに送信された値のみがサポートされます。 太字の日付:サポートされていません チェックアウトされている場合でも常に表示されます。 カレンダーの色:サポートされていません
日付時刻ピッカー		ユーザー設定に関係なく、直ちに送信された値のみがサポートされます。 Configuration: サポートされません。 ｶﾚﾝﾀﾞｰﾄﾞﾛｯﾌﾟﾀﾞｳﾝの色
リスト ボックス		ユーザー設定に関係なく、直ちに送信された値のみがサポートされます。
コネクタポイント		ConnectorPoint の移動はサポートされません。

サポートされるアニメーション

グラフィックを Web ブラウザで表示する際にランタイム時にサポートされるアニメーションを以下の表に示します。色プロパティを組み込むアニメーションを使用する場合、選択したアニメーションの色

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

には、FillColor、LineColor、およびその他の関連するグラフィック要素プロパティと同じ制限が適用されます。また、ブラウザの表示では値の書き込みがサポートされません。属性値を更新するアニメーションのタイプはサポートされません。

注意 : サポートされるすべてのアニメーションを以下の表に示します。**一覧に含まれないアニメーションは、InTouchWebClient でサポートされません。**

アニメーション	制限事項
アラーム輪郭	アラーム輪郭アニメーションは、表示される外観がグラフィック要素のアラーム タイプと確認ステータスを表すグラフィック要素の周囲にアラーム輪郭を表示します。各アラーム輪郭の外観は、関連付けられた要素スタイルによって定義されます。一部の要素スタイル プロパティはサポートされません。たとえば、アラーム輪郭要素スタイルの［線の色の上書き］オプションの値として［グラデーション］が指定されている場合、このオプションはサポートされません。
要素スタイル	色、線、およびフォントに関するグラフィック要素の制限を参照してください。グラフィック要素に適用されるものと同じ制限が要素スタイル アニメーションにも適用されます。 ［時間］および［経過時間］はサポートされません。 要素スタイルの定義の変更は、アプリケーションを再配置またはパブリッシュした後にのみ有効になります。 Windows 共通コントロールはサポートされません。
塗りつぶしスタイル 線スタイル テキスト スタイル	色、線、およびフォントに関するグラフィック要素の制限を参照してください。グラフィック要素に適用されるものと同じ制限が要素スタイル アニメーションにも適用されます。 ［時間］および［経過時間］状態はサポートされません。 Windows 共通コントロールはサポートされません。 純色のみがサポートされます。
水平塗りつぶしパーセント 垂直塗りつぶしパーセント	RelativeToGraphic のみがサポートされます。
水平スライダー 垂直スライダー	スライダーはタグにデータを書き込むことはできませんが、カスタム プロパティとのデータ バインディングをサポートできます。 カスタム プロパティへの書き込み時、スライダーは［マウス ボタンを離したとき］のみをサポートします。 スライダーは［カーソル アンカー］をサポートしません。。 スライダーは［ツールヒントを表示する］をサポートしません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アニメーション	制限事項
幅 高さ	Windows 共通コントロールはサポートされません。
向き	向きアニメーションを設定する際、RelativeAnchor および RelativeOrigin はサポートされません。（中央にアンカーのみがサポートされます）。
無効化	ラジオ ボタン グラフィック要素は無効化アニメーションをサポートしません。
ポイント	曲線および閉曲線のポイント アニメーションはサポートされません。
ツールヒント	ツールヒント アニメーションは、画像、ラジオ ボタン、ボタン、およびテキスト グラフィック要素以外でサポートされます。 ツールヒント アニメーションはグループおよび埋め込まれたシンボルではサポートされません。 Windows 共通コントロールはサポートされません。
アクション スクリプト	アクションスクリプトは、ダイアログを呼び出す関数以外でサポートされます。ShowGraphic アニメーションは枠ウィンドウでのみサポートされます。
ShowSymbol	デフォルトオプションのみがサポートされます。 ［閉じるボタン］オプションが選択されている場合、［モーダル］と［モードレス］がサポートされます。 クライアント領域 x,y の値が 0,0 の中心のみがサポートされます。 ［シンボルを基準］オプションのみがサポートされます。 ショートカット:サポートされていません。
グラフィックの表示	デフォルトオプションのみがサポートされます。 ［閉じるボタン］オプションが選択されている場合、［モーダル］および［モードレス］がサポートされます。 クライアント領域 x,y の値が 0,0 の中心のみがサポートされます。 ［シンボルを基準］オプションのみがサポートされます。 ショートカット:サポートされていません。
ハイパーリンク	Windows 共通コントロール、ボタン、テキスト、およびテキストボックス上でサポートされません。

Web Client モバイルアプリ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オペレーティングシステムの要件

Apple App Store または Google Play Store からモバイル アプリケーションをダウンロードします。AVEVA Mobile Operations を検索して、［**インストール**］を選択します。

Minimum Android version: Android 6.0 (API Level 23 – Marshmallow)

Target Android version: Android 9.0 (API Level 28 – Pie)

iOS の最低バージョン: iOS 9.0 以降

デバイスのサポートについては、「ブラウザとモバイルのサポート」を参照してください。

モバイルアプリケーションにログインする

アプリのインストール時にユーザー資格情報を入力してアプリにログインし、グラフィックを表示します。

1. モバイル アプリを起動します。
2. アプリケーションが実行するノードの**サーバー名**または **IP アドレス**を入力します。
3. **Web サイト**名を選択または入力ます。提供される Web サイト名は、InTouch HMI アプリケーション マネージャで指定した［*カスタム URL*］に一致する必要があります。
4. **ユーザー名**と**パスワード**を入力します。必要に応じて、ユーザー名のドメイン名を入力します。
5. アプリを全画面モードで表示するには、［**全画面モード**］チェック ボックスをオンにします。

   この設定は、ナビゲーション バーとヘッダー バーがデフォルトで非表示になっているときに TV などの大画面ディスプレイでフル サイズのグラフィックを表示する場合に便利です。
6. **[ログイン] を選択します。**

   正常にログインすると、Web Client のホーム ページが表示されます。

初回起動時にユーザー資格情報を入力する必要があります。その後の使用では、ユーザー資格情報がアプリケーションに記憶されます。ログイン画面から情報をクリアするには［**クリア**］ボタンを使用します。モバイル アプリケーションにアクセスできるのは、Web Client ユーザー グループに含まれるユーザーだけです。ログオフ ボタンを使用してアプリを終了します。

注意：Web Client モバイル アプリは、匿名ログインに加えて、AVEVA Identity Manager（AIM）経由のログインもサポートします。

InTouch Web Client モバイルアプリケーションを使用する

資格情報が認証され、有効な Web Server ライセンスが取得されるとアプリケーションが表示されます。デフォルトでホーム シンボルが表示されます（設定されている場合）。

1. アイコンを選択します。

   左側のオーバーレイに実行中のアプリケーション内のグラフィックの階層構造が表示されます。
2. フォルダ名を選択します

   フォルダ内のグラフィックが表示されます。
3. 画像を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

選択したグラフィックが表示されます。

［パン］および［ズーム］を使用して、グラフィックをさまざまなスケールで表示することもできます。

InTouch Web Client モバイル アプリはランタイムの言語切り替えをサポートします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 22 ランタイム時の言語の切り替え

ランタイム時に別の言語に切り替えることができるアプリケーションを開発できます。

ランタイム時の言語の切り替えを有効にするには、次のタスクを完了する必要があります。

- アプリケーションに対して複数の言語を設定します。
- オフライン翻訳用にアプリケーション テキストをエクスポートします。
- エクスポートされた 1 つまたは複数のディクショナリ ファイルを翻訳します。
- 翻訳された 1 つまたは複数のディクショナリ ファイルをインポートします。

ランタイム時の言語切り替えについて

ランタイム時に別の言語に切り替えることができるアプリケーションを開発できます。

ランタイム時の言語の切り替えを有効にするには、次のタスクを完了する必要があります。

- アプリケーションに対して複数の言語を設定します。
- オフライン翻訳用にアプリケーション テキストをエクスポートします。
- エクスポートされた 1 つまたは複数のディクショナリ ファイルを翻訳します。
- 翻訳された 1 つまたは複数のディクショナリ ファイルをインポートします。

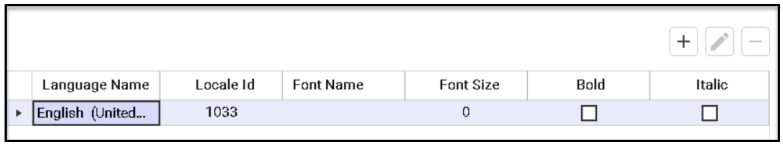
ランタイム時の言語の切り替えのための言語の設定

すべての InTouch アプリケーションは、アプリケーションを開発するために使用されるベース言語に関連しています。サポートする追加の言語を設定する必要があります。

注意：言語の切り替えとネットワークアプリケーション開発（NAD）を組み合わせて使用している場合、NAD クライアント ノードに対しては、変更モードは［WindowViewer に変更をロード］または［WindowViewer へ変更ロード時にメッセージ表示］ではなく、［WindowViewer の再起動］または［WindowViewer の再起動時にメッセージ表示］に設定することをお勧めします。

ランタイム時の言語の切り替え用に言語を設定するには

1. WindowMaker で、言語を設定するアプリケーションを開きます。
2. ［**ファイル**］メニューの［**設定**］グループで［**言語**］を選択します。［**言語**］ウィンドウが表示されます。



［**言語の設定**］ダイアログ ボックスに、アプリケーションのベース言語が表示されます。

3. **追加**を選択します。［**言語の追加**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. 言語およびフォント設定を指定します。フォントを設定すると、翻訳されたテキストのデフォルトのフォント プロパティが定義されます。
   - [名前] または [**ロケール ID**] リストで、追加する言語を選択します。名前で言語を選択した場合、対応するロケール ID が［**ロケール ID**］リストに表示され、ロケール ID で言語を選択した場合、対応する名前が表示されます。
   - フォントの選択［**フォント**］ダイアログ ボックスが表示されます。フォントを設定して、［**OK**］を選択します。
5. ［**OK**］をクリックして、**追加言語**ダイアログ ボックスを閉じます。設定した言語が［**言語の設定**］ダイアログ ボックスに一覧表示されます。
6. 他にも言語を追加する場合は、手順 3 ～ 5 を繰り返します。
7. 完了したら、[**閉じる**] を選択します。

言語を削除するには

a. 削除する言語を［**言語の設定**］ダイアログ ボックスで択します。

b. **削除**を選択

［**削除の確認**］ダイアログ ボックスが表示され、アプリケーションからの言語の削除の確認を求めるメッセージが表示されます。

c. ［**はい**］を選択します。

［**言語の設定**］ダイアログ ボックスが更新され、言語が削除されたことが示されます。

デフォルト言語を設定するには

a. アプリケーションのデフォルトとして設定する言語を［**言語の設定**］ダイアログ ボックスで選択します。

b. **デフォルトセット**を選択します。

［**言語の設定**］ダイアログ ボックスが更新され、左下隅にアプリケーションのデフォルト言語が表示されます。

設定言語のフォント設定を変更

すべての言語のデフォルトのフォントは Tahoma です。フォント スタイルおよびサイズは、ベース言語内の個々の語句の対応する設定によって異なります。既に設定されている言語のフォント設定は変更できます。異なる言語間ではテキスト表示が異なるため、翻訳されたテキストがボタンやその他のオブジェクトに正しく表示されるように適切なフォントを指定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

設定言語のフォント設定を変更

1. WindowMaker で、設定言語のフォント設定を変更するアプリケーションを開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［言語］を選択します。

   ［言語］画面が表示されます。

3. ［全般］タブの言語のリストから目的の言語を選択します。
4. ［フォント名］と［フォント サイズ］エントリをダブルクリックして値を編集します。
5. 必要に応じて、［太字］と［斜体］チェック ボックスをオンまたはオフにします。
6. 変更を加え、[OK]を選択 します。

ランタイム時の言語の切り替え機能の追加

ランタイム ユーザーは、WindowViewer の［システム］メニューの［言語］コマンドを使用してアプリケーション インターフェイスの言語を切り替えることができます。

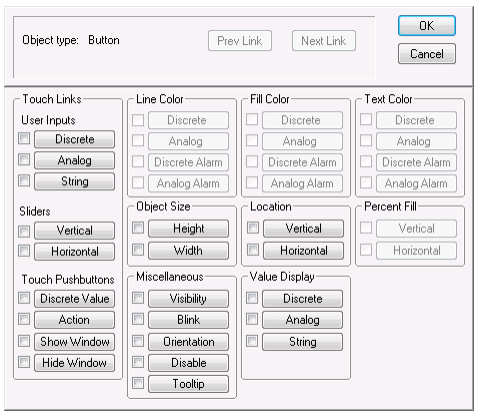
ランタイム ユーザーが言語を切り替えることができるように、アプリケーションにボタンを追加することもできます。開始する前に、アプリケーションで追加の言語を設定し、言語のロケール ID を知っておくようにしてください。アプリケーションの言語設定の詳細については、「ランタイム時の言語の切り替えのための言語の設定」を参照してください。

ランタイム時に言語を切り替えるためのボタンを追加

1. WindowMaker で、言語の切り替えボタンを追加するアプリケーション ウィンドウを開きます。
2. ボタンを描画し、選択したときに切り替えられる言語を示すテキスト ラベルをボタンに割り当てます。
3. ボタンをダブルクリックします。

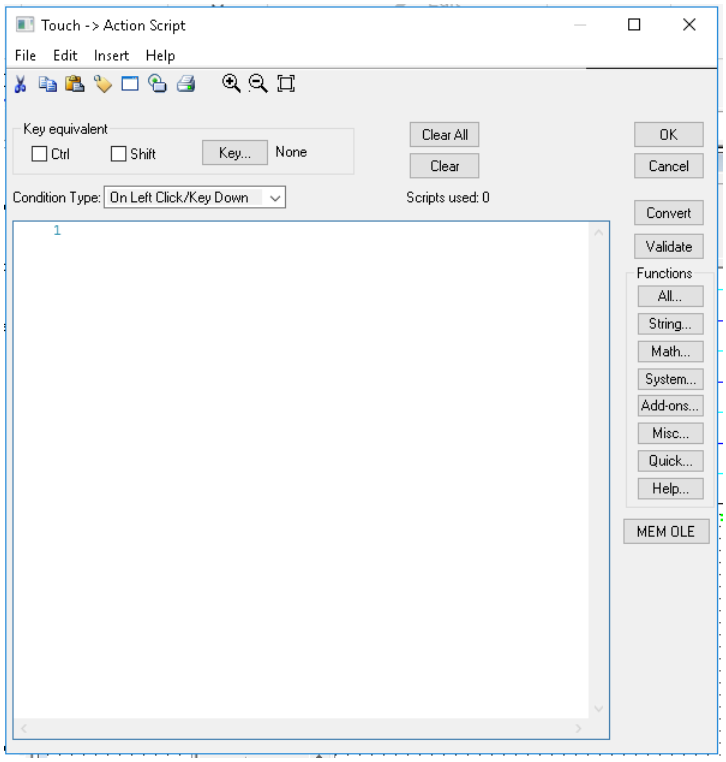
   アニメーションを選択するダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［押しボタン］領域で、［スクリプト ボタン］を選択します。

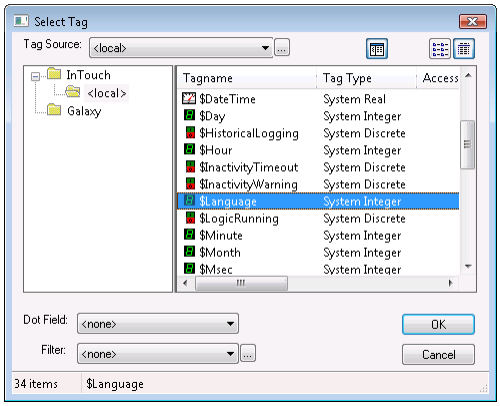
   ［タッチ -> アクション スクリプト］ダイアログ ボックスが表示されます。



5. ［タッチ -> アクション スクリプト］ダイアログ ボックスのスクリプト領域内をダブルクリックします。

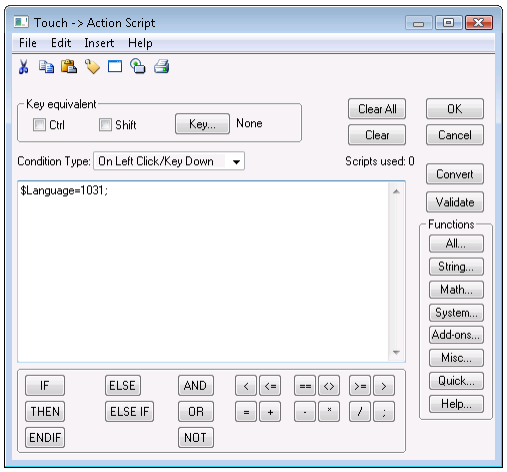
   ［タグの選択］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



1. **$Language** システムタグを選択し、 **[OK] を選択します。**

   $Language システム タグ変数を、ボタンに割り当てる言語のロケール ID に等しく設定し、［**OK**］をクリックします。



注意：$Language タグの代わりに、スクリプト関数 SwitchDisplayLanguage(LocaleID) を使用することもできます。

1. ［**OK**］をクリックして、ダイアログ ボックスを閉じます。

SwitchDisplayLanguage() 関数

翻訳された文字列とアラーム フィールドが表示される目的の言語で表示される、静的テキストの表示を切り替えます。

カテゴリ

その他

構文

```
SwitchDisplayLanguage(LocaleID);
```

パラメータ

ロケール ID
ランタイムに静的テキストの文字列とアラーム フィールドが表示される言語

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

例

この例では、ランタイムに表示される言語はドイツ語です。

```
SwitchDisplayLanguage(1031);
```

次もご確認ください。

$Language システム タグ変数

$Language システム タグ変数

InTouch アプリケーションで複数の言語が定義されている場合、$Language システム タグには、現在表示されている言語の言語 ID の値が反映されます。デフォルトでは、これは、ベースの InTouch システム（E/F/G/J/SC）の言語 ID（ロケール ID）です。これを他の ID に設定すると、新しい言語で定義された値を使用して文字列とアラーム フィールドが切り替わります。

注意 : $Language タグをローカルタグとして設定すると、Web Client で独立した言語変更が可能になります。複数のユーザー セッションで Web Client を異なる言語で表示できます。Web Client で言語を変更しても、WindowViewer で選択されている言語には影響しません（その逆も同様です）。

カテゴリ

システム

データ型

整数型（読み取り/書き込み）

オフライン翻訳用にアプリケーションテキストをエクスポート

InTouch アプリケーションに多くの文字列が含まれている場合、通常バルク翻訳用にテキスト文字列を送信します。アプリケーションの文字列を翻訳用にエクスポートし、テキスト エディタ、XML エディタ、または Microsoft Excel などのスプレッドシート プログラムを使用して整理できます。

静的テキストは、次のアイテムからエクスポートできます。

- テキスト オブジェクト
- ボタン テキスト
- SmartSymbol 内テキスト
- ツールヒント静的テキスト
- ユーザー メッセージ
- 入力リンク内のメッセージのオン/オフ
- 出力リンク内のメッセージのオン/オフ
- ウィザード上のテキスト

WindowMaker 内のすべてのウィンドウを閉じるまでは、ディクショナリをエクスポートできません。ディクショナリ ファイルをエクスポートした後にアプリケーションを変更する場合、ディクショナリ ファイルを再度エクスポートする必要があります。詳細については、「既存のディクショナリ ファイルへのテキストのエクスポート」を参照してください。

テキスト文字列は、一度に 1 つの言語に対してのみエクスポートできます。デフォルトでは、［マイ InTouch アプリケーション］フォルダが InTouch HMI で開かれます。他のフォルダを選択した場合、

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch HMI ではそのパスがデフォルトになります。各言語の語句をエクスポートするために新しいフォルダを作成すると、ディクショナリ ファイルを管理しやすくなります（たとえば、...\マイ InTouch アプリケーション\My German Files\ など）。

InTouch HMI によって、アプリケーションのディクショナリ ファイルが作成され、アプリケーション内の各 SmartSymbol 用に別のディクショナリ ファイルが作成されます。アプリケーション ディクショナリ名は、アプリケーション名_localeID の形式を持ち、SmartSymbol ディクショナリ ファイルは Symbol_localeID_GUID の SSD_Name の形式を持ちます。

アプリケーションのディクショナリをエクスポートすると、ファイルは Microsoft Excel 2003 以降を使用して編集できる .xml ファイルになります。

オフライン翻訳用にアプリケーションテキストをエクスポート

1. WindowMaker を起動し、オフライン翻訳のためにテキスト文字列をエクスポートするアプリケーションを開きます。
2. ［**ファイル**］メニューで、［**設定**］をポイントし、［**言語**］を選択します。

   ［**言語**］画面が表示されます。
3. **[エクスポート]**タブを選択します。［**ディクショナリのエクスポート**］ダイアログ ボックスが表示されます。

4. エクスポートの設定を行います。
   - ［**定義された言語**］リストで、エクスポートする言語ディクショナリを選択します。
   - ［**パス**］ボックスで、ディクショナリをエクスポートするフォルダを入力します。［**参照**］を選択して、既存のフォルダを選択するか、新しいフォルダを作成します。
5. **[Export]** を選択します。エクスポートの進行が表示されます。エクスポートが正常に終了すると、［**エクスポートに成功しました**］ダイアログ ボックスが表示されます。
6. ［**閉じる**］を選択して WindowMaker ウィンドウに戻るか、［**閉じてからエクスプローラを起動する**］を選択してディクショナリ ファイルを含むフォルダを開きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

既存のディクショナリ ファイルへのテキストのエクスポート

オフライン翻訳用にアプリケーション テキストをエクスポートした後、アプリケーションを変更する場合があります。アプリケーションを変更した場合は、テキストを再度エクスポートする必要があります。詳細については、「オフライン翻訳用にアプリケーションテキストをエクスポート」を参照してください。.

同じディレクトリに 2 回以上エクスポートすると、[**ファイル置換の確認**] メッセージが表示されます。

[**はい**] を選択すると、新しい文字列や最後にエクスポートしてから追加した言語情報で、既存の .xml ファイルが更新されます。既存のディクショナリ ファイルに語句の翻訳が含まれ、それを InTouch HMI に以前インポートしていた場合、これらの翻訳は維持されます。最後のエクスポート以来、アプリケーションから任意の語句を削除した場合、その語句はディクショナリ ファイルから削除されます。

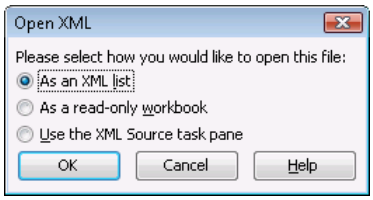
エクスポートされたディクショナリファイルを翻訳

アプリケーション テキストが含まれているディクショナリ ファイルをエクスポートした後に、Microsoft Excel 2003 以降を使用してテキストを編集します。

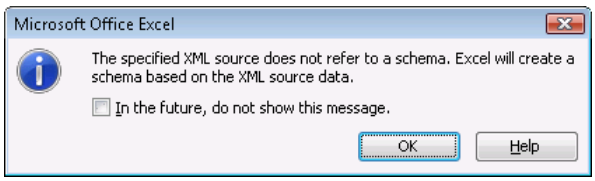
エクスポートした各言語に対する個別のディクショナリ ファイルが InTouch HMI によって作成されます。また、アプリケーションの各 SmartSymbol に対しても、個別のディクショナリ ファイルが InTouch HMI によって作成されます。すべての言語および SmartSymbol に対して、すべてのディクショナリ ファイルを翻訳してください。

エクスポートされたディクショナリ ファイルを翻訳するには

1. Excel で XML ファイルを開きます。[**XML を開く**] ダイアログ ボックスが表示されます。



2. [**XML リスト**] を 選択し、 [**OK**] を選択します。メッセージが表示されます。



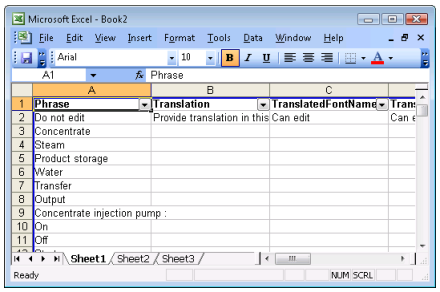
3. [**OK**] を選択します。

   Excel で XML ファイルが開き、次のカラムが表示されます。

   - アプリケーション内のフレーズ
   - 翻訳されたフレーズ
   - 翻訳されたフォント名
   - 翻訳されたフォント プロパティ
   - 翻訳されたフォント サイズ
   - ベース フォント プロパティ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ベース フォント サイズ
- コンテキスト、フレーズ ID、言語 ID および 外国語 ID



重要：［**Translation**］カラム、［**TranslatedFontSize**］カラム、［**TranslatedFontName**］カラムおよび「**TranslatedFontProperty**」カラムのデータのみを修正してください。カラム ヘッダーは変更しないでください。行を挿入または削除しないでください。

1. ［**語句**］カラムのベース言語の文字列に対応する行の［**翻訳**］カラムに、言語固有のテキストを入力します。
2. 必要に応じて、WindowViewer でのスペースにテキストを表示できるように、翻訳された文字列のフォント パラメータを変更します。
   - ［**TranslatedFontName**］カラムで、フォント名を入力します。
   - ［**TranslatedFontProperty**］カラムで、フォント プロパティの表記法を入力します。

     B = 太字

     I = *斜体*

     U = 下線

     たとえば、テキストを太字にする場合、［**TranslatedFontProperty**］カラムに **B** と入力します。テキストを太字および下線付きにするには、［**TranslatedFontProperty**］カラムに **BU** と入力します。
3. XML データのファイルのタイプを使用して、ファイルを保存します。

重要 : XML スプレッドシートなどの他のファイルのタイプで保存すると、Excel によってスキーマが変更され、InTouch HMI でファイルをロードできません。XML ファイルの名前を変更すると、ランタイム時の言語の切り替えが機能しません。

翻訳されたディクショナリファイルをインポート

エクスポートした各言語に対するディクショナリ ファイルが InTouch HMI によって作成されます。また、アプリケーションの各 SmartSymbol に対しても、個別のディクショナリ ファイルが InTouch HMI によって作成されます。翻訳後、各言語に対してディクショナリ ファイルをインポートし、これらの言語に対してランタイム時の言語の切り替えを有効にする必要があります。1 つの言語のすべてのディクショナリ ファイルは同じフォルダに配置する必要があります。

翻訳されたディクショナリファイルをインポート

1. WindowMaker を起動し、翻訳されたディクショナリ ファイルをインポートするアプリケーションを開きます。
2. ［**ファイル**］メニューで、［**設定**］をポイントし、［**言語**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

1. ［言語］画面の［インポート］タブを選択します。

1. インポート設定を構成します。
   - ［定義された言語］リストで、インポートする言語ディクショナリを選択します。
   - ［パス］ボックスで、インポートするディクショナリ ファイルへのパスを入力します。［参照］を選択し、ファイルを参照して選択します。
2. **Import** を選択します。
3. SmartSymbol ディクショナリ ファイルを再度インポートしている場合、既存のファイルを置換するよう求めるメッセージが表示されます。

   インポートが正常に終了すると、［インポートに成功しました］ダイアログ ボックスが表示されます。

アラームのためのランタイム時の言語切り替え

ランタイム時の言語の切り替えのセットアップの一部として、アラーム グループ名、アラーム コメント、アラーム フィールド、内部ステータス メッセージもローカライズできます。テキスト文字列のランタイム時の言語の切り替えのほかに、Alarm Viewer および Alarm DB View コントロールのアラーム コメント、アラーム状態、アラーム タイプ、およびアラーム クラスのランタイム時の言語の切り替えを設定することもできます。

翻訳されたアラーム グループ名を使用してスクリプト記述、クエリー、またはフィルタも行うことができます。

翻訳のためのアラームコメントのエクスポート

翻訳するために、アラーム コメントをエクスポートできます。

以下の項目のアラーム状態、アラーム タイプ、およびアラーム クラス フィールドをエクスポートできます。

- アラーム コメントを持つすべてのタグ
- タグ コメントを持つすべてのタグ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- システム タグによりイベントが発生したとき、クライアントに示されるコメントをローカライズできるようにするシステム タグ

2 文字のアプリケーション ＩＤ の理解

ローカライズ用にアラームおよびタグ変数コメントをエクスポートするとき、2 文字のアプリケーション ID を指定する必要があります。ID は、同じ名前のアプリケーションによって複数のアラームが生成された場合、個々のアラームを識別するためにシステムによって内部で使用されます。

タグ変数にはタグ変数コメントとアラーム コメントの両方を含めることができるため、2 文字のアプリケーション ID の最後に 1 および 2 が追加され、これらの 2 つのフィールド間が区別されます。タグ変数コメントでは、ID とタグ変数名の間に 1 が挿入されます。アラーム コメントでは、ID とタグ変数名の間に 2 が挿入されます。たとえば、AA1TankLevel はタグ変数コメントで、AA2TankLevel はアラーム コメントです。

アプリケーションをエクスポートすると、アプリケーション ID 情報は削除されます。

アラームデータベースに 2 文字のアプリケーション ID を持たない古いデータが含まれていて、新しいレコードの先頭には ID が付いている場合、Alarm DB View コントロール内のアラーム コメント クエリーでは、

アラーム コメントのエクスポート

翻訳するために、アラーム コメントをエクスポートできます。

注意 : アラームとタグ コメントをエクスポートする前に、データの破損やエラーの可能性に備えて、ターゲット ディレクトリにファイルをバックアップしてください。

オフライン翻訳のためにアラーム コメントをエクスポート

1. WindowMaker を起動し、オフライン翻訳のためにアラーム コメントをエクスポートするアプリケーションを開きます。
2. ［**ファイル**］メニューで、［**設定**］をポイントし、［**言語**］を選択します。

   ［**言語**］画面が表示されます。
3. ［**言語**］画面の［**アラームのエクスポート**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. ［パス］ボックスで、ディクショナリをエクスポートするフォルダを入力します。［参照］をクリックして、既存のフォルダを選択するか、新しいフォルダを作成します。
5. ［**2 つの文字は一意のアプリケーションを表します**］ボックスで、2 文字を入力します。ID には英数字のみが含まれ、大文字小文字が区別されます。

注意：このアプリケーションからアラームやタグ コメントを以前エクスポートしたことがある場合は、次回エクスポートする際にも同じ 2 文字のアプリケーション ID を使用する必要があります。新しい 2 文字のアプリケーション ID を入力すると、InTouch HMI ではすべてのアラームやタグに ID が再生成されるため、すべての既存の翻訳が失われる原因となります。

1. ［**エクスポート**］を選択して、XML ディクショナリ ファイルに情報をエクスポートします。

   InTouch HMI によって、それぞれの設定言語に対して個々のエクスポート ファイルが作成されます。異なる言語のすべてのディクショナリ ファイルが、指定する単一のディクショナリにエクスポートされます。

   エクスポートされる言語に対して重複するファイルが存在する場合、ファイルの名前でメッセージが表示されます。エクスポートをキャンセルするか、エクスポート操作を続行できます。

   エクスポートが正常に終了すると、［**エクスポートに成功しました**］ダイアログ ボックスが表示されます。

   注意：設定されたタグ名ディクショナリのアラーム コメントのサイズが 127 文字を超える場合、またはタグ コメントが 46 文字を超える場合、アラームまたはタグ コメントはエクスポートされません。エクスポート プロセスの最後に、ディクショナリ ファイルにコメントがエクスポートされなかったことが通知され、エクスポート ディクショナリに AlarmComment.log ファイルまたは TagComment.log ファイルが作成されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［閉じる］を選択して WindowMaker ウィンドウに戻るか、［閉じてからエクスプローラを起動する］をクリックしてディクショナリ ファイルを含むフォルダを開きます。

既存のアラーム コメント ファイルへのエクスポート

オフライン翻訳のためにアラームやタグ コメントをエクスポートした後で、アラームおよびタグ コメントを再びエクスポートする必要があるアプリケーションに変更を加える必要がある場合があります。詳細については、「翻訳のためのアラームコメントのエクスポート」を参照してください。.

同じディレクトリに 2 回以上エクスポートすると、［**ファイル置換の確認**］ダイアログ ボックスが表示されます。

既存のディクショナリ ファイルを、最後にエクスポートしてから追加された新しい文字列と言語情報で更新するには、［**はい**］を選択します。既存のディクショナリ ファイルに語句の翻訳が含まれ、それを InTouch に以前インポートしていた場合、これらの翻訳は維持されます。最後のエクスポート以来、アプリケーションから任意の語句を削除した場合、その語句はディクショナリ ファイルから削除されます。

InTouch HMI で設定されたすべての言語に対する既存のディクショナリ ファイルを更新するには、［**すべてにはい**］を選択します。

既存ファイルまたはすべての言語に対する複数の既存ファイルが上書きされることを防ぐには、［**いいえ**］または［**すべていいえ**］をクリックします。

再びエクスポートされる場合、アラーム コメント、アラーム フィールド、およびタグ コメントの既存の翻訳は保持されます。

ディクショナリ ファイルの編集

ディクショナリ ファイルを作成した後で、文字列を編集する必要があります。

ディクショナリ ファイルの名前は 2 文字のアプリケーション ID とエクスポートされる言語から作成されます。たとえば、設定言語が Chinese(PRC)-2052 で、2 文字のアプリケーション ID が **AA** の場合、ファイル名は **AA_2052_ AlarmComment.xml** となります。このファイルは、ランタイム言語の切り替えファイルにより使用されるものと同じ XML スキーマを使用して記述されます。

ディクショナリ ファイルの一般的な構造は、以下のとおりです。

```
  <?xml version="1.0" encoding="utf-8" ?>
- <ArrayOfAlarmCommentPhraseItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <AlarmCommentPhraseItem Phrase="Do not edit">
      <Translation>Provide translation in this column</Translation>
      <Context>Do not edit</Context>
      <PhraseID>0</PhraseID>
      <LanguageID>0</LanguageID>
      <ForeignLanguageID>0</ForeignLanguageID>
    </AlarmCommentPhraseItem>
  - <AlarmCommentPhraseItem Phrase="System">
      <Translation />
      <Context>$System : System Tag Comment</Context>
      <PhraseID>AA1$System</PhraseID>
      <LanguageID>1033</LanguageID>
      <ForeignLanguageID>2052</ForeignLanguageID>
    </AlarmCommentPhraseItem>
  </ArrayOfAlarmCommentPhraseItem>
```

翻訳文字列の翻訳を入力します。その他の情報は変更しないでください。

アラーム状況、アラーム タイプ、およびアラーム クラスの値の一部を上書きできます。これらの値に使用できる最大長は 50 文字です。

以下のアラーム状況の値は、InTouch が生成するアラームに対して上書きできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

上書きする値	デフォルトで表示される文字列
未確認/復帰	未確認/復帰
確認/復帰	確認/復帰
未確認ｱﾗｰﾑ	未確認ｱﾗｰﾑ
確認済みｱﾗｰﾑ	確認済みｱﾗｰﾑ

以下のアラームのタイプの値は、InTouch が生成するアラームに対して上書きできます。

上書きする値	デフォルトで表示される文字列
SPC	SPC
HiHi	HiHi
HI	HI
Lo	Lo
LOLO	LOLO
MINDEV	MINDEV
MAJDEV	MAJDEV
ROC	ROC
DSC	DSC
OPR	OPR
LGC	LGC
DDE	DDE
SYST	SYST
USER	USER
プロ	プロ
LOGON_FAILED	LOGON_FAILED

以下のアラーム クラスの値は、InTouch が生成するアラームに対して上書きできます。

上書きする値	デフォルトで表示される文字列
DEV	DEV
ROC	ROC

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

上書きする値	デフォルトで表示される文字列
DSC	DSC
イベント	イベント
値	値

翻訳されたアラーム コメントをインポート

文字列を翻訳した後で、各言語に対してディクショナリ ファイルをインポートし、これらの言語に対してランタイム時の言語の切り替えを有効にする必要があります。

翻訳されたアラーム コメント ディクショナリ ファイルをインポートすると、それらはアプリケーション ディレクトリ内の該当する言語フォルダにコピーされます。

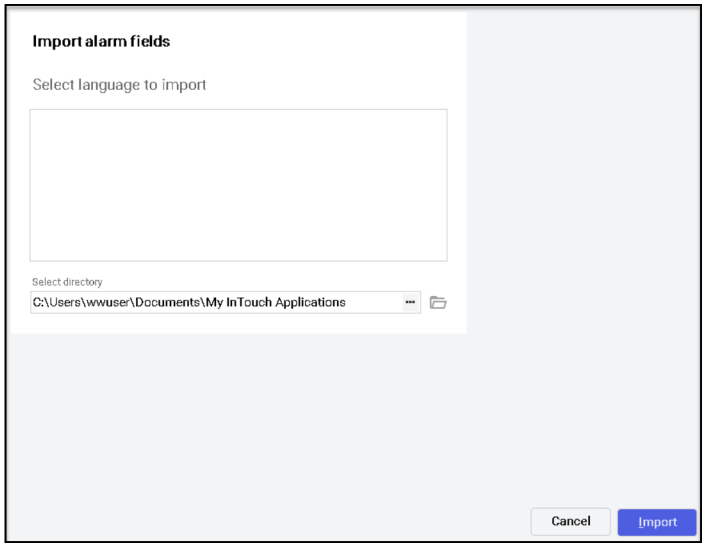
既存のアプリケーションに対して翻訳されたアラーム コメントは、インポートされたファイルの内容で上書きされ、アプリケーション バージョン（$AppVersion）は 1 ずつ増大します。

他のノードから複数のディクショナリ ファイルをインポートして、複数ノードからのアラーム フィールドのローカリゼーションをサポートするには、翻訳されたディクショナリ ファイルを他のノードから 1 つのディレクトリにコピーします。このディレクトリをインポート パスとして選択します。複数のディクショナリ ファイルが、1 回のインポート操作でインポートされます。InTouch HMI では、インポートされる言語に基づいて自動的にファイル パスが作成されます。

翻訳されたアラーム コメント ファイルをインポート

1. WindowMaker を起動し、翻訳されたディクショナリ ファイルをインポートするアプリケーションを開きます。
1. ［**ファイル**］メニューの［**設定**］を選択し、［**セキュリティ**］を選択します。

   ［**言語**］ダイアログ ボックスが表示されます。
1. ［言語］画面の［**アラームのインポート**］タブを選択します。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［パス］ボックスで、インポートするディクショナリ ファイルへのパスを入力するか、［参照］を選択してファイルを選択します。
3. **Import** を選択します。ディクショナリ ファイルで翻訳が行われていない場合、インポートする翻訳済みディクショナリ ファイルが存在しないことを示すダイアログ ボックスが表示されます。
4. ［**OK**］を選択します。インポートが正常に終了すると、［**インポートに成功しました**］ダイアログ ボックスが表示されます。

翻訳のためのアラーム グループ名のエクスポート

翻訳するためにアラーム グループ名をエクスポートできます。

アラームとタグ コメントをエクスポートする前に、データの破損やエラーの可能性に備えて、ターゲット ディレクトリにファイルをバックアップしてください。

オフライン翻訳のためにアラーム コメントをエクスポート

1. WindowMaker を起動し、オフライン翻訳のためにアラーム コメントをエクスポートするアプリケーションを開きます。
2. ［**ファイル**］メニューで、［**エクスポート**］をポイントします。
3. ［**ローカリゼーション**］を選択し、［**アラーム メッセージ**］を選択します。
4. ［**アラーム フィールドのエクスポート**］画面が表示されます。
5. ［エクスポートする言語の選択］リストで、翻訳用にエクスポートする言語を選択します。
6. ［ディレクトリの選択］ボックスで、ファイルをエクスポートする場所を選択します。
7. ［**2 つの文字は一意のアプリケーションを表します**］ボックスで、2 文字を入力します。ID には英数字のみが含まれ、大文字小文字が区別されます。

   注意：このアプリケーションからアラームやタグ コメントを以前エクスポートしたことがある場合は、次回エクスポートする際にも同じ 2 文字のアプリケーション ID を使用する必要があります。新しい 2 文字のアプリケーション ID を入力すると、InTouch HMI ではすべてのアラームやタグに ID が再生成されるため、すべての既存の翻訳が失われる原因となります。

8. ［**エクスポート**］を選択して、XML ディクショナリ ファイルに情報をエクスポートします。

   InTouch HMI によって、それぞれの設定言語に対して個々のエクスポート ファイルが作成されます。異なる言語のすべてのディクショナリ ファイルが、指定する単一のディクショナリにエクスポートされます。

   エクスポートされる言語に対して重複するファイルが存在する場合、ファイルの名前でメッセージが表示されます。エクスポートをキャンセルするか、エクスポート操作を続行できます。

   エクスポートが正常に終了すると、［**エクスポートに成功しました**］ダイアログ ボックスが表示されます。

   注意：設定されたタグ名ディクショナリのアラーム コメントのサイズが 127 文字を超える場合、またはタグ コメントが 46 文字を超える場合、アラームまたはタグ コメントはエクスポートされません。エクスポート プロセスの最後に、ディクショナリ ファイルにコメントがエクスポートされなかったことが通知され、エクスポート ディクショナリに AlarmComment.log ファイルまたは TagComment.log ファイルが作成されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

9. ［閉じる］を選択して WindowMaker ウィンドウに戻るか、［閉じてからエクスプローラを起動する］を選択してディクショナリ ファイルを含むフォルダを開きます。

翻訳されたアラーム グループ名のインポート

文字列を翻訳した後で、各言語に対してディクショナリ ファイルをインポートし、これらの言語に対してランタイム時の言語の切り替えを有効にする必要があります。

翻訳されたアラーム コメント ディクショナリ ファイルをインポートすると、それらはアプリケーション ディレクトリ内の該当する言語フォルダにコピーされます。

既存のアプリケーションに対して翻訳されたアラーム コメントは、インポートされたファイルの内容で上書きされ、アプリケーション バージョン（$AppVersion）は 1 ずつ増大します。

他のノードから複数のディクショナリ ファイルをインポートして、複数ノードからのアラーム フィールドのローカリゼーションをサポートするには、翻訳されたディクショナリ ファイルを他のノードから 1 つのディレクトリにコピーします。このディレクトリをインポート パスとして選択します。複数のディクショナリ ファイルが、1 回のインポート操作でインポートされます。InTouch HMI では、インポートされる言語に基づいて自動的にファイル パスが作成されます。

翻訳されたアラーム グループ ファイルをインポート

1. WindowMaker を起動し、翻訳されたディクショナリ ファイルをインポートするアプリケーションを開きます。
1. ［ファイル］メニューで、［保存］を選択します。
2. ローカライゼーションを選択し、 [**Alarm messages**] を選択します。

   ［アラーム フィールドのインポート］ダイアログ ボックスが表示されます。
1. ［インポートする言語の選択］リストで目的の言語を選択します。
2. ［パス］ボックスで、インポートするディクショナリ ファイルへのパスを入力するか、［参照］を選択してファイルを選択します。
3. **Import** を選択します。ディクショナリ ファイルで翻訳が行われていない場合、インポートする翻訳済みディクショナリ ファイルが存在しないことを示すダイアログ ボックスが表示されます。
4. ［**OK**］を選択します。インポートが正常に終了すると、［インポートに成功しました］ダイアログ ボックスが表示されます。

ランタイム時の言語の切り替え機能のテスト

アプリケーションでランタイム時の言語の切り替えを有効にした後で、言語の切り替え機能をテストします。アラームとタグ コメント、およびアラーム フィールドの言語の切り替えは、Alarm Viewer コントロールおよび Alarm DB View コントロールでのみ表示できます。

ローカライズされたアラームおよびタグ コメントを操作する場合は、以下の点に注意してください。

- アラームまたはタグ コメントが $Language によって指定されている言語に翻訳されていない場合、アラーム クライアントによりデフォルトのコメントが表示されます。
- Alarm Viewer コントロールが複数のプロバイダからクエリーを実行している場合、アプリケーションにリモート ノード アプリケーションの翻訳されたディクショナリ ファイルが存在すると、リモート ノードからのアラーム コメント、タグ コメント、およびアラーム フィールドも翻訳された状態で表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アラームを確認し、コメントを提供すると、ローカライズされたアラーム コメントの代わりに、アラーム クライアントによってこのコメントが表示されます。
- Alarm Viewer コントロールが停止モードの場合、言語を切り替えても、言語は切り替わった状態で表示されません。コントロールの停止モードを解除したときに、コントロールは翻訳された文字列で更新されます。
- Alarm Viewer コントロールのローカリゼーションは、コントロールの表示用のみを目的として行われています。言語が切り替えられても、すべてのスクリプト関数では通常どおりデフォルトの文字列が返されます。
- Alarm DB Logger は、データベースのデータのデフォルト言語文字列のみを保存します。データベースには、ローカライズされた文字列は保存されません。
- EVENT および ACK などのアラーム フィールドの固有 ID は事前定義され、異なるノードの複数のディクショナリ ファイルにわたり同じ ID を持ちます。アラーム クライアントは最初にロードされたディクショナリ ファイルから翻訳を選択し、他のディクショナリ ファイルの翻訳は無視されます。理想的には、1 つの言語のすべてのディクショナリ ファイルのアラーム フィールドの翻訳が同じである必要があります。指定された言語の同じアラーム状態に対しては、複数のアラーム クライアント（Alarm DB View コントロールおよび Alarm Viewer コントロール）で同じ翻訳が使用されます。
- 翻訳されたテキストは、アラーム コメントの場合は 131 文字、タグ コメントの場合は 160 文字に切り捨てられます。

言語の切り替え機能をテスト

1. WindowViewer でアプリケーションを開きます。
2. ［**ファイル**］メニューで、［**設定**］をポイントし、［**言語**］を選択します。次に、切り替える言語の名前を選択します。

   対応する翻訳されたディクショナリ ファイル（存在する場合）からの情報がロードされ、表示されます。
3. 言語を切り替えるためのボタンを追加している場合は、ボタンを選択して、スクリプトをテストします。

ネットワーク アプリケーション開発（NAD）クライアントへのローカライズされたファイルの配布

ローカライズされたアラーム コメント、タグ変数コメント、およびアラーム フィールドを含むファイルは、InTouch アプリケーションの一部としてネットワーク アプリケーション開発（NAD）クライアントに配布されます。アラーム コメントを含む更新ファイルを受け取った場合、サポートされているアラーム クライアントで翻訳されたアラーム コメントを表示するには、WindowViewer を再起動する必要があります。

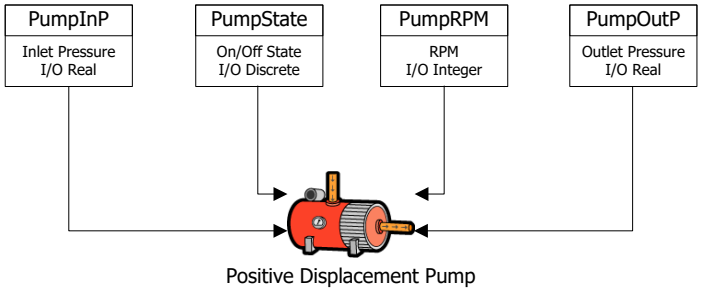
ネットワーク アプリケーション開発（NAD）と組み合わせた言語の切り替え機能を使用する場合は、NAD クライアント ノードの変更モードを、「WindowViewer の再起動」または「WindowViewer の再起動時にメッセージ表示」に設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 23 タグ

InTouch ヒューマン マシン インターフェイス（HMI）は、製造環境におけるコンポーネントをグラフィック表示するアプリケーションです。このグラフィカル インターフェイスを使用して、工場のオペレータは製造工程を監視および管理することができます。

以下の図は、製造工程のコンポーネントであるポンプの例を示しています。ポンプには、値に関連付けられているプロパティがあります。圧力、RPM、およびステータスはポンプのプロパティであり、その値は HMI から監視されます。



タグは InTouch HMI アプリケーションのデータ アイテムを表します。タグを使用すると、製造環境からの特定のデータ アイテムとして、特定のコンポーネント プロパティをアクセス可能にできます。上の図では、PumpState タグはポンプがオンであるかオフであるかを示しています。製造環境において、InTouch アプリケーションでプロパティを監視または制御するコンポーネントに対してタグを作成します。

製造コンポーネントから収集された異なるタイプのデータに対しては異なるタイプのタグを使用できます。たとえば、PumpState タグは、ポンプが実行中であるか停止中であるかを示すブール型のオン/オフ値を返します。アプリケーションの一部とするデータ型に対しては、適切なタイプの InTouch タグを割り当てます。

Tag Viewer を開始

Tag Viewer はタグを監視し、ランタイム時にタグの値を変更できる外部アプリケーションです。Tag Viewer では、アラームグループに基づいて、アプリケーションで使用可能なすべてのタグの階層の一覧を表示できます。Tag Viewer は WindowViewer が実行しているときにのみ実行できます。Tag Viewer にはローカルの InTouch アプリケーションで使用可能なタグだけが表示されます。リモート リファレンスはサポートされていません。

Tag Viewer を使用するには、最初に WindowMaker で Tag Viewer を有効にしておく必要があります。それ後、ランタイム時にアプリケーションを起動します。一度に開くことのできる Tag Viewer のインスタンスは 1 つだけです。

Tag Viewer の有効化

デザイン時に WindowViewer のプロパティを設定して Tag Viewer を有効にすることができます。また WindowViewer のメニューを設定して、［システム］メニューに Tag Viewer オプションを表示することもできます。

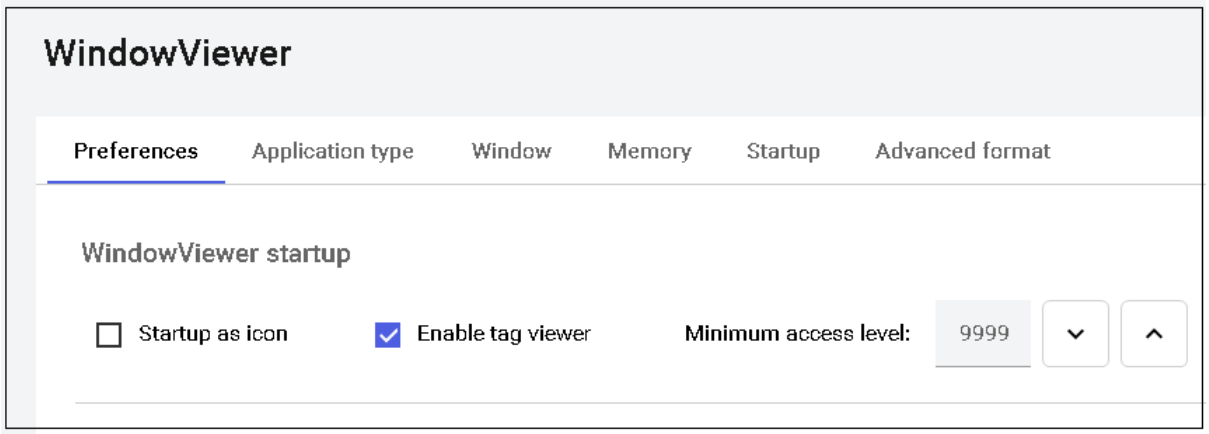
WindowViewer が開いている状態でこれらのプロパティを変更した場合、変更を適用するには WindowViewer を再起動する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Tag Viewer の有効化

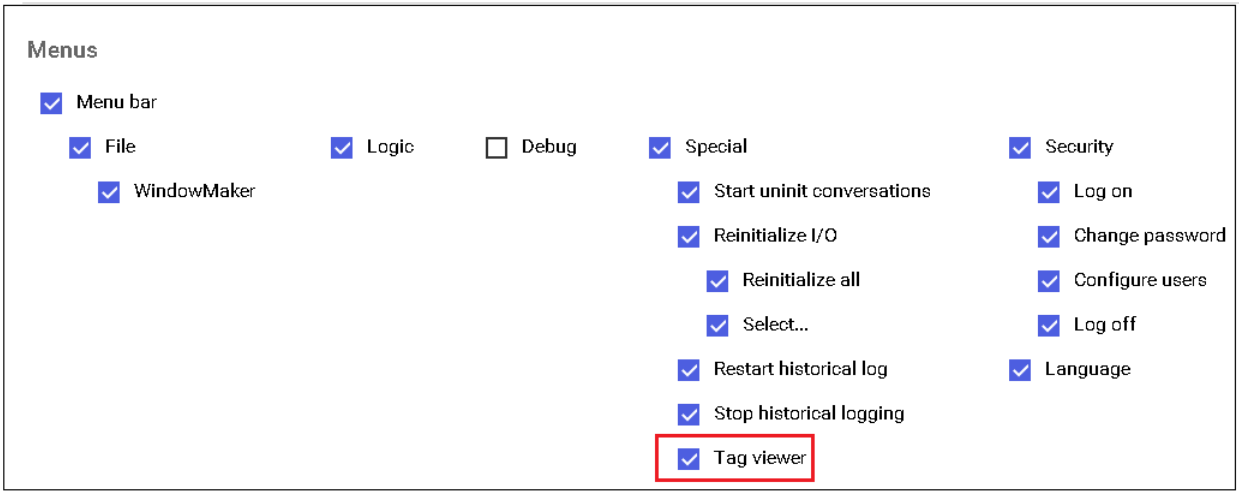
1. WindowMaker を開きます。
2. ［ファイル］メニューで、［設定］をポイントし、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［システム設定］タブを選択します。



4. ［**WindowViewer 起動時の設定**］領域で、［**Tag Viewer の有効化**］チェック ボックスをオンにして、Tag Viewer を起動できるようにします。デフォルトでは、このチェック ボックスはオンになっていません。
5. ［最小アクセス レベル］ボックスで、Tag Viewer アプリケーションを実行するために必要なセキュリティ アクセス レベルを設定します。デフォルトの値は、9999 ですが、0 ～ 9999 の値を入力できます。

   注意：ユーザーのアクセス レベルが最低限のアクセス レベルよりも低い場合、ランタイム時に Tag Viewer を起動することはできません。
6. ［ウィンドウ］タブを選択します。



1. ［メニュー］領域の［**Tag Viewer**］チェック ボックスをオンにして、WindowViewer の［システム］メニューの［**Tag Viewer**］オプションを有効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**OK**］ を選択します。
3. これらのパラメータのいずれかを変更した場合、変更を適用するには WindowViewer を再起動する必要があります。

Tag Viewer の起動

Tag Viewer は、WindowViewer が実行していて、デザイン時に WindowViewer が有効に設定されいる場合にのみ開始することができます。Tag Viewer を起動するには、WindowViewer のメニューから選択するか、LaunchTagViewer() スクリプト関数を呼び出します。この関数は OnStartup と OnShutdown 以外の、どのスクリプト タイプからでも実行することができます。Tag Viewer が有効化されていない場合、スクリプト関数を呼び出しても Tag Viewer は起動せず、警告メニューがログに記録されます。

セキュリティ

Tag Viewer を実行するには、適切なセキュリティ権限が必要です。使用可能な InTouch ユーザー セキュリティ レベルを次の表に示します。

セキュリティ レベル	説明
ArchestrAArchestrA	このオプションを選択した場合、以下の手順を実行する必要があります。 1. ［**WindowViewer のプロパティ**］画面で［**最小アクセス レベル**］以上のアクセス レベルを設定します。 2. WindowViewer にログオンして、Tag Viewer を起動します。 この手順は、マネージド InTouch アプリケーションを実行している場合に実行します。
グラフィック エディタでは、「InTouch:」と入力しなくても	このオプションを選択した場合、以下の手順を実行する必要があります。 1. ［**WindowViewer のプロパティ**］画面で［**最小アクセス レベル**］以上のアクセス レベルを設定します。 2. WindowViewer にログオンして、Tag Viewer を起動します。 この手順は、スタンドアロンの InTouch アプリケーションを実行している場合に実行します。
なし	このオプションを選択した場合、ユーザーは WindowViewer にログオンせずにランタイム時に Tag Viewer を起動することができます。
OS	このオプションを選択した場合、以下の手順を実行する必要があります。 1. ［**WindowViewer のプロパティ**］画面で［**最小アクセス レベル**］以上のアクセス レベルを設定します。 2. WindowViewer にログオンして、Tag Viewer を起動します。 この手順は、スタンドアロンの InTouch アプリケーションを実行している場合に実行します。
CONNECT	これは、スタンドアロンの InTouch アプリケーションを実行している場合、および Configurator の [**ライセンスモード**] タブで**接続済みエクスペリエンス**オプションを選択した場合に適用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アプリケーションを開いている場合、Tag Viewer を実行するには、再度ログオンして再認証を受ける必要があります。ユーザーの設定の詳細については、『AVEVA™ InTouch HMI 管理ガイド』の「ユーザーの管理と承認レベルの設定」を参照してください。

InTouch のセキュリティ レベルが［なし］の場合、再度ログオンする必要はありません。

ログオン

1. WindowViewer を開きます。
2. ［**システム**］ メニューで、［**セキュリティ**］をポイントして、［**ログオン**］を選択します。
3. 有効なユーザー ID とパスワード、および必要最低限のアクセス レベルを使用してログオンします。

注意：管理者としてログオンするか、適切な権限を持つユーザー ID を作成することができます。

Tag Viewer の起動

1. WindowViewer を開きます。
2. ［システム］メニューで、［Tag Viewer］を選択します。Tag Viewer が表示されます。

Tag Viewer をスクリプトで開始

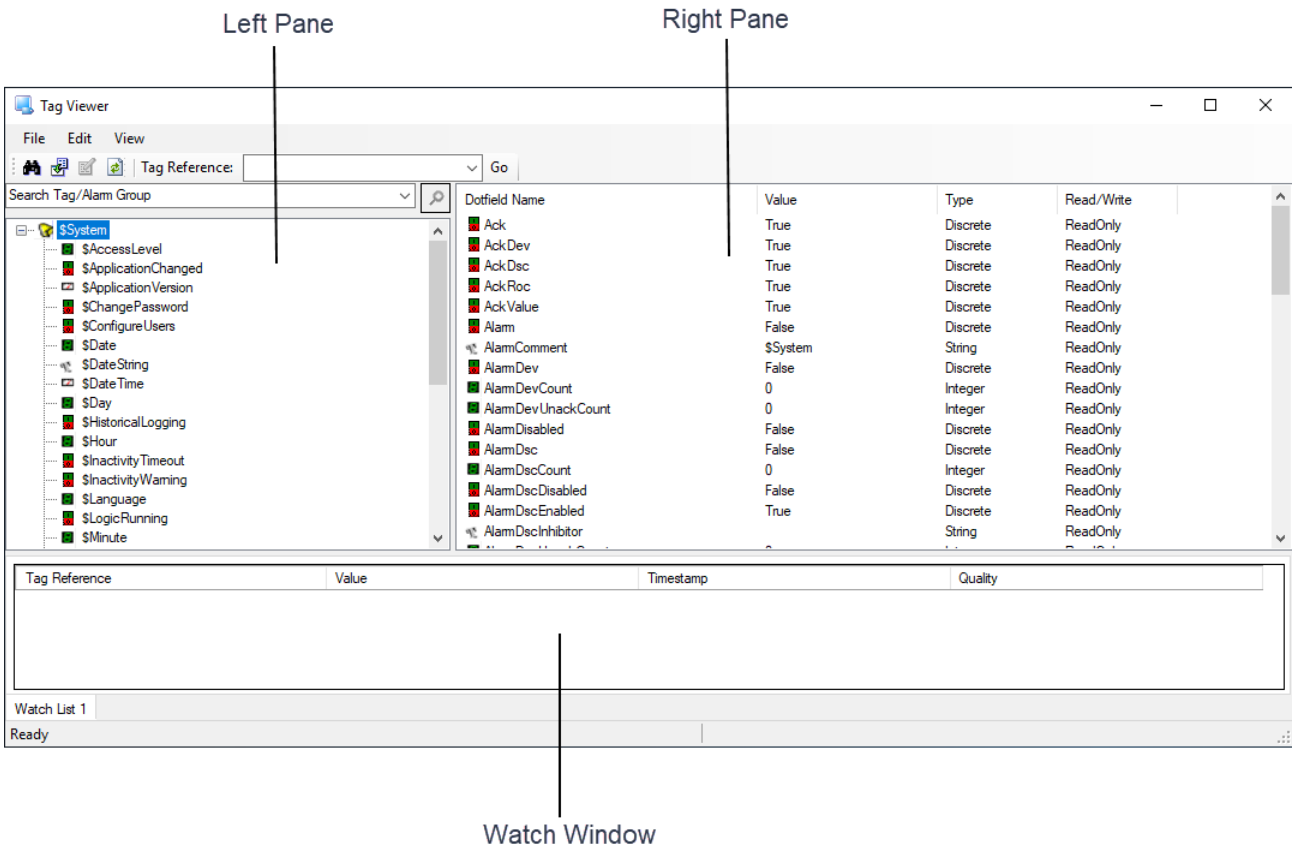
1. WindowMaker を開きます。
2. 「On Startup」および「On Shutdown」アプリケーション スクリプト以外の LaunchTagViewer() スクリプトを設定します。
3. WindowViewer を開きます。
4. スクリプトを実行して Tag Viewer を起動します。

Tag Viewer でのナビゲート

Tag Viewer ウィンドウは 3 つの部分で構成されています。

- 左ペインには、アラーム グループ別の階層形式でタグが表示されます。
- 右ペインには、左ペインで選択したタグまたはアラーム グループの使用可能なドットフィールドのリストが表示されます。
- 下部の監視ウィンドウには、監視するタグのランタイム時の値が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



Tag Viewer を閉じる

Tag Viewer は次の方法で閉じることができます。

- Tag Viewer を閉じる
- WindowViewer を閉じる
- ログオンしているユーザーを変更する

Tag Viewer の使用

Tag Viewer をランタイム時に使用して、以下のことを行うことができます。

- タグの検索
- タグの表示
- タグのプロパティの変更
- 監視ウィンドウの追加と監視ウィンドウへの情報入力
- 監視ウィンドウの管理

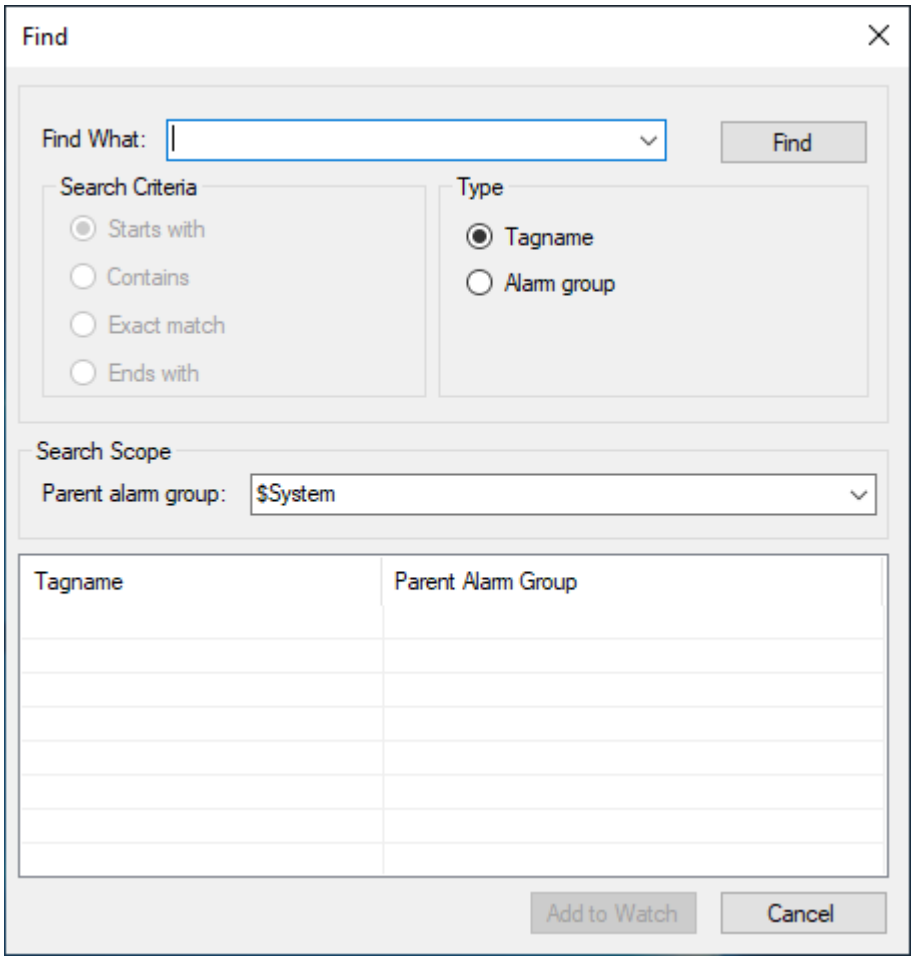
タグを検索

WindowViewer で特定のタグを検索するには、［**検索**］ダイアログ ボックスを使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグを検索

1. ［編集］メニューで、［タグを検索/アラーム グループ］を選択します。［**検索**］ダイアログ ボックスが表示されます。



2. 検索するタグまたはアラーム グループの名前を［**検索する文字列**］ボックスに入力します。デフォルトでは、最後に検索したタグまたはアラーム グループの名前が表示されます。［**検索する文字列**］ボックスには検索履歴も保持されています。タグまたはアラーム グループを初めて検索する場合、［**検索する文字列**］ボックスは空白です。

注意：検索履歴は Tag Viewer を閉じると削除されます。

3. ［**検索条件**］領域で、以下のいずれかを選択します。
   - ［**検索する文字列**］ボックスに入力したテキストで始まるタグ変数またはアラーム グループを検索する場合、このオプションを選択します。デフォルトでは、このオプションが選択されています。
   - ［**検索する文字列**］ボックスに入力したテキストを含むタグ変数またはアラーム グループを検索する場合、このオプションを選択します。
   - ［**検索する文字列**］ボックスに入力したテキストと一致するタグ変数またはアラーム グループを検索する場合、このオプションを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**検索する文字列**］ボックスに入力したテキストで終わるタグ変数またはアラーム グループを検索する場合、このオプションを選択します。

4. ［**タイプ**］領域で、以下のいずれかを選択します。
   - タグ名：タグを検索する場合、このオプションを選択します。デフォルトでは、このオプションが選択されています。
   - アラーム グループ：検索する場合、このオプションを選択します。
5. タグの所属するアラーム グループの名前を［**親のアラーム グループ**］ボックスに入力します。デフォルト値の **$System** を保持する場合、フィルタされていない検索結果がアラーム グループ別に表示されます。
6. [**Find(検索)]** を選択します。タグまたはアラーム グループのリストが表示されます。
7. 検索結果に表示されたタグの名前をダブル選択すると、そのタグが Tag Viewer で選択されます。

タグ変数を監視ウィンドウに追加するには、［監視リストに追加］を選択します。

クイック検索を行う

必要に応じて、Tag Viewer ではタグをすばやく検索することができます。

クイック検索を行う

1. 左ペインの上にある検索ボックスに、タグの完全な名前またはその一部を入力します。

   注意：Tag Viewer を閉じるまで、検索ボックスは検索履歴を保持します。

2. Enter キーを押すか、検索アイコンを選択します。入力したテキストで始まるタグが選択されます。

   注意：入力したテキストがワイルドカード（*）で始まる場合、テキストを含むすべてのタグが検索されます。

3. 検索アイコンをもう一度選択して、検索を続行します。

   注意：クイック検索を行うには、テキストを入力してから **F3** キーを押します。

タグの管理

Tag Viewer では、タグの監視やランタイム時のタグの値の監視を行うことができます。アプリケーションで使用可能なすべてのタグを表示することができます。タグはアラーム グループ別にツリー形式で表示されます。

タグの表示

Tag Viewer でタグの詳細を表示することができます。

タグの表示

1. Tag Viewer を開きます。
2. 左ペインには、アプリケーションで使用可能なタグのリストが表示されます。

   注意：Tag Viewer には InTouch アプリケーションで使用可能なタグだけが表示されます。リモート リファレンスはサポートされません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. 左ペインでタグを選択すると、右ペインにそのタグの詳細が表示されます。次の情報が表示されます。
   - ドットフィールド:選択したタグまたはアラーム グループのドットフィールドが表示されます。
   - 値 : ドットフィールドの値を表示します。ドットフィールドを選択すると、その値が更新されます。
   - タイプ : 整数型や論理型など、ドットフィールドのタイプを表示します。
   - 書き込み : タグが読み取り専用または書き込み可能かどうかを示します。
4. ランタイム時にドットフィールド値を表示するには、[監視] ウィンドウにタグを追加します。[監視] ウィンドウへのタグの追加の詳細については、この付録の「監視ウィンドウへのタグまたはドットフィールドの追加」を参照してください。

タグのプロパティの変更

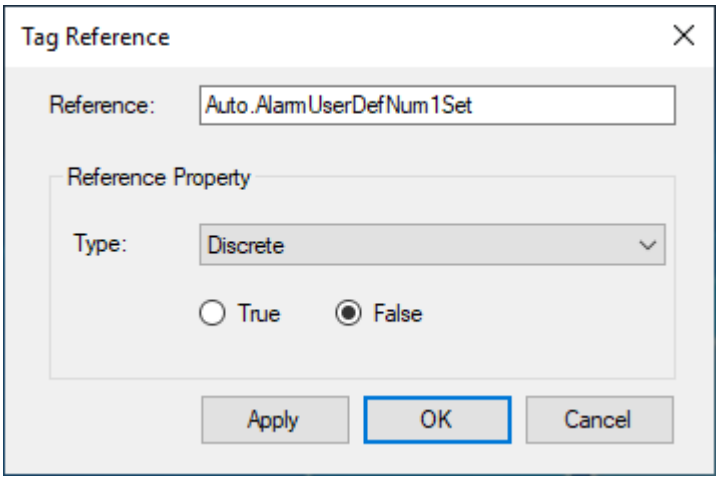
Tag Viewer でタグのプロパティを変更することができます。タグのプロパティは、ドットフィールドが [読み取り/書き込み] または [書き込み専用] としてマーク付けされている場合にのみ変更できます。WindowViewer を読み取り専用ライセンスで実行している場合、タグは変更できません。

以下のタグを変更しようとすると再認証が求められます。

- セキュリティ書き込みの設定された Galaxy 属性をポイントしている間接型タグ
- 確認書き込みの設定された Galaxy 属性をポイントしている間接型タグこの場合、確認者であるユーザーのログオン認証も求められます。

タグのプロパティの変更

1. Tag Viewer を開きます。
2. 左ペインで、プロパティを変更するタグを選択します。タグの詳細が右ペインに表示されます。
3. 以下のいずれかを実行します。
   - 右ペインでプロパティを選択して、ツールバーの [Go] を選択します。[タグ リファレンス] ダイアログ ボックスが表示されます。



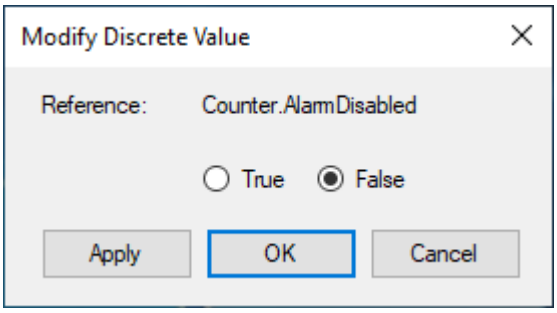
   - [タイプ] リストで、タグのタイプを選択して、値を変更します。

     ドットフィールドのデータ タイプを変更する場合は、右ペインの値をダブルクリックします。[変更] ダイアログ ボックスが表示されます。ドットフィールドのデータ タイプによって、表示

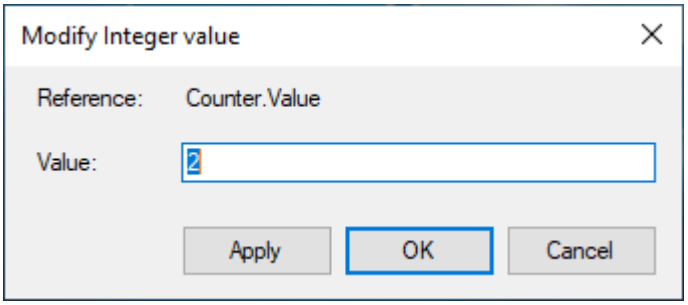
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

されるダイアログ ボックスが異なります。以下のデータ タイプの値を変更することができます。

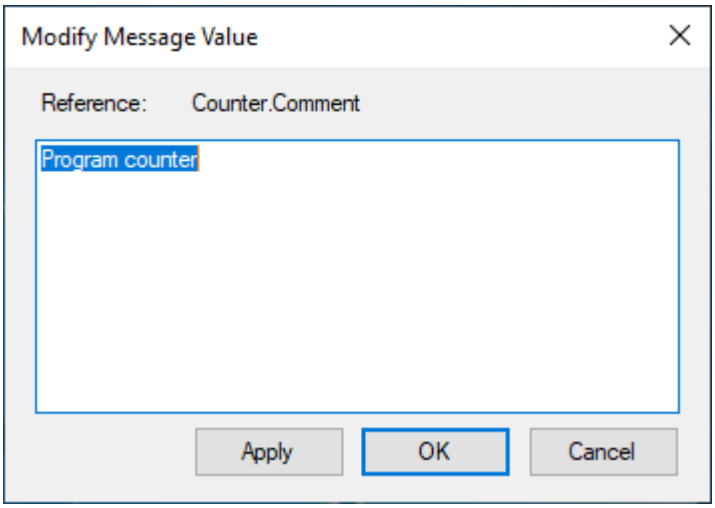
離散値: **True** または **False** を選択できます。



数字：
-2147483648 から 2147483647 の間の任意の数値を入力できます。



メッセージ：最大 131 文字のテキスト メッセージを入力できます。



- アイテムを右クリックし、［変更］を選択します。［**変更**］ダイアログ ボックスが表示されます。値を変更します。

4. **[適用]** を選択します。値が変更されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

監視ウィンドウの管理

デフォルトでは、監視ウィンドウは Tag Viewer の下部に表示されます。このウィンドウを使用して、ランタイム時にタグの値を表示することができます。追加の監視ウィンドウを作成して、関連性のあるタグをグループ化できます。必要に応じて、監視ウィンドウを削除することもできます。

監視ｳｨﾝﾄﾞｳを追加

追加の監視ウィンドウを作成して、グループ化したタグを監視することができます。新規に作成した監視ウィンドウは、デフォルトの監視ウィンドウの横にタブとして追加されます。監視ウィンドウは最大 50 個まで作成できます。

監視ｳｨﾝﾄﾞｳを追加

1. 監視ウィンドウを右クリックして、[**監視ウィンドウを追加**] を選択します。新しい監視ウィンドウが追加されます。デフォルトでは、このウィンドウには「監視リスト <n>」という名前が付けられます。

   監視ウィンドウの名前を変更するには、監視ウィンドウのタブを右クリックして［タブ名を変更］を選択します。

2. 必要に応じて、タグを監視ウィンドウに追加します。
3. タグを別の監視ウィンドウに移動するには、移動元と移動先のタブを選択します。

監視ウィンドウへのタグまたはドットフィールドの追加

タグまたはドットフィールドを監視ウィンドウに追加して、ランタイム時の値を監視することができます。1 つの監視ウィンドウには、最大 2000 個のタグまたはドットフィールドを追加できます。

注意：アラーム グループを監視ウィンドウに追加した場合、そのアラーム グループに属するすべてのタグが監視リストに追加されます。

タグを監視ウィンドウに追加

1. 以下のいずれかを実行します。
   - 左ペインでタグ変数を右クリックして、［監視に追加］を選択します。タグが監視ウィンドウに追加されます。
   - 左ペインでタグを選択してドラッグし、監視ウィンドウにドロップします。
   - 監視ウィンドウを右クリックして、［タグ リファレンスを追加］を選択します。
   - 右ペインでドットフィールドを右クリックして、［監視に追加］を選択します。ドットフィールドが監視ウィンドウに追加されます。

   複数のタグまたはドットフィールドを選択して、監視ウィンドウに追加することができます。

2. 昇順または降順でテーブルをソートするには、カラム ヘッダーをクリックします。

グループ タグへのセパレータの追加

複数のタグの後にセパレータ（区切り文字）を追加して、同類のタグをグループ化することができます。監視ウィンドウを保存すると、監視ウィンドウのセパレータも自動的に保存されます。

グループ タグへのセパレータの追加

1. 監視ウィンドウでセパレータを追加する行を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. 監視ウィンドウを右クリックして、[**セパレータを追加**] を選択します。選択した行の下にセパレータが追加されます。
3. タグ リファレンスと一緒にセパレータをドラッグして別の場所にドロップすると、監視ウィンドウに表示されるタグの順番を変更できます。

注意：監視ウィンドウの情報をソートした場合、セパレータは削除されます。

監視ウィンドウのバックアップとリストア

監視ウィンドウをバックアップとして保存しておき、必要に応じて再度読み込むことができます。

監視ウィンドウを保存

必要に応じて、開いているすべての監視ウィンドウを保存することができます。これらの監視ウィンドウを後で読み込んでタグを監視することができます。

監視ウィンドウを保存

- [ファイル] メニューの [監視リストを上書き保存] をクリックして、監視ウィンドウを保存します。デフォルトでは、監視ウィンドウを現在のアプリケーション ディレクトリに**監視リスト**フォルダに保存します。

注意：監視ウィンドウを別の場所に保存、[**ファイル**] メニューの [**監視リストに名前を付けて保存**] をクリックします。

監視ウィンドウを読み込む

保存した監視ウィンドウを開くことができます。

監視ウィンドウを読み込む

1. [**ファイル**] メニューで、[**監視リストをロード**] を選択します。[**ファイルの選択**] ダイアログ ボックスが表示されます。

   監視ウィンドウを右クリックして、 [監視リストをロード] をクリックすることもできます。

2. ファイルの保存されている場所を参照して、ファイルを開きます。

   注意：監視ウィンドウを開くと、既存のウィンドウが新しいウィンドウに置き換えられます。現在のアプリケーションで有効なリファレンスのみが、監視ウィンドウに追加されます。

I/O 参照

Tag Viewer に表示される I/O リファレンスまたは間接型タグは、WindowViewer のウィンドウのように動作します。

- I/O または間接型タグを監視ウィンドウで削除した場合、I/O リファレンスは要求されません。
- /O または間接型タグを監視ウィンドウに追加していない状態で、ドットフィールドが更新された場合、または左ペインのタグをクリックした場合は、Tag Viewer が I/O リファレンスをサブスクライブします。そのタグが選択されている間は、サブスクライブが持続します。
- Tag Viewer を閉じた場合、Tag Viewer が要求した I/O または間接型タグのサブスクライブが解除されます。
- Tag Viewer を閉じた場合や監視ウィンドウでタグを削除した場合、WindowViewer のタグには影響しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Tag Viewer の持続性

次の点において、Tag Viewer には InTouch アプリケーションの持続性があります。

- メイン ウィンドウのサイズと場所
- 左ペインに表示されるタグのツリー構造のサイズ、左ペインに表示されるドットフィールドのリスト、および監視ウィンドウ

次の点において、Tag Viewer には WindowViewer セッションの持続性があります。

- 左ペインで選択したタグ
- 監視ウィンドウ
- ステータス バーおよびツール バー

注意：WindowViewer を閉じると、WindowViewer の持続性は失われます。

監視ウィンドウのファイル形式

保存した監視ウィンドウのファイルは、XML ファイル形式で保存されます。XML スキーマを以下に示します。

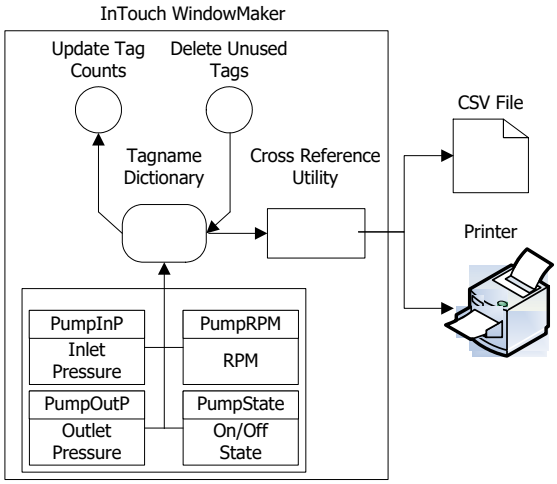
```
<?xmlversion=”1.0”encoding=”UTF-8”?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="InTouchTagViewer">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="DependentFiles">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ReferenceString" type="xs:string" minOccurs="0"
maxOccurs="unbounded"></xs:element>
              <xs:element minOccurs="1" maxOccurs="1" name="DependentFiles">
             <xs:sequence>
           <xs:complexType>
         </xs:element>
       <xs:sequence>
     <xs:complexType>
   </xs:element>
</xs:schema>
```

タグの使用数の削減

InTouch アプリケーションで使用可能なタグの最大数は、所有するライセンスによって異なります。実行中のアプリケーションで最大制限を超える数のタグを同時に使用することはできません。

InTouch アプリケーションでは、タグの使用数をトラッキングできます。以下の図は、クロス リファレンス ユーティリティがタグ名ディクショナリの内容を分析して、タグ使用数レポートを生成する方法を示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



InTouch アプリケーションを実行するためには、最低限のタグを保持している必要があります。タグの数を最小限にする方法の 1 つは、未使用のタグを削除することです。未使用のタグを削除する前に、タグの数を更新する必要があります。

InTouch と Historian

Historian は、工場のプロセス データを保存するリアルタイムな履歴データベースです。リレーショナル データベースの記憶容量と、リアルタイム システムのスピードを兼ね備えています。Historian は、Microsoft SQL Server の機能強化として工場のデータ取得を大幅に高速化します。また、InTouch HMI およびその他の関連製品のイベント、サマリ、生産、および履歴データが工場のデータと統合されます。

InTouch HMI では、履歴タグ変数データをリモートの履歴リポジトリに保存できます。この場合、InTouch HMI アプリケーション ディレクトリを共有し、InTouch アプリケーションにアクセスするよう Historian を設定する必要があります。Historian を使用して InTouch HMI 履歴データを保存する場合、WindowMaker から分散名前マネージャを使用して、データベースへの接続を指定する必要があります。InTouch HMI では、Historian データベースに保存された履歴データをトレンド ウィザードに表示できます。履歴タグ変数データにアクセスするには、SQL Server クライアント コンポーネントをインストールし、Historian コンピュータを参照する必要があります。

タグ変数の使用数の確認

InTouch HMI は、ローカル タグ変数ディクショナリで定義されているすべてのタグ変数の使用数を管理します。タグ変数の数に、内部システム タグ変数は含まれません。

リモート タグは、タグ変数ディクショナリでは定義しません。InTouch では、アプリケーション内でリモート タグが参照されるたびにタグ変数の数が増えます。InTouch でのリモート タグ変数リファレンスのカウント方法の詳細については、「ライセンスに基づくリモート タグの最大数を確認」を参照してください。

未使用のタグ変数を削除する前に、以下のタスクを完了してください。

- WindowViewer を閉じます。
- ローカル タグ変数およびリモート タグへの参照の数を検索します。
- クロス リファレンスのタグ変数レポートを生成します。
- クロス リファレンス ユーティリティでアプリケーション内のタグ変数の使用数を調べます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アプリケーションを WindowMaker に保存します。

タグの数の確認

アプリケーションのタグ名ディクショナリで現在定義されているローカル タグの数を更新できます。また、別のノードに置かれたリモート タグに対するアプリケーションの参照数も検索可能です。

ローカル タグの数を表示

1. WindowMaker で InTouch アプリケーションを開きます。
2. ［ファイル］メニューの［設定］を選択し、［**WindowMaker**］を選択します。

   WindowMaker の設定画面が表示されます。
3. ［タグの数の表示］を選択します。

   このオプションを選択すると、タグ名ディクショナリ全体が読み込まれ、表示されているタグの数が更新されます。タグ名ディクショナリの更新には時間がかかることがあります。
4. 保存を選択します。設定の変更を有効にするために WindowMaker を再起動するようメッセージが表示されます。
5. WindowMaker を閉じます。
6. WindowMaker でアプリケーションを再起動します。

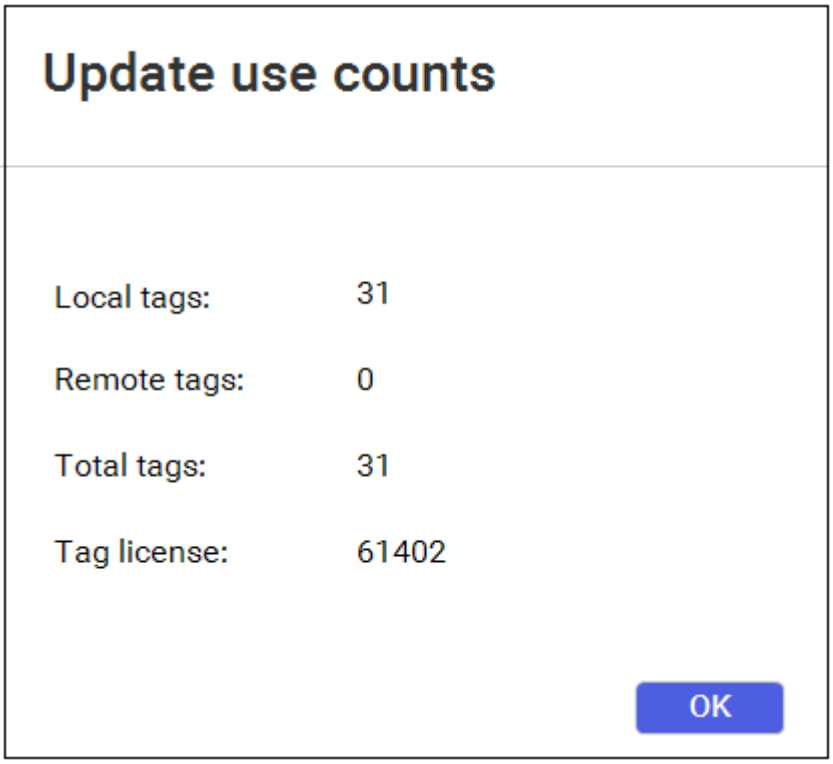
   アプリケーションのタグ名ディクショナリで定義されているローカル タグの数が、WindowMaker のメニュー バーの右側に表示されます。

アプリケーション内でリモート参照を含むタグの合計数を更新

1. WindowMaker で InTouch アプリケーションを開きます。
2. 開いているすべてのウィンドウを閉じます。
3. ［ホーム］メニューの［タグ］グループで［使用カウントを更新］をクリックします。

   計算が終了すると、ダイアログ ボックスにアプリケーションのローカル タグおよびリモート タグへの参照の数が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



このダイアログ ボックスには、タグの合計数と InTouch のライセンスに基づいたタグの最大使用数も表示されます。

ライセンスに基づくリモート タグの最大数を確認

製品のライセンスによって、InTouch アプリケーションで使用できるタグ変数の数が決定されます。ローカル タグ変数の数だけでなく、リモート ノードにあるタグ ソースを参照するタグ変数の数もライセンスによって施行されます。

アプリケーションで使用できるリモート リファレンスのタグ変数の数が InTouch ライセンスによってどのように施行されるのかについては、「InTouch ライセンス」を参照してください。

タグの使用箇所を特定

クロス リファレンス ユーティリティを使用すると、InTouch アプリケーション内のタグの使用数を示すオンライン レポートを生成できます。

クロス リファレンス ユーティリティのレポートには、次の中に含まれるローカル タグ、リモート タグ、スーパータグが表示されます。

- アニメーション リンク
- ウィザード
- すべてのタイプの InTouch スクリプト
- クイック関数
- ActiveX コントロール
- 最適な InTouch コンポーネント（SQL アクセス マネージャやレシピ マネージャなど）
- 産業用グラフィック参照

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クロス リファレンス ユーティリティでレポートを作成

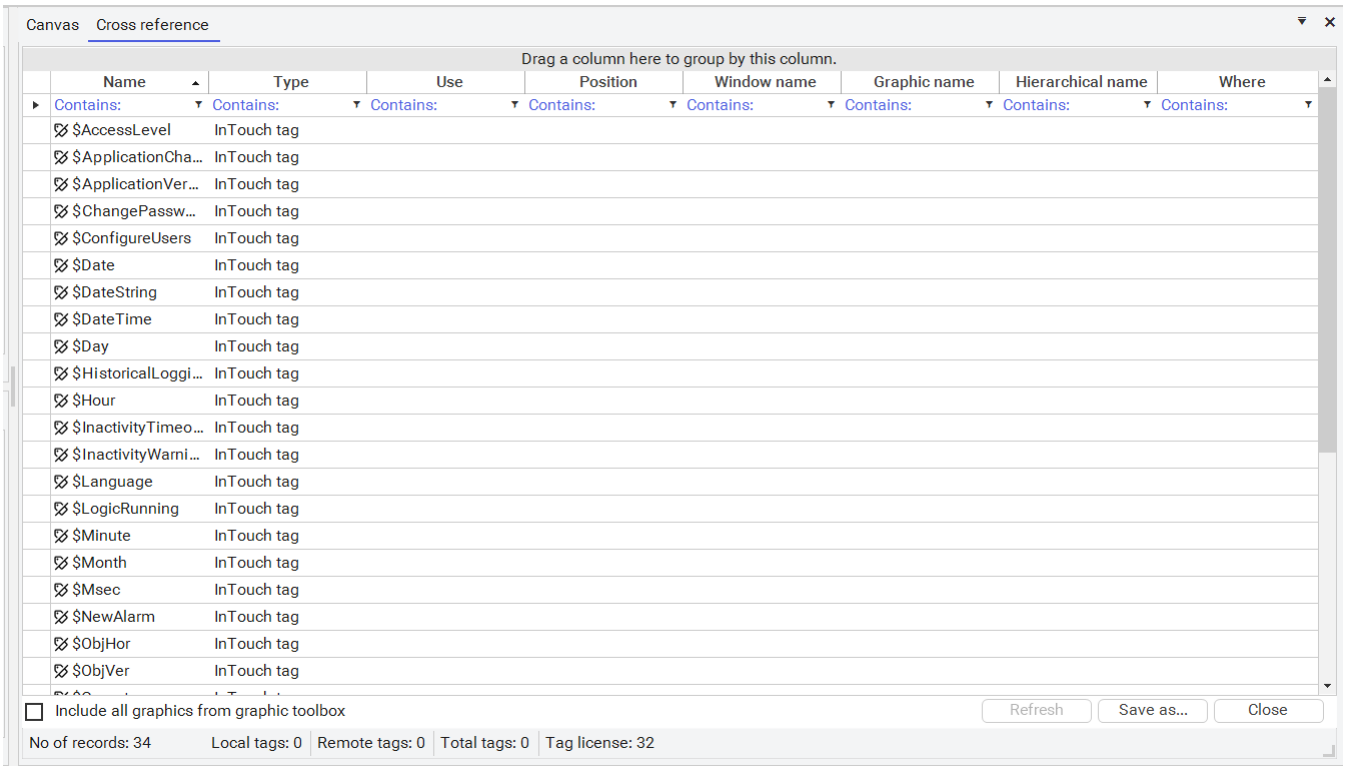
1. WindowMaker で InTouch アプリケーションを開きます。
2. ［**ホーム**］メニューの［**タグ**］グループで［**クロス リファレンス**］をクリックします。

   クロス リファレンス グリッドが表示されます。

   クロス リファレンス グリッドには、現在のアプリケーションのタグ名ディクショナリで定義されているローカルおよびリモート タグに関連するすべてのレコードが一覧表示されます。

クロス リファレンス ユーティリティ レポートの理解

グリッド形式では、レポートには、参照タイプの名前、タグ リファレンスのタイプ、使用場所、ウィンドウ上の位置、ウィンドウ名、グラフィック名、タグ リファレンスのある産業用グラフィックの階層のフル パス、および条件タイプ（スクリプトでタグが使用されている場合）が一覧表示されます。



以下に例を示します。

名前	タイプ	使用場所	位置	ウィンドウ名	グラフィック名	階層名	トリガ
NSelect	InTouch タグ	キー スクリプト					F6 を押した時
PLCSim.SP3	オブジェクト属性	産業用グラフィック	始点: (0,0) 終点: (826,455)	セクション 5	SA_Meters2	SA_Meters.SA_Tank_Vessel.SA_OperatingRange	

レポートの下部にあるステータス バーには、以下の内容が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アプリケーション内のタグに関連するレコードの合計数。この値は、適用したフィルタに基づいて動的に変更されます。
- **ローカルタグ**：このノード上で作成されたタグの合計数。
- **リモートタグ**：リモート タグを参照するタグの合計数。
- **タグ合計数**：InTouch アプリケーション内のタグの合計数。
- **タグライセンス**：ライセンスごとに使用可能なタグの合計数。

InTouch のクロス リファレンス ユーティリティのレポートには、ステータスおよび使用数を示すアイコンが表示されます。

アイコン	説明
	タグまたはスーパータグはアプリケーションのタグ名ディクショナリで定義されていますが、オブジェクトには割り当てられていません。
	タグまたはスーパータグは、アニメーション リンク、InTouch QuickScript、または産業用グラフィックで使用されています。
	産業用グラフィックは、タグまたはスーパータグを参照しています。
	タグまたはスーパータグはアニメーション リンクに割り当てられています。
	タグまたはスーパータグは、アプリケーション スクリプトで使用されています。
	選択されているすべてのスクリプトに表示されます。スクリプトを読み取り専用モードで表示するには、スクリプト名をダブルクリックします。
	タグまたはスーパータグはウィンドウ スクリプトで使用されています。
	タグまたはスーパータグはデータ変更スクリプトで使用されています。
	タグまたはスーパータグは条件スクリプトで使用されています。
	タグまたはスーパータグはキー スクリプトで使用されています。
	タグまたはスーパータグはクイック関数で使用されています。
	タグまたはスーパータグは、ActiveX イベント スクリプトで使用されています。
	ウィンドウによるクロス リファレンスを行うと、表示されているタグまたはスーパータグが使用されているウィンドウ名の前にこのアイコンが付きます。アイコンをダブルクリックすると、このウィンドウで使用されているすべてのタグが表示されます。
	表示されているタグまたはスーパータグは SQL アプリケーションで使用されています。
	表示されているタグまたはスーパータグはレシピ マネージャ アプリケーションで使用されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アイコン	説明
	タグは、アラーム抑止に使用されています。

グラフィック ツールボックスからのグラフィックの包含

- ［更新］を選択して、最新のタグ情報を表示します。クロス リファレンス ウィンドウが開いているときにタグ変数情報を編集した場合、またはウィンドウに埋め込まれたシンボルを編集した場合、ウィンドウ［更新］ボタンが有効になります。
- ［グラフィック ツールボックスのすべてのグラフィックを含める］チェックボックスを選択すると、グラフィック ツールボックスに存在していてもいずれのウィンドウでも使用されていないタグが表示されます。デフォルトでは、このチェックボックスは選択されていません。

  例えば、タグ ディクショナリに 10 個のタグが含まれていて、4 つのタグがグラフィック ツールボックスのグラフィックで参照されているとします（ウィンドウには埋め込まれていません）。その他の 4 つのタグは 1 つのウィンドウに配置されたグラフィックで参照されているとします。デフォルトでは、4 つのタグ（ウィンドウ上に配置されたタグ）が表示されます。チェックボックスを選択すると、8 つのタグ（ウィンドウの 4 つのタグ + グラフィックの 4 つのタグ）が表示されます。

表示されるグラフィックには、以下の条件が適用されます。

- InTouch ウィンドウで使用されていないシンボル内のリモート タグ リファレンスは表示されません。
- InTouch ウィンドウで使用されていないオブジェクト シンボルは表示されません。
- このチェックボックスがクロス リファレンス ユーティリティで選択されている場合、未使用タグ変数の検索や未使用タグの削除などのその他の操作にも同じ条件が設定されます。

ArchestrA または Situational Awareness Library シンボルでの InTouch タグのクロス リファレンス タグの使用

InTouch クロス リファレンス ユーティリティが強化され、産業用グラフィックがアプリケーションの標準の InTouch タグ クロス リファレンスのリストに含まれるようになりました。以下に示す条件を使用して産業用グラフィックのクロス リファレンス検索を実行できます。これらの条件は、クロス リファレンス ユーティリティの InTouch タグの使用の一部として含まれます。

- InTouch タグへの参照を含むカスタム プロパティ
- InTouch タグへの直接参照を含むアニメーション
- InTouch タグへの直接参照を含むクライアント スクリプト
- InTouch タグへの埋め込まれたシンボルの参照

クロス リファレンス ユーティリティでは、オブジェクト属性の一部である属性参照を検索する際に産業用グラフィックも検索されます。タグ使用数レポートの例外を以下に示します。

1. オブジェクト属性参照:
   - 相対参照（me.I1 など）を使用するオブジェクト属性はサポートされず、クロス リファレンス ユーティリティ タグ使用数レポートに表示されません。
   - <InTouchViewApp instance name>.<Tagname> の構文を使用する参照は InTouch タグ リファレンスと見なされません。このリファレンスはオブジェクト属性と見なされます。
     例: InTouchViewApp1.Tag1

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 「InTouch:<Tag>」の構文ではないリファレンスはオブジェクト属性参照と見なされます。

2. スクリプト関数パラメータとして使用されるクライアント スクリプト リファレンスは InTouch タグおよびオブジェクト属性参照とは見なされません。

   例:文字列パラメーターとして参照
   SetCustomPropertyValue("CP1", "InTouch:Tag1", false);

   例: Attribute parameter
   SetBad(InTouch:Tag1);

3. 現在の InTouch アプリケーションで定義されていない InTouch タグを参照する産業用グラフィックは「未定義」タイプとして表示されます。オブジェクト シンボル ArchestrA オブジェクト "UD_001" にはインスタンス シンボル "S1"、"S2"、および "S3" があります。

   インスタンス シンボル "UD_001.S1" には InTouch タグ "Tag51" への参照があります。UD_001.S1 を UD_001.S2 に埋め込みます。
   UD_001.S2 を UD_001.S3 に埋め込みます。
   UD_001.S3 を InTouch ウィンドウ "Win1" に埋め込みます。

   また、UD_001.S2 をウィンドウ "Win1" に埋め込みます。

インスタンス シンボルは、インスタンス内またはその派生テンプレート内で定義できます。インスタンスは、絶対または相対属性を使用して埋め込むことができます。

クロス リファレンス ユーティリティでのタグ使用数の表示

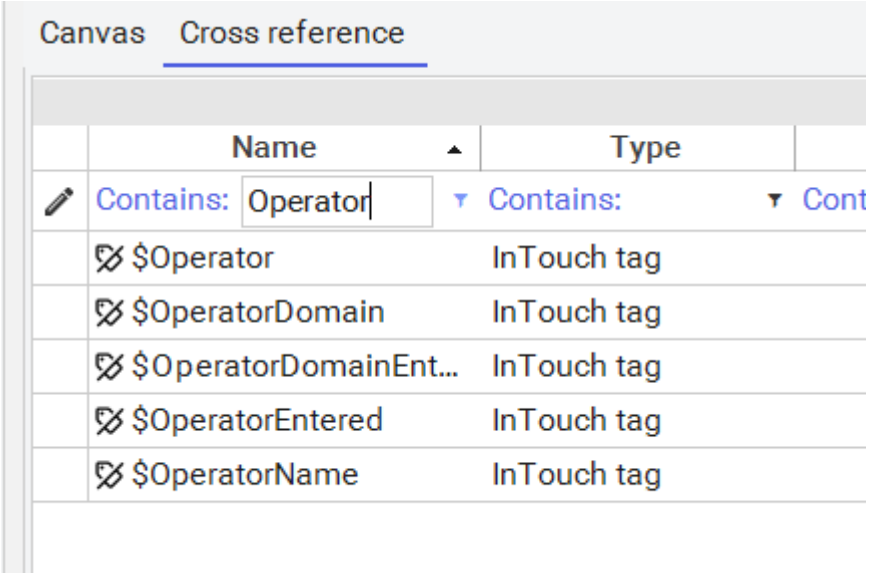
フィルタ、ソート、およびグループ化オプションを使用して、クロス リファレンス タグ使用数レポートの表示を変更できます。フィルタを適用すると、表示されるレコードの数が更新されます。ソートおよびグループ化では、グリッドでのレコードの表示方法だけが変更されます。

フィルタ オプションを使用したタグ使用数レポートの絞り込み

1. フィルタアイコン（ ）を使って、「含む」、「含まない」、「～で始まる」、「～で終わる」、「等しい」、「等しくない」、「NULL である」、「NULL でない」、「カスタム」などのオプションリストから選択します。
2. 選択したフィルタ オプションに対する検索条件を入力します。

   フィルタ オプションおよび検索条件に一致するすべてのレコードがグリッドに表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



カスタム フィルタ オプションの使用

カスタム フィルタ オプションを使用すると、複数のタグを追加して複雑な検索式を作成できます。

タグ使用数レポートのソート

タグ使用数レポートの各列は、昇順または降順でソートできます。

1. 列見出しを選択します。

   グリッド内のすべてのレコードが、その列を基準にソートされます。矢印の向きはソート順を示します。

2. 列見出しを再選択すると、ソート順が変更されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

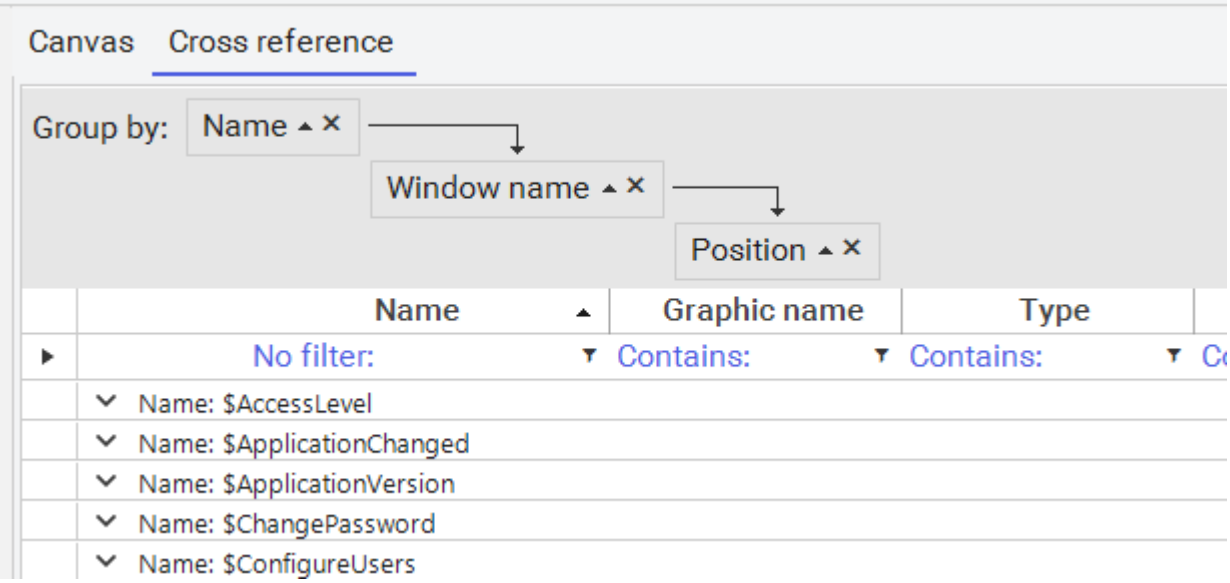
タグ使用数レポートのグループ化

列の組み合わせを使用してタグ使用数レポートをパターンで整理します。

1. 列名を選択して、列名の上の空白スペースにドロップします。

   列の名前を含むボックスが表示されます。
2. 別の列名を選択して、最初のボックスの上または下にドロップします。

   2 つの列の間に階層関係が自動的に作成されます。
3. 同じレベルに 2 つの列を作成するには、2 番目の列名を最初のボックスにドロップします。
4. 個々の列を昇順または降順にソート、パターン内の列名ボックスをクリックします。



タグのクロス リファレンス リストの保存と印刷

クロス リファレンス タグのリストをファイルに保存し、.csv ファイル形式をサポートしているアプリケーションで表示できます。

クロス リファレンス タグのリスト ファイルには、名前ごとのすべてのタグ、アプリケーション内でのタグの使用方法、タグの使用位置のウィンドウ名が一覧表示されます。クロス リファレンス タグのリスト ファイルは、Excel または .csv 形式ファイルをサポートしているその他のプログラムで開くことができます。

また、タグ名ディクショナリの内容を印刷することも可能です。タグ名ディクショナリの内容を印刷すると、アプリケーションで使用されているデータベース エントリを参照できます。表示される詳細のレベルを指定できます。

このレポートは、プリンタに送信したり、ファイルに保存したりできます。

クロス リファレンス ファイルを保存するには

1. ［クロス リファレンス ユーティリティ］ダイアログ ボックスの［ファイルを保存］をクリックします。［名前を付けて保存］ダイアログ ボックスが開きます。
   ファイルの名前と保存先を指定します。
2. 保存を選択します。タグの使用数を含むファイルが指定のフォルダに保存されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

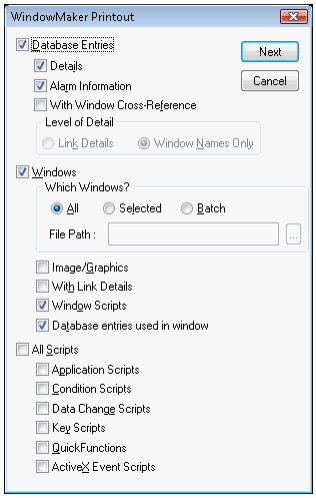
.csv ファイルに保存されるタグ使用数レポートは、以下の条件に基づきます。

- 検索条件を選択しない場合、すべてのレコードが .csv ファイルに保存されます。例えば、クロス リファレンス グリッド内のレコードの総数が 1000 で、検索を絞り込むとレコード数は 320 に削減されます。［名前を付けて保存］を使用すると、選択された 320 のレコードだけが .csv ファイルに保存されます。
- グループ化パターンをレコードに適用した場合、.csv は、そのグループ化パターンで保存されます。

タグ名ディクショナリの内容を印刷するには

1. WindowMaker で InTouch アプリケーションを開きます。
2. クイック アクセス ツールバーの［**印刷**］をクリックします。

   ［**アプリケーション情報の印刷**］ダイアログ ボックスが表示されます。



3. タグ名ディクショナリからタグの情報を印刷する場合は、［**データベース エントリ**］を選択します。

   ［**データベース エントリ**］を選択すると、以下のオプションが選択できるようになります。

   - ［**詳細**］を選択すると、タグの詳細が印刷されます。
   - ［**アラーム設定**］を選択すると、タグのアラーム設定情報が印刷されます。
   - ［**ウィンドウとの関連**］を選択すると、ウィンドウのクロス リファレンスを持つタグ名ディクショナリの内容がすべて印刷されます。このオプションを選択した場合は、印刷する詳細レベルを指定します。

     ［**リンク詳細**］を選択すると、タグが使用されている場所とアニメーション リンクの詳細が印刷されます。

     ［**ウィンドウ名のみ**］を選択すると、クロス リファレンス先のウィンドウの名前だけが印刷されます。

4. **[次へ]** を選択します。［**出力の送信先の選択**］ダイアログ ボックスが表示されます。
5. タグ名ディクショナリの内容を印刷するか、または出力結果をテキスト ファイルや .html ファイルに送信するかのオプションを選択します。
6. **[印刷]** を選択します。

クロス リファレンス タグのリストをファイルに保存し、.csv ファイル形式をサポートしているアプリケーションで表示できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

未使用タグの削除

使用回数を更新した後で、InTouch アプリケーションから未使用のタグを削除できます。タグ名ディクショナリから一度に 1 個のタグを削除するか、または一度に複数のタグを削除することが可能です。

タグを含むウィンドウがアプリケーションから削除されたり、またはリンク スクリプトでタグが変更された場合でも、タグの数は自動更新されません。InTouch はタグが使用中であると見なすため、削除されないようにします。

注意：産業用グラフィック参照で［**削除**］ボタンが無効化されている場合、［**使用カウントを更新**］を実行して［削除］ボタンを有効にします。

未使用のタグを削除する前に、タグの数を更新して、合計数から未使用のタグを取り除く必要があります。タグの数の更新については、「タグの数の確認」を参照してください。

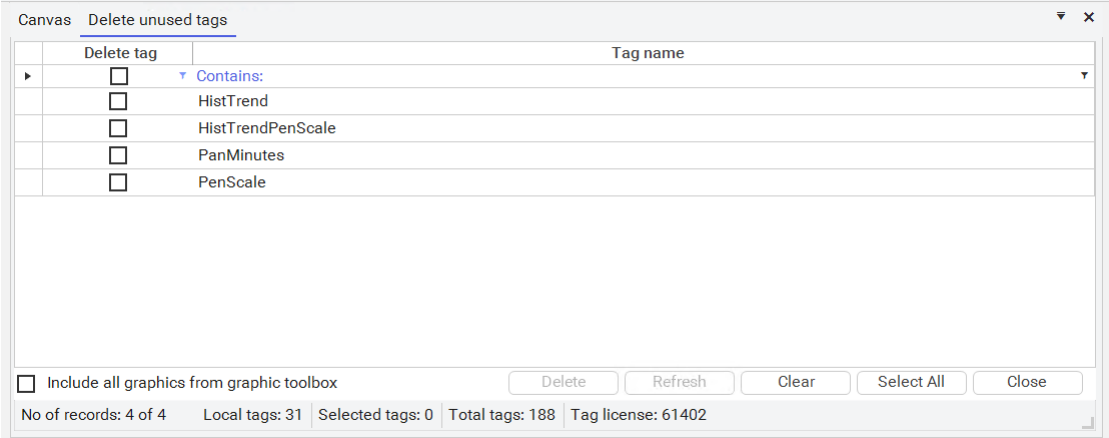
注意: アラーム状態になっているだけのタグは使用回数に含まれないため、誤って削除される可能性があります。このようなタグを使用回数に含めるには、ウィンドウまたは QuickScript でこれらのタグを使用する必要があります。

複数の未使用のタグを削除

1. WindowViewer が起動されている場合は閉じます。
2. WindowMaker で InTouch アプリケーションを開きます。
3. ［**ホーム**］メニューの［**タグ**］グループで［**未使用タグの削除**］をクリックします。

   未使用のタグが一覧表示された［**未使用タグの削除**］ダイアログ ボックスが表示されます。

   ダイアログ ボックスの下部にあるステータス バーには、レコードの数、ローカル タグの数、選択されたタグの数、タグの合計数、およびタグ ライセンス カウントが一覧表示されます。



4. フィルタ（ ）オプションを使用して、未使用タグのリストを絞り込みます。フィルタ オプションを選択して、検索条件を入力します。

   条件に一致するタグが表示されます。
5. 列をソート、列見出しをクリックします。
6. 削除するタグを選択します。現在のフィルタ条件に一致するすべてのタグを選択、［**すべて選択**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ステータス バーには、選択されたタグの数（タグの選択は、列の条件が変更された場合でも維持されます。「選択したタグ」の値には、ユーザーが選択したタグの合計数が反映されます。フィルタ条件には依存しません。選択されたタグの完全なリストを表示するには、すべてのフィルタ条件をクリアします。

7. ［**クリア**］を選択して、現在のフィルタのタグ選択をクリアします。
8. ［**グラフィック ツールボックスのすべてのグラフィックを含める**］チェック ボックスを選択すると、いずれのグラフィックまたはウィンドウでも使用されていないタグだけが表示されます。これらのタグはグラフィックやグラフィックを含むウィンドウで参照されていないので、安全に削除できます。

   タグ ディクショナリに 10 個のタグが含まれていて、4 つのタグがグラフィック ツールボックスのグラフィックで参照されているとします（ウィンドウには埋め込まれていません）。その他の 4 つのタグはウィンドウに配置されたグラフィックで参照されているとします。デフォルトでは、6 つのタグ（2 つの未使用タグ + ウィンドウに埋め込まれていないグラフィック ツールボックスのグラフィックの 4 つのタグ）が表示されます。チェック ボックスをオンにすると、未使用の 2 つのタグだけが表示されます。
9. ［**削除**］を選択して、すべてのタグ選択を削除します。フィルタ条件には依存しません。

   確認メッセージが表示されます。メッセージには、削除されるタグの数が示されます。
10. [**OK] を選択** して確認します。

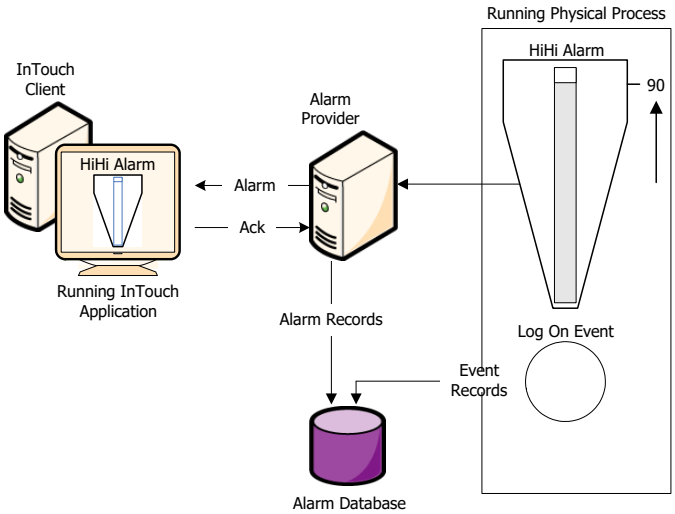
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 24 アラーム

アラームとイベントを生成してオペレータにプロセスの動作状況を通知する InTouch アプリケーションを作成できます。

- 問題を引き起こす可能性のあるプロセス状態に関する警告が、アラームによってランタイム オペレータに警告されます。通常は、プロセス値が定義済みのしきい値を超えた場合にトリガするようにアラームを設定します。オペレータは、アラームを確認する必要があります。
- イベントは、通常のシステム ステータス メッセージを表します。オペレータが InTouch アプリケーションにログオンするなど、システム条件が発生するときが典型的なイベントとなります。オペレータは、イベントを確認する必要はありません。

以下の図は、アプリケーションが実行されている間、InTouch HMI でアラームとイベントが処理される方法を示しています。アラームとイベントのデータは、アラーム データベースに保存されます。

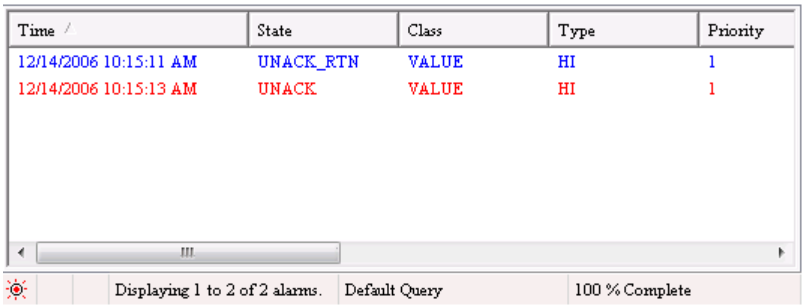


イベント監視用のタグを設定できます。イベント メッセージは、タグの値が変更するたびにアラーム システムにログ記録されます。イベント メッセージには、値がどのように変更されたかに加えて、オペレータ、I/O、スクリプト、またはシステムのいずれによってその変更が開始されたのかという情報が含まれます。

現在のアラームの表示

InTouch Alarm Viewer ActiveX コントロールを使用して、アラームを表示します。Alarm Viewer コントロールには、スクロール バー、サイズ変更可能なカラム、複数アラーム選択、更新ステータス バー、動的表示タイプ、アラームのタイプに基づいたカラー表示などの機能があります。

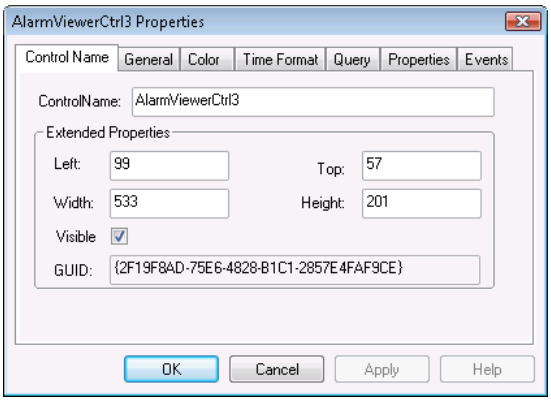
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



InTouch アラームを表示するには、Alarm Viewer コントロールを使用することが推奨されます。ただし、7.1 よりも古いバージョンの InTouch で作成されたアプリケーションのアラームを表示するには、分散アラーム オブジェクトの使用を続行できます。

Alarm Viewer コントロールの設定

WindowMaker から Alarm Viewer コントロール オプションを設定できます。また、Alarm Viewer コントロール の実行中にユーザーが変更できるオプションも設定できます。これらのオプションは、［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスで設定します。



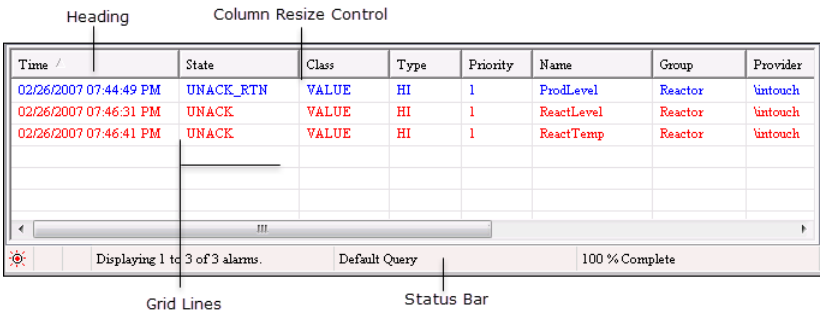
グリッドの外観を設定する

Alarm Viewer コントロールの表示外観を設定する場合、以下の操作を行うことができます。

- ステータス バーを含めます。
- カラム ヘッダーを含めます。
- 行とカラムを表示する水平と垂直のグリッド線を含めます。
- ユーザーのカラム幅を調整することを許可するランタイム オプションを含めます。
- 表示要素の色を設定します。

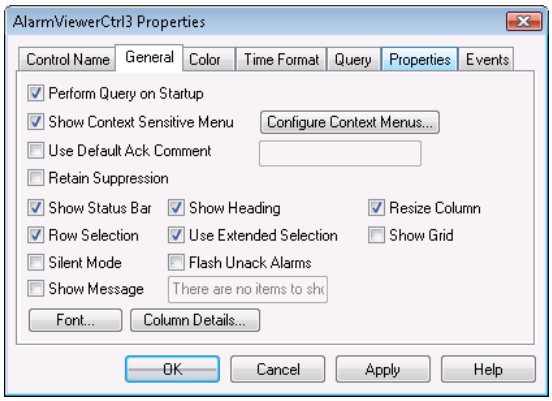
以下の図は、すべての表示プロパティがアクティブである場合に Alarm Viewer コントロールが表示される様子を示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



表示外観を設定する

1. Alarm Tree Viewer コントロールを右クリックして、［**プロパティ**］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. ［**全般**］タブを選択します。



3. 表示外観を設定します。以下のいずれかを設定します。

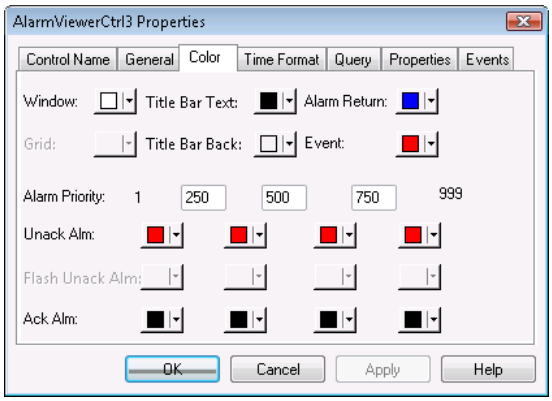
オプション	説明
起動時にクエリー実行	デフォルトのクエリー プロパティを使用して、コントロールの更新を自動的に開始します。アプリケーションの起動時にクエリーが実行されない場合は、Requery() 関数を使用したスクリプトを実行して、グリッドを更新する必要があります。ランタイム中には、グリッドのショートカット メニューでも［再クエリー］オプションを利用できます。
コンテキスト メニューを表示	ランタイム中に右クリックのショートカット メニューが有効になります。
デフォルトの確認コメントを使用	オペレータがアラームを確認する際にデフォルトのコメントを表示するかどうかを制御します。このオプションをオンにして文字列を入力すると、その文字列がランタイム中にデフォルトのコメントとして使用されます。 このチェック ボックスをオフにすると、オペレータがアラームを確認する場合に、オプションのコメントを入力するためのダイアログ ボックスが表示されます。このダイアログ ボックスはテキストを入力することも、空白のまま残すこともできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オプション	説明
非表示を保持	アラーム クエリーが変更された場合に、アラーム クエリー間のアラーム非表示を保持します。
ステータス バーの表示	Alarm Viewer コントロールの一番下にステータス バーを表示するかどうかを切り替えます。
行の選択	ランタイム中にユーザーが個々の行を選択できるようにします。各行はアラーム レコードを表しています。ユーザーは複数アラームを選択できます。
サイレント モード	［サイレント モード］チェック ボックスをオンにすると、Alarm Viewer コントロールではランタイム中にエラー メッセージは表示されません。このチェック ボックスをオフにすると、アラーム表示にエラー メッセージが表示されます。 いずれの場合も、エラー メッセージは、常に Log Viewer に送信されます。
メッセージの表示	テスト ボックスに入力したメッセージが表示されます。このメッセージは、アラームがない場合に表示されるメッセージです。
見出しの表示	Alarm Viewer コントロールの一番上に見出しを表示するかどうかを切り替えます。
拡張アラーム選択の使用	**CTRL** キーまたは **SHIFT** キーを押しながらマウス ボタンと共に使用することにより、ユーザーは複数のアラームを同時に選択できるようになります。［行の選択］チェック ボックスがオンの場合のみ選択できます。
点滅未確認アラーム	未確認アラームが確認されるまで、そのアラームが 1 秒間隔で点滅します。 WindowViewer でアラーム表示を停止しても、未確認アラームの点滅は停止しません。
カラムのサイズ変更	［カラムのサイズ変更］をオンにすると、ユーザーはランタイム中にカラム幅を変更できるようになります。それ以外の場合は、カラム幅は静的となり、WindowMaker でのみ設定できます。
グリッドの表示	［グリッドの表示］をオンにすると、アラーム表示の各行と各カラムを区切る縦線と横線が Alarm Viewer コントロールに表示されます。このチェック ボックスをオフにすると、グリッドは表示されません。

4. **[適用]** を選択します。
5. **色**タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



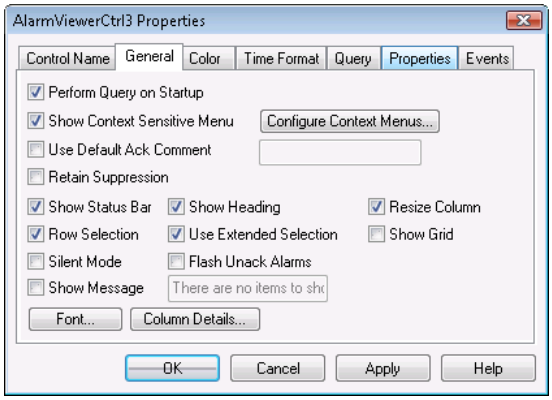
6. パレットのボタンを選択して、Alarm Viewer コントロールの表示要素に色を割り当てます。
7. ［適用］を選択して、色の選択を保存します。
8. ［**OK**］ を選択します。

フォント表示の設定

Alarm Viewer コントロールでテキストが表示される方法を設定できます。

フォントのプロパティを設定する

1. Alarm Tree Viewer コントロールを右クリックして、［**プロパティ**］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. [**全般**] タブを選択します。



3. フォントの選択 Windows 標準の［**フォント**］ダイアログ ボックスが表示されます。フォントを設定して、［**OK**］を選択します。
4. ［**OK**］ を選択します。

表示カラムの詳細の設定

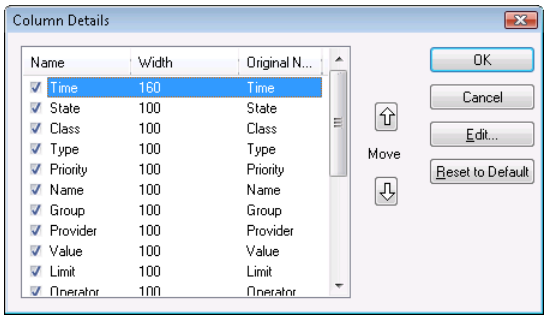
Alarm Viewer コントロールの場合、以下の操作を行えます。

- カラムを選択して、順序を設定します。
- カラム幅をピクセル単位で設定します。
- カラムの名前を変更します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

表示カラムの詳細を設定するには

1. Alarm Tree Viewer コントロールを右クリックして、［プロパティ］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. ［全般］タブを選択します。
3. **列の詳細**を選択します。［カラムの詳細］ダイアログ ボックスが表示されます。



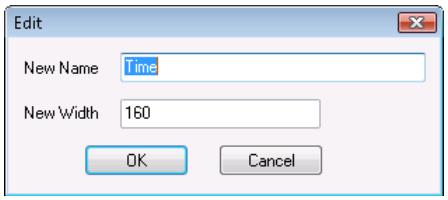
4. ［**アラーム名**］カラムで、表示するカラム名の横にあるチェック ボックスをオンにします。リストから少なくとも 1 つ以上のカラムを選択する必要があります。

カラム	内容
Time	［時間の形式］タブで指定された時間です。
状態	アラームの状態です。
クラス	アラーム カテゴリです。
Type	アラーム タイプです。
優先度	アラームの優先度です。
名前	タグ名です。
グループ	アラーム グループの名前です。
プロバイダ	アラーム プロバイダの名前です。
値	アラーム発生時のタグの値です。カラムの幅は、希望する正確なレベルを提供するために十分な大きさである必要があります。
しきい値	タグのアラームしきい値です。カラムの幅は、希望する正確なレベルを提供するために十分な大きさである必要があります。
Operator	アラーム状況に関連付けられているログオンしたオペレータの ID です。
オペレータ フルネーム	ログオンしたオペレータのフルネームです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オペレータ ノード	アラーム状況に関連付けられているログオンしたオペレータのノードです。 ターミナル サービス環境で、オペレータ ノードはオペレータがターミナル サービス セッションを確立したクライアント コンピュータの名前です。ノード名を取得できない場合、ノードの IP アドレスが代わりに使用されます。
オペレータ ドメイン	アラーム状況に関連付けられているログオンしたオペレータのドメインです。
タグ コメント	タグのコメントです。
アラーム コメント	タグのアラームに関連付けられているコメントです。これらのコメントは、タグのアラームが定義された際に［アラーム コメント］ボックスに入力されたものです。アラームに対して確認コメントが指定されると、新しいコメントはこのコメント カラムで更新されます。
ユーザー 1	アラームの AlarmUserDefNum1 プロパティの数値です。
ユーザー 2	アラームの AlarmUserDefNum2 プロパティの数値です。
User3	アラームの AlarmUserDefStr プロパティの文字列値です。

5. カラム名を選択し、上下の矢印を使用して、カラムを並べ替えます。［**カラムの詳細**］ダイアログ ボックスの一番上に表示されるカラム名は、アラーム コントロールの一番左のカラムの名前です。
6. カラムの名前または幅を変更する、カラムを選択して、［**編集**］を選択します。［編集］ダイアログ ボックスが表示されます。

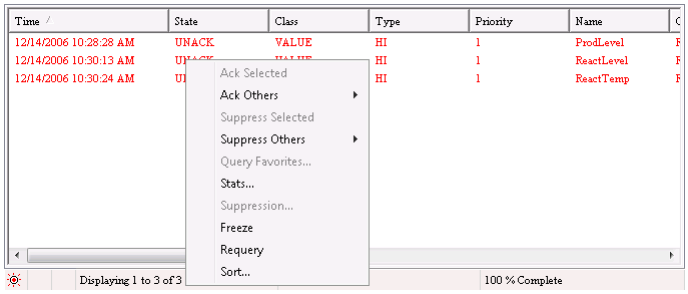


   a. ［**新しい名前**］ボックスに、新しいカラム名を入力します。
   b. ［**新しい幅**］ボックスに、カラム幅を入力します。カラム幅は、1 ～ 999 ピクセルの範囲で設定できます。
   c. ［**OK**］ を選択します。
7. ［カラムの詳細］ダイアログ ボックスの［**OK**］を選択します。
8. **[適用]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイム時における機能へのアクセスの制御

ランタイム中に Alarm Viewer コントロールを右クリックすると、コンテキスト（ショートカット）メニューが表示されます。



このショートカット メニューに、どのメニュー コマンドを表示するかを管理できます。

ショートカット メニューを設定

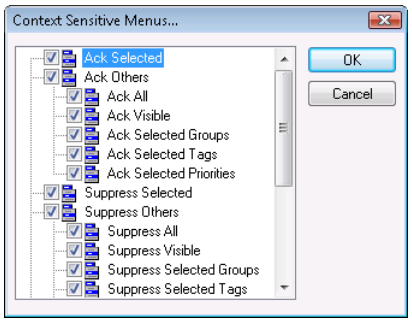
1. **Alarm Tree Viewer** コントロールを右クリックして、［**プロパティ**］を選択します。

   ［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。

2. ［**全般**］タブで［**コンテキスト メニューを表示**］チェック ボックスをオンにします。

3. ［**コンテキスト メニューの設定**］を選択します。

   ［**コンテキスト メニュー**］ダイアログ ボックスが表示されます。



   このダイアログ ボックスには、Alarm Viewer コントロールのショートカット メニューに表示できるコマンドのリストが階層表示されます。

4. ショートカット メニューのオプションを設定します。ショートカット メニューのコマンドは少なくとも 1 つ選択する必要があります。

コマンド名	ランタイム ユーザーに許可される内容
選択アラームを確認	選択したアラームを確認できます。［**選択アラームを確認**］および［**その他を確認**］のどちらのメニュー項目も選択しないと、［**デフォルトの確認コメントを使用**］チェック ボックスとテキスト ボックスは無効になります。
その他を確認	その他の方法によってアラームを確認します。どのアラームを確認するかは、ユーザーが選択できます。［**その他を確認**］ が選択されている場合、サブメニュー項目を少なくとも 1 つ選択する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

全てを確認	すべてのアクティブなアラームを確認します。
表示部分を確認	表示されているアラームを確認します。
選択グループを確認	選択したグループと同じグループ名および同じプロバイダ名を持つアラームをすべて確認します。
選択タグを確認	選択したタグとグループ名が同じで、同じプロバイダ、グループ、および優先度を持つアラームをすべて確認します。
選択優先度を確認	選択した優先度と同じ優先度を持ち、同じプロバイダおよびグループを持つアラームをすべて確認します。
選択項目を非表示	選択したアラームを非表示にします。
その他を非表示	ショートカット メニューに表示されるその他の方法で、アラームを非表示にします。
全て非表示	すべてのアラームを非表示にします。
表示部分を非表示	表示されているすべてのアラームを非表示にします。
選択グループを非表示	選択したグループと同じグループ名を持つすべてのアラームを非表示にします。
選択タグを非表示	選択したタグと同じタグ名を持つすべてのアラームを非表示にします。
選択優先度を非表示	選択した優先度と同じ優先度を持つすべてのアラームを非表示にします。
全て再表示	非表示のアラームをすべて再表示にします。
クエリー設定	［アラーム クエリー］ダイアログ ボックスを開きます。
統計	［アラームの統計］ダイアログ ボックスを開きます。
非表示	［非表示中のアラーム］ダイアログ ボックスを開きます。
停止	Alarm Viewer コントロールの停止/停止解除モードを切り替えます。
再クエリー	アラーム クエリーを再実行します。
ソート	［ソートのカスタマイズ］ダイアログ ボックスを開きます。

5. ［**OK**］ を選択します。
6. ［**行の選択**］をクリックすると、ユーザーはランタイム中に Alarm Viewer コントロールから行を選択できます。
7. ［**拡張アラーム選択の使用**］を選択すると、ユーザーは **Shift** キーまたは **Ctrl** キーを使用して、Alarm Viewer コントロールから複数のアラーム レコードを同時に選択できます。
8. **[適用]** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

表示するアラームの選択

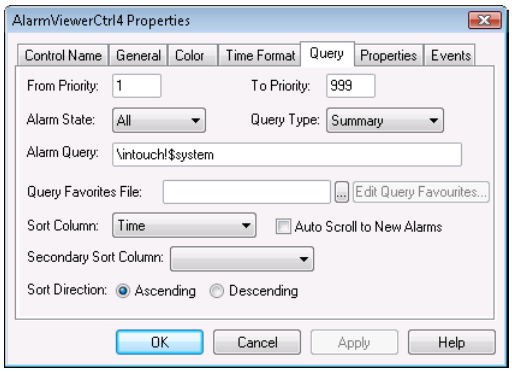
Alarm Viewer コントロールには、アクティブなアラームのサマリや履歴アラームのリストを表示できます。

一般のアラーム クエリー プロパティを設定

1. Alarm Tree Viewer コントロールを右クリックして、［プロパティ］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. [**全般**] タブを選択します。
3. ［**起動時にクエリー実行**］チェック ボックスをオンにすると、アプリケーションの起動時にデフォルトのクエリー プロパティを使用して Alarm Viewer コントロールが自動的に更新されます。
4. ［**メッセージの表示**］チェック ボックスをオンにすると、アラームがない場合にデフォルトのメッセージが表示されます。テキスト ボックスには、表示するメッセージを入力します。
5. **[適用]** を選択します。

クエリーのデフォルトを設定する

1. Alarm Tree Viewer コントロールを右クリックして、［**プロパティ**］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. **[クエリ]**タブを選択します。



3. ［**優先度始点**］ボックスに、最低アラーム優先度の値（1 ～ 999）を入力します。
4. ［**優先度終点**］ボックスに、最高アラーム優先度の値（1 ～ 999）を入力します。
5. ［**クエリータイプ**］の矢印をクリックして、デフォルトのランタイム アラーム表示として［**履歴**］または［**サマリ**］のいずれかを選択します。

   表示のデフォルト タイプは、クエリー関数を含む QuickScript を実行することによって、ランタイム中に変更できます。たとえば、スクリプトに Type パラメータが “Summary” に設定されている ApplyQuery() メソッドが含まれている場合、グリッドには現在のアラームのサマリが表示されます。反対に、同じグリッドに対して、Type パラメータが "Historical" に設定された ApplyQuery() メソッドが実行されている場合、履歴アラームが表示されます。QueryType プロパティは、アラーム表示の現在の状態を反映しています。
6. ［**アラーム クエリー**］ボックスに、有効なアラーム クエリーを入力します。たとえば、\InTouch!$System と入力すると、デフォルトの $System アラーム グループに属するすべてのアラームに対してクエリーされます。
7. ［**OK**］ を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

クエリー設定を使用したカスタム保存クエリーを作成

オペレータがショートカット メニューから選択できるクエリー設定のリストを設定できます。

Windows Vista または新しい OS の標準ユーザーがアクセスできるフォルダにクエリー設定ファイルを配置してください。WindowViewer が常に同じユーザー アカウントで実行される場合、以下を使用できます。

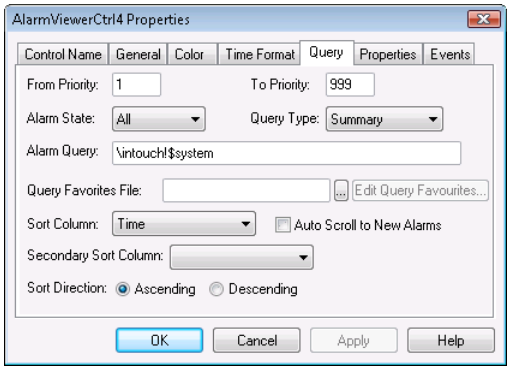
C:\Users\<ユーザー名>\AppData\Local\

異なるユーザーが WindowViewer を実行する場合、以下を使用できます。

C:\Users\Public\Documents\

クエリー設定ファイルを設定します。

1. Alarm Tree Viewer コントロールを右クリックして、［**プロパティ**］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。



2. **[クエリ]**タブを選択します。
3. クエリー設定ファイルを設定します。
   a. ［**クエリー設定ファイル**］ボックスに、ネットワーク パスとファイル名を入力するか、参照ボタンを選択してファイルを参照します。
   b. **フィルタ設定**ファイルを編集する、［設定ファイルの編集］ボタンを選択します。［**アラーム クエリー**］ウィンドウが開き、設定ファイルのフィルタを追加、変更、または削除できます。完了したら、［**OK**］を選択して変更を保存し、ウィンドウを閉じます。
4. ［**OK**］ を選択します。

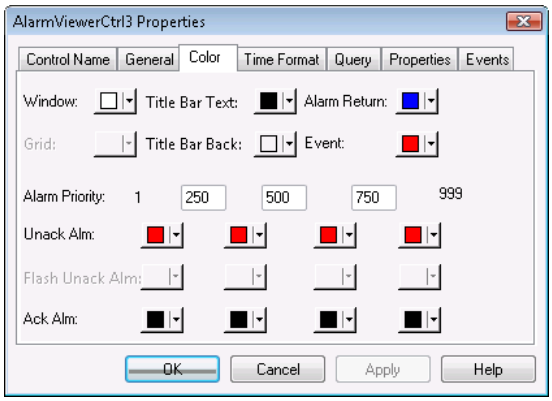
さまざまなタイプのアラーム レコードの色の使用

Alarm Viewer コントロールに表示されるさまざまなアラーム状態に対して、色オプションを設定できます。

アラーム表示の色を設定する

1. Alarm Tree Viewer コントロールを右クリックして、［プロパティ］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. **[マルチステート色]**タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. それぞれの色ボックスを選択すると、カラー パレットが開きます。以下の各パレットから使用したい色を選択します：

プロパティ	説明
ウィンドウ	背景色を設定します。
タイトル バーのテキスト	タイトル バーのテキストの色を設定します（[見出しの表示] オプションがオンである場合のみ）。
アラーム復帰	平常状態に復帰したアラーム（確認されずに平常に戻ったアラーム）の色を設定します。
グリッド	グリッドの色を設定します。デフォルトでは、グリッドは表示されません。デフォルトのグリッド色は薄いグレーです。アラーム オブジェクトのグリッドの色は、選択したウィンドウの色に対してコントラストのはっきりした色に自動設定されます。
タイトルバーの背景	タイトル バーの背景色を設定します（[見出しの表示] オプションがオンである場合のみ使用可能）。
イベント	イベントの色を設定します。

4. [アラーム優先度] ボックスに、未確認アラーム、確認アラーム、および点滅未確認アラームを識別するために使用される異なる色に対する境界として使用されるアラーム優先度の数値を入力します。
5. [未確認アラーム] と [確認アラーム] の色ボックスを選択して、カラー パレットを開きます。使用するパレットの色を選択します。
6. 点滅未確認アラームに対してアラーム クエリーを設定、[一般] タブを選択して、[点滅未確認アラーム] チェック ボックスをオンにしてから、[色] タブを選択し、[点滅未確認アラーム] の色ボックスを選択します。各アラーム優先度範囲に使用する色を選択します。

注意：Alarm Viewer コントロールには、1 秒未満の間に発生する変化は表示できません。1 秒間に 2 回アラーム状態が変化する場合、Alarm Viewer コントロールではその変化を認識できません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

7. **[適用]** を選択します。

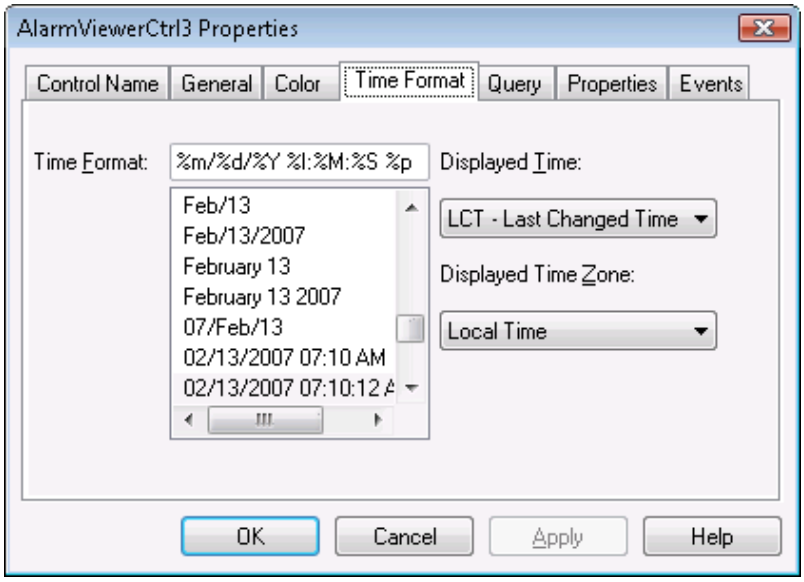
アラーム レコードの表示時間形式の設定

表示されているアラーム レコードの時間形式を設定できます。

Alarm Viewer コントロールの場合、元のアラーム時刻は、アラームが開始した日付/時刻スタンプです。タグが I/O 型タグであり、そのサーバーがタイムスタンプを渡すことができる場合は、I/O サーバーからのタイムスタンプです。

時間の形式を設定

1. **Alarm Tree Viewer** コントロールを右クリックして、［**プロパティ**］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. ［**書式設定**］タブを選択します。



3. ［**時間の形式**］リストで、使用する時間の形式を選択します。［**時間の形式**］ボックスには、選択した形式に対して、% 記号で区切られた文字で構成される一組の文字列が表示されます。

文字列	説明
d	2 桁の日付
b	月を示す 3 文字の略語
Y	年を表す 4 桁の数
m	月を表す 2 桁の数
y	年を表す 2 桁の数
#x	日付と曜日。例えば、2007 年 8 月 10 日 金曜日
B	月の名前
H	24 時間形式の時間
M	分

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列	説明
p	午前または午後（12 時間形式で使用）
S	秒
s	秒の端数
I	12 時間形式の時間

4. ［表示される時刻］リストで表示時間を選択します。

OAT	元のアラーム時刻です。アラームが開始した日付/時刻スタンプです。
LCT	最終変更時刻です。アラームのインスタンスに対して発生した最新の状態変化（アラームの開始、サブステートの変化、平常への復帰、または確認）の日付/時刻スタンプです。
LCT（確認時には OAT）	最終変更時刻ですが、確認時の元のアラーム時刻です。アラームが未確認である間は最終変更時刻、アラームが確認された後は元のアラーム時刻が使用されます。

5. ［表示されるタイム ゾーン］リストで、タイム ゾーンを選択します。

[GMT]	協定世界時（UTC または Zulu）とも呼ばれるグリニッジ標準時です。
ローカル時間	ローカルタイム ゾーンに対して調整されるアラーム時間です。
標準時間	アラーム ソースのタイム ゾーンに対して調整されるアラーム時間です。

6. **[適用]** を選択します。

アラーム レコードのソート順を設定

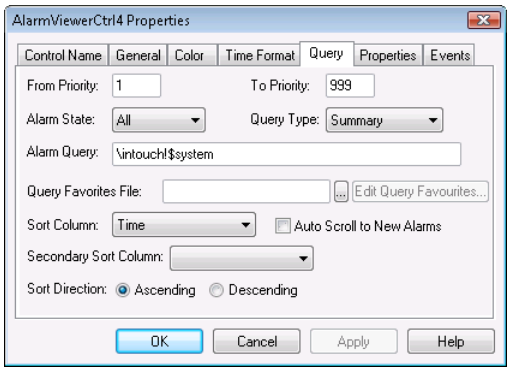
リストのアラーム レコードをソートできます。デフォルトで、Alarm Viewer コントロールにはアラーム レコードが時刻で昇順に一覧表示されます。

プライマリ カラムやオプションのセカンダリ ソート カラムに基づいて、アラーム レコードを昇順または降順にソートできます。

アラーム レコードのソート順を設定

1. Alarm Tree Viewer コントロールを右クリックして、［**プロパティ**］を選択します。［**AlarmViewerCtrl のプロパティ**］ダイアログ ボックスが表示されます。
2. **[XML]**タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 以下の手順に従って、ソート オプションを選択します。
   a. ［**カラムのソート**］リストから、プライマリ ソート カラムを選択します。［**カラムのソート**］リストに表示されるのは、表示されるカラムのみです。目的のカラムが見つからない場合は、［全般］タブに移動し、［カラムの詳細］からカラムを選択します。
   b. ［**セカンダリ ソート カラム**］リストから、セカンダリ ソート カラムを選択します。
   c. プライマリ ソート カラムとして時間を選択すると、［**新規アラームに自動スクロール**］チェックボックスが使用できるようになります。新しいアラームが発生した場合に、自動的にスクロールして、それらのアラームを表示する場合は、このオプションをオンにします。
   d. ［**昇順**］または［**降順**］を選択して、ソート順を設定します。
4. **[適用]** を選択します。

ランタイムでの Alarm Viewer コントロールの使用

Alarm Viewer コントロールには、1 つまたは複数の選択したアラーム、アラーム グループ、タグ変数、および優先度の表示オブジェクトに適用できるコマンドへの簡単アクセスをオペレータに提供するショートカット メニューが含まれています。

以下のリストは、Alarm Viewer コントロールのショートカット メニューから使用できるコマンドを一覧表示しています。

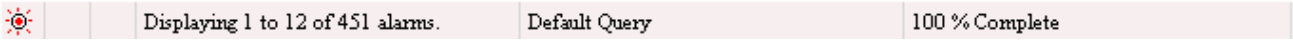
- **選択アラームを確認** - 選択したアラームを確認します。
- **その他を確認** - その他の確認コマンドを一覧するサブメニューが表示されます。
  - 全てを確認 - 現在のアラーム クエリーのすべてのアラームを確認します。アラーム グリッドの表示領域は制限されているため、［全てを確認］コマンドではグリッドで表示されていないアラームも確認されます。
  - 表示部分を確認 - アラーム グリッドに現在表示されているアラームのみを確認します。
  - 選択グループを確認 - 1 つまたは複数の選択したアラームと同じプロバイダからの同じグループ名を持つすべてのアラームを確認します。
  - 選択タグ変数を確認 - 1 つまたは複数の選択したアラームと同じ優先度を持つ、同じプロバイダおよびグループ名からの同じタグ変数を持つすべてのアラームを確認します。
  - 選択優先度を確認 - 1 つまたは複数の選択したアラームと同じプロバイダおよびグループ名からの同じ優先度を持つすべてのアラームを確認します。
- **選択アラームを非表示** - 選択したアラームを非表示にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **その他を非表示** - 非表示コマンドが含まれるサブメニューが表示されます。
  - 全て非表示 - すべてのアラームの現在および今後の発生を非表示にします。
  - 表示部分を非表示 - 表示されているアラームの現在および今後の発生を非表示にします。
  - 選択グループを非表示 - 同じプロバイダ名を持つ 1 つまたは複数の選択したアラームの同じグループに属す任意のアラームの現在および今後の発生を非表示にします。
  - 選択タグ変数を非表示 - 同じプロバイダ名、グループ名、および優先度範囲を持つ 1 つまたは複数の選択したアラームの同じタグ変数に属す任意のアラームの現在および今後の発生を非表示にします。
  - 選択優先度を非表示 - 同じプロバイダ名およびグループ名を持つ 1 つまたは複数の選択したアラームの同じ優先度に属す任意のアラームの現在および今後の発生を非表示にします。
  - 全て再表示 - 非表示の設定をクリアします。
- **クエリー設定** - 以前保存されたアラーム クエリーを選択するための［アラーム クエリー］ダイアログ ボックスを表示します。アラーム クエリーを追加、変更、および削除できます。
- **統計** - ［**アラームの統計**］ダイアログ ボックスを表示します。
- **非表示中** - ［**非表示中のアラーム**］ダイアログ ボックスを表示します。
- **停止** - 現在の表示をフリーズします。
- **再クエリー** - アラーム プロバイダを再びクエリーします。
- **ソート** - ［**ソートのカスタマイズ**］ダイアログ ボックスを表示します。

ステータス バー情報の表示

［一般］プロパティ ページから［ステータス バーの表示］オプションを選択すると、ランタイム中に Alarm Viewer コントロールの一番下にステータス バーが表示されます。



ステータス メッセージ、現在のアラーム クエリー、および進行状況バーの 3 つのインジケータが含まれているステータス バーです。このインジケータでは、表示クエリーの現在の状況の概要と Alarm Viewer コントロールで使用できる非表示機能に関する詳細が提供されます。コントロールが停止の状態である場合は、スータス バーの右ペインが赤となり、1 つまたは複数のアラームが非表示の状態である場合は、ステータス バーの左ペインが赤とまります。非表示の状態である場合、「非表示中」という単語が左ペインに表示されます。

ランタイム時のクエリー設定の使用

以前定義されたアラーム クエリーのリストから素早くアラーム クエリーを選択するには、Alarm Viewer コントロールのショートカット メニューの［クエリー設定］コマンドを使用します。また、新しい名前付きクエリーを作成したり、既存のクエリーを編集したり、既存のクエリーを削除したりすることもできます。

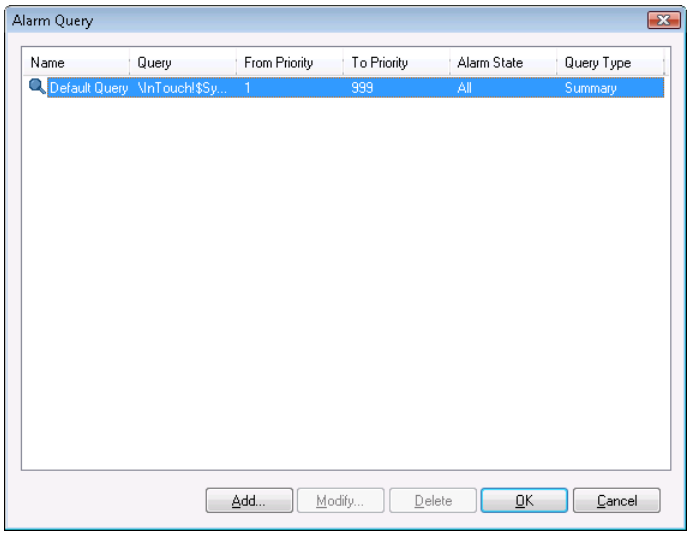
アラーム クエリーへの変更は、同じクエリーを使用する別の Alarm Viewer コントロールには、自動的に適用されません。アラーム クエリーを削除しても、同じクエリーを使用する他の Alarm Viewer コントロールからクエリーは自動的に削除されません。

注意：Alarm Viewer コントロールに複数行アラーム クエリーが表示される場合、改行部分が文字化けします。ただし、機能には影響ありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイム時にアラーム クエリーを選択

1. Alarm Viewer コントロールを右クリックして、［クエリー設定］を選択します。［アラーム クエリー］ダイアログ ボックスが表示されます。



2. 現在定義されているクエリーのリストから表示する名前付きクエリーを選択します。
3. ［**OK**］ を選択します。これで、クエリーによって取得されるアラーム情報が Alarm Viewer コントロールで表示されるようになります。

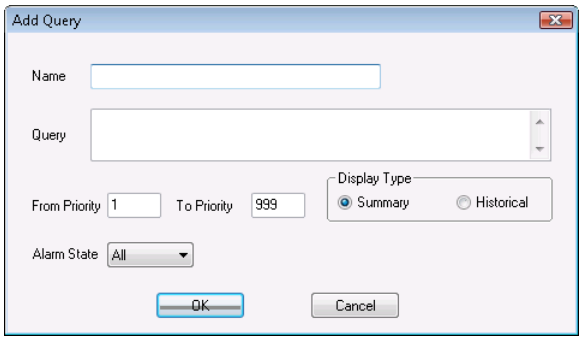
ランタイム時に新しい名前付きクエリーを追加

1. Alarm Viewer コントロールを右クリックして、［クエリー設定］を選択します。

   ［アラーム クエリー］ダイアログ ボックスが表示されます。

2. 追加を選択します。

   ［クエリーの追加］ダイアログ ボックスが表示されます。



3. クエリーを設定します。以下の手順を実行します。
   a. ［名前］ボックスに、クエリーを識別するために使用する名前を入力します。
   b. ［クエリー］ボックスに、実行する InTouch アラーム クエリーのセットを入力します。1 つまたは複数のアラーム プロバイダとグループを指定できます。
   c. ［優先度始点］ボックスに、最低アラーム優先度の値（1 ～ 999）を入力します。
   d. ［優先度終点］ボックスに、最高アラーム優先度の値（1 ～ 999）を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

e. ［アラーム状況］の矢印を選択して、アラーム クエリーで使用するアラーム状況（全て、確認、未確認）を選択します。

f. ［**表示タイプ**］領域で、クエリーするレコードのタイプとして［**サマリ**］または［**履歴**］を選択します。

4. ［**OK**］を選択して、［**クエリーの追加**］ダイアログ ボックスを閉じます。
5. ［アラーム クエリー］ダイアログ ボックスで［OK］を選択して、クエリーをクエリー設定に追加します。

ランタイム時に既存の名前付きクエリーを変更

1. Alarm Viewer コントロールを右クリックして、［クエリー設定］を選択します。

   ［**アラーム クエリー**］ダイアログ ボックスが表示されます。
2. 現在定義されているクエリーのリストで、変更する名前付きクエリーを選択します。
3. [**Modify**] を選択します。

   ［**クエリーの修正**］ダイアログ ボックスが表示されます。
4. 必要な変更を行い、［OK］をクリックします。
5. ［**アラーム クエリー**］ダイアログ ボックスで、［**OK**］を選択します。

ランタイム時に既存の名前付きクエリーを削除

1. Alarm Viewer コントロールを右クリックして、［クエリー設定］を選択します。

   ［**アラーム クエリー**］ダイアログ ボックスが表示されます。
2. 現在定義されているクエリーのリストで、削除する名前付きクエリーを選択します。
3. ［**削除**］を選択します。メッセージが表示されたら、［**はい**］を選択します。
4. ［アラーム クエリー］ダイアログ ボックスで、［**OK**］を選択します。

Alarm Viewer コントロールの ActiveX プロパティの使用

スクリプトを使用して Alarm Viewer コントロールのプロパティの値を直接設定するか、InTouch タグまたは I/O 参照に割り当てることができます。プロパティの設定の詳細については、『*AVEVA™ InTouch HMI アプリケーション開発ガイド*』の「ActiveX コントロールのスクリプト」を参照してください。

以下の表には、Alarm Viewer コントロール プロパティが一覧表示されています。

プロパティ	タイプ	説明
AckAllMenu	論理型	［**全てを確認**］メニュー項目を有効または無効にします。
AckAlmColorRange1	整数型	優先度範囲が 1 から ColorPriorityRange1 までの確認済みアラームを表示するために使用される色を設定します。デフォルトの優先度範囲は 1～250 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ	タイプ	説明
AckAlmColorRange2	整数型	優先度範囲が ColorPriorityRange1 から ColorPriorityRange2 までの確認済みアラームに対して使用される色を設定します。デフォルトの優先度範囲は 250～500 です。
AckAlmColorRange3	整数型	優先度範囲が ColorPriorityRange2 から ColorPriorityRange3 までの確認済みアラームに対して使用される色を設定します。デフォルトの優先度範囲は 500～700 です。
AckAlmColorRange4	整数型	優先度範囲が ColorPriorityRange3 から 999 までの確認済みアラームに対して使用される色を設定します。デフォルトの優先度範囲は 750～999 です。
AckOthersMenu	論理型	［その他を確認］メニュー項目を有効または無効にします。
AckSelectedGroupsMenu	論理型	［選択グループを確認］メニュー項目を有効または無効にします。
AckSelectedMenu	論理型	［選択アラームを確認］メニュー項目を有効または無効にします。
AckSelectedPrioritiesMenu	論理型	［選択優先度を確認］メニュー項目を有効または無効にします。
AckSelectedTagsMenu	論理型	［選択タグを確認］メニュー項目を有効または無効にします。
AckVisibleMenu	論理型	［表示部分を確認］メニュー項目を有効または無効にします。
AlarmQuery(アラーム クエリ)	メッセージ型	初期のアラーム クエリーを設定します。このフィールドはテキストのみを受け入れ、タグは受け入れられません。 以下の例は、アラーム グループへフル パスを使用しています。 \\ノード\InTouch!Group 以下の例では、ローカル アラーム グループへのフル パスを使用しています。 \InTouch!Group この例では、別のグループ リストを使用しています。 GroupList
AlarmState	メッセージ型	クエリーに対するデフォルトのアラーム状態（すべて、未確認、確認）です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ	タイプ	説明
AlmRtnColor	整数型	平常状態に復帰し、確認されていないアラームの色を設定します。この色は、確認状態から平常状態に復帰したものの、確認状態への移行が確認されていないアラームにも使用されます。
AutoScroll	論理型	ユーザーがリストを先頭からスクロールする場合、自動的に新しいアラームにジャンプします新しいアラームは、表示オブジェクト内に現在表示されていないアラームです。
ColorPriorityRange1	整数型	アラームが表示される優先度範囲の境界を設定します。このプロパティの値は 1 より大きく、ColorPriorityRange2 の値よりも小さい値を指定する必要があります。
ColorPriorityRange2	整数型	アラームが表示される優先度範囲の境界を設定します。このプロパティには、ColorPriorityRange1 より大きく、ColorPriorityRange3 よりも小さい値を指定する必要があります。
ColorPriorityRange3	整数型	アラームが表示される優先度範囲の境界を設定します。このプロパティの値には、ColorPriorityRange2 より大きく、999 よりも小さい値を指定する必要があります。
ColumnResize	論理型	ランタイム時にカラムのサイズを変更ができるかどうかを決定する値を返すか設定します。
CustomMessage	メッセージ型	アラームがない場合も表示されるデフォルト メッセージです。
DefaultAckComment	メッセージ型	アラームが確認され、"UseDefaultAckComment" が True の場合に、コメントとして使用されます。それ以外の場合には、ユーザーはコメントを入力することを要求されます。
DisplayedTime	メッセージ型	アラーム メッセージの時間を表示します。値は "OAT"（元のアラーム時刻）、"LCT"（最終変更時刻）、または "LCT But OAT on ACK"（最終変更時刻（確認時には元のアラーム時刻））になります。
DisplayedTimeZone	メッセージ型	現在のタイム ゾーンの文字列を取得または設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ	タイプ	説明
		値は "GMT"、"Origin Time"（標準時間）、または "Local Time"（ローカル時間）です。
EventColor	整数型	イベントの色を設定します。
ExtendedSelection	論理型	Ctrl キーまたは Shift キーを押しながらマウス ボタンと共に使用することにより、ユーザーは複数のアラームを選択できるようになります。 デフォルトでは、簡単に選択することによってアラームの選択を切り替えることができます（[行の選択］チェック ボックスがオンである場合のみ使用できます）。
FlashUnAckAlarms	論理型	未確認アラームの点滅を有効または無効にします。1 または 0 の論理型入力値を取ります。このプロパティが 1 に設定されると、未確認アラームは 1 秒ごとに点滅します。このプロパティが 0 に設定されると、未確認アラームは点滅しません。このプロパティは、Alarm Viewer コントロールの［一般］タブにある［点滅未確認アラーム］チェック ボックスに対応します。
フラッシュウナック **AlmColorRange1**	整数型	アラーム優先度範囲 1 に属す未確認アラームの点滅カラーを設定します。
フラッシュ **UnackAlmColorRange2**	整数型	アラーム優先度範囲 2 に属す未確認アラームの点滅カラーを設定します。
フラッシュ **UnackAlmColorRange3**	整数型	アラーム優先度範囲 3 に属す未確認アラームの点滅カラーを設定します。
フラッシュ **UnackAlmColorRange4**	整数型	アラーム優先度範囲 4 に属す未確認アラームの点滅カラーを設定します。
フォント	なし	コントロールのレコードと見出しのフォントを設定します。
FreezeMenu(フリーズメニュー)	論理型	［停止］メニュー項目を有効または無効にします。
FromPriority	整数型	デフォルトのクエリーの低い優先度値を設定します。
GridColor	整数型	背景グリッドの色を設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ	タイプ	説明
NewAlarmEventMode	整数型	NewAlarm イベントのトリガを制御します。 0 = NewAlarm イベントはトリガできません（デフォルト）。 1 = NewAlarm イベントがアクティブになります。 2 = NewAlarm イベントがアクティブになり、新しい未確認アラームが少なくとも 1 つ取得されると常にトリガされます。
QueryFavoritesFile(クエリ お気に入りファイル)	メッセージ型	クエリー設定ファイルの名前を返すか、設定します。
QueryFavoritesMenu(クエリ お気に入りメニュー)	論理型	［クエリー設定］メニュー項目を有効または無効にします。
クエリ名	文字列	現在のクエリー名を返します。
QueryStartup(クエリス タートアップ)	論理型	設定された場合、デフォルトのクエリー プロパティを使用して、グリッドの更新を自動的に開始します。 選択しない場合、スクリプトで ApplyDefaultQuery メソッドまたは ApplyQuery メソッドを実行して、グリッドを更新する必要があります。
QueryType	メッセージ型	表示タイプを［サマリ］または［履歴］のいずれかに設定します。
RequeryMenu	論理型	［再クエリー］ショートカットメニュー項目を有効または無効にします。
RetainSuppression	論理型	アラーム クエリーが変更された場合に、アラーム クエリー間のアラーム非表示を保持します。
RowSelection	論理型	ランタイム中にユーザーがアラームを選択できるようになります。
SecondarySortColumn	メッセージ型	現在のセカンダリ ソート カラムを返すか、設定します。
SelectedCount	整数型	選択したアラームの合計数を返します。
ShowContextMenu	論理型	ショートカット メニューの起動を有効にします。
ShowGrid	論理型	コントロールでグリッド ラインが表示されるかどうかを決定する値を返すか、設定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ	タイプ	説明
ShowHeading(見出しの表示)	論理型	コントロールのタイトル バーを表示します。
ShowMessage	論理型	コントロールでエラー メッセージを表示されるかどうかを決定する値を返すか、設定します。
ShowStatusBar(ステータスバーの表示)	論理型	ステータス バーが表示されるかどうか決定する値を取得するか、設定します。
SilentMode(サイレントモード)	論理型	コントロールがサイレント モードであるかどうかを決定する値を取得または設定します。
SortColumn	メッセージ型	現在のソート カラムを取得または設定します。
SortMenu(並べ替えメニュー)	論理型	［ソート］メニュー項目を有効または無効にします。
SortOrder(ソート順)	論理型	ソート順を取得または設定します。可能な値は「昇順」および「降順」です。それぞれ、0 および 1 で示されます。
StatsMenu(統計メニュー)	論理型	［統計］メニュー項目を有効または無効にします。
SuppressAllMenu	論理型	［全て非表示］メニュー項目を有効または無効にします。
SuppressedAlarms	整数型	非表示アラームの合計数を取得します。
SuppressionMenu	論理型	［非表示中］メニュー項目を有効または無効にします。
SuppressOthersMenu	論理型	［その他を非表示］メニュー項目を有効または無効にします。
SuppressSelectedGroupsMenu	論理型	［選択グループを非表示］メニュー項目を有効または無効にします。
SuppressSelectedMenu	論理型	［選択アラームを非表示］メニュー項目を有効または無効にします。
SuppressSelectedPrioritiesMenu	論理型	［選択優先度を非表示］メニュー項目を有効または無効にします。
SuppressSelectedTagsMenu	論理型	［選択タグを非表示］メニュー項目を有効または無効にします。
SuppressVisibleMenu	論理型	［表示部分を非表示］メニュー項目を有効または無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

プロパティ	タイプ	説明
TimeFormat	メッセージ型	アラームのタイムスタンプの形式を設定します。
ﾀｲﾄﾙ背景色	整数型	タイトルバーの背景色を設定します。
ﾀｲﾄﾙ前景色	整数型	タイトルバーの前景色を設定します。
ToPriority	整数型	アラーム クエリーの最大優先度を設定します。
TotalAlarms	整数型	アラームの数を取得します。
UnackAlarms	整数型	未確認アラームの合計数を取得します。
UnAckAlmColorRange1	整数型	優先度範囲が 1 から ColorPriorityRange1 までの未確認アラームを表示するために使用される色を設定します。
UnAckAlmColorRange2	整数型	優先度範囲が ColorPriorityRange1 から ColorPriorityRange2 までの未確認アラームを表示するために使用される色を設定します。
UnAckAlmColorRange3	整数型	優先度範囲が ColorPriorityRange2 から ColorPriorityRange3 の未確認アラームを表示するために使用される色を設定します。
UnAckAlmColorRange4	整数型	優先度範囲が ColorPriorityRange3 から 999 までの未確認アラームを表示するために使用される色を設定します。
UnsuppressAllMenu	論理型	［全て再表示］メニュー項目を有効または無効にします。
UseDefaultAckComment	論理型	True に設定すると、アラームが確認されるときにデフォルトの確認コメントが使用されます。それ以外の場合には、オペレータはコメントを入力することを要求されます。
ｳｨﾝﾄﾞｳの色	整数型	グリッドの背景色を設定します。

ActiveX コントロールの色の設定

整数値として色を指定します。アラーム ActiveX コントロールでは、ABGR モデルを使用して 32 ビットの整数値として色を指定しています。

A = 透明度

B = 青

G = 緑

R = 赤

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

透明度はアラーム ActiveX コントロールではサポートされていません。上位 8 ビットの値はすべて無視されます。

たとえば、色を青に設定するには、以下のような ABGR 値を使用します。

A = 0

B = 255

G = 0

R = 0

この色の 16 進数値は 0x00FF0000 です。10 進数値は 16711680 です。

次の表は、使用できる色の例を示しています。

色	16 進数値	10 進数値
白	0x00FFFFFF	16777215
黒	0000000	0
青	0x00FF0000	16711680
赤	0x000000E1	225
緑	0x0000FF00	65280

Alarm Viewer コントロールの ActiveX メソッドの使用

Alarm Viewer コントロールの ActiveX メソッドをスクリプトで使用すると、以下の操作を行えます。

- アラームを確認する
- アラームを非表示にする
- アラームに関する情報を取得する
- アラーム クエリーを実行する
- 表示を移動および停止する
- アラーム レコードをソートする
- 特定のアラームを選択する
- ショートカット メニュー、［**バージョン情報**］ダイアログ ボックス、および［**アラームの統計**］ダイアログ ボックスを表示します。

メソッドの呼び出しの詳細については、『*AVEVA™ InTouch HMI アプリケーション開発ガイド*』の「ActiveX コントロールのスクリプト」を参照してください。

実行時にアラームを確認する

以下のメソッドを使用して、ランタイム中にアラームを確認します。

- AckSelected() メソッド AckSelected めそっど
- AckAll() メソッド AckAll めそっど

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- AckVisible() メソッド AckVisible めそっど
- AckSelectedGroup() メソッド AckSelectedGroup めそっど
- AckSelectedTag() メソッド AckSelectedTag めそっど
- AckSelectedPriority() メソッド AckSelectedPriority めそっど
- AckGroup() メソッド AckGroup めそっど
- AckPriority() メソッド AckPriority めそっど
- AckTag() メソッド AckTag めそっど

AckSelected() メソッド AckSelected めそっど

ランタイムで Alarm Viewer コントロールで選択されたアラームを確認します。

構文

```
オブジェクト。AckSelected(コメント)
```

パラメータ

コメント
アラーム確認コメントです。

例

Tag1 がメッセージ型タグ変数として定義され、コントロールの名前は AlarmViewerCtrl1 です。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckSelected (Tag1);
```

AckAll() メソッド AckAll めそっど

現在のアラーム クエリーにあるすべてのアラームを確認します。Alarm Viewer コントロールの表示領域は制限されているため、AckAll() メソッドでは画面に表示されていないアラームも確認します。

構文

```
オブジェクト。AckAll (コメント)
```

パラメータ

コメント
アラーム確認コメントです。

例

Tag1 がメッセージ型タグ変数として定義され、コントロールの名前は AlarmViewerCtrl1 です。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckAll (Tag1);
```

AckVisible() メソッド AckVisible めそっど

Alarm Viewer コントロールに現在表示されているアラームのみを確認します。

構文

```
オブジェクト。AckVisible (コメント)
```

パラメータ

コメント
アラーム確認コメントです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

Tag1 がメッセージ型タグ変数として定義され、コントロールの名前は AlarmViewerCtrl1 です。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckAll (Tag1);
```

AckSelectedGroup() メソッド AckSelectedGroup めそっど

選択した 1 つまたは複数のアラームと同じグループ名を持つすべてのアラームを確認します。

構文

オブジェクト。AckSelectedGroup(*コメント*)

パラメータ

コメント
アラーム確認コメントです。

例

Tag1 がメッセージ型タグ変数として定義され、コントロールの名前は AlarmViewerCtrl1 です。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckAll (Tag1);
```

AckSelectedTag() メソッド AckSelectedTag めそっど

選択した 1 つまたは複数のアラームと同じタグ変数、グループ名、および優先度を持つすべてのアラームを確認します。

構文

オブジェクト。AckSelectedTag (*コメント*)

パラメータ

コメント
アラーム確認コメントです。

例

Tag1 がメッセージ型タグ変数として定義され、コントロールの名前は AlarmViewerCtrl1 です。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckAll (Tag1);
```

AckSelectedPriority() メソッド AckSelectedPriority めそっど

選択した 1 つまたは複数のアラームと同じ優先度範囲を持つすべてのアラームを確認します。

構文

オブジェクト。AckSelectedPriority (*コメント*)

パラメータ

コメント
アラーム確認コメントです。

例

Tag1 がメッセージ型タグ変数として定義され、コントロールの名前は AlarmViewerCtrl1 です。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckSelectedPriority (Tag1);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

AckGroup() メソッド AckGroup めそっど

指定のグループ名およびプロバイダに対するすべてのアラームを確認します。

構文

```
オブジェクト。AckGroup(ApplicationName、GroupName、Comment)
```

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

コメント
アラーム確認コメントです。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.AckGroup("\Intouch","Turbine", "Turbine acknowledgement Comment");
```

AckPriority() メソッド AckPriority めそっど

同じプロバイダ名とグループ名を持つ指定した優先度範囲のすべてのアラームを確認します。

構文

```
オブジェクト。AckPriority(ApplicationName、GroupName、FromPriority、ToPriority、Comment)
```

パラメータ

アプリケーション名
アプリケーションの名前です。たとえば、\\node1\Intouch です。

グループ名
グループの名前です。たとえば、Turbine です。

***FromPriority** プロパティ*
アラームの開始優先度です。たとえば、100 です。

***ToPriority** プロパティ*
アラームの終点優先度です。例: 900。

コメント
アラーム確認コメントです。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.AckPriority("\Intouch"、"Turbine"、100、900、"Turbine acknowledgement
Comment");
```

AckTag() メソッド AckTag めそっど

指定した優先度範囲で同じプロバイダ名およびグループ名を持つ指定のタグ変数のアラームを確認します。

構文

```
オブジェクト。AckTag(ApplicationName、 GroupName、 tag、 FromPriority、 ToPriority、 Comment)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

タグ
アラーム タグ変数の名前です。たとえば、Valve1 です。

***FromPriority** プロパティ*
アラームの開始優先度です。たとえば、100 です。

***ToPriority** プロパティ*
アラームの終点優先度です。例: 900。

コメント
アラーム確認コメントです。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.AckTag("\Intouch"、"Turbine"、"Valve1"、100、900、"Turbine acknowledgement Comment");
```

リアル タイムでのアラームの非表示

以下のメソッドを使用して、ランタイム中にアラームを非表示にします。

- ShowSuppression() メソッド
- SuppressSelected() メソッド SuppressSelected()めそっど
- SuppressAll() メソッド SuppressAll めそっど
- SuppressVisible() メソッド SuppressVisible めそっど
- SuppressSelectedGroup() メソッド SuppressSelectedGroup()めそっど
- SuppressSelectedTag() メソッド SuppressSelectedTag()めそっど
- SuppressSelectedPriority() メソッド SuppressSelectedPriority()めそっど
- UnSuppressAll() メソッド UnSuppressAll()めそっど
- SuppressGroup() メソッド SuppressGroup()めそっど
- SuppressPriority() メソッド SuppressPriority()めそっど
- SuppressTag() メソッド SuppressTag()めそっど

ShowSuppression() メソッド

非表示にされているすべてのアラームが含まれる［非表示中のアラーム］ダイアログ ボックスを表示します。

構文

オブジェクト。ShowSuppression()

例

コントロールの名前は AlarmViewerCtrl1 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
#AlarmViewerCtrl1.ShowSuppression();
```

SuppressSelected() メソッド SuppressSelected()めそっど

選択したアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressSelected()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressSelected();
```

SuppressAll() メソッド SuppressAll めそっど

すべてのアクティブなアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressAll()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressAll();
```

SuppressVisible() メソッド SuppressVisible めそっど

表示されているアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressVisible()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressVisible();
```

SuppressSelectedGroup() メソッド SuppressSelectedGroup()めそっど

選択した 1 つまたは複数のアラームと同じグループおよびプロバイダに属すアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressSelectedGroup()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressSelectedGroup();
```

SuppressSelectedTag() メソッド SuppressSelectedTag()めそっど

同じグループ名、プロバイダ名、および優先度範囲を持つ選択した 1 つまたは複数のアラームの同じタグ変数名に属す任意のアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressSelectedTag()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
#AlarmViewerCtrl1.SuppressSelectedTag();
```

SuppressSelectedPriority() メソッド SuppressSelectedPriority()めそっど

1 つまたは複数の選択したアラームの同じ優先度範囲に属す任意のアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressSelectedPriority()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressSelectedPriority();
```

UnSuppressAll() メソッド UnSuppressAll()めそっど

アラームの非表示をクリアします。

構文

```
オブジェクト。UnSuppressAll()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.UnSuppressAll();
```

SuppressGroup() メソッド SuppressGroup()めそっど

指定のグループ名に属す任意のアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressGroup(ApplicationName,GroupName)
```

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressGroup("\Intouch", "Turbine");
```

SuppressPriority() メソッド SuppressPriority()めそっど

同じプロバイダ名およびグループ名を持つ指定した優先度範囲の任意のアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressPriority(ApplicationName、GroupName、FromPriority、ToPriority)
```

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

***FromPriority** プロパティ*
アラームの開始優先度です。たとえば、100 です。

***ToPriority** プロパティ*
アラームの終点優先度です。例: 900。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressPriority("\Intouch","Turbine", 100, 900);
```

SuppressTag() メソッド SuppressTag()めそっど

指定した優先度範囲の指定したタグ変数名またはグループ名によって発行される任意のアラームの現在および今後の発生を非表示にします。

構文

```
オブジェクト。SuppressTag(ApplicationName、GroupName、tag、FromPriority、ToPriority)
```

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などのアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

タグ
アラーム タグ変数の名前です。たとえば、valve 1 です。

***FromPriority** プロパティ*
アラームの開始優先度です。たとえば、100 です。

***ToPriority** プロパティ*
アラームの終点優先度です。例: 900。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressTag("\Intouch"、"Turbine"、"Valve1"、100、900);
```

特定のアラームに関する情報の取得

アラームに関する情報を取得するには、GetItem() 関数を使用します。

GetItem() メソッド

指定された行とカラムのデータを文字列として返します。

構文

```
オブジェクト。GetItem(Integer、Message)
```

パラメータ

整数型
コントロールの特定の行を評価する整数式です。

メッセージ型
コントロールのカラム名を評価する文字列式です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

コントロール名は AlmDbView1 です。タグはメッセージ型タグとして定義されます。

```
tag = #AlmDbView1.GetItem(1, "Group");
```

アラーム クエリーを実行する

クエリーを実行するには、以下のメソッドを使用します。

- ShowQueryFavorites() メソッド ShowQueryFavorites()めそっど
- Requery() メソッド Requery()めそっど
- ApplyQuery() メソッド ApplyQuery()めそっど
- ApplyDefaultQuery() メソッド ApplyDefaultQuery()めそっど
- SetQueryByName() メソッド SetQueryByName()めそっど, SetQueryByName() メソッド SetQueryByName()めそっど

ShowQueryFavorites() メソッド ShowQueryFavorites()めそっど

QueryFavoritesFile プロパティに .xml 形式で有効なクエリー設定ファイルが含まれている場合に、[**クエリー設定**] ダイアログ ボックスを表示します。

構文

```
オブジェクト。ShowQueryFavorites()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.ShowQueryFavorites();
```

Requery() メソッド Requery()めそっど

アラーム プロバイダを再びクエリーします。

構文

```
オブジェクト。Requery()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.Requery();
```

ApplyQuery() メソッド ApplyQuery()めそっど

アラーム クエリー、優先度始点、優先度終点、クエリー対象のアラームの状況、および取得するアラームのタイプなど、パラメータによって指定されているようにクエリーを実行します。

構文

```
オブジェクト。ApplyQuery(AlarmQuery、FromPriority、ToPriority、State、Type)
```

パラメータ

AlarmQuery(アラーム クエリ)
アラーム クエリーです。例： \InTouch!$System

FromPriority プロパティ
アラームの開始優先度です。たとえば、100 です。

ToPriority プロパティ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラームの終点優先度です。例: 900。

状態
表示するアラームのタイプを指定します。たとえば、"UnAck" またはメッセージ型タグ変数です。有効な状態は、All、UnAck、および Ack です。

タイプ
クエリーのタイプを指定します。たとえば、Historical（履歴）（履歴アラーム）または Summary（サマリ）（サマリ アラーム）です。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.ApplyQuery ("\InTouch!$System",100,900,"All", "Historical");
```

ApplyDefaultQuery() メソッド ApplyDefaultQuery()めそっど

デザイン時に指定されているように、優先度始点、優先度終点、アラーム状況、クエリー タイプ、およびアラーム クエリーのプロパティを使用してクエリーを実行します。デフォルトのプロパティは開発時でのみ変更でき、その他のアラーム クエリーによって上書きされることはありません。

構文

オブジェクト。デフォルトクエリ()を適用する

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.ApplyDefaultQuery();
```

SetQueryByName() メソッド SetQueryByName()めそっど

渡されるクエリー名によって指定されている通りに現在のクエリーを設定します。クエリーはクエリー設定ファイルに存在する必要があります。

構文

オブジェクト。SetQueryByName(*QueryName*)

パラメータ

クエリ名
クエリー設定を使用して作成されたクエリーの名前です。たとえば、Turbine Queries です。

例

コントロールの名前は AlarmTreeViewerCtrl1 です。

```
#AlarmTreeViewerCtrl1.SetQueryByName("Turbine Queries");
```

表示を移動および停止する

表示を移動または停止するには、以下の関数を使用します。

- MoveWindow() メソッド MoveWindow()めそっど
- FreezeDisplay() メソッド FreezeDisplay()めそっど

MoveWindow() メソッド MoveWindow()めそっど

コントロールのアラームを指定した方法でスクロールします。

構文

オブジェクト。MoveWindow(*オプション*、*繰り返し*)

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

オプション
実行する動作のタイプです。

タイプ	説明
LineDn	次の行へ移動。Repeat パラメータによって、スクロールされる行数が制御されます。
ラインナップ	前の行へ移動。Repeat パラメータによって、スクロールされる行数が制御されます。
PageDn	次ページへ移動。Repeat パラメータによって、スクロールされるページ数が制御されます。
PageUp	前ページへ移動。Repeat パラメータによって、スクロールされるページ数が制御されます。
最上部	コントロールの一番上へ移動。
最下部	コントロールの一番下へ移動。
ページ Rt	右ページへ移動。Repeat パラメータによって、スクロールされるページ数が制御されます。
PageLf	左ページへ移動。Repeat パラメータによって、スクロールされるページ数が制御されます。
右	右へスクロール。Repeat パラメータによって、スクロールされるカラム数が制御されます。
左	左へスクロール。Repeat パラメータによって、スクロールされるカラム数が制御されます。
ﾎｰﾑ	コントロールの一番上の行の最も左のカラムまでスクロールします。

繰り返し
操作が繰り返される回数です。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.MoveWindow ("Bottom", 0);
#AlarmViewerCtrl1.MoveWindow("LineUp", 3);
#AlarmViewerCtrl1.MoveWindow(「PageLf」、7);
```

FreezeDisplay() メソッド FreezeDisplay()めそっど

表示を停止します。

構文

オブジェクト。フリーズ画面(*フリーズ*)

パラメータ

停止
True = 表示を停止します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

False = 表示の停止を解除します。

例

Tag1 はメモリ論理型タグ変数として定義されており、コントロールの名前は AlarmViewerCtrl1 です。

```
タグ＝1
#AlarmViewerCtrl1.FreezeDisplay(Tag1);
```

アラーム レコードをソートする

アラーム レコードをソートするには、以下の関数を使用します。

- ShowSort() メソッド, ShowSort() メソッド
- SetSort() メソッド SetSort()めそっど

ShowSort() メソッド

SortMenu プロパティが有効である場合、［**セカンダリ ソート**］ダイアログ ボックスを表示します。

構文

```
オブジェクト。ShowSort()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.ShowSort();
```

SetSort() メソッド SetSort()めそっど

SortColumn プロパティおよび SortOrder プロパティで指定されているソート条件を設定します。

構文

```
オブジェクト。SetSort()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SetSort();
```

追加情報の表示

［バージョン情報］ダイアログ ボックスと［アラームの統計］ダイアログ ボックスを表示するには、以下の関数を使用します。

- AboutBox() メソッド, AboutBox() メソッド, AboutBox() メソッド
- ShowStatistics() メソッド ShowStatistics()めそっど

AboutBox() メソッド

［**バージョン情報**］ダイアログ ボックスを表示します。

構文

```
オブジェクト。AboutBox()
```

例

コントロールの名前は AlarmPareto1 です。

```
#AlarmPareto1.AboutBox();
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ShowStatistics() メソッド ShowStatistics()めそっど

［アラームの統計］ダイアログ ボックスを表示します。

構文

```
オブジェクト。ShowStatistics()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.ShowStatistics();
```

特定のアラームを選択する

特定のアラームを選択するには、以下の関数を使用します。

- SelectGroup() メソッド SelectGroup()めそっど
- SelectPriority() メソッド SelectPriority()めそっど
- SelectTag() メソッド SelectTag()めそっど
- SelectAll() メソッド SelectAll()めそっど
- SelectItem() メソッド SelectItem()めそっど
- UnSelectAll() メソッド UnSelectAll()めそっど

SelectGroup() メソッド SelectGroup()めそっど

同じアラーム グループ名とプロバイダ名が含まれるすべてのアラームを選択します。

構文

```
オブジェクト。SelectGroup(ApplicationName, GroupName)
```

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SelectGroup("\Intouch", "Turbine");
```

SelectPriority() メソッド SelectPriority()めそっど

同じプロバイダ名とグループ名を持つ指定した優先度範囲のアラームをすべて選択します。

構文

```
オブジェクト。SelectPriority(ApplicationName、GroupName、FromPriority、ToPriority)
```

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

***FromPriority** プロパティ*
アラームの開始優先度です。たとえば、100 です。

***ToPriority** プロパティ*
アラームの終点優先度です。例: 900。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SelectPriority("\Intouch", "Turbine", 100, 900);
```

SelectTag() メソッド SelectTag()めそっど

特定のプロバイダ、グループ、またはタグ変数からのすべてのアラームを選択します。優先度範囲または 1～999 の値を指定できます。

構文

オブジェクト。SelectTag(*ApplicationName*、*GroupName*、*tag*、*FromPriority*、*ToPriority*)

パラメータ

アプリケーション名
たとえば、\\node1\Intouch などアプリケーション名です。

グループ名
グループの名前です。たとえば、Turbine です。

タグ
アラーム タグ変数の名前です。たとえば、valve 1 です。

***FromPriority** プロパティ*
アラームの開始優先度です。たとえば、100 です。

***ToPriority** プロパティ*
アラームの終点優先度です。例: 900。

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SelectTag("\Intouch"、"Turbine"、"Valve1"、100、900);
```

SelectAll() メソッド SelectAll()めそっど

表示のすべてのアラームの選択を切り替えます。アラーム表示の表示領域は制限されているため、SelectAll() 関数では画面に表示されていないアラームも選択します。

構文

オブジェクト。SelectAll()

例

```
#AlarmViewerCtrl1.SelectAll();
```

SelectItem() メソッド SelectItem()めそっど

指定した行にあるアラーム レコードの選択を切り替えます。

構文

オブジェクト。SelectItem(*RowNumber*)

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

行番号
選択するアラーム レコードの行番号である整数値です。コントロールの最初の行は 0 です。

例

コントロールの名前は AlarmViewerCtrl1 で、Tag1 はメモリ整数型として定義されています。Alarm Viewer コントロールの 10 行目のアラーム レコードの選択を切り替えます。

```
Tag1 = 9。
#AlarmViewerCtrl1.AckSelected (Tag1);
```

UnSelectAll() メソッド UnSelectAll()めそっど

選択したすべてのレコードを選択解除します。

構文

```
オブジェクト。UnSelectAll()
```

例

コントロールの名前は AlarmViewerCtrl1 です。

```
#AlarmViewerCtrl1.SuppressAll();
```

ランタイム時のコンテキスト メニューの表示

ランタイムでにショートカット メニューを表示するには、ShowContext() メソッドを使用します。

ShowContext() メソッド ShowContext()めそっど

RefreshMenu、ResetMenu、SortMenu の 3 つのプロパティのうち 1 つを有効にすると、ショートカット メニューが表示されます。

構文

```
オブジェクト。ShowContext()
```

例

コントロールの名前は AlmDbView1 です。

```
#AlmDbView1.Show Replicas();
```

メソッドとプロパティを使用するときのエラー処理

SilentMode プロパティを使用して、ランタイム中にエラーを非表示にできます。SilentMode プロパティを 1 に設定すると、Alarm Viewer コントロールではランタイム中にエラー メッセージは表示されません。0 に設定すると、Alarm Viewer コントロールではエラー メッセージが表示されます。エラー メッセージは、常に Operations Control Log Viewer に送信されます。

ActiveX イベントを使用したスクリプトのトリガ

マウス クリックやダブルクリックなどの Alarm Viewer コントロール イベントに、QuickScript を割り当てることができます。イベントが発生すると、QuickScript が実行されます。

Alarm Viewer コントロールは、以下のイベントをサポートしています。

- 選択
- ダブルクリック
- NewAlarm

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 終了
- 起動

Click イベントは ClicknRow というパラメータを 1 つ持ち、これはランタイム時に選択された行を識別します。

DoubleClick イベントは DoubleClicknRow というパラメータを 1 つ持ち、これはランタイム時にダブルクリックされた行を識別します。

選択イベントおよび DoubleClick イベントはゼロを基準としています。選択イベントおよび DoubleClick イベントがユーザーに対して発行されると、表示される行数は 0 から始まります。

注意：ユーザー インターフェイス メソッドが StartUp イベントから呼び出される場合、コントロールはまだ表示されていないため、Alarm Viewer コントロールでは無視されます。これらには、ShowSort()、ShowContext()、GetSelectedItem()、GetNext()、GetPrevious()、および AboutBox() があります。

ActiveX イベントのスクリプトの詳細については、『*AVEVA™ InTouch HMI アプリケーション開発ガイド*』の「ActiveX コントロールのスクリプト」を参照してください。

新しいアラームが検出されたときのスクリプトの実行あたらしいあらーむがけんしゅつされたときのすくりぷとのじっこう

Alarm Viewer コントロールで新しい未確認アラーム（つまり、平常状態から未確認状態に移行し、コントロールのクエリーや優先度のフィルタ条件に一致するアラーム）が検出された場合に、ActiveX イベント スクリプトを実行するように Alarm Viewer コントロールを設定できます。

NewAlarmEventMode プロパティは、NewAlarm イベントのトリガを制御します。

- NewAlarmEventMode プロパティを 0 に設定すると、NewAlarm イベントはトリガされません。これがデフォルトです。
- NewAlarmEventMode プロパティを 1 に設定すると、新しいアラームが発生します。
  - イベントがトリガされます。
  - NewAlarm イベントに関連付けられた ActiveX イベント スクリプトが実行されます。
  - NewAlarmEventMode プロパティが 0 に設定されます。

  後続のイベントを処理するには、NewAlarmEventMode プロパティを 1 に戻す必要があります。

  この設定は、状況が改善されて確認されるまでは実行するべきではない動作をアプリケーションで実行する場合に使用されます。たとえば、イベントがトリガされた場合に、アラームが確認されるまで ActiveX スクリプトでアラーム音を鳴らすことができます。新しいアラームを次回受け取ったときに、再びアラーム音を再生できます。
- NewAlarmEventMode プロパティを 2 に設定すると、NewAlarm イベントがアクティブとなり、新しい未確認アラームが少なくとも 1 つ受け取られると続けてトリガされます。後続のイベントを処理するために値を変更する必要はありません。1 秒間に受け取る新しい追加のアラームの数に関わらず、1 秒間にトリガされる新しいイベントは最大でも 1 回のみです。

リアル タイムでのアラームの確認

タグ データが通常状態からアラーム状態に移行すると、アラームの新しいインスタンスが生成されます。InTouch の分散アラーム システムでは、各アラーム インスタンスの以下の状態を追跡しています。

- タグが初めてアラーム状態になったとき

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アラームがマルチステートのアラームである場合、アラームがサブステートへの移行を行ったとき
- アラームが通常状態に戻ったとき
- アラームが確認を待機しているかどうか
- アラームが確認されたとき

アラーム インスタンスのライフサイクルは、アラームに関連付けられているタグ値が通常状態、つまりアラーム解除の状態に戻った時点で終了します。その後でアラーム状態に移行すると、新しいアラーム インスタンスが発生します。

アラーム確認モデルの理解

InTouch HMI では、3 つのアラーム確認モデルをサポートしています。

- 条件指向のアラームの場合、確認の時点までのアラーム状況へのすべてのエントリに対して確認はカウントされます。
- 拡張サマリ のアラームの場合、アラーム状況やサブステートへの移行、または平常状態への復帰などの場合でも、確認は特定の移行に対してのみ行われます。アラーム全体が確認されたとみなされるためには、異なるアラーム サブステートへのすべての移行が確認される必要があります。
- イベント指向のアラーム（OPC など）の場合、最後のアクティベーション イベントを参照している場合のみ確認が受け入れられます。

条件、確認アラーム モデルじょうけんかくにんあらーむもでる

条件指向のアラームの場合、確認の時点までのアラーム状況へのすべてのエントリに対して確認はカウントされます。

確認はアラームのインスタンスに対して行われます。アラーム インスタンスは初めてアラーム状況になると、確認を待機します。アラームが確認され、新しいアラーム サブステートにその後で移行する場合（たとえば「Hi」から「HiHi」に移行）、別の確認の待機を開始します。確認は受け取られるたびに、受け入れられ、これまでに発生したアラームのすべての移行に適用されます。

最後のインスタンスが確認されたときに、アラームは確認されたとみなされます。

拡張サマリ アラーム モデル

拡張サマリ のアラームの場合、アラーム状況やサブステートへの移行、または平常状態への復帰などの場合でも、確認は特定の移行に対してのみ行われます。アラーム全体が確認されたとみなされるためには、異なるアラーム サブステートへのすべての移行が確認される必要があります。

アラーム状況への最初のエントリ、および平常状態への復帰は個別に確認される必要があります。

新しいアラーム サブステートへの移行はすべて確認される必要のある新しい発生と見なされ、平常への復帰も確認する必要があります。サブステート間の移行は、アイテムが以前平常であったときのアラーム状況への最初のエントリで始まる「平常への復帰グループ」に属すると見なされます。

アイテムが平常に戻った後で再びアラーム状況に入ると、新しい平常への復帰グループが作成されます。

それぞれの移行は個別に明示的に確認される必要があり、アラームはアイテムが平常に戻り、保留状態にある平常への復帰グループがすべて確認されて初めて確認されたと見なされます。

注意：「サマリ」という単語は、「確認の待機状態」を意味します。このモデルのアラームは、「リングバック」アラームとも呼ばれます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

拡張サマリ アラーム レコード

拡張サマリ アラームの場合、アラームが発生すると、アラーム表示オブジェクトにレコードが生成され、アラーム状況が発生したことが表示されます。レコードには、アラームの日付／時間スタンプが表示されます。このレコードは、オペレータがアラームを確認し、平常への復帰状態になるまで表示されたまま残ります。アラームが確認される前に平常への復帰状態が発生した場合、アラーム オブジェクトに 2 つのレコードが表示されます。

たとえば、ボイラーの温度が上限値を超えて、アラームがトリガされるとします。その後で、オペレータがアラームを確認する前にボイラーの温度が通常の範囲内に戻ります。アラーム システムは上限アラームの発生を示すレコードと、アラームが未確認であることを示す別のレコードを生成します。

拡張サマリ アラームの使用

タグ変数を定義する際に確認モデルとして［**拡張サマリ**］を選択した場合、アラームをトリガする状況がすでに平常状態に戻っていたとしても、オペレータはそのアラームが発生したことを確認する必要があります。アラームを確認すると、アラーム エントリの色は変更されますが、時刻スタンプは変更されません。アラームが確認され、平常状態に戻ると、アラームの表示だけがクリアされます。

注意：拡張サマリの確認モードを使用してタグ変数を定義する場合、［**アラームのプロパティ**］ダイアログ ボックスの［アラーム復旧時に確認する］オプションはタグ変数には適用されません。

イベント指定アラーム モデル

イベント指向のアラーム（OPC など）の場合、最後のアクティベーション イベントを参照している場合のみ確認が受け入れられます。

アラーム インスタンスは初めてアラーム状況になると、確認の待機を開始します。インスタンスが確認され、その後で新しいアラーム サブステートに移行する場合、別の確認の待機を開始します。それ以降の移行には連番が割り当てられます。確認は応答している移行の連番を付ける必要があります。

確認は最後の移行に対してのみ受け入れられます。受け入れられる場合、これまでに発生したアラームのすべての移行に対して適用されます。最後のインスタンスが確認されたときに、アラームは確認されたとみなされます。

拒否された確認は、診断のためにログ記録されますが、システムでは追跡されません。

イベント指向のモデルでは、アラームが 2 つの状態の間で保留となっている場合、確認は現在の情報に応答します。遅延時間の短いシステムでは、このモデルは条件指向のアラームのように見えます。インターネットなどその他の環境では、このモデルの特徴が重要となります。

ランタイムでのタグ変数の確認モデルの確認

タグ変数に関連付けられた確認モデルの監視には、.AlarmAckModel ドットフィールドを使用します。

.AlarmAckModel ドットフィールド.AlarmAckModel どっとふぃーるど

タグ変数に関連付けられている確認モデルを以下のように監視します。

0 = 条件（デフォルト）

1 = イベント指向

2 = 拡張サマリ

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

```
Tagname.AlarmAckModel
```

パラメータ

タグ名
論理型、整数型、実数型、間接論理型、およびアナログ型の任意のタグ変数

備考

このドットフィールドのデフォルト値は 0 です（条件確認モデル）。

データ型

アナログ型（読み取り専用）

有効値

0、1、または 2

例

PumpStation タグ変数がイベント アラームに関連付けられている場合に、この IF-THEN ステートメントの本文が処理されます。

```
IF (PumpStation.AlarmAckModel == 1) その後
   MyAlarmMessage="PumpStation はイベント アラームです";
ENDIF;
```

参照項目

.Alarm、.Ack、.UnAck、.AckDev、.AckDSC、.AckROC

アラームを確認するためのドットフィールドの使用

現在のすべてのアラーム、選択したアラーム、または指定タイプのアラームのみを確認するドットフィールドを使用するスクリプトを作成できます。

アラームまたはアラーム グループの確認

指定したローカル タグ変数のアラームか、指定したアラーム グループ内のアラームを確認する以下のドットフィールドを使用するスクリプトを作成できます。

- .Ack ドットフィールド.Ack どっとふぃーるど
- .UnAck ドットフィールド UnAck どっとふぃーるど

これらのドットフィールドを使用して、Application Server から生成されるアラームは確認できません。

実行している InTouch アプリケーションのすべてのローカル アラームを確認するには、$System アラーム グループを適切な .Ack ドットフィールドとの組み合わせで使用します。

.Ack ドットフィールド.Ack どっとふぃーるど

すべてのタイプのローカル アラームのアラーム確認状況を監視または制御します。

カテゴリ

アラーム

使用法

```
TagName。Ack=1。
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

TagName
論理型、整数型、実数型、間接論理型、およびアナログ型のタグ、またはアラーム グループ

備考

指定したタグ変数またはアラーム グループに関連付けられるすべての未処理のアラームを確認するには、このドットフィールドの値を 1 に設定します。指定したタグ変数がアラーム グループである場合、指定したグループ内のタグ変数に関連付けられるすべての未確認アラームが確認されます。指定したタグ変数がアラーム グループ以外の場合、そのタグ変数に関連付けられている未確認アラームだけが確認されます。.Ack ドットフィールドに 1 以外の値を設定しても意味がありません。

データ型

論理型（読み取り/書き込み）

有効値

1

例

以下のステートメントでは、Tag1 のタグ変数に関連付けられているアラームが確認されます。

```
Tag1.Ack=1;
```

この次の例は、PumpStation アラーム グループ内のすべての未確認アラームを確認するために使用されます。

```
PumpStation.Ack=1;
```

注意：Ack ドットフィールドには、.UnAck と呼ばれる逆のドットフィールドがあります。未確認アラームが発生すると、.UnAck は 1 に設定されます。.UnAck はその後でアニメーション リンクや条件スクリプトで使用して、未確認アラームのアナンシエーターをトリガできます。

参照項目

Alarm、UnAck、AckDev、AckROC、AckDSC、AckValue、AlarmAckModel

.UnAck ドットフィールド UnAck どっとふぃーるど

ローカル アラームのアラーム確認状況を監視または制御します。

カテゴリ

アラーム

使用法

```
TagName。UnAck=0。
```

パラメータ

TagName
論理型、整数型、実数型、間接論理型、およびアナログ型のタグ、またはアラーム グループ

備考

指定したタグ変数またはアラーム グループに関連付けられるすべての未処理のアラームを確認するには、このドットフィールドの値を 0 に設定します。指定したタグ変数がアラーム グループである場合、指定したグループ内のタグ変数に関連付けられるすべての未確認アラームが確認されます。指定した

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ変数がそれ以外のタイプの場合、そのタグ変数に関連付けられる未確認のアラームだけが確認されます。このドットフィールドに 0 以外の値を設定しても意味がありません。

データ型

論理型（読み取り／再設定）専用

有効値

0

例

以下のステートメントでは、Tag1 タグ変数に関連付けられているアラームが確認されます。

```
Tag1.UnAck=0;
```

以下のステートメントは、PumpStation という名前のアラーム グループ内のすべての未確認アラームを確認します。

```
PumpStation.UnAck=0;
```

.UnAck には . Ack と呼ばれる逆ドットフィールドがあります。アラームが確認されている場合、.Ack ドットフィールドの値は 1 に設定されます。

参照項目

.Ack、Ack()、.Alarm、.AlarmAckModel

値アラームを確認する

.AckValue ドットフィールドを使用して、指定したローカル タグ変数のすべての値型アラームか、指定したアラーム グループ内のすべての値型アラームを確認するスクリプトを作成できます。

.AckValue ドットフィールド.AckValue どっとふぃーるど

ローカルの値型アラームの確認を監視および制御します。

使用法

```
TagName。AckValue=1;
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

を設定します。指定したタグ変数／グループに関連付けられる未処理の値型アラームをすべて確認、.AckValue ドットフィールドを 1 に設定します。指定したタグ変数がアラーム グループである場合、指定したグループ内のタグ変数に関連付けられるすべての未確認の値型アラームが確認されます。指定したタグ変数が任意のタイプの場合、そのタグ変数に関連付けられる未確認の値型アラームだけが確認されます。

このドットフィールドに 1 以外の値を設定しても意味がありません。

データ型

論理型（読み取り/書き込み）

有効値

1

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のステートメントでは、Tag1 タグ変数に関連付けられている値型アラームが確認されます。

```
Tag1.AckValue=1;
```

以下の例では、PumpStation という名前のアラーム グループ内のすべての未確認の値型アラームが確認されます。

```
PumpStation.AckValue=1;
```

間接アラーム グループ（GroupVar の使用）を使用して、値型アラームを確認できます。たとえば、以下のような割り当てを使用します。

```
StationAlarms.Name = "PumpStation";
```

ここで StationAlarms は、アラーム グループ型タグ変数として定義され、PumpStation に関連付けられます。このように、以下のステートメントは、StationAlarms アラーム グループ変数タグに現在関連付けられているアラーム グループ PumpStation 内の未確認の値型アラームを確認するために使用される以外は、上の例と類似しています。

```
StationAlarms.AckValue=1;
```

参照項目

.Alarm、.AlarmValue、.Ack、.UnAck、.AckDev、.AckDSC、.AckROC、.AlarmAckModel

論理値アラームの確認

指定したタグ変数の論理値アラームまたは指定したアラーム グループのすべての論理値アラームを確認するために .AckDsc ドットフィールドを使用するスクリプトを作成できます。

.AckDsc ドットフィールド AckDsc どっとふぃーるど

指定したタグ変数の論理値アラームか、指定したアラーム グループのすべての論理値アラームを確認します。

使用法

```
TagName。AckDsc=1;
```

パラメータ

TagName
論理型タグ変数に割り当てられている名前か、アラーム グループの名前です。

備考

指定したタグ変数またはアラーム グループに関連付けられるアクティブな論理値アラームを確認するには、1 に設定します。指定したタグ変数がアラーム グループである場合、指定したグループ内のタグ変数に関連付けられるすべての未確認の論理値アラームが確認されます。指定したタグ変数がアラーム グループ以外の場合、そのタグ変数に関連付けられている未確認の論理値アラームだけが確認されます。このドットフィールドに 1 以外の値を設定しても意味がありません。

データ型

論理型（読み取り/書き込み）

有効値

0 または 1。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のステートメントでは、Tag1 にアクティブなアラームが関連付けられているかどうかを検証します。

```
IF (Tag1.AlarmDsc == 1) THEN
   MyAlarmMessage="現在ポンプ ステーションにアラームが発生しています!";
ENDIF;
```

このドットフィールドは、.Ack ドットフィールドや .UnAck ドットフィールドにはリンクされていません。このため、アクティブなアラームが確認された場合も、このドットフィールドの値は 1 のまま維持されます。

参照項目

.Alarm、.AlarmDSC、.Ack、.UnAck、.AckDev、.AckDSC、.AckROC、.AckValue、.AlarmAckModel

偏差アラームの確認

指定したローカル タグ変数の小偏差アラームまたは大偏差アラーム、または指定したアラーム グループの偏差アラームを確認するために .AckDev ドットフィールドを使用するスクリプトを作成できます。

.AckDev ドットフィールド AckDev どっとふぃーるど

指定したローカル タグ変数の小偏差アラームまたは大偏差アラーム、または指定したアラーム グループのすべての偏差アラームを確認します。

カテゴリ

アラーム

使用法

```
TagName。AckDev=1;
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

指定したタグ変数またはアラーム グループに関連付けられるすべての未処理の偏差アラームを確認するには、このドットフィールドの値を 1 に設定します。指定したタグ変数がアラーム グループである場合、指定したグループ内のタグ変数に関連付けられるすべての未確認の偏差アラームが確認されます。このドットフィールドに 1 以外の値を設定しても意味がありません。

データ型

論理型（読み取り/書き込み）

有効値

1

例

以下のステートメントでは、Tag1 タグ変数に関連付けられている偏差アラームが確認されます。

```
Tag1。AckDev=1;
```

次の例は、PumpStation という名前のアラーム グループ内のすべての未確認の偏差アラームを確認するために使用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
PumpStation.AckDev=1;
```

参照項目

.Alarm、.AlarmDev、.Ack、.UnAck、.AckDSC、.AckROC、.AckValue、.AlarmAckModel

変化率アラームの確認

指定したローカル タグ変数の変更率アラーム、または指定したアラーム グループの変更率アラームを確認するために .AckROC ドットフィールドを使用するスクリプトを作成できます。

.AckROC ドットフィールド.AckROC どっとふぃーるど

指定したローカル タグ変数の変化率アラームまたは指定したアラーム グループのすべての変化率アラームを確認します。

使用法

```
TagName。AckROC=1;
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数に割り当てられた名前、またはアラーム グループの名前

備考

指定したタグ変数またはアラーム グループに関連付けられるすべての未処理の変化率アラームを確認するには、このドットフィールドの値を 1 に設定します。指定したタグ変数がアラーム グループの場合、指定したグループ内のタグ変数に関連付けられるすべての未確認の変化率アラームが確認されます。指定したタグ変数がアラーム グループ以外の場合は、そのタグ変数に関連付けられている未確認の変化率アラームだけが確認されます。このドットフィールドに 1 以外の値を設定しても意味がありません。

データ型

論理型（読み取り/書き込み）

有効値

1

例

以下のステートメントでは、Tag1 という名前のタグ変数に関連付けられている変化率アラームが確認されます。

```
Tag1。AckROC=1;
```

この次の例では、PumpStation という名前のアラーム グループ内のすべての未確認の変化率アラームが確認されます。

```
PumpStation.AckROC=1;
```

参照項目

.Alarm、.AlarmROC、.Ack、.UnAck、.AckDev、.AckDSC、.AckValue、.AlarmAckModel

アラームを確認するためのスクリプト関数の使用

Ack() スクリプト関数を使用して、タグ変数またはグループのすべてのアラームを確認できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Alarm Viewer コントロールを使用している場合は、メソッドを使用してアラームを確認できます。詳細については、「アラームを確認する」を参照してください。

分散アラーム表示オブジェクトを使用している場合は、スクリプト関数を使用してアラームを確認します。詳細については、「アラームを確認する」を参照してください。

Ack() 関数 Ack()かんすう

未確認の InTouch アラームを確認します。

カテゴリ

アラーム

構文

```
Ack TagName;
```

引数

TagName
任意の InTouch タグ変数、アラーム グループ、またはグループ変数です。

備考

この関数は、タグ変数またはアラーム グループに適用できます。

例

以下のステートメントは押しボタンで使用して、未確認のアラームを確認するために使用されます。

```
Ack $System; {すべてのアラーム}
Ack Tagname;
Ack GroupName;
```

参照項目

almAckAll()、almAckGroup()、almAckTag()、almAckDisplay()、almAckRecent()、almAckPriority()、almAckSelect()、almAckSelectedGroup()、almAckSelectedPriority()、almAckSelectedTag()

タグ値が平常に復帰するときの自動確認の使用

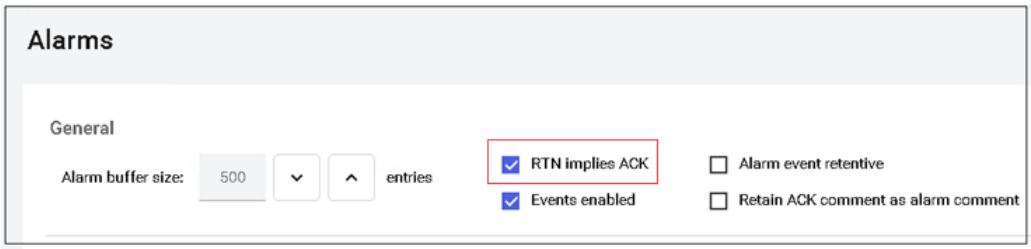
InTouch HMI では、アラーム状態のタグ値が平常状態に復帰する際に、アラームを自動的に確認できます。このオプションは、拡張サマリ アラームには適用されません。

復帰時のアラーム確認を設定

1. WindowMaker でアプリケーションを開きます。
2. ［ファイル］メニューでで、［設定］をポイントし、［アラーム］を選択します。

   ［アラーム］設定画面が表示されます。
3. ［全般］領域で、［アラーム復旧時に確認する］を選択すると、値が平常状態に復帰（RTN）したアラームが InTouch HMI によって自動的に確認されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

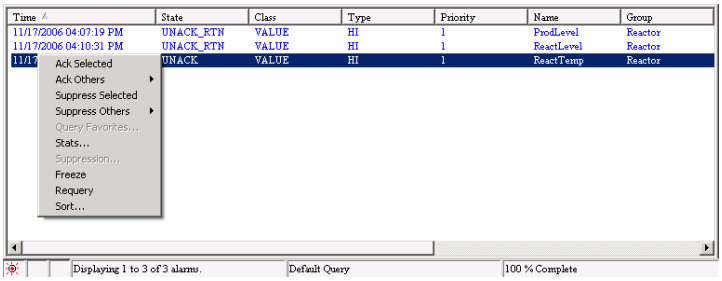


1. ［アラーム復帰時に確認する］チェック ボックスをオンにします。
2. ［OK］ を選択します。

アラームを確認するためのアラーム クライアントの使用

オペレータは Alarm Viewer コントロールに表示されるアラームを WindowViewer から確認します。

表示の［状況］カラムには、選択したアラーム レコードの現在の確認状況が表示されます。また、レコードのテキストには色が付けられており、確認状況を示しています。



確認するアラームは、表示から削除されます。

すべてのアラームを確認する

1. 表示を右クリックし、［その他を確認］をポイントし、適切なコマンドを選択します。
   - 現在のアラームをすべて確認、［全てを確認］を選択します。
   - 表示されているすべてのアラームを確認、［表示部分を確認］を選択します。

   ［確認コメント］ダイアログ ボックスが表示されます。
2. オプションの確認コメントを入力して、［OK］を選択します。

選択したアラームを確認できます。

1. 1 つまたは複数のアラームを選択します。
2. 右クリックして、［選択アラームを確認］を選択します。［確認コメント］ダイアログ ボックスが表示されます。
3. オプションの確認コメントを入力して、［OK］を選択します。

グループ、タグ変数、または優先度を確認

1. アラームを選択します。
2. 表示を右クリックし、［その他を確認］をポイントし、適切なコマンドを選択します。
   - 選択したアラーム グループに属するすべてのアラームを確認、［選択グループを確認］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 選択したアラームと同じ名前を持つすべてのタグ変数のアラームを確認、[選択タグ変数を確認] を選択します。
- 選択したアラームと同じ優先度を持つすべてのアラームを確認、[選択優先度を確認] を選択します。

[**確認コメント**] ダイアログ ボックスが表示されます。

3. オプションの確認コメントを入力して、[OK] を選択します。

アラーム コメントと確認コメントの使用

コメントには、アラーム コメントと確認コメントという 2 種類のコメントがあります。

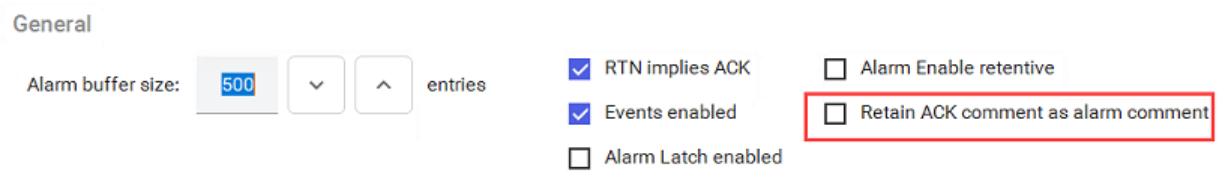
- アラーム コメントは、新しいアラーム インスタンスが発生したときに設定されます。アラーム コメントに使用される .AlarmComment ドットフィールドは、InTouc スクリプトで設定したり、読み取ったりできます。このコメントのデフォルト値は、タグ名ディクショナリのタグ定義で指定します。アラーム コメントの最大文字数は 131 文字です。
- 確認コメントは、アラームを確認するときにオペレータによって提供されます。

確認コメントを使用すると、タグ データベースのアラーム コメントを更新できます。

アラーム確認コメントで .AlarmComment ドットフィールドを更新することを許可

1. WindowMaker でアプリケーションを開きます。
2. [**ファイル**] メニューでで、[**設定**] をポイントし、[**アラーム**] を選択します。

   [**アラーム**] 画面が表示されます。



3. アラーム確認で入力されたコメントでタグの .AlarmComment ドットフィールドとタグ名ディクショナリを更新するには、[**確認コメントをアラーム コメントとして保持**] チェック ボックスをオンにします。

   このチェック ボックスをオンにしない場合、確認コメントは確認したアラームと共に表示されますが（データベース、印刷、および画面）、.AlarmComment ドットフィールドは変更されません。

4. [**OK**] を選択します。

ランタイム時のアラームの解除

アラームが発生すると、ランタイム オペレータ（またはシステム）がアラームを確認して、アラームを認識したことを示します。Galaxy に対してラッチ機能が有効になっている場合、標準値に戻った確認済みアラームは引き続き表示されます。ラッチが有効な場合、通常状態に戻った確認済みアラームは、LATCHED アラーム状態になります。

LATCHED アラームは、アラームが発生したことを示すために現在のアラーム モードで表示されることがあります。次の場合、アラームは LATCHED 状態になります。

- アラームが UNACK_RTN 状態から確認された場合。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

または

- アラームが ACK 状態から通常状態に戻った場合。

LATCHED 状態を表示するには、グローバル設定で LATCHED 状態を有効にする必要があります。詳細については、 *AVEVA OMI* ヘルプの「ラッチステートを有効にする」を参照してください。

LATCHED アラームを解除して、アラーム クライアント コントロール グリッドの現在のモードから LATCHED アラームを削除できます。解除された LATCHED アラームは、アラーム クライアント コントロールの最近のモードで表示されます。

選択したアラームを解除するには

1. **LATCHED** 状態のアラームを 1 つまたは複数選択します。
2. グリッドの任意の場所を右クリックして、［**選択項目を閉じる**］を選択します。

選択したアラームがグリッドから削除されます。

その他のアラームを解除するには

1. **LATCHED** 状態のアラームを 1 つまたは複数選択します。
2. グリッドの任意の場所を右クリックし、［**その他を閉じる**］をポイントして次のいずれかのオプションを選択します。
   - ［**すべて閉じる**］をクリックするとアラーム状態のすべてのアラームが解除されます。
   - ［**表示項目を閉じる**］をクリックすると表示されているすべてのアラームが解除されます。
   - ［**選択したグループを閉じる**］をクリックすると、選択した 1 つまたは複数のアクティブなアラームと同じプロバイダ名とグループ名のアラームが解除されます。
   - ［**選択したタグを閉じる**］をクリックすると、選択した 1 つまたは複数のアクティブなアラームと同じプロバイダ名、グループ名、およびタグ名のアラームが解除されます。

関連するアラームがグリッドから削除されます。

ドットフィールドを使用したアラームの解除

ドットフィールドを使用して LATCHED 状態のすべてのアラームを解除するスクリプトを作成できます。

カテゴリ

アラーム

使用法

```
TagName.Dismiss=1;
```

パラメータ

TagName
論理型、整数型、実数型、間接論理型、およびアナログ型のタグ、またはアラーム グループ

備考

指定したタグまたはアラーム グループに関連付けられるすべての未処理のアラームを解除するには、このドットフィールドの値を 1 に設定します。指定したタグがアラーム グループである場合、指定したグループ内のタグに関連付けられたすべての LATCHED アラームが解除されます。指定したタグがアラーム グループ以外のタイプの場合、そのタグに関連付けられている LATCHED アラームだけが解除されます。.Dismiss ドットフィールドを 1 以外の値にしても何の意味もありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データ型

論理型（読み取り/書き込み）

有効値

1

例

以下のステートメントでは、Tag1 タグに関連付けられているアラームが解除されます。

```
Tag1.Dismiss=1;
```

次の例は、PumpStation アラーム グループ内のすべての LATCHED アラームを解除するために使用されます。

```
PumpStation.Dismiss=1;
```

ランタイム時のタグとグループのアラーム プロパティの制御

アラーム ドットフィールドを使用して、アラーム条件を動的に管理できます。これらの多くのドットフィールドは、I/O、式、およびスクリプトを使用してアクセスできます。I/O アクセスを使用して、Excel やリモート ノードで実行している WindowViewer などの他の Windows アプリケーションを使用して、特定のタグのアラーム情報を監視したり、制御したりできます。

タグに関連付けられているドットフィールドにアクセスするには、以下の構文を使用します。

tag.dotfield

たとえば、Analog_tag という名前のタグの HiHi アラームしきい値のランタイム変更を許可するには、ボタンへのアナログ ユーザー入力タッチ リンクを作成して、そのリンクのダイアログ ボックスで Analog_tag.HiHiLimit を式として入力します。ランタイム時、オペレータはそのボタンを選択するだけで、Analog_tag に割り当てられた HiHi アラームしきい値に新しい値を入力できます。

以下の表では、各アラーム ドットフィールドの概要を説明しています。

ドットフィールド	説明
.Ack	タグとアラーム グループのアラーム確認状況を監視または制御します。 .Ack には、.UnAck と呼ばれる逆タグドットフィールドがあります。未確認アラームが発生すると、.UnAck は 1 に設定されます。.UnAck はアニメーション リンクや条件スクリプトで使用して、未確認アラームのアナンシエーターをトリガできます。
.AckDev	アナログ型タグまたはアラーム グループでアクティブな偏差型アラームのアラーム確認状況を監視または制御します。
.AckDsc	論理型タグの現在の確認状況を監視または制御します。
.AckROC	タグでアクティブな変化率タイプのアラームのアラーム確認状況を監視または制御します。
.AckValue	タグでアクティブな値型アラームのアラーム確認状況を監視または制御します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	説明
.Alarm	アラーム条件が存在する場合に通知します。
.AlarmAckModel	タグに関連付けられている確認モデルを次のように監視または制御します。 0 = 条件（デフォルト） 1 = イベント指向 2 = 拡張サマリ アラームを持つ論理型またはアナログ型タグに適用されます。読み取り専用ですが、WindowMaker で設定できます。
.AlarmDev	偏差アラームの存在を通知します。
.AlarmDevCount	指定されたタグまたはアラーム グループでアクティブな偏差アラームの合計数を追跡します。
.AlarmDevDeadband	大偏差アラームおよび小偏差アラームの偏差率デッドバンドを監視または制御します。
.AlarmDevUnAckCount	指定されたタグまたはアラーム グループでアクティブな未確認の偏差アラームの合計数を追跡します。
.AlarmDisabled	イベントやアラームを有効または無効にします。アラームを持つ論理型タグとアナログ型タグ、またはアラーム グループに適用されます。
.AlarmDsc	論理型アラーム条件が現在アクティブであることを示します。
.AlarmDscCount	指定されたタグまたはアラーム グループでアクティブな論理型アラームの合計数を追跡します。
.AlarmDscDisabled	タグが論理型アラームを生成できるかどうかを示します。 **注意 :** このドットフィールドは論理型タグに対しては .AlarmDisabled ドットフィールドと同じです。
.AlarmDscEnabled	タグが論理型アラームを生成できるかどうかを示します。 **注意 :** このドットフィールドは論理型タグに対しては .AlarmEnabled ドットフィールドと同じです。
.AlarmDscInhibitor ど	このタグの論理型アラーム（存在する場合）に割り当てられている抑止タグ名を返します。
.AlarmDscUnAckCount	指定されたタグまたはアラーム グループでアクティブな未確認の論理型アラームの合計数を追跡します。
.AlarmEnabled	イベントやアラームを有効または無効にします。
.AlarmHiDisabled	アラームを持つアナログ型タグの Hi 限界値を有効または無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	説明
.AlarmHiHiDisabled	アラームを持つアナログ型タグの HiHi 限界値を有効または無効にします。
.AlarmHiHiEnabled	アラームを持つアナログ型タグの HiHi 限界値を有効または無効にします。
.AlarmHiHiInhibitor	HiHi 限界値に対する抑止タグ リファレンスを返します。アラームを持つアナログ型タグに適用されます。読み取り専用ですが、WindowMaker で設定できます。
をクリックします。 **AlarmHiInhibitor**	Hi 限界値に対する抑止タグ リファレンスを返します。アラームを持つアナログ型タグに適用されます。 読み取り専用ですが、WindowMaker で設定できます。
.AlarmLoDisabled	アラームを持つアナログ型タグの Lo 限界値を有効または無効にします。
.AlarmLoEnabled	アラームを持つアナログ型タグの Lo 限界値を有効または無効にします。
をクリックします。 **AlarmLoInhibitor**	Lo 限界値に対する抑止タグ リファレンスを返します。アラームを持つアナログ型タグに適用されます。 読み取り専用ですが、WindowMaker で設定できます。
.AlarmLoLoDisabled	アラームを持つアナログ型タグの LoLo 限界値を有効または無効にします。
.AlarmLoLoEnabled	アラームを持つアナログ型タグの LoLo 限界値を有効または無効にします。
をクリックします。 **AlarmLoLoInhibitor**	LoLo 限界値に対する抑止タグ リファレンスを返します。アラームを持つアナログ型タグに適用されます。 読み取り専用ですが、WindowMaker で設定できます。
.AlarmMajDevDisabled	アラームを持つアナログ型タグの大偏差限界を有効または無効にします。
.AlarmMajDevEnabled	アラームを持つアナログ型タグの大偏差限界を有効または無効にします。
をクリックします。 **1AlarmMajDevInhibitor**	大偏差限界に対する抑止タグ リファレンスを返します。アラームを持つアナログ型タグに適用されます。 読み取り専用ですが、WindowMaker で設定できます。
.AlarmMinDevDisabled	アラームを持つアナログ型タグの小偏差限界を有効または無効にします。
.AlarmMinDevEnabled	アラームを持つアナログ型タグの小偏差限界を有効または無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	説明
をクリックします。**AlarmMinDevInhibitor**	小偏差限界に対する抑止タグ リファレンスを返します。アラームを持つアナログ型タグに適用されます。 読み取り専用ですが、WindowMaker で設定できます。
.AlarmROC	変化率アラームが存在することを通知します。
.AlarmROCCount	指定されたタグまたはアラーム グループでアクティブな変化率アラームの合計数を追跡します。
.AlarmROCDisabled	アラームを持つアナログ型タグの変化率の限界を有効または無効にします。
.AlarmROCEnabled	アラームを持つアナログ型タグの変化率の限界を有効または無効にします。
をクリックします。**.AlarmROCInhibitor**	変化率の限界に対する抑止タグ リファレンスを返します。アラームを持つアナログ型タグに適用されます。 読み取り専用ですが、WindowMaker で設定できます。
.AlarmROCUnAckCount	指定されたタグまたはアラーム グループで未確認の変化率アラームの合計数を追跡します。
.AlarmTotalCount	指定されたタグまたはアラーム グループでアクティブなアラームの合計数を追跡します。
.AlarmUnAckCount	任意のタグまたはアラーム グループでアクティブな未確認アラームの合計数を追跡します。
.AlarmUserDefNum1	読み取り/書き込みの実数値（浮動小数点）です。デフォルトは 0 で、値は設定されていません。アラームを持つ論理型タグやアナログ型タグ、またはアラーム グループに適用されます。 **注意：**このドットフィールドの値は、スクリプトや POKE などによって設定済みの場合のみ、アラームに付加されます。
.AlarmUserDefNum2	読み取り/書き込みの実数値（浮動小数点）です。デフォルトは 0 で、値は設定されていません。アラームを持つ論理型タグやアナログ型タグ、またはアラーム グループに適用されます。 **注意：**このドットフィールドの値は、スクリプトや POKE などによって設定済みの場合のみ、アラームに付加されます。
.AlarmUserDefNum2	読み取り/書き込みの論理型。対応するタグの .AlarmUserDefNum2 がスクリプトで定義されている場合に、TRUE に設定されます。タグの .AlarmUserDefNum2 の値の関連付けを解除するには、このドットフィールドを FALSE に設定します。デフォルトは FALSE です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	説明
.AlarmUserDefStr	読み取り/書き込みのテキスト文字列です。デフォルトは "" で、値は設定されていません。 アラームを持つ論理型タグやアナログ型タグ、またはアラーム グループに適用されます。 **注意：**このドットフィールドの値は、スクリプトや POKE などによって設定済みの場合のみ、アラームに付加されます。
.AlarmUserDefStrSet	読み取り/書き込みの論理型。対応するタグの .AlarmUserDefStr がスクリプトで定義されている場合に、TRUE に設定されます。タグの .AlarmUserDefStr の値の関連付けを解除するには、このドットフィールドを FALSE に設定します。デフォルトは FALSE です。
.AlarmValDeadban	アラーム値のデッドバンドの値を監視または制御します。
.AlarmValueCount	指定されたタグまたはアラーム グループでアクティブな値型アラームの合計数を追跡します。
.AlarmValueUnAckCount	指定されたタグまたはアラーム グループでアクティブな未確認の値型アラームの合計数を追跡します。
.DevTarget	大偏差アラームおよび小偏差アラームのターゲットを監視または制御します。
.HiLimit、.HiHiLimit、.LoLimit、.LoLoLimit	値型アラーム チェックのしきい値を監視または制御する読み取り/書き込みのアナログ型タグ名ドットフィールドです。これらのフィールドは、整数型および実数型タグに対してのみ有効です。
.HiStatus、.HiHiStatus、.LoStatus、.LoLoStatus	指定したタイプのアラームが存在するかどうかを決定する読み取り専用の論理型ドットフィールドです。これらのフィールドは、整数型および実数型のタグに対してのみ有効です。
.MajorDevPct	アラーム チェックのための大偏差率を監視または制御する読み取り/書き込みの整数型ドットフィールドです。
.MajorDevStatus	指定したタグに対して大偏差アラームが存在するかどうかを判断する読み取り専用の論理型ドットフィールドです。
.MinorDevPct	アラーム チェックのための小偏差率を監視または制御する読み取り/書き込みの整数型ドットフィールドです。
.MinorDevStatus	指定したタグに対して小偏差アラームが存在するかどうかを判断する読み取り専用の論理型ドットフィールドです。
.Name	タグの実際の名前を表示する読み取り/書き込みのメッセージ型ドットフィールドです。たとえば、グループ変数がポイントしているアラーム グループの名前、または TagID タグの名前を判断する場合に使用できます。また、グループ変数がポイントしているアラーム グループを変更するために書き込むこともできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	説明
.Normal	指定したタグに対してアラームが存在しない場合、読み取り専用の論理型ドットフィールドは 1 に等しくなります。このドットフィールドは、普通のタグだけでなく、アラーム グループやグループ変数に対しても有効です。
.ROCPct	アラーム チェックの変化率を監視または制御する読み取り/書き込みのドットフィールドです。
.ROCStatus	指定したタグに対して変化率アラームが存在するかどうかを判断する読み取り専用の論理型ドットフィールドです。

タグ変数またはアラーム グループがアラーム状況にあるかどうかの確認

実行中のアプリケーションのアラーム状況を確認するには、以下のドットフィールドとシステム タグ変数を使用します。新しいアラームが発生したか、タグ変数またはアラーム グループがアラーム状況と正常状況のいずれにあるかを確認できます。また、論理値、偏差、変化率の各アラーム状況を確認することもできます。

- $NewAlarm システム タグ変数
- $System システム タグ変数
- .Alarm ドットフィールド.Alarm どっとふぃーるど
- .Normal ドットフィールド
- .AlarmDsc ドットフィールド.AlarmDsc どっとふぃーるど
- .AlarmDev ドットフィールド.AlarmDev どっとふぃーるど
- .AlarmROC ドットフィールド.AlarmROC どっとふぃーるど
- .LoStatus ドットフィールド.LoStatus どっとふぃーるど
- .LoLoStatus ドットフィールド
- .HiStatus ドットフィールド.HiStatus どっとふぃーるど
- .HiHiStatus ドットフィールド.HiHiStatus どっとふぃーるど
- .MinorDevStatus ドットフィールド.MinorDevStatus どっとふぃーるど
- .MajorDevStatus ドットフィールド.MajorDevStatus どっとふぃーるど
- .ROCStatus ドットフィールド.ROCStatus どっとふぃーるど

$NewAlarm システム タグ変数

実行中のアプリケーションで新しいアラームが発生した場合 1 に設定します。**$NewAlarm** システム タグ変数は、リモート アラームに対しては設定されません。

構文

```
$NewAlarm=value;
```

データ型

論理型（読み取り/書き込み）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

0 または 1。

備考

$NewAlarm システム タグ変数をアプリケーション ウィンドウのアニメーション オブジェクトに関連付けます。たとえば、タグ変数の値を 0 にリセットして、アラームを確認するためにオペレータが選択する確認ボタンにタグ変数を関連付けます。また、**$NewAlarm** システム タグ変数を **PlaySound** ロジック関数にリンク付けて、アラームが発生したときに警告音を鳴らすこともできます。

例

ボタンをアラーム確認ウィンドウに追加して、オペレータがこのボタンを選択すると実行される以下のアクション スクリプトを添付します。

```
Ack $System;
$NewAlarm=0;
HideSelf;
```

オペレータがボタンを選択すると、すべてのアラームが確認され、$NewAlarm システム タグ変数は 0 にリセットされ、アラーム確認ウィンドウがビューから非表示となります。

$System システム タグ変数

デフォルトのアラーム グループです。

カテゴリ

アラーム

使用法

```
$System
```

備考

デフォルトでは、タグ変数はこのルート アラーム グループに割り当てられます。定義されたすべてのアラーム グループは、$System の下位に作成されます。

データ型

システム アラーム グループ

例

```
$System.Ack = 1;{すべてのアラームを確認します}
```

.Alarm ドットフィールド.Alarm どっとふぃーるど

指定したタグ変数またはアラーム グループが現在アラーム状況でない場合に 0 を返します。アラームが発生すると、**Alarm ドットフィールド**は 1 を返します。アラーム状況が存在しなくなるまで、この値は 1 のまま維持されます。The.**Alarm** ドットフィールドには、**Normal** という逆ドットフィールドがあります。

タグ変数にアラーム グループの名前を指定すると、そのグループに属する何らかのタグ変数がアラーム状況にある場合に **.Alarm** ドットフィールドは 1 を返します。

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

TagName。警報

パラメータ

TagName
論理型、整数型、実数型のタグ変数、間接論理型、アナログ型のタグ変数、またはアラーム グループタグ変数

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下のステートメントでは、タグ変数 **Tag1** に関連付けられたアラームがアクティブかどうかを検証します。

```
IF (Tag1.Alarm == 1) THEN
```

以下の例では、**PumpStation** アラーム グループにアクティブなアラームが存在する場合に、IF-THEN ステートメントの本文が処理されます。

```
IF(PumpStation.Alarm == 1)
   MyAlarmMessage="現在ポンプ ステーションにアラームが発生しています!";
ENDIF;
```

このドットフィールドは、**アック** またはドットフィールドや .UnAck ドットフィールドにはリンクされていません。**UnAck** ドットフィールドそのため、アクティブなアラームが確認された後も、**アラーム** の値は 1 のまま維持されます。

.Normal ドットフィールド.Normal どっとふぃーるど

指定したタグ変数がアラーム状況でない場合に 1 を返します。アラームが発生すると、**通常のドットフィールド** は 0 を返します。**.Normal** ドットフィールドには、**.Alarm**

カテゴリ

アラーム

構文

TagName。正常があります

パラメータ

TagName
論理型、整数型、実数型のタグ変数、間接論理型、アナログ型のタグ変数、またはアラーム グループタグ変数

データ型

論理型（読み取り専用）

有効値

0 または 1。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下の IF-THEN ステートメントは、**Tag1** タグ変数に関連付けられたアラームが存在しない場合に実行されます。**Tag1** に対して 1 つまたは複数のアラームがアクティブになると、「ELSE」の本文が実行されます。

```
IF (Tag1.Normal==1) THEN
   MyOperatorMessage="Tag1 に関連付けられているアラームはありません。";
ELSE
   MyOperatorMessage="Tag1 に 1 つまたは複数のアクティブなアラームがあります!";
ENDIF;
```

.AlarmDsc ドットフィールド.AlarmDsc どっとふぃーるど

指定したタグ変数またはアラーム グループにアラーム状況が存在するかどうかを示します。デフォルト値は 0 です。指定したタグ変数に論理値アラーム状況が存在する場合は 1 の値が設定されます。アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

指定したタグがアラーム グループの名前の場合、. そのグループ内の何らかのタグ変数がアクティブな論理型アラームである場合に **.AlarmDsc** ドットフィールドに 1 が設定されます。

カテゴリ

アラーム

使用法

```
TagName。AlarmDsc
```

パラメータ

TagName
任意の論理型タグ変数、間接論理型タグ変数、またはアラーム グループ

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下のステートメントでは、Tag1 にアクティブなアラームが関連付けられているかどうかを検証します。

```
IF (Tag1.AlarmDsc == 1) THEN
   MyAlarmMessage="現在ポンプ ステーションにアラームが発生しています!";
ENDIF;
```

このドットフィールドは、**.Ack** や **.UnAck** ドットフィールドにはリンクされていません。このため、アクティブなアラームが確認された後も、**.AlarmDsc** ドットフィールドの値は 1 のまま維持されます。

参照項目

.Ack、.UnAck、.Alarm、.AlarmDsc、.AckDsc

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.AlarmDev ドットフィールド.AlarmDev どっとふぃーるど

指定したタグ変数またはアラーム グループの偏差アラームがアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に偏差アラーム状況が存在する場合は 1 の値が設定されます。アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

タグ変数にアラーム グループの名前を指定すると、そのグループ内の何らかのタグ変数がアクティブなアラーム状況にある場合に **.AlarmDev** ドットフィールドに 1 が設定されます。

カテゴリ

アラーム

使用法

```
TagName。AlarmDev
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下のステートメントでは、タグ変数 Tag1 に関連付けられた偏差アラームがアクティブかどうかを検証します。

```
IF (Tag1.AlarmDev == 1) THEN
```

以下の例では、PumpStation アラーム グループにアクティブな偏差アラームが存在する場合に、IF-THEN ステートメントの本文が処理されます。

```
IF (PumpStation.AlarmDev == 1) THEN
   MyAlarmMessage="現在ポンプ ステーションにアラームが発生しています!";
ENDIF;
```

このドットフィールドは、.Ack や .UnAck ドットフィールドにはリンクされていません。このため、アクティブなアラームが確認された場合も、このドットフィールドの値は 1 のまま維持されます。

参照項目

.Ack、.UnAck、.Alarm、.AckDev

.AlarmROC ドットフィールド.AlarmROC どっとふぃーるど

指定したタグ変数またはアラーム グループの変化率アラーム状況がアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に変化率アラーム状況が存在する場合は 1 の値が設定されます。変化率アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

タグ変数にアラーム グループの名前を指定すると、そのグループ内の何らかのタグ変数が変化率アラーム状況にある場合に **.AlarmROC** ドットフィールドに 1 が設定されます。

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

```
TagName。AlarmROC
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下のステートメントは、タグ変数 Tag1 に関連付けられている変化率アラームがアクティブかどうかを検証します。

```
IF (Tag1.AlarmROC == 1) THEN
```

PumpStation というアラーム グループ内にアクティブな変化率アラームが存在する場合、この IF-THEN ステートメントの本文が処理されます。

```
IF (PumpStation.AlarmROC == 1) THEN
   MyAlarmMessage="現在ポンプ ステーションにアラームが発生しています!";
ENDIF;
```

このドットフィールドは、.Ack ドットフィールドや .UnAck ドットフィールドにはリンクされていません。このため、アクティブな変化率アラームが確認された後も、このドットフィールドの値は 1 のまま維持されます。

参照項目

.Ack、.AckROC、.Alarm、.AlarmROCEnabled、.AlarmROCDisabled

.LoStatus ドットフィールド.LoStatus どっとふぃーるど

指定したタグ変数またはアラーム グループの Lo アラーム状況がアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に Lo アラーム状況が存在する場合は 1 の値が設定されます。Lo アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

このドットフィールドは、特定のタグ変数が指定のアラーム状況にあるかを調べるために、**.Alarm** や **.Ack** ドットフィールドと組み合わせてよく使用されます。

カテゴリ

アラーム

使用法

```
TagName。LoStatus
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

0 または 1。

例

以下の IF-THEN ステートメントは、タグ変数 MyTag の .LoStatus（Lo アラーム状況）が 1 の場合に実行されます。

```
IF (MyTag.LoStatus == 1) THEN
   OperatorMessage=" MyTag で Lo アラームが発生しました";
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.LoLimit、.LoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled、.AlarmLoEnabled、.AlarmLoInhibitor

.LoLoStatus ドットフィールド.LoLoStatus どっとふぃーるど

指定したタグ変数またはアラーム グループの LoLo アラーム状況がアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に LoLo アラーム状況が存在する場合は 1 の値が設定されます。LoLo アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

このドットフィールドは、システム内の特定のタグ変数で発生したアラーム状況の正確な内容を調べるために、**.Alarm** や **.Ack** フィールドと組み合わせてよく使用されます。

カテゴリ

アラーム

使用法

```
TagName.LoLoStatus
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下の IF-THEN ステートメントは、タグ変数 MyTag の .LoLoStatus（LoLo アラームしきい値）が 1 の場合に実行されます。

```
IF (MyTag.LoLoStatus == 1) THEN
   OperatorMessage=" MyTag で LoLo アラームが発生しました";
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.LoLoLimit、.LoLoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled、.AlarmLoLoEnabled、.AlarmLoLoInhibitor

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.HiStatus ドットフィールド.HiStatus どっとふぃーるど

指定したタグ変数またはアラーム グループの Hi アラーム状況がアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に Hi アラーム状況が存在する場合は 1 の値が設定されます。Hi アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

このドットフィールドは、タグ変数が指定のアラーム状況にあるかを調べるために、**.Alarm** や **.Ack** ドットフィールドと組み合わせてよく使用されます。

カテゴリ

アラーム

使用法

```
TagName.HiStatus
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下のスクリプトは、MotorAmps タグ変数が上限アラーム状況になった場合に別のスクリプトを呼び出して、ポンプ モータの出力を停止します。

```
IF (MotorAmps.HiStatus == 1) THEN
   CALL PumpShutdown( );
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.HiLimit、.HiSet、.AlarmDisabled、.AlarmEnabled、.AlarmHiDisabled、.AlarmHiEnabled、.AlarmHiInhibitor

.HiHiStatus ドットフィールド.HiHiStatus どっとふぃーるど

指定したタグ変数またはアラーム グループの HiHi アラーム状況がアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に HiHi アラーム状況が存在する場合は 1 の値が設定されます。HiHi アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

このドットフィールドは、タグ変数が指定のアラーム状況にあるかを調べるために、**.Alarm** や **.Ack** ドットフィールドと組み合わせてよく使用されます。

カテゴリ

アラーム

使用法

```
TagName.HiHiStatus
```

パラメータ

TagName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下の IF-THEN ステートメントは、タグ変数 MyTag の .HiHiStatus（HiHi アラーム）が 1 の場合に実行されます。

```
IF (MyTag.HiHiStatus == 1) THEN
   OperatorMessage=" MyTag で HiHi アラームが発生しました";
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.HiHiLimit、.HiHiSet、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

.MinorDevStatus ドットフィールド.MinorDevStatus どっとふぃーるど

指定したタグ変数またはアラーム グループの小偏差アラームがアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に小偏差アラーム状況が存在する場合は 1 の値が設定されます。小偏差アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

このドットフィールドは、タグ変数が指定のアラーム状況にあるかを調べるために、**.Alarm** や **.Ack** ドットフィールドと組み合わせてよく使用されます。

カテゴリ

アラーム

使用法

```
TagName.MinorDevStatus
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下の IF-THEN ステートメントは、タグ変数 MyTag の .MinorDevStatus（小偏差アラーム）が 1 の場合に実行されます。

```
IF (MyTag.MinorDevStatus == 1) THEN
   OperatorMessage=" MyTag で小偏差アラームが発生しました";
ENDIF;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.AckDev、.AlarmDev、.AlarmMinDevDisabled、.AlarmMinDevEnabled、.AlarmMinDevInhibitor、.MinorDevPct、.MajorDevStatus

.MajorDevStatus ドットフィールド.MajorDevStatus どっとふぃーるど

指定したタグ変数またはアラーム グループの大偏差アラームがアクティブになったことを示します。デフォルト値は 0 です。大偏差アラーム状況が存在する場合は、指定したドットフィールドに 1 が設定されます。大偏差アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

このドットフィールドは、タグ変数が指定のアラーム状況にあるかを調べるために、**.Alarm** や **.Ack** ドットフィールドと組み合わせてよく使用されます。

カテゴリ

アラーム

使用法

```
TagName.MajorDevStatus
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下の IF-THEN ステートメントは、タグ変数 MyTag の .MajorDevStatus（大偏差アラーム）が 1 の場合に実行されます。

```
IF (MyTag.MajorDevStatus == 1) THEN
   OperatorMessage=" MyTag で大偏差アラームが発生しました";
ENDIF;
```

参照項目

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、,AlarmMajDevInhibitor、.MajorDevPct、.MajorDevSet、.MinorDevStatus

.ROCStatus ドットフィールド.ROCStatus どっとふぃーるど

指定したタグ変数またはアラーム グループの変化率アラームがアクティブになったことを示します。デフォルト値は 0 です。指定したタグ変数に変化率アラーム状況が存在する場合は 1 の値が設定されます。変化率アラーム状況が存在しなくなるまでこの値は 1 のまま維持されます。

このドットフィールドは、タグ変数が指定のアラーム状況にあるかを調べるために、**.Alarm** や **.Ack** ドットフィールドと組み合わせてよく使用されます。

カテゴリ

アラーム

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

```
TagName.ROCStatus
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

以下の IF-THEN ステートメントは、タグ変数 MyTag の .ROCStatus（変化率アラーム）が 1 の場合に実行されます。

```
IF (MyTag.ROCStatus == 1) THEN
   OperatorMessage=" MyTag で変化率アラームが発生しました";
ENDIF;
```

参照項目

.ROCPct、.ROCSet

InTouch 7.1 の動作にアラーム状況を戻す

InTouch 7.11 移行では、HiHi、LoLo、MinDev（小偏差）、または MajDev（大偏差）アラームの発生は、それぞれ Hi、Lo、MajDev、または MinDev アラームの In Alarm、ACK、および $NewAlarm 状態にリセットされます。InTouch 7.1 およびそれ以前のバージョンの場合、これらのアラーム状況は HiHi、LoLo、MinDev、および MajDev アラームの発生による影響を受けません。

その他のアラームの発生に関係なく In Alarm、ACK、および $NewAlarm 状態を維持するために Hi、Lo、MajDev、または MinDev アラームを予期する従来のアプリケーションをサポートするには、InTouch.ini ファイルに IT71StatusFlag パラメータを含めることができます。

InTouch 7.1 およびそれ以前のバージョンと同様に動作するようにアラーム処理を強制するには

- 次の行を InTouch.ini ファイルに入力します。
  ```
  IT71StatusFlags=1
  ```

InTouch 7.11 およびそれ以降のデフォルト動作に戻すには

- 次の行を InTouch.ini ファイルに入力します。
  ```
  IT71StatusFlags=0
  ```

InTouch.ini ファイルを編集して、このパラメータおよびその他の InTouch の動作パラメータを設定する方法の詳細については、『*AVEVA™ InTouch HMI アプリケーション開発ガイド*』を参照してください。

タグ変数のアラームしきい値設定の確認

アプリケーションの実行中に、タグ変数にアラームしきい値が設定されているかどうかを調べるには、以下のドットフィールドを使用します。

- .LoLoSet ドットフィールド.LoLoSet どっとふぃーるど

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- .LoSet ドットフィールド.LoSet どっとふぃーるど
- .HiSet ドットフィールド.HiSet どっとふぃーるど
- .HiHiSet ドットフィールド
- .MinorDevSet ドットフィールド.MinorDevSet どっとふぃーるど
- .MajorDevSet ドットフィールド.MajorDevSet どっとふぃーるど
- .ROCSet ドットフィールド.ROCSet どっとふぃーるど

.LoLoSet ドットフィールド.LoLoSet どっとふぃーるど

整数型または実数型タグ変数に LoLo アラームしきい値が設定されているかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName.LoLoSet
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

THEN ブロックは、タグ変数 MyTag に対して LoLo アラームしきい値が設定されている場合に実行されます。

```
IF (MyTag.LoLoSet== 1) THEN
   MsgTag="MyTag には LoLo アラームしきい値が設定されています";
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.LoLoStatus、.LoLoLimit、.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled、.AlarmLoLoEnabled、.AlarmLoLoInhibitor

.LoSet ドットフィールド.LoSet どっとふぃーるど

整数型または実数型タグ変数に Lo アラームしきい値が設定されているかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName。[LoSet]
```

パラメータ

TagName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

THEN ブロックは、タグ変数 MyTag に対して Lo アラームしきい値が設定されている場合に実行されます。

```
IF (MyTag.LoSet== 1) THEN
   MsgTag="MyTag には Lo アラームしきい値が設定されています";
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.LoStatus、.LoLimit、.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled、.AlarmLoEnabled、.AlarmLoInhibitor

.HiSet ドットフィールド.HiSet どっとふぃーるど

整数型または実数型タグ変数に Hi アラームしきい値が設定されているかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName.HiSet
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

THEN ブロックは、タグ変数 MyTag に対して Hi アラームしきい値が設定されている場合に実行されます。

```
IF (MyTag.HiSet== 1) THEN
   MsgTag="MyTag には Hi アラームしきい値が設定されています";
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.HiHiStatus、.HiHiLimit、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.HiHiSet ドットフィールド

整数型または実数型タグ変数に HiHi アラームしきい値が設定されているかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName.HiHiSet
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

THEN ブロックは、タグ変数 MyTag に対して HiHi アラームしきい値が設定されている場合に実行されます。

```
IF (MyTag.HiHiSet== 1) THEN
   MsgTag="MyTag には HiHi しきい値が設定されています";
ENDIF;
```

参照項目

.Alarm、.AlarmValue、.Ack、.HiHiStatus、.HiHiLimit、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

.MinorDevSet ドットフィールド.MinorDevSet どっとふぃーるど

整数型または実数型タグ変数に小偏差アラームしきい値が設定されているかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName.MinorDevSet
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

THEN ブロックは、タグ変数 MyTag に対して小偏差パーセント値のアラームしきい値が設定されている場合に実行されます。

```
IF (MyTag.MinorDevSet== 1) THEN
   MsgTag="MyTag には小偏差しきい値が設定されています";
ENDIF;
```

参照項目

.AckDev、.AlarmDev、.AlarmMinDevDisabled、.AlarmMinDevEnabled、.AlarmMinDevInhibitor、.MinorDevPct、.MinorDevStatus

.MajorDevSet ドットフィールド.MajorDevSet どっとふぃーるど

整数型または実数型タグ変数に大偏差アラームしきい値が設定されているかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName.MajorDevSet
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

THEN ブロックは、タグ変数 MyTag に対して大偏差パーセント値のアラームしきい値が設定されている場合に実行されます。

```
IF (MyTag.MajorDevSet== 1) THEN
   MsgTag="MyTag には大偏差しきい値が設定されています";
ENDIF;
```

参照項目

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、,AlarmMajDevInhibitor、.MajorDevPct、.MajorDevStatus

.ROCSet ドットフィールド.ROCSet どっとふぃーるど

整数型または実数型タグ変数に変化率アラームしきい値が設定されているかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName。ROCSet
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

データ型

論理型（読み取り専用）

有効値

0 または 1。

例

THEN ブロックは、タグ変数 MyTag に対して変化率アラームしきい値が設定されている場合に実行されます。

```
IF (MyTag.ROCSet == 1) THEN
   MsgTag="MyTag には変化率しきい値が設定されています";
ENDIF;
```

参照項目

.Alarm、.Ack、.LoLimit、.LoLoLimit、.HiHiLimit、.HiLimit、.HiSet、.LoSet、LoLoSet、.HiStatus、.HiHiStatus、ROCPct、ROCStatus

タグ変数またはアラームグループのアラームの有効化と無効化

InTouch HMI ではドットフィールドを使用して、タグ変数に設定されているアラームをアプリケーションの実行中に有効または無効にできます。

すべてのアラームの有効化/無効化

.AlarmEnabled および **.AlarmDisabled** ドットフィールドにより、タグ変数またはアラーム グループのアラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反するアラーム状況が関連付けられます。**.AlarmEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループのアラームを有効にします。**.AlarmDisabled** の値を 1 にすると、タグ変数またはアラーム グループのアラームを無効にします。

いずれかのドットフィールドによってアラーム グループのアラームを有効化すると、そのグループに属するすべてのタグ変数のアラームが有効になります。いずれかのドットフィールドによってアラームを無効にすると、すべてのイベントとアラームは無視されます。この場合、アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmEnabled ドットフィールド.AlarmEnabled どっとふぃーるど

タグ変数またはアラーム グループに対するアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmEnabled
```

パラメータ

TagName
論理型、整数型、実数型、間接論理型、間接アナログ型のタグ変数、またはアラーム グループ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

.AlarmEnabled を 0 に設定すると、イベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

指定されたタグ変数がアラーム グループの場合、指定アラーム グループ内のタグ変数に関連付けられているすべてのアラームが有効になります。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下のステートメントは、タグ変数 Tag1 のアラームを無効にします。

```
Tag1.AlarmEnabled=0;
```

参照項目

.AlarmDisabled

.AlarmDisabled ドットフィールド.AlarmDisabled どっとふぃーるど

タグ変数またはアラーム グループに対するアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmDisabled
```

パラメータ

TagName
論理型、整数型、実数型、間接論理型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmDisabled を 1 に設定すると、イベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

指定されたタグ変数がアラーム グループの場合、指定アラーム グループ内のタグ変数に関連付けられているすべてのアラームが無効になります。

.AlarmEnabled ドットフィールドとは逆のドットフィールドです。

例

以下のステートメントは、タグ変数 Tag1 のアラームを有効にします。

```
Tag1.AlarmDisabled=0;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.AlarmEnabled

LoLo アラームの有効化または無効化

.AlarmLoLoEnabled ドットフィールドおよび **.AlarmLoLoDisabled** ドットフィールドにより、タグ変数またはアラーム グループのアラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する LoLo アラーム状況が関連付けられます。**.AlarmLoLoEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの LoLo アラームを有効にします。**.AlarmLoLoDisabled** の値を 1 にすると、タグ変数またはアラーム グループの LoLo アラームを無効にします。

いずれかのドットフィールドによってアラーム グループの LoLo アラームを有効化すると、そのグループに属するすべてのタグ変数のアラームが有効になります。いずれかのドットフィールドによって LoLo アラームを無効にすると、すべての LoLo アラームは無視されます。この場合、アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmLoLoEnabled ドットフィールド.AlarmLoLoEnabled どっとふぃーるど

LoLo 条件のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmLoLoEnabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmLoLoEnabled を 0 に設定すると、LoLo 条件のイベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下のステートメントは、タグ変数 Tag1 の LoLo アラームを無効にします。

```
Tag1.AlarmLoLoEnabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled

.AlarmLoLoDisabled ドットフィールド.AlarmLoLoDisabled どっとふぃーるど

LoLo 条件のイベントおよびアラームを有効または無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

アラーム

使用法

```
TagName.AlarmLoLoDisabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmLoLoDisabled を 1 に設定すると、LoLo 条件のイベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

データ型

論理型（読み取り/書き込み）

有効値

1 = アラームを無効化

0 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag2 の LoLo アラームを有効にします。

```
Tag2.AlarmLoLoDisabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmLoLoEnabled

Lo アラームの有効化または無効化

.AlarmLoEnabled ドットフィールドおよび **.AlarmLoDisabled** ドットフィールドにより、タグ変数またはアラーム グループの Lo アラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する Lo アラーム状況が関連付けられます。**.AlarmLoEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの Lo アラームを有効にします。**.AlarmLoDisabled** の値を 1 にすると、タグ変数またはアラーム グループの Lo アラームを無効にします。

いずれかのドットフィールドによってアラーム グループの Lo アラームを有効化すると、そのグループに属するすべてのタグ変数の Lo アラームが有効になります。いずれかのドットフィールドによって Lo アラームを無効にすると、すべての Lo アラームは無視されます。この場合、アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmLoEnabled ドットフィールド.AlarmLoEnabled どっとふぃーるど

Lo 条件のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmLoEnabled
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmLoEnabled を 0 に設定すると、Lo 条件のイベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag1 の Lo アラームを無効にします。

```
Tag1.AlarmLoEnabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled

.AlarmLoDisabled ドットフィールド.AlarmLoDisabled どっとふぃーるど

Lo 条件のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmLoDisabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmLoDisabled を 1 に設定すると、Lo 条件のイベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

データ型

論理型（読み取り/書き込み）

有効値

１= アラームを無効化

0 = アラームを有効化（デフォルト）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下の例は、タグ変数 Tag2 の Lo アラームを有効にします。

```
Tag2.AlarmLoDisabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、AlarmLoEnabled

Hi アラームの有効化または無効化

.AlarmHiEnabled ドットフィールドおよび **.AlarmHiDisabled** ドットフィールドにより、タグ変数またはアラーム グループの Hi アラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する Hi アラーム状況が関連付けられます。**.AlarmHiEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの Hi アラームを有効にします。**.AlarmHiDisabled** の値を 1 にすると、タグ変数またはアラーム グループのアラームを無効にします。

いずれかのドットフィールドによってアラーム グループの Hi アラームを有効化すると、そのグループに属するすべてのタグ変数の Hi アラームが有効になります。いずれかのドットフィールドによって Hi アラームを無効にすると、すべての Hi アラームは無視されます。この場合、Hi アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmHiEnabled ドットフィールド

Hi 条件のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmHiEnabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmHiEnabled を 0 に設定すると、Hi 条件のイベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmHiDisabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag1 の Hi アラームを無効にします。

```
Tag1.AlarmHiEnabled=0;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.AlarmHiDisabled、.AlarmEnabled

.AlarmHiDisabled ドットフィールド

Hi 条件のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
.AlarmHiDisabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmHiDisabled を 1 に設定すると、Hi 条件のイベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmHiEnabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

1 = アラームを無効化

0 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag2 の Hi アラームを有効にします。

```
Tag2.AlarmHiDisabled=0;
```

参照項目

.AlarmHiEnabled、.AlarmDisabled

HiHi アラームの有効化または無効化

.AlarmHiHiEnabled および **.AlarmHiHiDisabled** ドットフィールドにより、タグ変数またはアラーム グループの HiHi アラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する HiHi アラーム状況が関連付けられます。**.AlarmHiHiEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの HiHi アラームを有効にします。**.AlarmHiHiDisabled** の値を 1 にすると、タグ変数またはアラーム グループの HiHi アラームを無効にします。

いずれかのドットフィールドによってアラーム グループの HiHi アラームを有効化すると、そのグループに属するすべてのタグ変数の HiHi アラームが有効になります。いずれかのドットフィールドによって HiHi アラームを無効にすると、すべての HiHi アラームは無視されます。この場合、HiHi アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.AlarmHiHiEnabled ドットフィールド

HiHi 条件のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmHiHiEnabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmHiHiEnabled を 0 に設定すると、HiHi 条件のイベントとアラームはすべて無視されます。アラームメモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag1 の HiHi アラームを無効にします。

```
Tag1.AlarmHiHiEnabled=0;
```

参照項目

.AlarmHiHiDisabled、.AlarmEnabled

.AlarmHiHiDisabled ドットフィールド.AlarmHiHiDisabled どっとふぃーるど

HiHi 条件のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmHiHiDisabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmHiHiDisabled を 1 に設定すると、HiHi 条件のイベントとアラームはすべて無視されます。アラームメモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.AlarmHiHiEnabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

1 = アラームを無効化

0 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag2 の HiHi アラームを有効にします。

```
Tag2.AlarmHiHiDisabled=0;
```

参照項目

.AlarmHiHiEnabled、.AlarmDisabled

論理値アラームの有効化または無効化

.AlarmDscEnabled および **.AlarmDscDisabled** ドットフィールドにより、タグ変数またはアラーム グループの論理値アラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する論理値アラーム状況が関連付けられます。**.AlarmDscEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの論理値アラームを有効にします。**.AlarmDscDisabled** の値を 1 にすると、タグ変数またはアラーム グループの論理値アラームを無効にします。

いずれかのドットフィールドによってアラーム グループの論理値アラームを有効化すると、そのグループに属するすべてのタグ変数の論理値アラームが有効になります。いずれかのドットフィールドによって論理値アラームを無効にすると、すべての論理値アラームは無視されます。この場合、論理値アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmDscEnabled ドットフィールド.AlarmDscEnabled どっとふぃーるど

タグ変数が論理値アラームを生成できるかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName.AlarmDscEnabled
```

パラメータ

TagName
任意の論理型、間接論理型タグ変数またはアラーム グループ

備考

.AlarmDscEnabled を 0 に設定すると、論理値条件のアラームとイベントはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmDscDisabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag1 の論理値アラームを無効にします。

```
Tag1.AlarmDscEnabled=0;
```

参照項目

.AlarmDscDisabled

.AlarmDscDisabled ドットフィールド.AlarmDscDisabled どっとふぃーるど

タグが論理型アラームを生成できるかどうかを示します。

カテゴリ

アラーム

使用法

```
TagName.AlarmDscDisabled
```

パラメータ

TagName
任意の論理型、間接論理型タグ変数またはアラーム グループ

備考

.AlarmDscDisabled を 1 に設定すると、すべての論理型条件のアラームとイベントが無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmDscEnabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

1 = アラームを無効化

0 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag2 の論理値アラームを有効にします。

```
Tag2.AlarmDscDisabled=0;
```

小偏差アラームの有効化または無効化

.AlarmMinDevEnabled および **.AlarmMinDevDisabled** ドットフィールドにより、タグ変数またはアラーム グループの小偏差アラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する小偏差アラーム状況が関連付けられます。**.AlarmMinDevEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの小偏差アラームを有効にします。**.AlarmMinDevDisabled** の値を 1 にすると、タグ変数またはアラーム グループの小偏差アラームを無効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

いずれかのドットフィールドによってアラーム グループの小偏差アラームを有効化すると、そのグループに属するすべてのタグ変数の小偏差アラームが有効になります。いずれかのドットフィールドによって小偏差アラームを無効にすると、すべての小偏差アラームは無視されます。この場合、小偏差アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmMinDevEnabled ドットフィールド.AlarmMinDevEnabled どっとふぃーるど

小偏差のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmMinDevEnabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmMinDevEnabled を 0 に設定すると、小偏差のイベントおよびアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag1 の小偏差アラームを無効にします。

```
Tag1.AlarmMinDevEnabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmMinDevDisabled

.AlarmMinDevDisabled ドットフィールド.AlarmMinDevDisabled どっとふぃーるど

小偏差のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmMinDevDisabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

.AlarmMinDevDisabled を 1 に設定すると、小偏差のイベントおよびアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmMinDevEnabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

1 = アラームを無効化

0 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag2 の小偏差アラームを有効にします。

```
Tag2.AlarmMinDevDisabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmMinDevEnabled

大偏差アラームの有効化または無効化

.AlarmMajDevEnabled ドットフィールドおよび **.AlarmMajDevDisabled** ドットフィールドにより、タグ変数またはアラーム グループの大偏差アラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する大偏差アラーム状況が関連付けられます。**.AlarmMajDevEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの大偏差アラームを有効にします。**.AlarmMajDevDisabled** の値を 1 にすると、タグ変数またはアラーム グループの大偏差アラームを無効にします。

いずれかのドットフィールドによってアラーム グループの大偏差アラームを有効化すると、そのグループに属するすべてのタグ変数の大偏差アラームが有効になります。いずれかのドットフィールドによって大偏差アラームを無効にすると、すべての大偏差アラームは無視されます。この場合、大偏差アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmMajDevEnabled ドットフィールド

大偏差のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmMajDevEnabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

.AlarmMajDevEnabled を 0 に設定すると、大偏差のイベントとアラームはすべて無視されます。アラームメモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmMajDevDisabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag1 の大偏差アラームを無効にします。

```
Tag1.AlarmMajDevEnabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmMajDevDisabled

.AlarmMajDevDisabled ドットフィールド.AlarmMajDevDisabled どっとふぃーるど

大偏差のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmMajDevDisabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmMajDevDisabled を 1 に設定すると、大偏差のイベントとアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmMajDevEnabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

1 = アラームを無効化

0 = アラームを有効化（デフォルト）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下の例は、タグ変数 Tag2 の大偏差アラームを有効にします。

```
Tag2.AlarmMajDevDisabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、AlarmMajDevEnabled

変化率アラームの有効化／無効化

.AlarmROCEnabled および **.AlarmROCDisabled** ドットフィールドにより、タグ変数またはアラーム グループの変化率アラームを設定値に応じて有効または無効にできます。この 2 つのドットフィールドには、相反する変化率アラーム状況が関連付けられます。**.AlarmROCEnabled** の値を 1 に設定すると、タグ変数またはアラーム グループの変化率アラームを有効にします。**.AlarmROCDisabled** の値を 1 にすると、タグ変数またはアラーム グループの変化率アラームを無効にします。

いずれかのドットフィールドによってアラーム グループの変化率アラームを有効化すると、そのグループに属するすべてのタグ変数の変化率アラームが有効になります。いずれかのドットフィールドによって変化率アラームを無効にすると、すべての変化率アラームは無視されます。この場合、変化率アラームはアラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。

.AlarmROCEnabled ドットフィールド.AlarmROCEnabled どっとふぃーるど

変化率のイベントおよびアラームを有効または無効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmROCEnabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmROCEnabled を 0 に設定すると、変化率条件のイベントおよびアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmROCDisabled ドットフィールドとは逆のドットフィールドです。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

1 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag1 の変化率アラームを無効にします。

```
Tag1.AlarmROCEnabled=0;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmROCDisabled

.AlarmROCDisabled ドットフィールド.AlarmROCDisabled どっとふぃーるど

変化率のイベントおよびアラームを無効または有効にします。

カテゴリ

アラーム

使用法

```
TagName.AlarmROCDisabled
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

.AlarmROCDisabled を 1 に設定すると、変化率条件のイベントおよびアラームはすべて無視されます。アラーム メモリに保持されないだけでなく、ディスクにも書き込まれません。データの損失を避けるには、イベントやアラームをできるだけ有効にしてください。

.AlarmROCEnabled のプロパティとは逆のプロパティです。

データ型

論理型（読み取り/書き込み）

有効値

0 = アラームを無効化

0 = アラームを有効化（デフォルト）

例

以下の例は、タグ変数 Tag2 の変化率アラームを有効にします。

```
Tag2.AlarmROCDisabled=0;
```

参照項目

.AlarmDisabled、.AlarmEnabled、.AlarmROCEnabled

タグ変数のアラームしきい値の変更

アプリケーションの実行中にタグ変数のアラームしきい値を変更するには、以下のドットフィールドを使用します。変更できるのは、LoLo、Lo、Hi、および HiHi アラームのしきい値、大偏差および小偏差のパーセント値と基準値、 および変化率の偏差です。

- .LoLoLimit ドットフィールド.LoLoLimit どっとふぃーるど
- .LoLimit ドットフィールド.LoLimit どっとふぃーるど
- .HiLimit ドットフィールド.HiLimit どっとふぃーるど
- .HiHiLimit ドットフィールド.HiHiLimit どっとふぃーるど
- .MinorDevPct ドットフィールド.MinorDevPct どっとふぃーるど

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- .MajorDevPct ドットフィールド.MajorDevPct どっとふぃーるど
- .DevTarget ドットフィールド.DevTarget どっとふぃーるど
- .ROCPct ドットフィールド.ROCPct どっとふぃーるど

.LoLoLimit ドットフィールド.LoLoLimit どっとふぃーるど

タグ変数の LoLo アラームしきい値を変更します。

カテゴリ

アラーム

使用法

```
TagName.LoLoLimit
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

アナログ型（読み取り／書き込み）

有効値

指定タグ変数に対する設定範囲内の値

例

このステートメントにより、タグ変数 MyTag に対する LoLo アラームしきい値が 10 だけ減少します。

```
MyTag1.LoLoLimit=MyTag1.LoLoLimit - 10;
```

参照項目

.Alarm、.AlarmValue、.Ack、.LoLoStatus、.LoLoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled、.AlarmLoLoEnabled、.AlarmLoLoInhibitor

.LoLimit ドットフィールド.LoLimit どっとふぃーるど

タグ変数の Lo アラームしきい値を変更します。

カテゴリ

アラーム

使用法

```
タグ名です。ロリミット
```

パラメータ

タグ名
任意の整数型、実数型、または間接アナログ型のタグ。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

アナログ型（読み取り／書き込み）

有効値

指定タグ変数に対する設定範囲内の値

例

このステートメントは、タグ変数 MyTag に対する Lo アラームしきい値を 10 だけ減少させます。

```
MyTag1.LoLoLimit=MyTag1.LoLoLimit - 10;
```

参照項目

.Alarm、.AlarmValue、.Ack、.LoStatus、.LoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled、.AlarmLoEnabled、.AlarmLoInhibitor

.HiLimit ドットフィールド.HiLimit どっとふぃーるど

タグ変数の Hi アラームしきい値を変更します。

カテゴリ

アラーム

使用法

```
TagName。HiLimit
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

アナログ型（読み取り／書き込み）

有効値

指定タグ変数に対する設定範囲内の値

例

このステートメントは、タグ変数 **PumpTemp** の Hi 限界アラームを 212 に設定します。

```
PumpTemp.HiLimit = 212;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.Alarm, .AlarmValue, .Ack, .HiHiStatus, .HiHiSet, .AlarmDisabled, .AlarmEnabled, .AlarmHiHiDisabled, .AlarmHiHiEnabled, .AlarmHiHiInhibitor

.HiHiLimit ドットフィールド.HiHiLimit どっとふぃーるど

タグ変数の HiHi アラームしきい値を変更します。

カテゴリ

アラーム

使用法

```
TagName。HiHiLimit
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

アナログ型（読み取り／書き込み）

有効値

指定タグ変数に対する設定範囲内の値

例

以下のステートメントにより、タグ変数 MyTag に対する HiHi アラームしきい値が 5 だけ増加します。

```
MyTag.HiHiLimit=MyTag.HiHiLimit + 5;
```

参照項目

.Alarm, .AlarmValue, .Ack, .HiHiStatus, .HiHiSet, .AlarmDisabled, .AlarmEnabled, .AlarmHiHiDisabled, .AlarmHiHiEnabled, .AlarmHiHiInhibitor

.MinorDevPct ドットフィールド.MinorDevPct どっとふぃーるど

タグ変数の小偏差アラームのしきい値を変更します。

カテゴリ

アラーム

使用法

```
TagName。MinorDevPct
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

実数型（読み取り／書き込み）

有効値

0~100

例

以下のステートメントは、タグ変数 MyTag の小偏差しきい値のプロパティを 25 パーセントに設定します。

```
MyTag.MinorDevPct=25;
```

参照項目

.AckDev、.AlarmDev、.AlarmMinDevDisabled、.AlarmMinDevEnabled、.AlarmMinDevInhibitor、.MinorDevSet、.MinorDevStatus

.MajorDevPct ドットフィールド.MajorDevPct どっとふぃーるど

タグ変数の大偏差アラームのしきい値を変更します。

カテゴリ

アラーム

使用法

```
TagName。MajorDevPct
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

実数型（読み取り／書き込み）

有効値

0~100

例

以下のステートメントは、タグ変数 MyTag の大偏差しきい値のプロパティを 25 パーセントに設定します。

```
MyTag.MajorDevPct=25;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、,AlarmMajDevInhibitor、.MajorDevSet、.MajorDevStatus

.DevTarget ドットフィールド.DevTarget どっとふぃーるど

タグの小偏差および大偏差アラームの基準値を変更します。

カテゴリ

アラーム

使用法

```
TagName.DevTarget
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

実数型（読み取り／書き込み）

有効値

タグに指定された範囲内の値である必要があります。

例

以下のステートメントは、タグ MyTag の偏差基準値を 500 に設定します。

```
MyTag.DevTarget=500;
```

参照項目

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、,AlarmMajDevInhibitor、.MajorDevSet、.MajorDevStatus

.ROCPct ドットフィールド.ROCPct どっとふぃーるど

タグ変数の変化率アラームしきい値を変更します。

カテゴリ

アラーム

使用法

```
TagName。ROCPct
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

備考

タグ変数ディクショナリの［アラーム］で設定されている同じフィールドに直接対応しています。

データ型

整数型（読み取り/書き込み）

有効値

0~100

例

以下のステートメントは、タグ変数 MyTag の変化率アラームしきい値を 25 パーセントに設定します。

```
MyTag.ROCPct=25;
```

参照項目

.ROCStatus、.ROCSet

タグ変数のアラーム デッドバンドの変更

アプリケーションの実行中にタグ変数のアラーム デッドバンドの範囲を変更するには、以下のドットフィールドを使用します。

- .AlarmValDeadband ドットフィールド.AlarmValDeadband どっとふぃーるど
- .AlarmDevDeadband ドットフィールド.AlarmDevDeadband どっとふぃーるど

.AlarmValDeadband ドットフィールド.AlarmValDeadband どっとふぃーるど

InTouch アプリケーションの実行中に、タグのデッドバンド値を変更します。

カテゴリ

アラーム

使用法

```
TagName.AlarmValDeadband
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

アナログ型（読み取り／書き込み）

有効値

タグに指定された範囲内の値である必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

以下のステートメントは、タグ Tag1 に対するデッドバンド値を 25 に変更します。

```
Tag1.AlarmValDeadband=25;
```

参照項目

.AlarmDevDeadband

.AlarmDevDeadband ドットフィールド.AlarmDevDeadband どっとふぃーるど

大小両方の偏差アラームに対するタグ変数の偏差率デッドバンドを変更します。

カテゴリ

アラーム

使用法

```
TagName.AlarmDevDeadband
```

パラメータ

TagName
任意の整数型、実数型、または間接アナログ型のタグ。

備考

WindowViewer を意図的にまたは誤ってシャットダウンした後も継続してこのドットフィールドのランタイム値を使用するには、タグ名ディクショナリで［**パラメータ値保持**］オプションを選択してください。

データ型

整数型（読み取り/書き込み）

有効値

0~100

例

以下のステートメントは、偏差デッドバンドのパーセント率を 25 パーセントに変更します。

```
tag.AlarmDevDeadband=25;
```

参照項目

.AlarmValDeadband、.AlarmDev

タグ変数に関連付けられたアラーム コメントの変更

.AlarmComment ドットフィールドは、タグ変数またはアラーム グループのアラームに関連付けられたコメントの文字列を返します。

.AlarmComment ドットフィールド

タグ変数またはアラーム グループのアラームに関連付けられたコメントの文字列を返します。新規アプリケーションのデフォルト値は空白です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザー定義情報とアラーム インスタンスとの関連付け

アラーム レコードには、3 つの項目（2 つの数値と 1 つの文字列）を付加することができます。アラーム レコードに情報を追加するには、以下のドットフィールドを使用します。

- .AlarmUserDefNumX ドットフィールド AlarmUserDefNumX どっとふーるど
- .AlarmUserDefStr ドットフィールド.AlarmUserDefStr どっとふぃーるど

.AlarmUserDefNumX ドットフィールド AlarmUserDefNumX どっとふーるど

特定のタグ変数だけでなくアラーム グループにドットフィールドを設定できるので、ユーザー値の設定が簡単化できます。たとえば、InTouch で $System アラーム グループに対して .AlarmUserDefNum1 を使ってバッチ番号を設定し、すべてのアラームにバッチ番号を付加することができます。

.AlarmUserDefNum1 と .AlarmUserDefNum2 はそれぞれ、Alarm Viewer コントロールのユーザー 1 とユーザー 2 のカラムに対応しています。

アラーム グループに対して .AlarmUserDefNum1 を設定した場合、この値はこのアラーム グループとサブ グループ内のすべてのアラームに適用されます。また、特定のタグ変数に対して .AlarmUserDefNum1 の値を設定することもできます。この場合、設定した値はそのタグ変数にのみ適用され、そのタグ変数のアラーム グループ内にある .AlarmUserDefNum1 の設定は、この値で上書きされます。

カテゴリ

アラーム

使用法

```
TagName.AlarmUserDefNum1
TagName.AlarmUserDefNum2
```

パラメータ

TagName
論理型、整数型、実数型、間接論理型、間接アナログ型のタグ変数、またはアラーム グループ

備考

このユーザー定義のドットフィールドは、さまざまなタグ変数、特に論理型タグ変数、アナログ型タグ変数、およびアラーム グループに対して（定義されたアラームの有無にかかわらず）有効です。これらのアイテムを未設定のままにすること、すべてのアイテムを設定すること、あるいは個々のタグ変数、グループ、親グループに対する一部のアイテムだけを設定することが可能です。

このドットフィールドの値は、スクリプトや POKE などによって設定済みの場合のみ、アラームに付加されます。

データ タイプ

アナログ型（読み取り／書き込み）

有効値

任意の実数値、または未設定（デフォルト）

例

以下の例では、定数値が使用されています。ただし、InTouch QuickScript を使用すれば、他のタグ変数の値をこれらのユーザー定義フィールドのいずれかにコピーすることができます。また、PtAcc を使用し

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

て設定や確認をしたり、InTouch を I/O Server として使用して値を取得したり、設定したりすることもできます。

```
$System.AlarmUserDefNum1 = 4;
GroupA.AlarmUserDefNum1 = 27649;
```

アラーム通知が分散アラーム システムに送信される場合は、最低レベルの設定を優先するコンセプトになっています。つまり、タグ変数の .AlarmUserDefNum1 に値が設定されている場合、アラーム レコードはこの設定を使用して記録されます。ただし、タグ変数にこの設定がない場合は、タグ変数のアラーム グループに設定があるかがルート グループの $System に達するまでさかのぼって WindowViewer によってチェックされます。タグ変数の設定がいずれのレベルにもない場合、アラーム レコード内のエントリは空のままとなります（数値に対してはゼロ、文字列に対しては空の文字列）。

注意：この階層検索は、アイテムごとに独立して処理されます。このため、タグ変数に .AlarmUserDefNum2 だけが設定されていて .AlarmUserDefNum1 は設定されておらず、親グループで .AlarmUserDefNum1 が設定されている場合は、タグ変数は親グループから .AlarmUserDefNum1 の設定を受け継ぎます。

参照項目

.AlarmUserDefStr

.AlarmUserDefStr ドットフィールド.AlarmUserDefStr どっとふぃーるど

.AlarmUserDefStr ドットフィールドは、Alarm DB Logger によってアラームごとにアラーム データベースに記録される情報に付加されます。.AlarmUserDefStr ドットフィールドは、データベースの User3 フィールドに対応しています。SELECT ステートメントで「ユーザー定義」カラムを使用すると、特定のアラームの集合を選択してデータベース操作を行うことができます。たとえば、$System.AlarmUserDefStr をバッチ文字列に設定し、バッチが変化するたびにこの値も変化するようにした場合、データベースの User3 フィールドを含む選択を使用して特定のバッチのアラームを選択できます。

カテゴリ

アラーム

使用法

```
Tagname.AlarmUserDefStr
```

パラメータ

タグ名
論理型、整数型、実数型、間接論理型、間接アナログ型のタグ変数、またはアラーム グループ

備考

このユーザー定義のドットフィールドは、さまざまなタグ変数、特に論理型タグ変数、アナログ型タグ変数、およびアラーム グループに対して（定義されたアラームの有無にかかわらず）有効です。これらのアイテムを未設定のままにすること、すべてのアイテムを設定すること、あるいは個々のタグ変数、グループ、親グループに対する一部のアイテムだけを設定することが可能です。

このドットフィールドの値は、スクリプトや Poke などによって設定されている場合にのみアラームに付加されます。

データ型

メッセージ型（読み取り／書き込み）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

NULL および任意の有効な文字列

例

上記の例では、定数値が使用されています。ただし、InTouch QuickScript を使用すれば、他のタグ変数の値をこれらのユーザー定義フィールドのいずれかにコピーすることができます。また、PtAcc を使用して設定や確認をしたり、InTouch を I/O Server として使用して値を取得したり、設定したりすることもできます。

```
Tag04.AlarmUserDefStr = "Csv";
```

アラーム通知が分散アラーム システムに送信される場合は、最低レベルの設定を優先するコンセプトになっています。つまり、タグ変数の .AlarmUserDefStr に値が設定されている場合、アラーム レコードはこの設定を使用して記録されます。ただし、タグ変数にこの設定がない場合は、タグ変数のアラーム グループに設定があるかがルート グループの $System に達するまでさかのぼって WindowViewer によってチェックされます。タグ変数の設定がいずれのレベルにもない場合、アラーム レコード内のエントリは空のままとなります（数値に対してはゼロ、文字列に対しては空の文字列）。

また、この階層検索は、アイテムごとに独立して処理されます。したがって、タグ変数に .AlarmUserDefNum1 だけが設定されていて .AlarmUserDefStr が設定されておらず、その親グループには .AlarmUserDefStr の設定がある場合は、この設定がアラーム レコードで使用されます。

参照項目

.AlarmUserDefNumX ドットフィールド

タグ変数またはアラーム グループの抑止タグ変数の確認

各タイプのアラームの抑止タグ変数を調べるには、以下のドットフィールドを使用します。

- .AlarmDscInhibitor ドットフィールド.AlarmDscInhibitor どっとふぃーるど
- .AlarmLoLoInhibitor ドットフィールド.AlarmLoLoInhibitor どっとふぃーるど
- .AlarmLoInhibitor ドットフィールド.AlarmLoInhibitor どっとふぃーるど
- .AlarmHiInhibitor ドットフィールド.AlarmHiInhibitor どっとふぃーるど
- .AlarmHiHiInhibitor ドットフィールド.AlarmHiHiInhibitor どっとふぃーるど
- .AlarmMinDevInhibitor ドットフィールド.AlarmMinDevInhibitor どっとふぃーるど
- .AlarmMajDevInhibitor ドットフィールド.AlarmMajDevInhibitor どっとふぃーるど
- .AlarmROCInhibitor ドットフィールド.AlarmROCInhibitor どっとふぃーるど

.AlarmDscInhibitor ドットフィールド.AlarmDscInhibitor どっとふぃーるど

論理値アラームに割り当てられた抑止タグ変数の名前を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmDscInhibitor
```

パラメータ

TagName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

任意の論理型タグ変数またはアラーム グループ

備考

WindowMaker で設定します。ランタイム中は変更できません。

データ タイプ

メッセージ型（読み取り専用）

例

.AlarmDSCInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name に タグ変数の .AlarmDSCInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作します。

以下のステートメントは、論理値アラームに対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmDscInhibitor;
```

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1;
```

抑止をオンにします。AlarmedTag の論理値アラームは無効になります。

```
SomeIndirectTag = 0;
```

抑制をオフ
AlarmedTag に対して個別アラームを生成可能

.AlarmLoLoInhibitor ドットフィールド.AlarmLoLoInhibitor どっとふぃーるど

LoLo アラームに割り当てられた抑止タグ変数の名前を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmLoLoInhibitor
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループタグ変数

備考

WindowMaker で設定します。ランタイム中は変更できません。

データ タイプ

メッセージ型（読み取り専用）

例

.AlarmLoLoInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name に タグ変数の .AlarmLoLoInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作します。

以下のステートメントは、LoLo アラームしきい値に対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmLoInhibitor;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1;
```

抑制をオン
AlarmedTag の LoLo アラームが無効になります。

```
SomeIndirectTag = 0;
```

抑制をオフ
AlarmedTag の LoLo アラームが生成できるようになります。

参照項目

.AlarmHiInhibitor、.AlarmHiHiInhibitor、.AlarmLoInhibitor

.AlarmLoInhibitor ドットフィールド.AlarmLoInhibitor どっとふぃーるど

Lo アラームに割り当てられた抑止タグ変数の名前を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmLoInhibitor
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループタグ変数

備考

WindowMaker で設定します。ランタイム中は変更できません。

データ タイプ

メッセージ型（読み取り専用）

例

.AlarmLoInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name に タグ変数の .AlarmLoInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作 します。

以下のステートメントは、Lo アラームしきい値に対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmLoInhibitor;
```

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1;
```

抑制をオン
AlarmedTag の Lo アラームが無効になります。

```
SomeIndirectTag = 0;
```

抑制をオフ
AlarmedTag の Lo アラームが生成できるようになります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

.AlarmHiInhibitor、.AlarmHiHiInhibitor、.AlarmLoLoInhibitor

.AlarmHiInhibitor ドットフィールド.AlarmHiInhibitor どっとふぃーるど

Hi アラームに割り当てられた抑止タグ変数の名前を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmHiInhibitor
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループタグ変数

備考

WindowMaker で設定します。ランタイム中は変更できません。

データ タイプ

メッセージ型（読み取り専用）

例

.AlarmHiInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name にタグ変数の.AlarmHiInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作します。

以下のステートメントは、Hi アラームしきい値に対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmHiInhibitor;
```

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1;
```

抑制をオン
AlarmedTag の Hi アラームが無効になります。

```
SomeIndirectTag = 0;
```

抑制をオフ
AlarmedTag の Hi アラームが生成できるようになります。

参照項目

.AlarmHiHiInhibitor、.AlarmLoInhibitor、.AlarmLoLoInhibitor

.AlarmHiHiInhibitor ドットフィールド.AlarmHiHiInhibitor どっとふぃーるど

HiHi アラーム状況に割り当てられた抑止タグ変数の名前を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmHiHiInhibitor
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループタグ変数

備考

WindowMaker で設定します。ランタイム中は変更できません。

データ タイプ

メッセージ型（読み取り専用）

例

.AlarmHiHiInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name に タグ変数の.AlarmHiHiInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作 します。以下のステートメントは、HiHi アラームしきい値に対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmHiHiInhibitor;
```

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1;
```

抑制をオン
AlarmedTag の HiHi アラームが無効になります。

```
SomeIndirectTag = 0;
```

抑制をオフ
AlarmedTag の HiHi アラームが生成できるようになります。

参照項目

.AlarmHiInhibitor、.AlarmLoInhibitor、.AlarmLoLoInhibitor

.AlarmMinDevInhibitor ドットフィールド.AlarmMinDevInhibitor どっとふぃーるど

小偏差アラーム状況に関連付けられているアラーム抑止タグ変数名を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmMinDevInhibitor
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループタグ変数

備考

WindowMaker で設定します。ランタイム中は変更できません。

データ タイプ

メッセージ型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

.AlarmMinDevInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name に タグ変数の .AlarmMinDevInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作 します。以下のステートメントは、小偏差アラームしきい値に対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmMinDevInhibitor;
```

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1。
```

抑制をオン
AlarmedTag の 小偏差アラームが無効になります。

```
SomeIndirectTag = 0;
```

抑制をオフ
AlarmedTag の 小偏差アラームが生成できるようになります。

参照項目

.AlarmMajDevInhibitor

.AlarmMajDevInhibitor ドットフィールド.AlarmMajDevInhibitor どっとふぃーるど

大偏差アラーム状況に関連付けられているアラーム抑止タグ変数名を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmMajDevInhibitor
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループタグ変数

データ タイプ

メッセージ型（読み取り専用）

例

.AlarmMajDevInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name に タグ変数の.AlarmMajDevInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作します。以下のステートメントは、大偏差アラームしきい値に対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmMajDevInhibitor;
```

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1;
```

抑制をオン
AlarmedTag の大偏差アラームが無効になります。

```
SomeIndirectTag = 0;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

抑制をオフ
AlarmedTag の大偏差アラームが生成できるようになります。

参照項目

.AlarmMinDevInhibitor

.AlarmROCInhibitor ドットフィールド.AlarmROCInhibitor どっとふぃーるど

変化率アラーム状況に関連付けられているアラーム抑止タグ変数を返します。

カテゴリ

アラーム

使用法

```
TagName.AlarmROCInhibitor
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループタグ変数

データ タイプ

メッセージ型（読み取り専用）

例

.AlarmROCInhibitor ドットフィールドを使用するには、任意の間接型タグ変数の .Name に タグ変数の.AlarmROCInhibitor と同じ値をセットしてから、その間接型タグ変数の値を操作します。以下のステートメントは、変化率アラームしきい値に対するアラーム抑止タグ変数名を返します（SomeIndirectTag はアナログ型間接タグ変数とする）。

```
SomeIndirectTag.Name = AlarmedTag.AlarmROCInhibitor;
```

アラーム状況にあるタグ変数の抑止状態は、間接型タグ変数の値を次のように設定することによって制御できます。

```
SomeIndirectTag = 1;
```

抑制をオン
AlarmedTag の変化率アラームが無効になります。

```
SomeIndirectTag = 0;
```

抑制をオフ
AlarmedTag の変化率アラームが生成できるようになります。

アクティブまたは未確認のアラーム数の確認

アクティブまたは未確認のアラーム数をアプリケーションの実行中に調べるには、以下のドットフィールドを使用します。

ドットフィールド	説明
.AlarmTotalCount ドットフィールド.AlarmTotalCount どっとふぃーるど	タグ変数またはアラーム グループに関連付けられているアラームの数をカウントします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ドットフィールド	説明
.AlarmUnAckCount ドットフィールド.AlarmUnAckCount どっとふぃーるど	タグ変数またはアラーム グループに関連付けられている未確認アラームの数をカウントします。
.AlarmValueCount ドットフィールド.AlarmValueCount どっとふぃーるど	タグ変数に関連付けられた値型アラームの数をカウントします。
.AlarmValueUnAckCount ドットフィールド.AlarmValueUnAckCount どっとふぃーるど	タグ変数に関連付けられた未確認の値型アラームの数をカウントします。
.AlarmDscCount ドットフィールド.AlarmDscCount どっとふぃーるど	論理値アラームの数をカウントします。
.AlarmDscUnAckCount ドットフィールド.AlarmDscUnAckCount どっとふぃーるど	未確認の論理値アラームの数をカウントします。
.AlarmDevCount ドットフィールド.AlarmDevCount どっとふぃーるど	偏差アラームの数をカウントします。
.AlarmDevUnAckCount ドットフィールド.AlarmDevUnAckCount どっとふぃーるど	未確認の偏差アラームの数をカウントします。
.AlarmROCCount ドットフィールド.AlarmROCCount どっとふぃーるど	変化率アラームの数をカウントします。
.AlarmROCUnAckCount ドットフィールド	未確認の変化率アラームの数をカウントします。

.AlarmTotalCount ドットフィールド.AlarmTotalCount どっとふぃーるど

指定されたタグ変数またはアラーム グループでアクティブなアラームの合計数を追跡します。

カテゴリ

アラーム

使用法

```
TagName.AlarmTotalCount
```

パラメータ

TagName
任意のタイプのタグ変数、またはアラーム グループ

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

備考

カウントには、値型、偏差、変化率、および論理型の各アラームが含まれます。これには、確認済みと未確認の両方のアラームが含まれます。

データ タイプ

整数型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 はアラームに対して設定されたアナログ型タグ変数です。ATC もまたアナログ型タグ変数で、Tag1 に存在するすべてのアクティブなアラーム（未確認と確認の両方）の合計数を取得します。

```
ATC = Tag1.AlarmTotalCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDSCCount、.AlarmDSCUnAckCount、.AlarmValueCount、.AlarmUnAckCount、.AlarmValueUnAckCount、.AlarmROCCount、.AlarmROCUnACkCount

.AlarmUnAckCount ドットフィールド.AlarmUnAckCount どっとふぃーるど

指定されたタグ変数またはアラーム グループの未確認アラームの合計数を追跡します。

カテゴリ

アラーム

使用法

```
TagName.AlarmUnAckCount
```

パラメータ

TagName
任意のタイプのタグ変数、またはアラーム グループ

備考

カウントには、値型、偏差、変化率、および論理型の未確認アラームが含まれます。

データ タイプ

整数型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 はアラームに対して設定されたアナログ型または論理型のタグ変数です。AUC はアナログ型タグ変数で、Tag1 に存在する未確認アラームの合計数を取得します。

```
AUC = Tag1.AlarmUnAckCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmValueCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCCount、.AlarmROCUnACkCount

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

.AlarmValueCount ドットフィールド.AlarmValueCount どっとふぃーるど

指定されたタグ変数またはアラーム グループでアクティブな値型アラームの合計数を追跡します。

カテゴリ

アラーム

使用法

```
TagName.AlarmValueCount
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

備考

これには、HiHi、Hi、Lo、LoLo アラームの数が含まれます。これには、確認済みと未確認の両方のアラームが含まれます。非拡張サマリ型のアラーム タグ変数の場合、このカウント数は 1 以下です。ただし、アラーム グループによってはカウント数が変わることがあります。

データ タイプ

整数型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 は値型アラームに対して設定されたアナログ型タグ変数です。AVC もアナログ型タグ変数で、Tag1 に存在するすべてのアラーム値の合計数を取得します。

```
AVC = Tag1.AlarmValueCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmROCCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmValueUnAckCount ドットフィールド.AlarmValueUnAckCount どっとふぃーるど

指定されたタグ変数またはアラーム グループの未確認値型アラームの合計数を追跡します。これには、HiHi、Hi、Lo、LoLo アラームの数が含まれます。

カテゴリ

アラーム

使用法

```
TagName.AlarmValueUnAckCount
```

パラメータ

TagName
整数型、実数型、間接アナログ型のタグ変数、またはアラーム グループ

データ タイプ

整数型（読み取り専用）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

有効値

0 または任意の正の整数

例

この例で、Tag1 は値型アラームに対して設定されたアナログ型タグ変数です。AVUC もアナログ型タグ変数で、Tag1 に存在するすべての未確認値型アラームの合計数を取得します。

```
AVUC = Tag1.AlarmValueUnAckCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmROCCount、.AlarmTotalCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDscCount ドットフィールド.AlarmDscCount どっとふぃーるど

指定されたタグ変数またはアラーム グループでアクティブな論理値アラームの合計数を追跡します。

カテゴリ

アラーム

使用法

```
TagName.AlarmDscCount;
```

パラメータ

TagName
任意の論理型タグ変数、間接論理型タグ変数、またはアラーム グループ

備考

AlarmDscCount ドットフィールドに割り当てらたこのカウント数には、確認済みと未確認の両方のアラームが含まれています。非拡張サマリ型のアラーム タグ変数の場合、このカウント数は常に 1 です。ただし、アラーム グループによってはカウント数が変わることがあります。

データ タイプ

整数型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 は論理値アラームに対して設定された論理型タグ変数です。ADC はアナログ型タグ変数で、Tag1 に存在するアクティブな論理値アラーム（未確認と確認の両方）の合計数を取得します。

```
ADC = Tag1.AlarmDSCCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmTotalCount、.AlarmDscUnAckCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDscUnAckCount ドットフィールド.AlarmDscUnAckCount どっとふぃーるど

指定されたタグ変数またはアラーム グループの未確認の論理値アラームの合計数を追跡します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

アラーム

使用法

```
TagName.AlarmDscUnAckCount
```

パラメータ

TagName
任意の論理型タグ変数、間接論理型タグ変数、またはアラーム グループ

データ タイプ

整数型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 は論理値アラームに対して設定された論理型タグ変数です。ADUC はアナログ型タグ変数で、Tag1 に存在する未確認の論理値アラームの合計数を取得します。

```
ADUC = Tag1.AlarmDscUnAckCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmValueCount、.AlarmROCCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDevCount ドットフィールド.AlarmDevCount どっとふぃーるど

指定されたタグ変数またはアラーム グループでアクティブな偏差アラームの合計数を追跡します。

カテゴリ

アラーム

使用法

```
TagName.AlarmDevCount
```

パラメータ

TagName
実数型タグ変数、整数型タグ変数、間接アナログ型のタグ変数、またはアラーム グループ

備考

これには、大偏差アラームおよび小偏差アラームのカウント数が含まれます。これには、確認済みと未確認の両方のアラームが含まれます。非拡張サマリ型のアラーム タグ変数の場合、このカウント数は常に 1 です。ただし、アラーム グループによってはカウント数が変わることがあります。

データ タイプ

アナログ型（読み取り専用）

有効値

0 または任意の正の整数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例で、Tag1 は偏差アラームに対して設定されたアナログ型タグ変数です。ADC もまたアナログ型タグ変数で、Tag1 に存在するアクティブな偏差アラーム（未確認と確認の両方）の合計数を取得します。

```
ADC=Tag1.AlarmDevCount;
```

参照項目

.AlarmDSCCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmTotalCount、.AlarmDSCUnAckCount、.AlarmValueUnAckCount、.AlarmDevUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDevUnAckCount ドットフィールド.AlarmDevUnAckCount どっとふぃーるど

指定されたタグ変数またはアラーム グループの未確認の偏差アラームの合計数を追跡します。これには、大偏差アラームおよび小偏差アラームのカウント数が含まれます。

カテゴリ

アラーム

使用法

```
TagName.AlarmDevUnAckCount
```

パラメータ

TagName
実数型タグ変数、整数型タグ変数、間接アナログ型のタグ変数、またはアラーム グループ

データ タイプ

アナログ型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 は偏差アラームに対して設定されたアナログ型タグ変数です。ADUC もまたアナログ型タグ変数で、Tag1 に存在する未確認の偏差アラームの合計数を取得します。

```
ADUC = Tag1.AlarmDevUnAckCount;
```

参照項目

.AlarmDevCount、.AlarmDSCCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmTotalCount、.AlarmDSCUnAckCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmROCCount ドットフィールド.AlarmROCCount どっとふぃーるど

指定されたタグ変数またはアラーム グループでアクティブな変化率アラームの合計数を追跡します。これには、確認済みと未確認の両方のアラームが含まれます。非拡張サマリ型のアラーム タグ変数の場合、このカウント数は常に 1 です。ただし、アラーム グループによってはカウント数が変わることがあります。

カテゴリ

アラーム

使用法

```
TagName.AlarmROCCount
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ

TagName
実数型タグ変数、整数型タグ変数、間接アナログ型のタグ変数、またはアラーム グループ

データ タイプ

整数型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 は変化率アラームに対して設定されたアナログ型タグ変数です。ARC もまたアナログ型タグ変数で、Tag1 に存在するアクティブな変化率アラーム（未確認と確認の両方）の合計数を取得します。

```
ARC = Tag1.AlarmROCCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmValueCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmROCUnAckCount ドットフィールド

指定されたアナログ型タグ変数またはアラーム グループの未確認の変化率アラームの合計数を追跡します。

カテゴリ

アラーム

使用法

```
TagName.AlarmROCUnAckCount
```

パラメータ

TagName
実数型タグ変数、整数型タグ変数、間接アナログ型のタグ変数、またはアラーム グループ

データ タイプ

整数型（読み取り専用）

有効値

0 または任意の正の整数

例

この例で、Tag1 は変化率アラームに対して設定されたアナログ型タグ変数です。ARUC もアナログ型タグ変数で、Tag1 に存在する未確認の変化率アラームの合計数を取得します。

```
ARUC = Tag1.AlarmROCUnAckCount;
```

参照項目

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmValueCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCCount、.AlarmUnAckCount

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Maintain

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

章 25 アプリケーションの移行およびアップグレード

このセクションでは、さまざまな InTouch アプリケーションの移行とアップグレードについて説明します。

従来のアプリケーションから新しいスタンドアロン アプリケーションへの移動

System Platform 2020 以前では、InTouch HMI ユーザーは、以下のタイプのアプリケーションを作成することができました。

- スタンドアロン
- モダン
- マネージド
- パブリッシュ済み

スタンドアロン アプリケーションは、従来のシンボルおよびコントロールを使用して構築されました。モダン アプリケーションは、従来のシンボルに加えて産業用グラフィック（以前の名称は ArchestrA グラフィック/シンボル）をサポートしました。マネージド アプリケーションは IDE および Galaxy オブジェクトを使用して構築されました。スタンドアロン アプリケーションはパッケージにパブリッシュして、他のノードに配布することができました（パブリッシュ済みアプリケーション）。

InTouch HMI 2020 では、モダン アプリケーションは、より包括的なスタンドアロン アプリケーションとして再設計されました。新しいスタンドアロン アプリケーションは、従来のスタンドアロン アプリケーションに比べて多くの機能強化が行われています。

- 容易な配布 – アプリケーション フォルダをコピーして別のノードにコピーできます。インポートやエクスポートの操作は必要ありません。
- 産業用グラフィックの使用 – 新しいスタンドアロン アプリケーションでは、従来のアプリケーションの使いやすさと最新の産業用グラフィックが組み合わされています。
- クラウド対応 – オンプレミス ノードで作成されたアプリケーションが HTML5 互換ブラウザで表示できるようになりました。
- 軽量 – アプリケーション ファイルは軽量なので、パフォーマンスと使用性が向上しています。

マネージド アプリケーションとパブリッシュ済みアプリケーションの動作は変更されていません。

古いアプリケーションの移行およびアップグレード

以前のバージョンの InTouch HMI で作成されたアプリケーションをサポートするために、2 つのワークフローを使用して新しいスタンドアロン アプリケーションに移行できます。

- 古いモダン アプリケーションのインプレース移行 : ノードに古いモダン アプリケーションが含まれていて、ノードの製品バージョンをアップグレードする場合、アプリケーション マネージャを使用してアプリケーションを移行します。
- 以前のバージョンの InTouch HMI からエクスポートされたモダン アプリケーション .aapkg ファイルのインポート

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

以前の InTouch アプリケーションの現在のバージョンへの移行

古いバージョンの InTouch HMI で開発したアプリケーションを現在のバージョンに移行できます。WindowMaker または WindowViewer のどちらかで古いアプリケーションを開こうとすると、［アプリケーションの移行］ダイアログ ボックスが表示されます。このダイアログ ボックスでは、以下の操作を行うことができます。

- アプリケーションの解像度を変換する。
- 古いアプリケーションを現在のバージョンの InTouch HMI に移行する前に、バックアップ コピーを作成する。

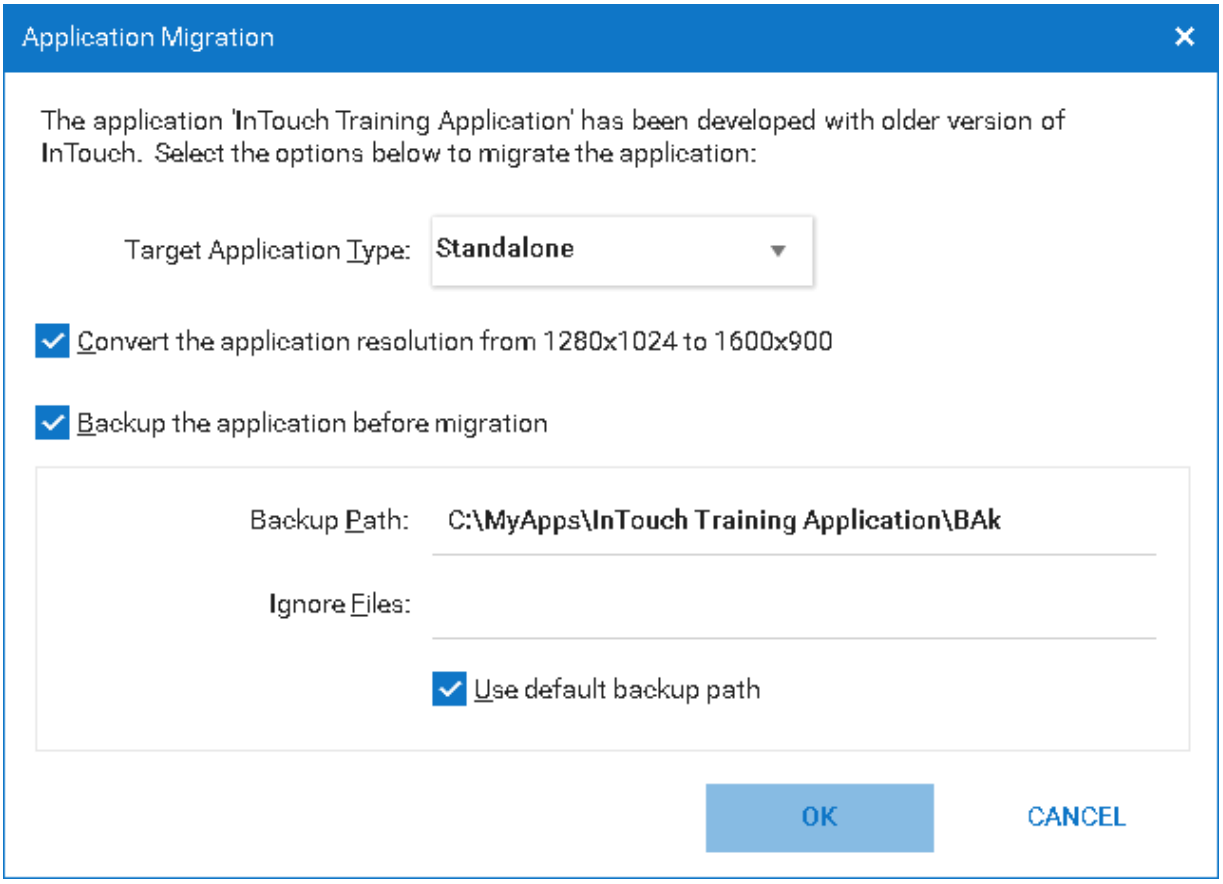
既存のスタンドアロン、モダン、またはパブリッシュ済み InTouch アプリケーションを現在のバージョンの InTouch に移行できます。バックアップ コピーを作成するフォルダを指定したり、バックアップから除外するファイルを指定する必要があります。

1. アプリケーション リストから目的のアプリケーションをダブルクリックします。

   ［**アプリケーションの移行**］ダイアログ ボックスが表示されます。

2. アプリケーションの解像度を元の解像度から現在の解像度に変換するには、［**アプリケーションの解像度を <既存の解像度> から <新しい解像度> に変換**］チェック ボックスをオンにします。
3. デフォルトのバックアップ パス（<アプリケーション ディレクトリ>\Bak）を変更するには、［**移行前にアプリケーションをバックアップ**］チェック ボックスをオフにします。次に、［**バックアップのパス**］ボックスに、バックアップを保存するフォルダへのパスを入力します。フォルダが存在しない場合、フォルダを作成してからバックアップを作成する必要があります。
4. ［**ファイルを無視**］ボックスで、バックアップから除外するファイルを指定できます。デフォルトでは、アプリケーションのディレクトリにあるすべてのファイルがバックアップされます。徐外するファイルの名前を入力します。複数のファイル名を入力する場合は、各ファイル名をセミコロンで区切ります。または、名前の中の共通の文字を使って一連のファイルを除外するには、標準のワイルド カード文字（「*」および「?」）を使用します。
5. 必要なオプションを設定したら、［**OK**］ を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



古いアラーム表示の変換

InTouch 7.11 より前のバージョンで構築されたアプリケーションを WindowViewer で開く場合、WindowMaker を実行してアプリケーションを変換することを要求するダイアログ ボックスが表示されます。変換を続行すると、デフォルト値ですべての標準のアラーム オブジェクトが分散アラーム オブジェクトに変換されます。色、フォント、式、およびアラーム クエリーの設定内容は保持されません。

アプリケーション設定の管理

アプリケーション パスなどの InTouch アプリケーション設定は、Win.ini ファイルに保存されます。Win.ini ファイルは以下のディレクトリにあります。

C:\Users\<ユーザー名>\AppData\Local\Wonderware

WindowMaker は管理者ユーザーとして実行されます。WindowViewer は管理者ユーザーまたは標準ユーザーとして実行することができます。標準ユーザーは管理者ユーザー プロファイルの Win.ini ディレクトリにアクセスできません。したがって、アプリケーション開発者は、アプリケーションを開発するときに共通の Win.ini 属性を標準ユーザーの Win.ini プロファイルにコピーする必要があります。これにより、標準ユーザーが WindowViewer を起動したときに、管理者ユーザーの下で設定されているすべての属性も使用可能になります。共通の Win.ini 属性を変更するたびに、属性をコピーする必要があります。

InTouch アプリケーションのインポート

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーション マネージャを使用して既存のモダン アプリケーションをインポートして、スタンドアロン アプリケーションに変換できます。スタンドアロン アプリケーションは別のノードにコピーできます。別のノードにコピーしたスタンドアロン アプリケーションは、［アプリケーションの検索］オプションを使用して見つけることができます。インポートやエクスポートの操作は必要ありません。

既存のモダン アプリケーションのインポート：

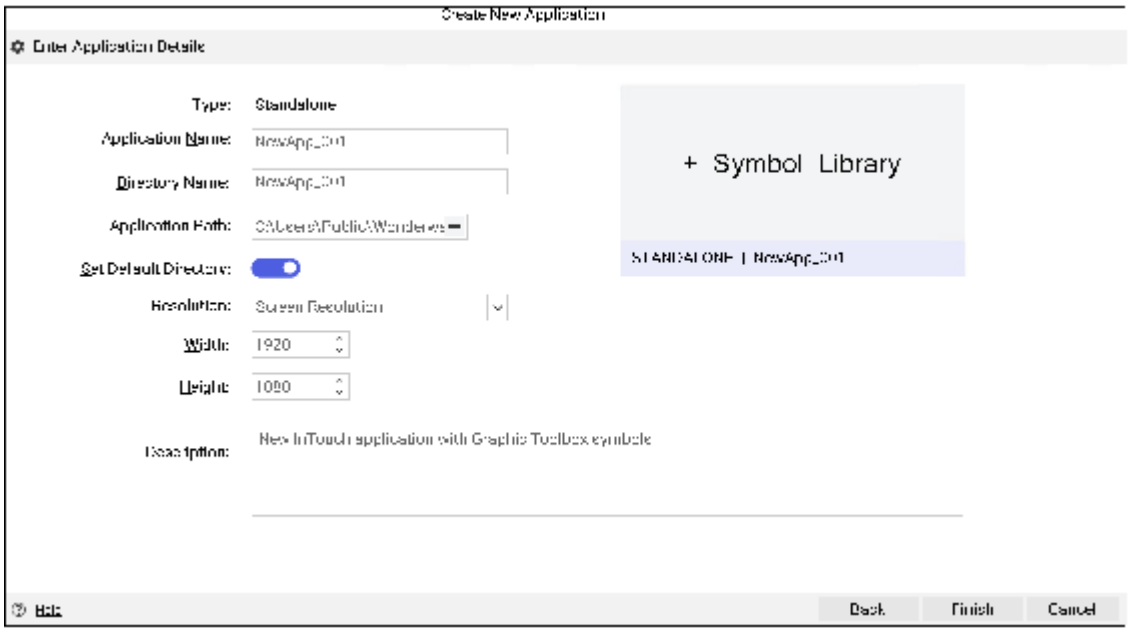
1. ［**ファイル**］メニューの［**インポート**］グループで［**インポート**］を選択します。

   ［**新規アプリケーションの作成: インポートするアプリケーションの選択**］画面が表示されます。

2. ［**アプリケーションの検索**］セクションを使用して、インポートするアプリケーションを検索します。インポートするフォルダまたはファイルを検索します。

3. アプリケーションを選択して［**次へ**］を選択します。

   ［アプリケーションの詳細を入力］画面が表示されます。



4. 必要に応じて設定を変更します。

   [完了] を選択します。

   新しいアプリケーションが作成され、アプリケーション マネージャに表示されます。

注意：すべてのモダン アプリケーションはスタンドアロン アプリケーションとしてインポートされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

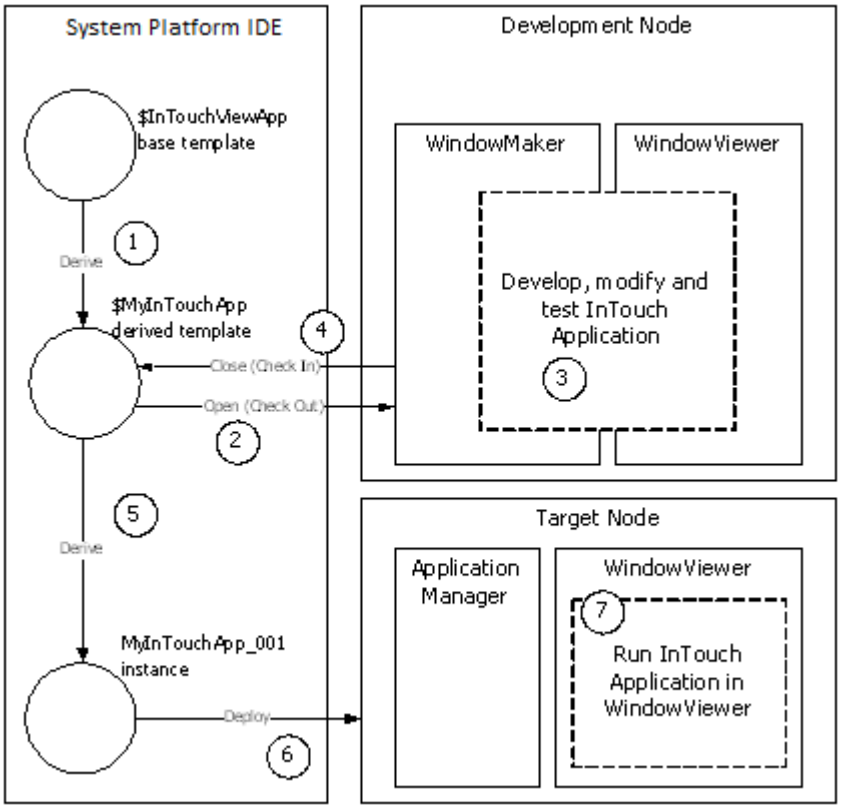
章 26 System Platform IDE を使用した InTouch アプリケーションの管理

IDE を使用して、InTouch アプリケーションを管理できます。以下の手順は、通常の方法を示しています。詳細については、「IDE を使用した InTouch アプリケーションの管理」を参照してください。

System Platform IDE の InTouch の機能は、次の 2 つのオートメーション オブジェクトによって処理されます。

- InTouchViewApp オブジェクトは、デザイン時およびランタイム時の InTouch アプリケーションを表します。
- ViewEngine オブジェクトは、Galaxy 内の対象ノードでの InTouch アプリケーションの実行方法を制御します。

以下の図は、System Platform IDE を使用して InTouch アプリケーションがどのように管理されるのかを示しています。



IDE を使用して InTouch アプリケーションを管理する

1. マネージド InTouch アプリケーションを System Platform IDE で作成します。
2. 作成した InTouch アプリケーションを WindowMaker で開きます。
3. WindowMaker で InTouch アプリケーションを設定します。アプリケーションをテストするために、WindowViewer に切り替えることができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. InTouch アプリケーションを保存し、WindowMaker と WindowViewer を終了します。
5. InTouch アプリケーションを配置するノードを決めます。
6. Galaxy 内の対象ノードに InTouch を配置します。
7. 対象ノード上の WindowViewer で InTouch アプリケーションを実行します。

InTouchViewApp オブジェクト InTouchViewApp おぶじぇくと

Application Server は、InTouchViewApp オブジェクトと呼ばれる特定のタイプの Application Server オブジェクトを使用して InTouch アプリケーションを管理します。

InTouchViewApp テンプレートは、デザイン時には 1 つの特定のマネージド InTouch アプリケーションを参照しており、ランタイム時には実行できません。

InTouchViewApp テンプレートのインスタンスを作成する必要があります。このインスタンスは、対象ノードに配置できます。ターゲット ノードとは、WindowViewer 内でマネージド InTouch アプリケーションが実行するノードのことです。

InTouch アプリケーションを分散させるには、同じテンプレートの複数のインスタンスを作成し、複数のノードに配置します。

オプションで、以下を行うことができます。

- Galaxy 間でマネージド InTouch アプリケーションを交換するために、InTouchViewApp オブジェクトをエクスポートおよびインポートします。
- タグのディクショナリ データを .csv ファイルとしてエクスポートおよびインポートします。
- 異なるタイプの InTouch アプリケーション間で、ウィンドウをエクスポートおよびインポートします。
- マネージド InTouch アプリケーションをパブリッシュします。パブリッシュ済み InTouch アプリケーションはスタンドアロン InTouch アプリケーションのように実行されますが、産業用グラフィックを含めることができます。
- ArchestrA 属性を持つ InTouch タグの読み取りおよび書き込みを行うには、配置された InTouchViewApp オブジェクトの属性を使用します。

InTouchViewApp オブジェクトを使用する

1. InTouchViewApp テンプレートを $InTouchViewApp base テンプレートから派生させます。
2. 新しい InTouch アプリケーションを作成するか、またはスタンドアロン InTouch アプリケーションをインポートすることにより、派生テンプレートを InTouch アプリケーションに関連付けます。
3. WindowMaker でアプリケーションを開きます。
4. WindowMaker でおよびアプリケーションを設定し、WindowViewer でテストします。
5. WindowMaker を保存して閉じます。InTouchViewApp テンプレートがチェック インされます。
6. InTouchViewApp テンプレートからインスタンスを派生させます。
7. これらのインスタンスを、Galaxy 内で選択した対象ノードに配置します。
8. 対象ノードでアプリケーション マネージャを実行し、WindowViewer でマネージド InTouch アプリケーションを実行します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouchViewApp テンプレートと InTouch アプリケーションの関連付け

新しい InTouchViewApp テンプレートを作成した後、以下を行うことにより、InTouchViewApp テンプレートを InTouch アプリケーションに関連付けることができます。

- 新しい InTouch アプリケーションの作成。
- スタンドアロン InTouch アプリケーションのインポート

InTouchViewApp テンプレートには、タグ変数の設定および値などの InTouch アプリケーションのデータ自体は含まれていません。単にアプリケーションを参照しているだけです。

マネージド InTouch アプリケーションの編集

スタンドアロン InTouch アプリケーションで行う場合と同様に、WindowMaker を使用してマネージド InTouch アプリケーションを編集します。ただし、WindowMaker で関連付けられている InTouch アプリケーションを起動するために、InTouchViewApp テンプレートのエディタを開く場合を除きます。

InTouch アプリケーションへの変更を加えた後に WindowMaker を終了すると、InTouchViewApp オブジェクトが自動的にチェック インされます。

マネージド InTouch アプリケーションのテスト

スタンドアロン InTouch アプリケーションで行う場合と同様に、WindowViewer を使用してマネージド InTouch アプリケーションをテストできます。

System Platform IDE から WindowMaker を開いた場合、WindowMaker と WindowViewer をすばやく切り替えてマネージド アプリケーションをテストできます。

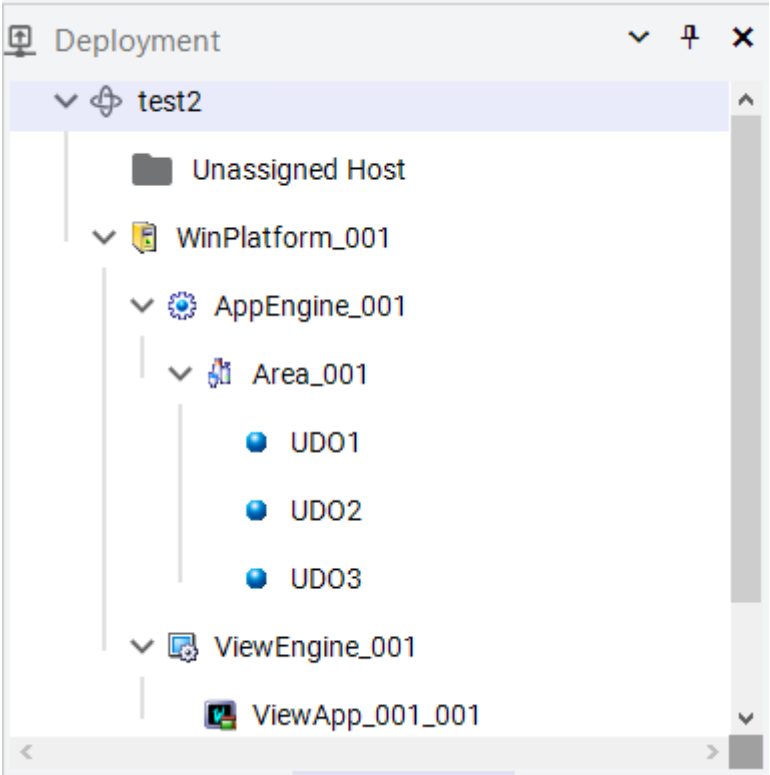
マネージド InTouch アプリケーションに、`galaxy:UDA` などの Application Server データへの参照が含まれている場合、InTouch アプリケーションを編集しているノードに WinPlatform オブジェクトを配置する必要があります。そうしないと、データに値が表示されません。

InTouchViewApp オブジェクトの配置

InTouchViewApp テンプレートのインスタンスを派生させた後、ViewEngine オブジェクト下にある対象プラットフォームに割り当てることができます。

1 つの ViewEngine 下で同じ親を持つ複数の InTouchViewApp インスタンスを割り当てることはできません。その場合、同じ親を持つ、追加の InTouchViewApp インスタンスをホストする 2 つ目の ViewEngine インスタンスを作成します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



InTouchViewApp オブジェクトを配置した後、対象ノードの InTouch アプリケーション マネージャを開くことができます。関連付けられているマネージド InTouch アプリケーションと、[修正された日付] カラムでの最新の配置のタイム スタンプがリストに表示されます。

InTouchViewApp インスタンスを対象ノードに配置すると、InTouch アプリケーションが以下に含まれます。

- 開発ノード上のフォルダ。これには、InTouchViewApp テンプレートのソースが含まれます。
- InTouch アプリケーションが実行されている対象ノード上のフォルダ。これには、InTouch アプリケーションのインスタンスのコピーが含まれます。

InTouchViewApp オブジェクトのエクスポートとインポート

InTouchViewApp オブジェクトをエクスポートできます。これは、たとえば、マネージド InTouch アプリケーションと、それをホストする他の Galaxy 内にある InTouchViewApp オブジェクトを一緒に使用するために行います。

オブジェクトをエクスポートすると、オブジェクト、関連付けられているマネージド InTouch アプリケーション、およびアプリケーションによって使用されているすべての産業用グラフィックに関する情報が含まれているパッケージ ファイル（.aaPKG）が作成されます。

InTouchViewApp オブジェクトをインポートすると、System Platform IDE によって、マネージド InTouch アプリケーションもインポートされます。

InTouchViewApp オブジェクトの属性

InTouchViewApp オブジェクトの Application Server 属性を使用して、関連付けられている InTouch アプリケーションのタグのランタイム データにアクセスできます。これは、Galaxy 名前空間で直接 InTouch デ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ータの読み取り、および書き込みを行ったり、InTouchProxy オブジェクトと同じ機能を提供する場合に便利です

この例では、配置されたマネージド InTouch アプリケーションでは、タンクの充てん量レベルを報告するために実数型のタグ TankLevel が使用され、バルブの状態を制御するために論理型タグ ValveOut が使用されています。

InTouch タグを InTouchViewApp オブジェクト インスタンスから読み取ったり書き込んだりする

1. 配置された InTouchViewApp オブジェクトを右クリックし、次に［**モニタ**］を選択します。

   ［**オブジェクト ビューア**］画面が表示されます。

2. ［表示］領域を右クリックし、次に［**属性参照の追加**］を選択します。

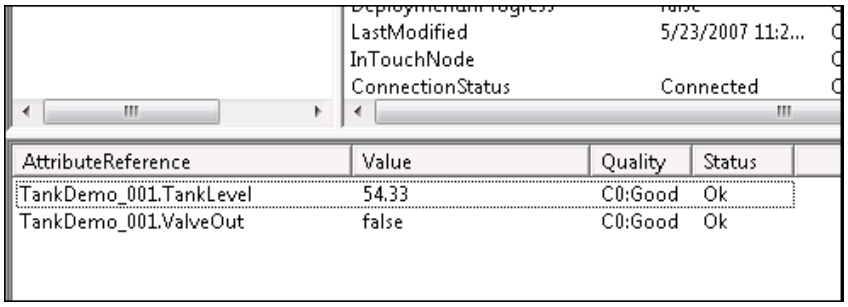
   ［**属性参照の追加**］ダイアログ ボックスが表示されます。

3. ［**属性参照**］ボックスに、InTouchViewApp オブジェクトの名前、ドット、読み取りまたは書き込みを行う InTouch タグの名前の順で入力します。たとえば、`TankDemo_001.TankLevel` などのように入力します。

4. ［**OK**］ を選択します。［表示］領域に属性が追加されます。

5. 読み取りまたは書き込みを行うその他の InTouch タグに対しても、手順 2 から 4 を繰り返します。

6. InTouch タグの値が表示されます。



7. 以下の手順を実行して、InTouch タグの値を書き込みます。

   a. タグをダブルクリックします。［**値の変更**］ダイアログ ボックスが表示されます。

   b. 新しい値を入力し、［**OK**］ を選択します。実行中の InTouch アプリケーションのタグに値が書き戻されます。

InTouchViewApp オブジェクトと他のオートメーション オブジェクトとの違い

InTouchViewApp オブジェクトは､他のオートメーション オブジェクトとは異なります。他のオートメーション オブジェクトで通常行うことができるいくつかの操作を実行することができません。

- InTouchViewApp インスタンスを設定しようとすると、親テンプレートを代わりに開くかどうかを確認するメッセージが表示されます。インスタンスは直接設定できません。親テンプレートのみ設定できます。
- 設定するために複数の InTouchViewApp テンプレートを 1 つのノードで一度に開こうとすると、IDE によって阻止されます。WindowMaker、WindowViewer、およびアプリケーション マネージャを終了して再試行してください。または、InTouch WindowMaker を使用して、異なるノード上にある InTouchViewApp オブジェクトを編集できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- WindowMaker を使用して InTouch アプリケーションを編集している間に IDE を終了した場合、変更を保存するよう要求するメッセージが WindowMaker に表示されます。終了後、InTouchViewApp テンプレートがチェック インされます。
- WindowViewer を使用して InTouch アプリケーションをテストしている間に IDE を終了した場合、WindowViewer が終了します。

以下のような場合、次の操作を実行します。

- InTouchViewApp と InTouch アプリケーション間での関連付けを変更するには、新しい派生した InTouchViewApp テンプレートを代わりに作成します。
- InTouch で ArchestrA セキュリティ（Galaxy セキュリティ）を使用するには、配置されたマネージド InTouch アプリケーションが実行されているノードに WinPlatform インスタンスを配置します。

ViewEngine オブジェクト ViewEngine おぶじぇくと

ViewEngine は、配置された InTouchViewApp オブジェクトをホストおよび実行する Application Server オブジェクトです。

InTouchViewApp インスタンスを対象プラットフォームに配置するには、まず ViewEngine オブジェクトに割り当てる必要があります。すると ViewEngine オブジェクトが対象 WinPlatform オブジェクトに割り当てられます。

AppEngine インスタンスがアプリケーション オブジェクトに対して行う場合と同様に、ViewEngine によって InTouchViewApp インスタンスと同じ機能が実行されます。ViewEngine によって、

- InTouchViewApp オブジェクトが最初に配置、起動されたときに、Galaxy 内の他のオブジェクトと通信できるようにするために、InTouchViewApp オブジェクトが設定および初期化されます。
- 監視、警告、履歴の記録が可能な、属性の診断が実行されます。
- Historian にデータの履歴が記録されます。

異なる ViewEngine オブジェクトを使用して、以下を行うことができます。

- 別の Historian へのデータの履歴の記録。
- 異なる走査速度での、配置された InTouch アプリケーションとの対話。これにより、InTouch タグ データが Galaxy 名前空間と対話できる頻度が設定されます。

プラットフォームは、複数の ViewEngine オブジェクトをホストできます。すべての InTouchViewApp を ViewEngine に割り当てる必要があります。

同じ ViewEngine オブジェクト下で実行する、同じ InTouchViewApp テンプレートの複数のインスタンスを作成することはできません。ただし、異なる ViewEngine オブジェクト下で、同じテンプレートの複数のインスタンスを実行することはできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 27 InTouch コンポーネントのエクスポートとインポート

既存のアプリケーションのコンポーネントの一部またはすべてをインポートまたはエクスポートすることによって、より迅速に InTouch アプリケーションを構築できます。既存のアプリケーションから新しいアプリケーションにタグ変数定義、ウィンドウ、スクリプト、アプリケーション スタイル ライブラリ、産業用グラフィック、クライアント コントロール、ローカリゼーション文字列、HTML5 ウィジェット、およびスクリプト関数ライブラリをインポートできます。タグ変数定義はアプリケーション マネージャからインポートおよびエクスポートできます。その他のコンポーネントは WindowMaker からインポートおよびエクスポートできます。

マネージド InTouch アプリケーションに関連付けられているタグ変数データのエクスポートとインポート

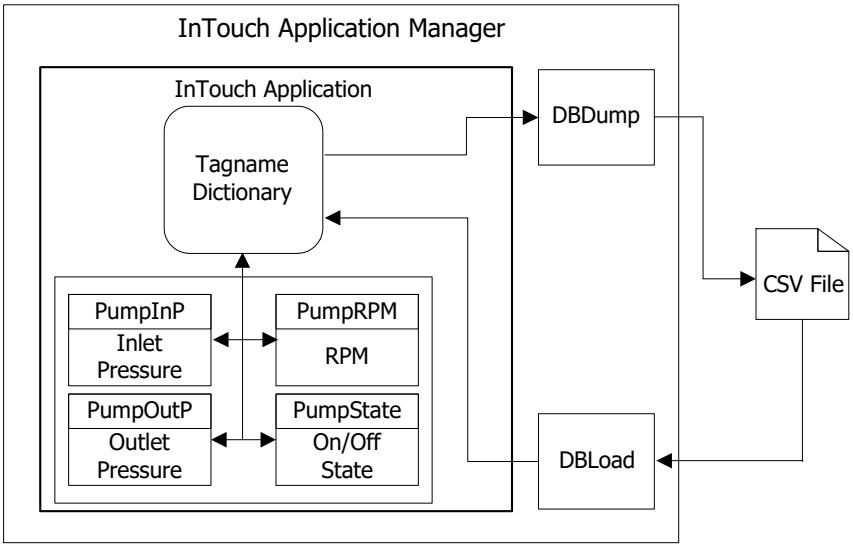
マネージド InTouch アプリケーションに関連付けられているタグ変数データを、.csv ファイルにエクスポートできます。これは、InTouch アプリケーション マネージャの DB ダンプ関数と同等です。

.csv ファイルからエクスポートされたタグ変数データは、DB ロード関数と同じ方法で、マネージド InTouch アプリケーションにインポートできます。

マネージド InTouch アプリケーションからエクスポートされた .csv ファイルと、スタンドアロン InTouch アプリケーションからエクスポートされた .csv ファイルはまったく同じです。

タグ定義をエクスポートする

以下の図は、中間エクスポート ファイルとアプリケーションのタグ名ディクショナリ間のタグ定義をエクスポートおよびインポートする手順を示しています。



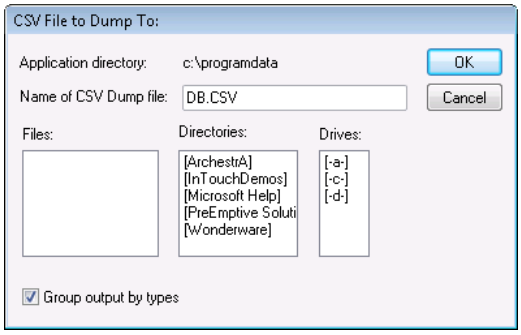
アプリケーション マネージャ内の DBDump ユーティリティを使用して、タグ名ディクショナリの内容をカンマ区切り値（CSV）ファイルにエクスポートします。エクスポートされたファイルを Microsoft メモ帳または Microsoft Excel で表示したり、編集したりできます。編集を行った後で、アプリケーション マネージャ ユーティリティでもある DBLoad ユーティリティを使用して、タグ定義を InTouch アプリケーションにインポートします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ定義をエクスポートする前に、InTouch HMI ソフトウェアの現在のバージョンにアプリケーションを変換する必要があります。

タグ定義をエクスポートする

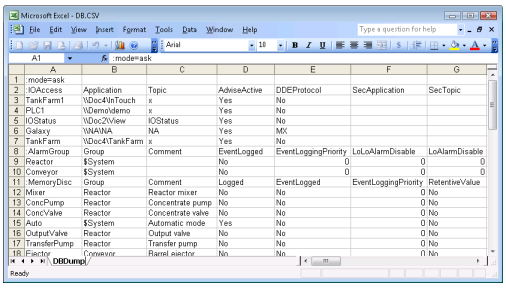
1. WindowMaker と WindowViewer を閉じます。
2. アプリケーション マネージャを起動します。［**アプリケーション マネージャ**］ダイアログ ボックスには、InTouch アプリケーションのリストが表示されます。
3. リストからアプリケーションを選択します。
4. ［**ファイル**］メニューの［**データ**］グループで［**DBDump**］をクリックします。［**ダンプ先 CSV ファイル名**］ダイアログ ボックスが表示されます。



5. ［**ダンプ先 CSV ファイル名**］ボックスに、.csv ファイル名の拡張子を持つファイルの名前を入力します。
6. エクスポート ファイルのデータ グループ化のタイプを選択します。
   - ［**タグ タイプ毎にグループ出力**］チェック ボックスをオンにしてし、エクスポート ファイルのタグのタイプによってデータをグループ化します。これがデフォルトです。
   - ［**タグ タイプ毎にグループ出力**］をクリアして、タグ名によって出力をアルファベット順にエクスポート ファイルに保存します。
7. ［**OK**］を選択して、タグ名ディクショナリの内容を選択したファイルに保存します。内容がファイルに正常に保存されたことを示すメッセージが表示されます。

エクスポートされたタグ定義の表示

Microsoft Excel を使用して、DBDump ユーティリティを使用して作成されたエクスポート ファイルを表示する場合、各データ レコードが個別のスプレッドシート セルに表示されます。



	A	B	C	D	E	F	G
1	mode=ask						
2	IOAccess	Application	Topic	AdviseActive	DDEProtocol	SecApplication	SecTopic
3	TankFarm1	\\Doc4\InTouch	x	Yes	No		
4	PLC1	\\Demo\demo	x	Yes	No		
5	IOStatus	\\Doc2\View	IOStatus	Yes	No		
6	Galaxy	\\NA\NA	NA	Yes	MX		
7	TankFarm	\\Doc4\TankFarm	x	Yes	No		
8	AlarmGroup	Group	Comment	EventLogged	EventLoggingPriority	LoLoAlarmDisable	LoAlarmDisable
9	Reactor	$System		No	0	0	0
10	Conveyor	$System		No	0	0	0
11	MemoryDisc	Group	Comment	Logged	EventLogged	EventLoggingPriority	RetentieValue
12	Mixer	Reactor	Reactor mixer	No	No	0	No
13	ConcPump	Reactor	Concentrate pump	No	No	0	No
14	ConcValve	Reactor	Concentrate valve	No	No	0	No
15	Auto	$System	Automatic mode	Yes	No	0	No
16	OutputValve	Reactor	Output valve	No	No	0	No
17	TransferPump	Reactor	Transfer pump	No	No	0	No
18	Ejector	Conveyor	Barrel ejector	No	No	0	No

このファイルでは、キーワード、キーワードの属性が配置され、またタグ名ディクショナリからのデータがカラム順にキーボード属性の下に配置された状態で構成されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Excel スプレッドシートの例にある :MemoryDisc キーワードをご覧ください。このキーワードはタグ名ディクショナリからエクスポートされたメモリ論理型タグを識別します。同じスプレッドシート行で、メモリ論理型タグの属性は個別のスプレッドシート カラムに表示されています。たとえば、ログ記録された属性カラムでは、メモリ論理型タグのデータがログ記録されるかどうかが示されています。

キーワードと属性の行のすぐ下には、エクスポートされたタグとそれに関連付けられたプロパティが配置されています。Excel スプレッドシートの例で、OutputValve はデータがログ記録されていないメモリ論理型タグです。

DBDump によって作成されたエクスポート ファイルは、.csv ファイル形式をサポートする任意のプログラムで表示したり、編集したりできます。一般的には、縦欄式スプレッドシート形式によって、タグ データを整理することが簡単であるため Excel が使用されます。ただし、ファイルの内容をネイティブのカンマ区切り文字列形式で表示したり、編集したりする場合は、Microsoft メモ帳を使用することもできます。

タグ定義のインポート

アプリケーション マネージャ内の DBLoad ユーティリティを使用すると、タグ定義の .csv ファイルをアプリケーションのタグ名ディクショナリにインポートできます。DBDump ユーティリティを使用して元々作成された定義ファイルをインポートできます。または、独自のインポート ファイルを作成することもできます。

DBLoad ユーティリティを使用して、スーパータグ インスタンスを作成することもできます。詳細については、「スーパータグ インスタンスの作成」を参照してください。

タグ名ディクショナリ インポート ファイルの形式

.csv ファイル形式をサポートする任意のプログラムを使用して、DBLoad インポート ファイルを手動で作成できます。Excel を使用してインポート ファイルを作成する場合、各エントリは個別のスプレッドシート セルに配置されます。この処理によって内容が読みやすくなり、エラーの可能性も少なくなります。

インポート ファイルの作成の詳細については、「インポート ファイル テンプレートの作成」を参照してください。

DBLoad インポート ファイルには、ファイル内のアクセス名、アラーム グループ、およびタグ変数データを整理する一組のキーワードが含まれています。

- すべてのキーワードの前には、コロン（:）が付きます。
- 行を続ける場合は、行末に円記号（\）を入力します。
- コメントを入力するには、その前にセミコロン（;）を付けます。

以下の表には、DBLoad インポート ファイル内のキーワードが一覧表示されています。この表には、DBDump を使用してファイルを作成するときに指定した順序で、キーワードが一覧表示されています。ただし、ファイル内では任意の順序でキーワードを指定できます。

キーワード	説明
:mode	DBLoad ファイルの内容をアプリケーションのタグ名ディクショナリにインポートしているとき、重複したタグ レコードを処理する方法を指定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

キーワード	説明
:IOAccess	InTouch アプリケーションに対して定義されたアクセス名
:AlarmGroup	InTouch アプリケーションに対して定義されたアラーム グループ
:MemoryDisc	メモリ論理型タグ
:IODisc	I/O 論理型タグ
:MemoryInt	メモリ整数型タグ
:IOInt	I/O 整数型タグ
:MemoryReal	メモリ実数型タグ
:IOReal	I/O 実数型タグ
:MemoryMsg	メモリ メッセージ型タグ
:IOMsg	I/O メッセージ型タグ
:GroupVar	グループ変数タグ変数
:HistoryTrend	履歴トレンド タグ変数
:TagID	タグ変数 ID タグ変数
:IndirectDisc	間接論理型タグ
:IndirectAnalog	間接アナログ型タグ
:IndirectMsg	間接メッセージ型タグ

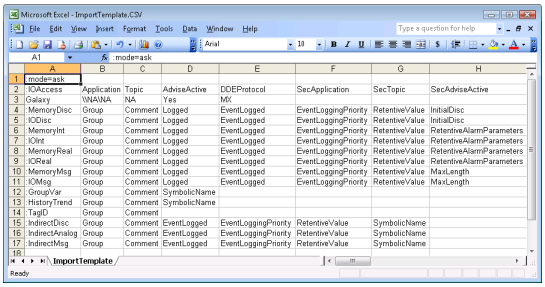
各キーワードには、アクセス名、アラーム グループ、およびタグ変数のプロパティを指定する、関連付けられた一組の属性が含まれています。たとえば、:IOAccess キーワードには、すべての InTouch アクセス名のプロパティであるアプリケーション、トピック、および通信プロトコルを指定する属性が含まれています。

インポート ファイル テンプレートの作成

.csv ファイル形式をサポートする任意のアプリケーションを使用して、タグ名ディクショナリ インポート ファイルを手動で作成できます。ただし、インポート ファイル全体を作成することは時間のかかる作業であり、エラーが発生しやすくなります。テンプレートとして既存の .csv ファイルを使用する方が、処理が速く信頼性も高くなります。

次の図は、DBDump で作成されたテンプレート インポート ファイルを示しています。この図は、ウィンドウもタグも持たない InTouch アプリケーションから作成されたファイルを示しています。作成されたファイルには、必須キーワードとタグ データのない属性のみが含まれます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



テンプレートを作成した後で、タグのタイプを識別するキーワードの下にタグ データを手動で追加します。タグ タイプのキーワードに関連付けられた対応する属性カラムに、タグのプロパティを挿入します。

テンプレート インポート ファイルを作成する

1. アプリケーション マネージャを開きます。
2. 新しい InTouch アプリケーションを作成します。

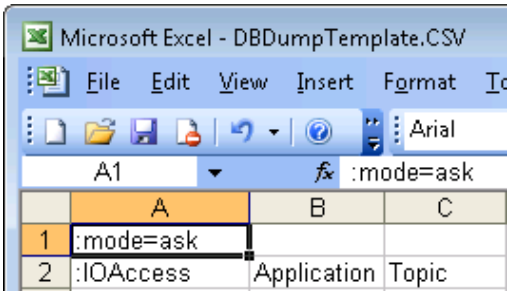
   アプリケーションを作成する手順の詳細については、「InTouch アプリケーションの作成」を参照してください。
3. アプリケーション マネージャに表示されるリストから、新しいアプリケーションを選択します。
4. DBDump ユーティリティを使用してアプリケーションのタグ名ディクショナリの内容をエクスポートします。

   タグのエクスポートの詳細については、「タグ定義をエクスポートする」を参照してください。
5. インポートするタグ データを挿入するファイルを編集します。

ディクショナリ インポート ファイルの操作モードの設定

インポート ファイルからアプリケーションのタグ変数ディクショナリにデータをロードしている間に、重複したタグ変数レコードを DBLoad が処理する方法を指定する必要があります。

DBDump を使用して作成されたインポート ファイル テンプレートを使用する場合、ファイルの最初の行には **:mode** キーワードが含まれます。たとえば、Excel アプリケーションのセル A1 で値 ask を :mode キーワードに割り当てることができます。



次の値を :mode キーワードに割り当てることができます。

```
:MODE=REPLACE
:MODE=UPDATE
:MODE=ASK
:MODE=IGNORE
:MODE=TERMINATE
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
:MODE=TEST
```

:MODE=REPLACE

重複したタグ変数が見つかった場合、DBLoad ユーティリティはタグ変数ディクショナリで既存のタグ変数を削除して、同じ名前を持つインポート ファイルのタグ変数で置き換えます。

:MODE=UPDATE

重複したタグ変数が見つかった場合、DBLoad ユーティリティはタグ変数ディクショナリの既存のタグ変数定義をインポート ファイルから明示的に指定されたデータのみで上書きします。タグ変数に関連付けられたその他のすべてのデータは、タグ変数ディクショナリでそのまま維持されます。

フィールドがレコード内にあり、ユーザーによって入力されたか、「:KEYWORD=value」メカニズムで設定されている場合、そのフィールドは明示的に定義されたものとみなされます。フィールドがレコードで指定されておらず、キーワードが「:KEYWORD=」コマンドを使用して再設定されている場合、現在のフィールド値は更新されません。

以下の例は、update モードのインポート ファイルがタグ変数ディクショナリにロード／マージされたときに何が起こるかを示したものです。

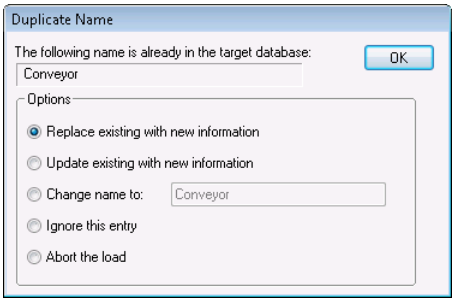
```
:Mode=update
:Group=Group1
:IODisc,Group,DConversion
Tagname1,Group2,
; Tagname1 の Group のみ Group2 に更新します
Tagname2,,
; Tagname2 の Group を Group1 に更新し、DConversion はそのままにします
Tagname3,,Reverse
; Tagname3 の Group を Group1 に更新し、DConversion を "Reverse" に更新します
; 以下の行は Group フィールドをデフォルト値に再設定します
:Group=
; データ フィールド "Group" がデフォルト値に再設定されます
Tagname4,,
; Tagname4 はそのまま残ります
```

タグ変数を変更中で、そのタグ変数が使用中である場合、タグ変数タイプは互換性がある必要があります。たとえば、タグ変数がアプリケーションによって使用されている場合は、既存の履歴トレンド タグ変数を I/O 整数型に変更することはできません。また、タグ変数がアプリケーションの入力リンクに使用されている場合、このタグ変数を ReadOnly=yes に変更することはできません。これらの制限により、DBLoad ユーティリティを使用する前にインポート先アプリケーションで使用タグの検索をしてください。

:MODE=ASK

タグ変数ディクショナリをロードしている間に重複したタグ変数が見つかったとき、DBLoad が停止します。［**同一名称タグ変数の処理**］ダイアログ ボックスが表示され、重複したタグ変数を処理するオプションのリストが表示されます。これは、デフォルトのインポート モードです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



重複を処理するオプション：

- 既存のタグ変数レコードをインポート ファイルのレコードに置き換えるには、［**新しい情報で置き換える**］を選択します。
- 既存のタグ変数レコードをインポート ファイルで明示的に定義されたフィールドでのみ上書きするには、［**新しい情報のみ書き込む**］を選択します。
- インポートされたタグ変数の名前を［**同一名称タグ変数の処理**］ダイアログ ボックスのボックスに入力した名前で置き換えるには、［**名称を変更**］を選択します。
- タグ変数を無視して、ファイルの内容のインポートを続行するには、［**入力を無視**］を選択します。
- インポート処理をキャンセルするには、［**ロード処理を中止する**］を選択します。

:MODE=IGNORE

DBLoad インポート ユーティリティは重複したタグ変数を無視し、インポート ファイルの残りのレコードの処理を続行します。

:MODE=TERMINATE

重複したタグ変数が見つかったとき、DBLoad インポート操作は停止します。

:MODE=TEST

DBLoad はインポート ファイルをスキャンしてエラーをチェックしますが、タグ変数定義をタグ変数ディクショナリにロードしません。DBLoad は、インポート ファイル内の行番号と場所の書式設定エラーを識別するレポートを生成します。

最初に、**:mode=test** を使用して DBLoad を実行し、インポート ファイルのエラーを識別します。エラーをで修正した後、DBLoad を実行する前に、**mode** キーワード値を **:mode=replace** または **:mode=update** に変更します。

アクセス名とアラーム グループの設定

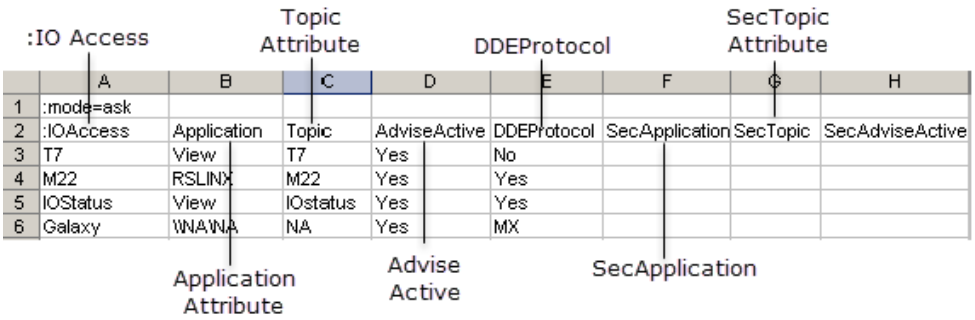
DBLoad インポート ファイルには、InTouch アプリケーションの定義済みアクセス名とアラーム グループを指定するキーワードが含まれています。

:IOAccess キーワード属性

:IOAccess キーワードは、InTouch アプリケーションで定義されたアクセス名を識別します。**:IOAccess** キーワードには、定義された InTouch アクセス名の特性を記述する一組の属性が含まれています。

以下の図は、アクセス名が :IOAccess キーワードを使用して Excel スプレッドシートで定義される方法を示しています。属性は個別のスプレッドシート カラムで左から右へ指定されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



	A	B	C	D	E	F	G	H
1	:mode=ask							
2	:IOAccess	Application	Topic	AdviseActive	DDEProtocol	SecApplication	SecTopic	SecAdviseActive
3	T7	View	T7	Yes	No			
4	M22	RSLINX	M22	Yes	Yes			
5	IOStatus	View	IOstatus	Yes	Yes			
6	Galaxy	WNAWNA	NA	Yes	MX			

以下の表は、:IOAccess キーワードに関連付けられた属性のリストを示しています。この表では、DBDump ユーティリティを使用して作成されたテンプレート インポート ファイルを使用するときに指定された順序で属性を一覧表示しています。

文字列位置	属性	許可できる値	デフォルト値
1	アプリケーション	アクセス名に対して定義されたアプリケーション名	なし
2	トピック	アクセス名に対して定義されたトピック名	なし
3	AdviseActive	サーバーからポーリングする情報 No = 全てのアイテムを要求 Yes = 有効アイテムのみ要求	可
4	DDEProtocol	アクセス名に対して定義された通信プロトコル No = Suitelink Yes = DDE <1>MX = </1>Message Exchange	不可
5	SecApplication	アクセス名のセカンダリ ソースに対して定義されたアプリケーション名	なし
6	SecTopic	アクセス名のセカンダリ ソースに対して定義されたトピック名	なし
7	SecAdviseActive	セカンダリ サーバーに保存された情報をポーリングするとき No = 全てのアイテムを要求 Yes = 有効アイテムのみ要求	なし

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性	許可できる値	デフォルト値
8	SecDDEProtocol	アクセス名のセカンダリ ソースに対して定義された通信プロトコル No = Suitelink Yes = DDE <1>MX = </1>Message Exchange	なし
9	FailoverExpression	TRUE の場合にアクセス名をセカンダリ ソースにを切り換えるフェイルオーバー式	なし
10	FailoverDeadband	アクセス名によって定義されたセカンダリ ソースに対してフェイルオーバーを開始する前の秒を表す整数	なし
11	DFOFlag	フェイルオーバー不可フラグ Yes = フェイルオーバー不可フラグの設定 No = フェイルオーバー不可フラグの設定なし	なし
12	FBDFlag	メインへ切り替えるフラグ Yes = フェイルオーバー条件の解除時にメインへ切り替える No = フェイルオーバー条件の解除時にメインへ切り替えない	なし
13	FailbackDeadband	フェイルオーバー条件の解除時にメインのアクセス名ソースに切り替える前の秒を表す整数	値なし

:AlarmGroup キーワード属性

DBLoad インポート ファイルには、InTouch アプリケーション用に定義されたアラーム グループを識別するキーワードが含まれています。**:AlarmGroup** キーワードには、InTouch アプリケーションのアラーム グループの特性を記述する一組の属性が含まれています。

以下の表は、:AccessGroup キーワードに関連付けられた属性のリストを示しています。この表では、DBDump ユーティリティを使用して作成されたテンプレート インポート ファイルを使用するときに指定された順序で属性を一覧表示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性	許可できる値	デフォルト値
1	グループ	アラーム グループの名前	$System
2	コメント	アラーム グループに割り当てられたコメント 任意のテキスト文字列	なし
3	EventLogged	イベント ログ記録の有効化または無効化 Yes または On = イベントログ記録が有効 No または Off = イベント ログ記録が無効	不可
4	EventLoggingPriority	イベントに割り当てられた優先度 1 ～ 999、0 = ログ記録なし	0
5	LoLoAlarmDisable	LoLo アラームの無効化または有効化 0 = LoLo アラームが有効 1 = LoLo アラームが無効	0
6	LoAlarmDisable	Lo アラームの無効化または有効化 0 = Lo アラームが有効 1 = Lo アラームが無効	0
7	HiAlarmDisable	Hi アラームの無効化または有効化 0 = Hi アラームが有効 1 = Hi アラームが無効	0
8	HiHiAlarmDisable	HiHi アラームの無効化または有効化 0 = HiHi アラームが有効 1 = HiHi アラームが無効	0
9	MinDevAlarmDisable	小偏差アラームの無効化または有効化 0 = 小偏差アラームが有効 1 = 小偏差アラームが無効	0

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性	許可できる値	デフォルト値
10	MajDevAlarmDisable	大偏差アラームの無効化または有効化 0 = 大偏差アラームが有効 1 = 大偏差アラームが無効	0
11	RocAlarmDisable	変化率（ROC）アラームの無効化または有効化 0 = ROC アラームが有効 1 = ROC アラームが無効	0
12	DSCAlarmDisable	論理型アラームの無効化または有効化 0 = 論理型アラームが有効 1 = 論理型アラームが無効	0
13	LoLoAlarmInhibititor	LoLo アラームの抑止に使用されるタグ変数の名前 タグ リファレンス : 任意の論理型またはアナログ型タグ変数	なし
14	LoAlarmInhibitor	Lo アラームの抑止に使用されるタグ変数の名前 タグ リファレンス : 任意の論理型またはアナログ型タグ変数	なし
15	HiAlarmInhibitor	Hi アラームの抑止に使用されるタグ変数の名前 タグ リファレンス : 任意の論理型またはアナログ型タグ変数	なし
16	HiHiAlarmInhibitor	HiHi アラームの抑止に使用されるタグ変数の名前 タグ リファレンス : 任意の論理型またはアナログ型タグ変数	なし
17	MinDevAlarmInhibitor	小偏差アラームの抑止に使用されるタグ変数の名前 タグ リファレンス : 任意の論理型またはアナログ型タグ変数	なし

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性	許可できる値	デフォルト値
18	MajDevAlarmInhibitor	大偏差アラームの抑止に使用されるタグ変数の名前 タグ リファレンス：任意の論理型またはアナログ型タグ変数	なし
19	RocAlarmInhibitor	変更率アラームの抑止に使用されるタグ変数の名前 タグ リファレンス：任意の論理型またはアナログ型タグ変数	なし
20	DSCAlarmInhibitor	論理型アラームの抑止に使用されるタグ変数の名前 タグ リファレンス：任意の論理型またはアナログ型タグ変数	なし

タグ変数タイプのキーワードと属性の定義

タグ変数レコードは、タグ変数のタイプを識別するキーワード行で開始します。各タグ変数キーワードには、タグ変数のタイプに関連付けられたデータの特性を指定するための属性の固有セットが含まれています。

以下の例では、**:IODisc** キーワードが I/O 論理型タグ変数タイプを識別します。キーワード行の残りの値は、I/O 論理型タグ変数に関連付けられたデータの属性を識別します。この例では、メモ帳を使用してネイティブのカンマ区切り文字列形式でファイルの内容が示されています。

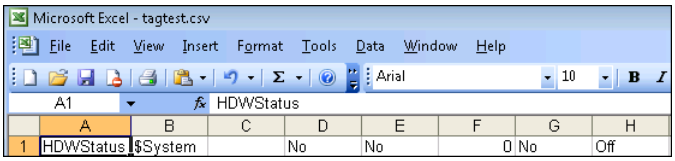
```
:IODisc,Group,Comment,Logged,EventLogged, EventLoggingPriority,RetentiveValue,InitialDis,
OffMsg,OnMsg,AlarmState,AlarmPri,DConversion, AccessName,ItemUseTagname,ItemName,ReadOnly,
AlarmComment,AlarmAckModel,DSCAlarmDisable, DSCAlarmInhibitor,SymbolicName
```

タグ変数タイプのキーワード行の下では、個別の行がそのタイプのタグ変数を一組の属性値で指定しています。以下の例では、**HDWStatus** タグ変数はインポート ファイルの I/O 論理型タグ変数タイプに属しています。

```
"HDWStatus","$System","",No,No,0,No,Off,"","",,1,Direct,"HistdataViewstr",No,"Status",No,""
,0,0,"",""
```

このレコードでは、引用符を使用して空白文字列を識別しています。

以下の図は、Excel スプレッドシートの同じインポート ファイル データを示しています。**Comment** セルは、タグ変数コメントがインポート ファイルで指定されていないため空白です。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タグ変数キーワード属性

以下の表は、InTouch タグ変数キーワードに関連付けられたすべての属性を一覧表示しています。この表には、各タグ変数属性とそのデフォルト値に関連付けられたデータのタイプを記述するカラムが含まれています。

これらのタグ変数属性は、付随するタグ変数データが対応する属性に一致する限り、DBLoad インポート ファイルに任意の順序で指定できます。たとえば、Excel インポート ファイルに :IODisc キーワードを挿入する場合、すべての I/O 論理型タグ変数の工学単位を EngUnits 属性と同じ Excel カラムに配置する必要があります。

属性	許可できる値	デフォルト値
アクセス名	タグ変数に割り当てられた InTouch アクセス名	なし
AlarmAckModel	アラーム確認モデル 整数型 0 = 条件 1 = イベント指向 2 = 拡張サマリ	0
AlarmComment	タグ変数に割り当てられたアラーム コメント テキスト文字列	なし
AlarmDevDeadband	タグ変数偏差アラーム デッドバンド 実数型	なし
AlarmPri	タグ変数に割り当てられたアラーム優先度 1 <1>～ </1>999	1
AlarmState	タグ変数のアラーム状況 オン（On）、オフ（Off）、なし（None）	なし
AlarmValueDeadband	タグ変数アラーム デッドバンド 実数型	0
コメント	タグ変数に割り当てられたコメント テキスト文字列	なし
コンバージョン	タグ変数値変換 線形（Linear）または二乗根（Square Root）	線形
デッドバンド	タグ変数に割り当てられた値のデッドバンド 実数型	0

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性	許可できる値	デフォルト値
DevTarget	タグ変数の偏差目標値 実数型	0
DSCAlarmDisable	論理型アラームの無効化または有効化 0 = 論理型アラームが有効 1 = 論理型アラームが無効	0
DSCAlarmInhibitor	論理型アラームの抑止に使用されるタグ変数の名前	なし
工学単位	タグ変数に割り当てられた工学単位 テキスト文字列	なし
EventLogged	イベント ログ記録の有効化または無効化 Yes または On = イベントログ記録が有効 No または Off = イベント ログ記録が無効	不可
EventLogging	タグ変数イベントログ記録の有効化または無効化 No または Off = ログ記録が無効 Yes または On = ログ記録が有効	不可
EventLoggingPriority	イベントに割り当てられた優先度 1 ～ 999、0 = ログ記録なし	0
グループ	タグ変数が属すアラーム グループの名前	$System
HiAlarmDisable	Hi アラームの無効化または有効化 0 = Hi アラームが有効 1 = Hi アラームが無効	0
HiAlarmInhibitor	Hi アラームの抑止に使用されるタグ変数の名前 任意の論理型またはアナログ型タグ変数	なし
HiAlarmPri	Hi アラームに割り当てられた優先度 1 <1>～ </1>999	1
HiAlarmState	Hi アラームの有効化または無効化 No または Off = 無効 Yes または On = 有効	不可

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性	許可できる値	デフォルト値
HiAlarmValue	タグ変数に割り当てられた Hi アラーム ポイント 実数型	0
HiHiAlarmDisable	HiHi アラームの無効化または有効化 0 = HiHi アラームが有効 1 = HiHi アラームが無効	0
HiHiAlarmInhibitor	HiHi アラームの抑止に使用されるタグ変数の名前 任意の論理型またはアナログ型タグ変数	なし
HiHiAlarmPri	HiHi アラームに割り当てられた優先度 1 <1>～ </1>999	1
HiHiAlarmState	HiHi アラームの有効化または無効化 No または Off = 無効 Yes または On = 有効	不可
HiHiAlarmValue	タグ変数に割り当てられた HiHi アラーム ポイント 実数型	0
InitialDisc	論理型タグ変数に割り当てられた初期値 0、Off、False、または No = オフ 1、On、True、または Yes = オン	0
InitialMessage	初期タグ変数メッセージ テキスト文字列	なし
InitialValue	タグ変数に割り当てられた初期値 実数型	0
ItemName	タグ変数に割り当てられたアイテムの名前 テキスト文字列	なし
ItemUseTagname	［タグ変数名をアイテム名として使用］オプションの有効化または無効化 No または False = 無効 Yes または True = 有効	不可

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性	許可できる値	デフォルト値
LoAlarmDisable	Lo アラームの無効化または有効化 0 = Lo アラームが有効 1 = Lo アラームが無効	0
LoAlarmInhibitor	Lo アラームの抑止に使用されるタグ変数の名前 任意の論理型またはアナログ型タグ変数	なし
LoAlarmPri	Lo アラームに割り当てられた優先度 1 <1>～ </1>999	1
LoAlarmState	Lo アラームの有効化または無効化 No または Off = 無効 Yes または On = 有効	不可
LoAlarmValue	タグ変数に割り当てられた Lo アラーム ポイント 実数型	0
LogDeadband	タグ変数に割り当てられたログ記録のデッドバンド 実数型	0
Logged	タグ変数値のログ記録の有効化または無効化 No または Off = ログ記録が無効 Yes または On = ログ記録が有効	不可
LoLoAlarmDisable	LoLo アラームの無効化または有効化 0 = LoLo アラームが有効 1 = LoLo アラームが無効	0
LoLoAlarmInhibitor	LoLo アラームの抑止に使用されるタグ変数の名前 任意の論理型またはアナログ型タグ変数	なし
LoLoAlarmPri	LoLo アラームに割り当てられた優先度 1 <1>～ </1>999	1
LoLoAlarmState	LoLo アラームの有効化または無効化 No または Off = 無効 Yes または On = 有効	不可

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性	許可できる値	デフォルト値
LoLoAlarmValue	タグ変数に割り当てられた LoLo アラームポイント 実数型	0
MajDevAlarmDisable	大偏差アラームの無効化または有効化 0 = 大偏差アラームが有効 1 = 大偏差アラームが無効	0
MajDevAlarmInhibitor	大偏差アラームの抑止に使用されるタグ変数の名前 任意の論理型またはアナログ型タグ変数	なし
MajorDevAlarmPri	大偏差アラームに割り当てられた優先度 1 <1>～ </1>999	1
MajorDevAlarmState	大偏差アラームの無効化または有効化 No または Off = 無効 Yes または On = 有効	不可
MajorDevAlarmValue	タグ変数に割り当てられた大偏差アラームのパーセンテージ 実数型	0
MaxEU	タグ変数に割り当てられた最大工学単位値 実数型	32767
最大長	メッセージの最大文字数 実数型	131
MaxRaw	タグ変数に割り当てられた生データ最大値 実数型	32767
最大値	タグ変数に割り当てられた最大値 実数型	32767
MinDevAlarmDisable	小偏差アラームの無効化または有効化 0 = 小偏差アラームが有効 1 = 小偏差アラームが無効	0

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性	許可できる値	デフォルト値
MinDevAlarmInhibitor	小偏差アラームの抑止に使用されるタグ変数の名前 任意の論理型またはアナログ型タグ変数	なし
MinEU	タグ変数に割り当てられた最小工学単位値 実数型	-32768
MinorDevAlarmPri	小偏差アラームに割り当てられた優先度 1 <1>～ </1>999	1
MinorDevAlarmState	小偏差アラームの有効化または無効化 No または Off = 無効 Yes または On = 有効	不可
MinorDevAlarmValue	タグ変数に割り当てられた小偏差アラームのパーセンテージ 実数型	0
MinRaw	タグ変数に割り当てられた生データ最小値 実数型	-32768
最小値	タグ変数に割り当てられた最小値 実数型	-32768
OffMsg	論理型タグ変数オフ メッセージ テキスト文字列	なし
OnMsg	論理型タグ変数オン メッセージ テキスト文字列	なし
読み取り専用	タグ変数値読み取り専用または読み取り／書き込み Yes = 読み取り専用 No = 読み取り／書き込み	不可
RetentiveAlarmParameters	タグ変数のパラメータ値保持の有効化または無効化 No または Off = 無効 Yes または On = 有効	不可

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性	許可できる値	デフォルト値
RetentiveValue	タグ変数の変数値保持の有効化または無効化 0、Off、False、または No = 無効 1、On、True、または Yes = 有効	不可
RocAlarmDisable	変化率（ROC）アラームの無効化または有効化 0 = ROC アラームが有効 1 = ROC アラームが無効	0
RocAlarmInhibitor	変更率アラームの抑止に使用されるタグ変数の名前 任意の論理型またはアナログ型タグ変数	なし
ROCAlarmPri	変化率アラームに割り当てられた優先度 1 <1>～ </1>999	1
ROCAlarmState	変化率アラームの有効化または無効化 No または Off = 無効 Yes または On = 有効	不可
ROCAlarmValue	パーセントによるタグ変数値の変更 実数型	0
ROCTimeBase	変化率を計算する測定期間 秒（Sec）、分（Min）、時（Hr）	最小
SymbolicName	S7 Tag Creator 製品によって入力データ ブロックに割り当てられたシンボリック名。シンボリック名は S7 Tag Creator のシンボル表にリストされています。	なし

:MemoryDisc キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできるメモリ論理型タグ変数を定義するための :MemoryDisc キーワードが含まれています。以下の表は、メモリ論理型タグ変数のプロパティに関連付けられた :MemoryDisc キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:MemoryDisc キーワード属性が指定される順序を示しています。属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
1	グループ
2	コメント
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	InitialDisc
8	OffMsg
9	OnMsg
10	AlarmState
11	AlarmPri
12	AlarmComment
13	AlarmAckModel
14	DSCAlarmDisable
15	DSCAlarmInhibitor
16	SymbolicName

:IODisc キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできる I/O 論理型タグ変数を定義するための :IODisc キーワードが含まれています。以下の表は、I/O 論理型タグ変数のプロパティに関連付けられた :IODisc キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:IODisc キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
7	InitialDisc
8	OffMsg
9	OnMsg
10	AlarmState
11	AlarmPri
12	コンバージョン
13	アクセス名
14	ItemUseTagname
15	ItemName
16	読み取り専用
17	AlarmComment
18	AlarmAckModel
19	DSCAlarmDisable
20	DSCAlarmInhibitor
21	SymbolicName

:MemoryInt キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできるメモリ整数型タグ変数を定義するための :MemoryInt キーワードが含まれています。以下の表は、メモリ整数型タグ変数のプロパティに関連付けられた :MemoryInt キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump ユーティリティが使用されるときに、:MemoryInt キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
8	AlarmValueDeadband
9	AlarmDevDeadband
10	工学単位
11	InitialValue
12	最小値
13	最大値
14	デッドバンド
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHIAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

文字列位置	属性
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable
42	LoAlarmDisable
43	HiAlarmDisable
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor
55	SymbolicName

:IOInt キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできる I/O 整数型タグ変数を定義するための :IOInt キーワードが含まれています。以下の表は、I/O 整数型タグ変数のプロパティに関連付けられた :IOInt キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:IOInt キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	Logged

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	工学単位
11	InitialValue
12	MinEU
13	MaxEU
14	デッドバンド
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHIAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

文字列位置	属性
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
39	MinRaw
40	MaxRaw
41	コンバージョン
42	アクセス名
43	ItemUseTagname
44	ItemName
45	読み取り専用
46	AlarmComment
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

文字列位置	属性
61	RocAlarmInhibitor
62	SymbolicName

:MemoryReal キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできるメモリ実数型タグ変数を定義するための :MemoryReal キーワードが含まれています。以下の表は、メモリ実数型タグ変数のプロパティに関連付けられた :MemoryReal キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:MemoryReal キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	工学単位
11	InitialValue
12	最小値
13	最大値
14	デッドバンド
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable
42	LoAlarmDisable
43	HiAlarmDisable
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

文字列位置	属性
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor
55	SymbolicName

:IOReal キーワード属性

DBLoad インポート ファイルには、タグ名ディクショナリにインポートできる I/O 実数型タグ変数を定義するための :IOReal キーワードが含まれています。以下の表は、I/O 実数型タグ変数のプロパティに関連付けられた :IOReal キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:IOReal キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	工学単位
11	InitialValue
12	MinEU
13	MaxEU
14	デッドバンド
15	LogDeadband
16	LoLoAlarmState

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	MinRaw
40	MaxRaw
41	コンバージョン
42	アクセス名
43	ItemUseTagname
44	ItemName
45	読み取り専用

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

文字列位置	属性
46	AlarmComment
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor
61	RocAlarmInhibitor
62	SymbolicName

:MemoryMsg キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできるメモリ メッセージ型タグ変数を定義するための :MemoryMsg キーワードが含まれています。以下の表は、メモリメッセージ型タグ変数のプロパティに関連付けられた :MemoryMsg キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:MemoryMsg キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	Logged
4	EventLogged
5	EventLoggingPriority

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
6	RetentiveValue
7	最大長
8	InitialMessage
9	AlarmComment
10	SymbolicName

:IOMsg キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできる I/O メッセージ型タグ変数を定義するための :IOMsg キーワードが含まれています。以下の表は、I/O メッセージ型タグ変数のプロパティに関連付けられた :IOMsg キーワードの属性を一覧表示しています。

この表には、インポート ファイルを作成するために DBDump が使用されるときに、:IOMsg キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	最大長
8	InitialMessage
9	アクセス名
10	ItemUseTagname
11	ItemName
12	読み取り専用
13	AlarmComment
14	SymbolicName

:GroupVar キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできるグループ変数タグ変数を定義するための :GroupVar キーワードが含まれています。以下の表は、グループ変数タグ変数のプロパティに関連付けられた :GroupVar キーワードの属性を一覧表示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：InTouch グループ変数タグ変数は現在使用されていません。:GroupVar キーワードは、従来のアプリケーションのみをサポートするために含まれています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:GroupVar キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	SymbolicName

:HistoryTrend キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできる履歴トレンド タグ変数を定義するための :HistoryTrend キーワードが含まれています。以下の表は、履歴トレンド タグ変数のプロパティに関連付けられた :HistoryTrend キーワードの属性を一覧表示しています。

この表には、インポート ファイルを作成するために DBDump が使用されるときに、:HistoryTrend キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	SymbolicName

:TagID キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできるタグ変数 ID タグ変数を定義するための :TagID キーワードが含まれています。以下の表は、タグ変数 ID タグ変数のプロパティに関連付けられた :TagID キーワードの属性を一覧表示しています。

この表には、インポート ファイルを作成するために DBDump が使用されるときに、:TagID キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント

:IndirectDisc キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできる間接論理型タグ変数を定義するための :IndirectDisc キーワードが含まれています。以下の表は、間接論理型タグ変数のプロパティに関連付けられた :IndirectDisc キーワードの属性を一覧表示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:IndirectDisc キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

:IndirectAnalog キーワード属性

DBLoad インポートファイルには、タグ名ディクショナリにインポートできる間接アナログ型タグ変数を定義するための :IndirectAnalog キーワードが含まれています。以下の表は、間接アナログ型タグ変数のプロパティに関連付けられた :IndirectAnalog キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:IndirectAnalog キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

:IndirectMsg キーワード属性

DBLoad インポート ファイルには、タグ変数ディクショナリにインポートできる間接メッセージ型タグ変数を定義するための :IndirectMsg キーワードが含まれています。以下の表は、間接メッセージ型タグ変数のプロパティに関連付けられた :IndirectMsg キーワードの属性を一覧表示しています。

この表は、インポート ファイルを作成するために DBDump が使用されるときに、:IndirectMsg キーワード属性が指定される順序を示しています。これらの属性とそのデフォルト値に関連付けられているデータについては、「タグ変数キーワード属性」を参照してください。

文字列位置	属性
1	グループ
2	コメント

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列位置	属性
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

インポート ファイルでの空白文字列の使用

ディクショナリ インポート ファイルの場合、空白文字列を含むフィールドとデータを含まないフィールドの間には違いがあります。空白文字列を割り当てられるキーワード属性：

コメント	工学単位	**OffMsg**
InitialMessage	OnMsg	アプリケーション
ItemName	トピック	

以下の例で、空白文字列は引用符（“ “）によって示されています。

```
:Comment="Hi"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,"",$System
```

ここで、

Tagname1 の Comment フィールドの値は Hi で、Tagname2 の Comment フィールドの値は空白コメントです。

Microsoft Excel はファイルを保存するとき空白文字列を示す引用符を無視し、以下の結果となります。

```
:Comment="Hi"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,,$System
```

空白文字列が Excel で使用されていることを確認するには、属性値としてセルにスペースを入力します。

フィールドに対するデフォルト値の使用

キーワードを使用して、レコードの特定フィールドに対するデフォルト値を設定できます。デフォルト値は、タグ変数タイプの元々の InTouch の値です。たとえば、メモリ論理型タグ変数はデフォルト値として Group=$System、EventLogging=Off、InitialValue=Off を使用します。

以下に例を示します。

```
:KEYWORD=値
```

これによって、後続のデータ レコードすべてに対して参照されるフィールドのデフォルト値が設定されます。この機能は、多数のレコードに対して変化しないまま維持されるべきであるフィールドのデフォルト値を設定するために使用します。フィールドに定義されたデフォルト値が存在する場合、その値のレコードにデータが存在しない場合にデフォルト値が使用されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

たとえば、`:GROUP=Reactor_Site` を設定すると、GROUP カラムに空白エントリを持つすべてのタグ変数が Reactor_Site アラーム グループに割り当てられます。たとえば、タグ変数の GROUP カラムに $System が入力されている場合、このタグ変数はアラーム グループ $System に割り当てられます。

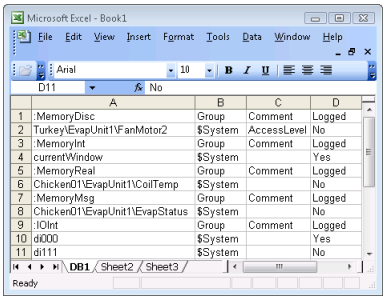
方程式の値を省略すると、個々のキーワードを元々のデフォルト値にリセットできます。たとえば、:GROUP= などです。

すべてのキーワードをリセットするには、:RESET コマンドを使用します。このコマンドは引数を持たず、コマンド後に発生するファイルのすべてのエントリに影響を与えます。

スーパータグ インスタンスの作成

アプリケーション マネージャ内の DBLoad ユーティリティを使用してスーパータグを作成できます。ただし、作成するスーパータグ インスタンスはスーパータグ テンプレート定義には反映されません。

有効なスーパー タグ形式を使用する必要があり、スーパータグ インスタンスのデータ レコードはタグ タイプの有効なキーワードで開始する必要があります。以下に例を示します。



次の例は有効な構文を示しています。

```
ParentInstance\ChildMember
ParentInstance\ChildMember\Submember
```

次の例は無効な構文を示しています。

```
ParentInstance\
ParentInstance\ChildMember\
```

無効な形式を使用すると、エラー メッセージが表示されます。

スーパータグ インスタンスを含む CSV ファイルをインポートすると、これらのインスタンスは自動的にタグ名ディクショナリに追加され、アニメーション リンクや InTouch QuickScript ですぐに使用できるようになります。

DBLoad を使用したタグ定義のインポート

ファイルの内容を DBLoad を使用してインポートするとき、すべてのタグ定義は選択した InTouch アプリケーションのタグ名ディクショナリにインポートされます。

インポートが失敗すると、失敗の理由を説明するメッセージが表示されます。このエラー メッセージは Logger に記述されます。

タグ定義を InTouch アプリケーションにインポートする

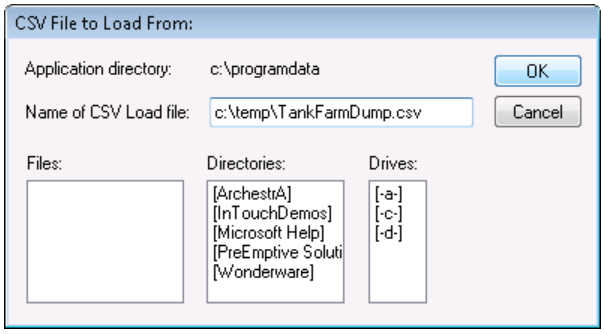
1. WindowMaker と WindowViewer を閉じます。
2. タグ名ディクショナリがインポート ファイルからのタグ定義でロードされるアプリケーションをバックアップします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. **アプリケーション マネージャ**を起動します。
4. タグ名ディクショナリがインポートされたタグ定義を受け取るリストからアプリケーションを選択します。
5. ［**ファイル**］メニューの［**データ**］グループで［**DBLoad**］を選択します。

   InTouch アプリケーションのバックアップを確認するよう要求するメッセージが表示されます。
6. ［**はい**］を選択して、アプリケーションがバックアップされたことを確認します。

   ［**ロード元 CSV ファイル名**］ダイアログ ボックスが表示されます。



7. ［**CSV ロード ファイル名**］ボックスで、インポートするファイルを探して、選択します。
8. ［**OK**］ を選択します。

   次の手順は、DBLoad が新しいタグ定義または既存のタグ定義のどちらをタグ名ディクショナリにインポートするかに基づいて異なります。

   - 新しいタグ定義をインポートする場合、新しいタグ データがアプリケーションのタグ名ディクショナリにロードされます。データが正常にロードされマージされたことを確認するメッセージが表示されます。
   - 既存のタグ定義をインポートする場合、:mode キーワードが :mode=ask に設定され、インポートファイルに重複したタグが含まれる場合、インポートは停止します。重複したタグを処理するオプションが表示されます。またはインポートをキャンセルすることもできます。キーワード オプションの詳細については、「ディクショナリ インポート ファイルの操作モードの設定」を参照してください。

DBLoad ユーティリティを使用したタグ定義のインポートにおける既知の制限

ファイル サイズが 2 GB を超える大きなタグ カウント newtag.tag ファイルをインポートする場合、DBLoad ユーティリティでファイルのインポートが失敗することや、パフォーマンスの問題が発生することがあります。

制限:

- newtags.tag ファイルのサイズが 2GB の制限を超える場合、DBLoad で大きなタグ カウント ファイルのインポートが失敗することがあります。
- 大きなタグ カウント CSV ファイルをインポートする場合、DBLoad でパフォーマンスの問題が発生します。

回避策:

以下のレジストリ エントリを使用して、newtags.tag ファイルへのタグのインポートを回避し、Tagname.x ファイルへのインポートを行います。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

1. レジストリ エディタを開きます。
2. **HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Wonderware\InTouch\Installation** に移動します。
3. ナビゲーション ペインのインストール フォルダを右クリックするか、メイン領域の任意の場所を右クリックして、［**新規**］、［**DWORD 値**］の順に選択します。
4. 作成されたファイルの名前を `DisableWriteToNewTag` に変更します。
5. “DisableWriteToNewTag” キーを 1 に設定します。
6. DBLoad を使用して CSV ファイルからタグをインポートし、インポートが正常に完了することを確認します。

インポート機能における DisableWriteToNewTag レジストリ キーのさまざまな値の影響を以下に示します。

DisableWriteToNewTag の DWORD 値	**インポートへの影響**
DisableWriteToNewTag レジストリ キーが存在し、DWORD 値が 1 に設定されている	DBLoad ユーティリティは newtags.tag ファイルへのインポートをスキップします
DisableWriteToNewTag レジストリ キーが存在し、DWORD 値が 0 に設定されている	DBLoad ユーティリティはタグを newtags.tag ファイルにインポートします

注意：InTouch HMI バージョン 2023 以降、レジストリ キー DisableWriteToNewTag はデフォルトで存在しないため、DBLoad ユーティリティは newtags.tag ファイルへのインポートをスキップします。

InTouch アプリケーション間での InTouch ウィンドウのエクスポートとインポート

WindowMaker の 3 つのすべての InTouch アプリケーション タイプからウィンドウをエクスポートできますが、エクスポートされたウィンドウをインポートする場合、または InTouch アプリケーションから直接のウィンドウをインポートする場合に、いくつかの制限があります。

- スタンドアロンの場合、産業用グラフィックが含まれているパブリッシュ済みおよびマネージド InTouch アプリケーションからはウィンドウをインポートできません。警告メッセージが表示され、インポートされなかったウィンドウに関する情報がロガーに書き込まれます。

  産業用グラフィックを含むマネージドまたはパブリッシュ済みの InTouch アプリケーションからウィンドウをインポートする場合、ウィンドウはインポートされますが、産業用グラフィックは機能せず、「見つかりません」と表示されます。

- マネージド InTouch アプリケーションの場合、パブリッシュ済み、スタンドアロン、およびその他のマネージド InTouch アプリケーションからウィンドウをインポートできます。埋め込まれた産業用グラフィックはインポートされません。
- パブリッシュ済み InTouch アプリケーションの場合、スタンドアロン InTouch アプリケーションからウィンドウをインポートできます。埋め込まれた産業用グラフィックはインポートされません。

ウィンドウのインポート

既存の InTouch アプリケーションから現在のアプリケーションにウィンドウをインポートすることによって、以前作成したウィンドウ、オブジェクト、およびウィンドウ スクリプトを再使用できるため、開発時間を削減できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウをインポートする前に、InTouch HMI ソフトウェアの現在のバージョンにアプリケーションを変換する必要があります。

デフォルトでは、インポートされたウィンドウに関連付けられたタグに対してプレースホルダが作成されます。インポート後、プレースホルダはローカル タグまたはリモート タグ参照に変換できます。詳細については、「インポートされたウィンドウとスクリプト用のタグ変数プレースホルダ」を参照してください。関連付けられたタグがインポート先アプリケーションにすでに存在する場合、インポート中にこれらのタグを代わりに使用することを選択できます。

SmartSymbol を含むウィンドウをインポートする際、既存のタグを使用することを選択する場合、タグがインポート先アプリケーションで使用できる場合でも、InTouch HMI は回復されたシンボルに対してプレースホルダを維持します。

スーパータグを含むアプリケーションからウィンドウをインポートする際、ウィンドウで実際に使用されたスーパータグのインスタンスのみが新しいアプリケーションにインポートされます。スーパータグのテンプレート構造全体はインポートされません。たとえば、アプリケーションにスーパータグ メンバーのタグが何百個も定義されている場合で、インポートされたウィンドウでは 50 個しかそれらのタグが使用されていない場合、それらの 50 個のタグのみがインポートされます。

重要：インポートまたはエクスポート以外の方法を使用して InTouch ウィンドウ ファイルを移動する場合、アプリケーションのタグ名ディクショナリの内容が壊れる可能性があります。

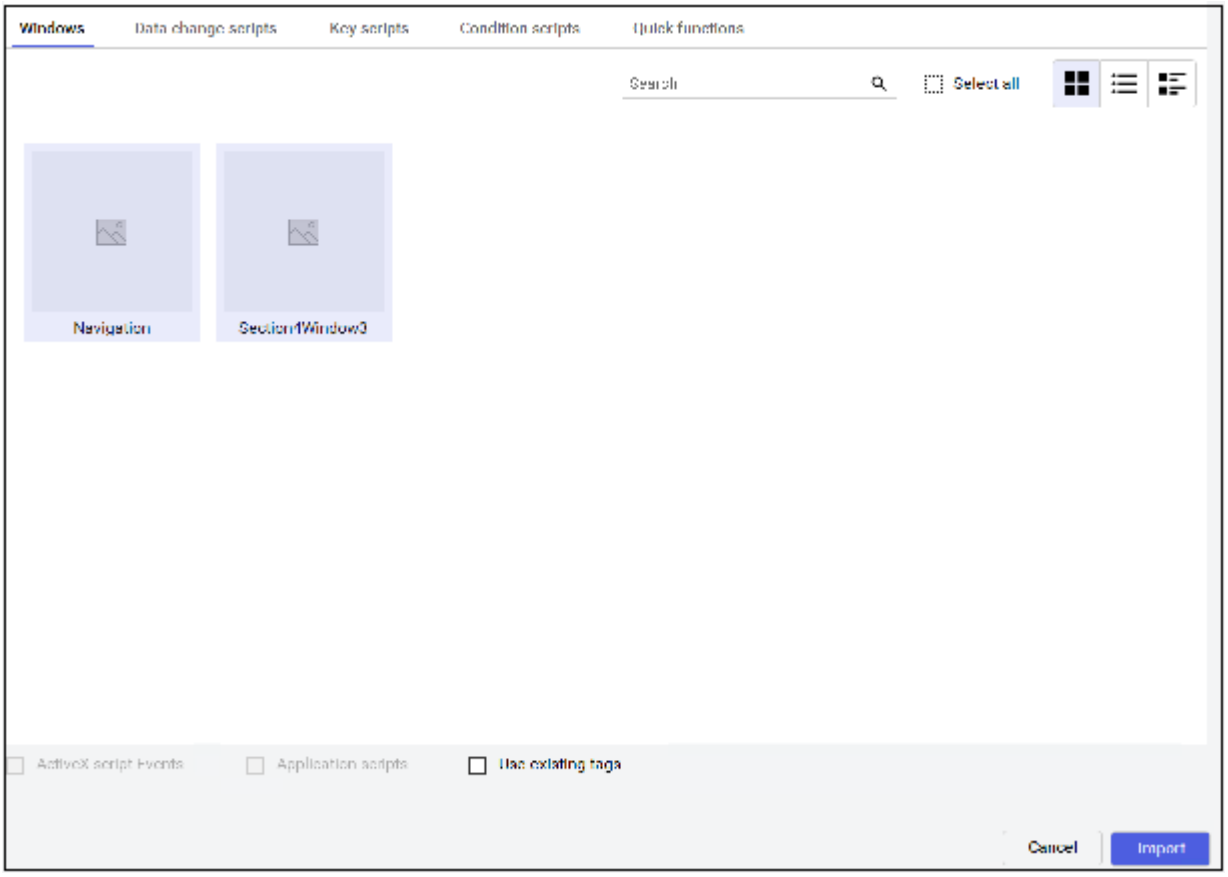
ウィンドウをインポートします

1. 現在使用中のアプリケーションのウィンドウをすべて閉じます。
2. ［**ファイル**］メニューで［**インポート**］、［**視覚化**］の順に選択してから［**ウィンドウとスクリプト**］を選択します。

   ［**フォルダを開く**］ダイアログ ボックスが表示されます。
3. インポートするウィンドウを含むアプリケーションのフォルダを選択します。
4. ［**OK**］ を選択します。

   ［**アプリケーション データ インポート オプション**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. ［ウィンドウ］タブでインポートする個々のウィンドウを選択します。
6. インポートされたウィンドウに関連付けられたタグがアプリケーションに既に存在し、それらのウィンドウをプレースホルダの代わりに使用する場合、［既存のタグの使用］チェック ボックスをオンにします。
7. **Import** を選択します。
8. プレースホルダのタグをローカル タグまたはリモート タグ参照に変換します。詳細については、「インポートされたウィンドウのプレースホルダ タグの変換」を参照してください。
9. インポートされたウィンドウに 1 つまたは複数のウィザードが含まれている場合、各ウィザードをダブルクリックして、プロパティ パネルを開きます。インポートされたウィンドウに 1 つまたは複数の SmartSymbol が含まれている場合、各 SmartSymbol を編集して、新しいインスタンスを作成します。

インポートされたウィンドウのプレースホルダ タグの変換

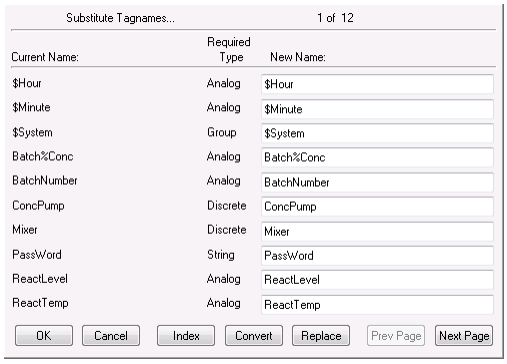
ウィンドウまたは QuickScript を現在のアプリケーションに対してインポートまたはエクスポートする際、そのウィンドウまたは QuickScript に関連付けられているすべてのタグはウィンドウと共に転送されます。ただし、タグは新しいアプリケーションのタグ名ディクショナリには追加されません。その代わりに、［プレースホルダの節約］オプションがインポート時に選択されていない限り、タグがプレースホルダのタグとして自動的にマーク付けされます。これらのプレースホルダ タグは変換して、必要に応じて新しいアプリケーションのタグ名ディクショナリで定義する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

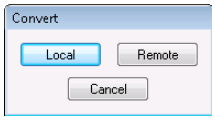
ウィンドウのタグを変換するには

1. WindowMaker でウィンドウを開きます。
2. F2 キーを押して、ウィンドウのすべてのオブジェクトを選択します。
3. ［**アニメーション**］メニューの［**置換**］グループで［**タグ**］を選択します。

   ［**タグ名の置換**］ダイアログ ボックスが表示されます。

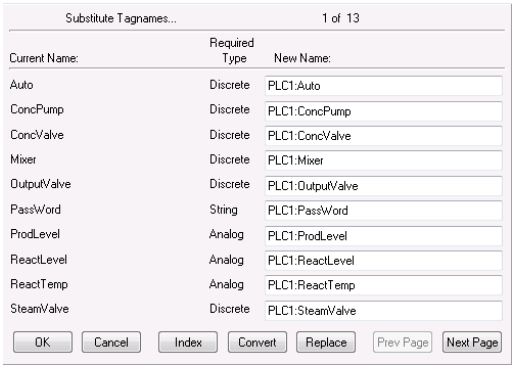


4. **変換**を選択します。［**変換**］ダイアログ ボックスが表示されます。



5. タグを変換します。
   - ［**ローカル**］を選択して、プレースホルダ タグ変数をローカル タグ変数に変換します。タグ名ディクショナリで各タグを定義するよう求めるメッセージが表示されます。
   - ［**リモート**］を選択して、プレースホルダ タグ変数をリモート タグ リファレンスに変換します。［**アクセス名**］ダイアログ ボックスが表示されます。アクセス名を選択して、［**閉じる**］を選択します。

   変換した後、［**タグ名の置換**］ダイアログ ボックスに新しいタグが表示されます。



6. ［**OK**］ を選択します。

ウィンドウのエクスポート

アプリケーション ウィンドウをエクスポートすると、以下の操作を実行できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- すべてのウィンドウのライブラリ アプリケーションを作成または維持する
- 別のアプリケーションでリモート タグ参照を作成する

ウィンドウをエクスポートする前に、InTouch HMI ソフトウェアの現在のバージョンにアプリケーションを変換する必要があります。

ウィンドウをエクスポートすると、そのウィンドウに関連付けられているオブジェクトやアニメーション リンクもすべてエクスポートされます。ウィンドウのオブジェクトに関連付けられたタグはプレースホルダ タグに変換され、エクスポート先アプリケーションの既存のタグが上書きされることを防ぎます。プレースホルダ タグの詳細については、「インポートされたウィンドウのプレースホルダ タグの変換」を参照してください。

重要：インポートまたはエクスポート以外の方法を使用して InTouch ウィンドウ ファイルを移動する場合、アプリケーションのタグ名ディクショナリが壊れる可能性があります。

ウィンドウをエクスポートします

1. 現在使用中のアプリケーションのウィンドウをすべて閉じます。
2. ［**ファイル**］メニューで［**エクスポート**］、［**視覚化**］の順に選択してから［**ウィンドウ**］を選択します。
3. エクスポートするウィンドウを選択して、［**エクスポート**］を選択します。

   ［**フォルダを開く**］ダイアログ ボックスが表示されます。
4. ウィンドウをエクスポートするアプリケーションのフォルダを選択します。
5. ［**OK**］ を選択します。
6. 問題が発生した場合、［**エクスポート実行時の問題**］ダイアログ ボックスが表示されます。実行する処理のオプションを選択し、［**OK**］ を選択します。

スクリプトのインポート

開発時間を節約するために、既存の QuickScript は InTouch アプリケーションから現在のアプリケーションにインポートできます。

スクリプトをインポートする前に、InTouch HMI ソフトウェアの現在のバージョンにアプリケーションを変換する必要があります。

デフォルトで、インポートされた QuickScript に関連付けられたタグに対してプレースホルダが作成されます。インポート後、プレースホルダはローカル タグまたはリモート タグ参照に変換できます。詳細については、「インポートされたウィンドウとスクリプト用のタグ変数プレースホルダ」を参照してください。関連付けられたタグがインポート先アプリケーションにすでに存在する場合、インポート中にこれらのタグを代わりに使用することを選択できます。

ウィンドウ スクリプトをインポートするには、ウィンドウ全体をインポートする必要があります。

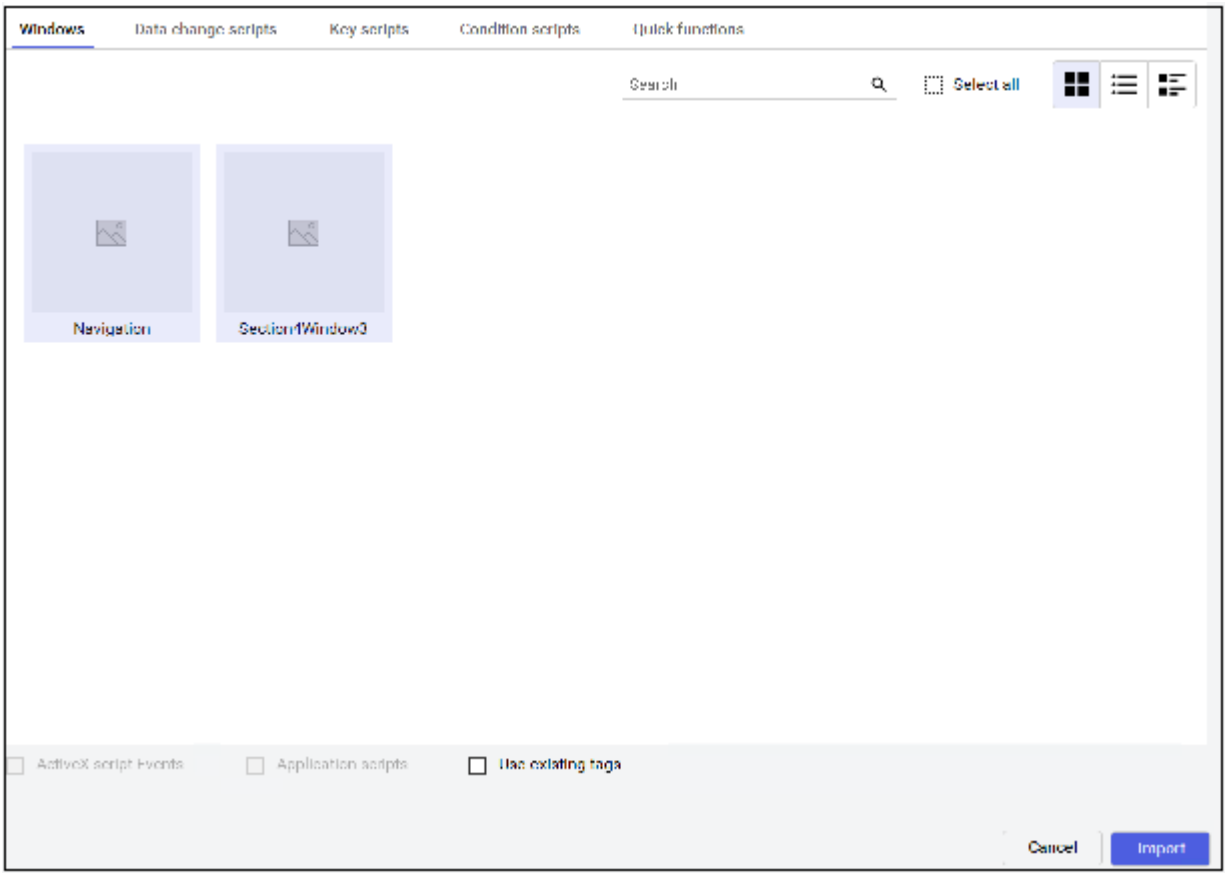
インポートされた ActiveX イベント スクリプトがインポート先のアプリケーションで正しく機能するためには、スクリプトが元々作成されたときと同じ ActiveX コントロールとイベントがインポート先のアプリケーションで使用される必要があるため、メモリにロードする必要があります。ActiveX コントロールを含むウィンドウが閉じられると、それに関連付けられたスクリプト（ActiveX イベント スクリプトまたは QuickScript）は正常に動作しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

QuickScript をインポートする

1. 現在使用中のアプリケーションのウィンドウをすべて閉じます。
2. ［ファイル］メニューで［インポート］、［視覚化］の順に選択してから［ウィンドウとスクリプト］を選択します。

   ［フォルダを開く］ダイアログ ボックスが表示されます。
3. インポートするスクリプトを含むアプリケーションのフォルダを選択します。
4. ［OK］ を選択します。［アプリケーション データ インポート オプション］ダイアログ ボックスが表示されます。



5. インポートするクイック関数タイプのチェック ボックスをオンにし、［選択］を選択して、インポートする個々のスクリプトを選択します。

   注意：ウィンドウ スクリプトをインポートするには、ウィンドウ全体をインポートする必要があります。詳細については、「ウィンドウのインポート」を参照してください。
6. インポートされたスクリプトに関連付けられたタグがアプリケーションに既に存在し、それらのスクリプトをプレースホルダの代わりに使用する場合、［既存のタグの使用］チェック ボックスをオンにします。
7. **Import** を選択します。アプリケーションに同じ名前を持つスクリプトがある場合、上書き、省略、または名前の変更などの操作を行うよう求めるメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

8. プレースホルダのタグをローカル タグまたはリモート タグ参照に変換します。詳細については、「インポートされたスクリプトでのプレースホルダ タグの変換」を参照してください。

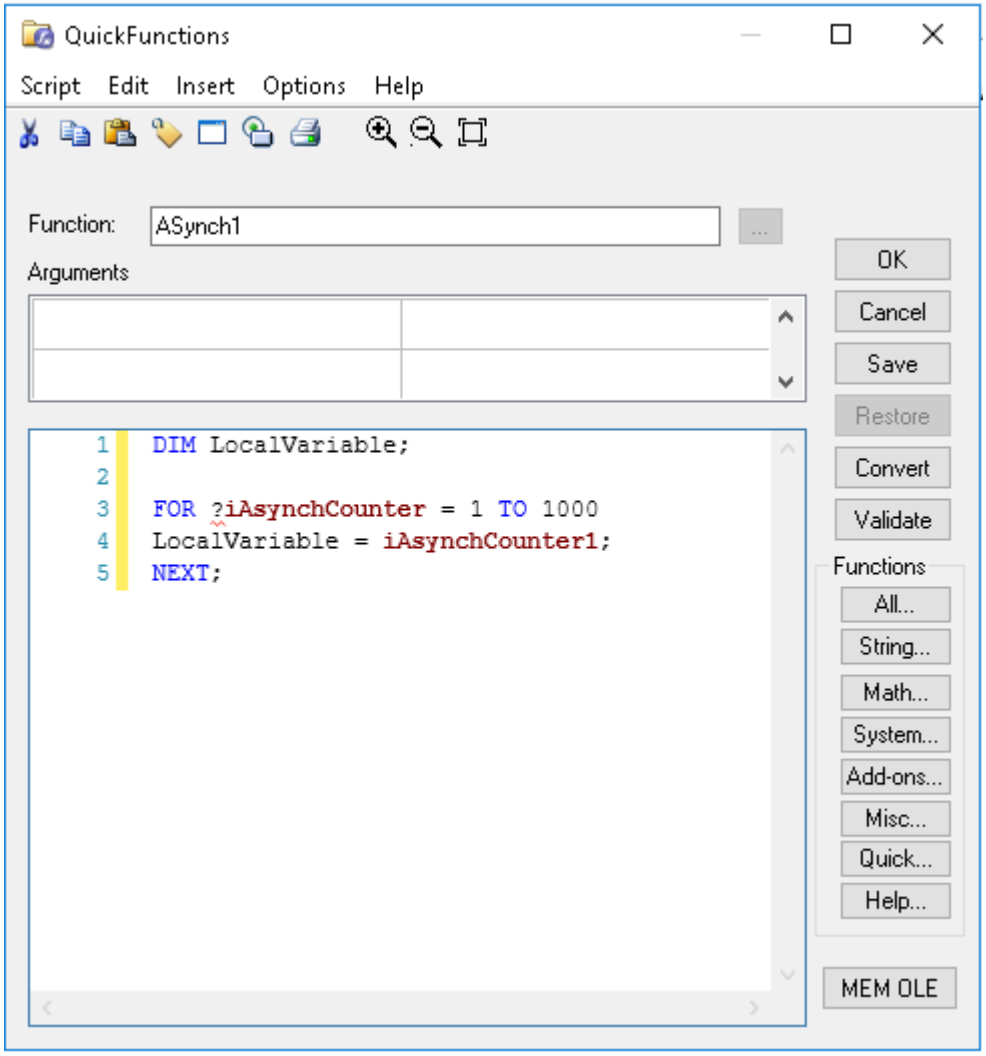
インポートされたスクリプトでのプレースホルダ タグの変換

クイック関数を現在のアプリケーションに対してインポートまたはエクスポートする際、そのクイック関数に関連付けられているすべてのタグは転送されます。ただし、タグは新しいアプリケーションのタグ名ディクショナリには追加されません。その代わりに、これらのタグは自動的にプレースホルダ タグとしてマーク付けされます。これらのプレースホルダ タグは変換して、必要に応じて新しいアプリケーションのタグ名ディクショナリで定義する必要があります。

インポートされたスクリプトでのプレースホルダ タグの変換

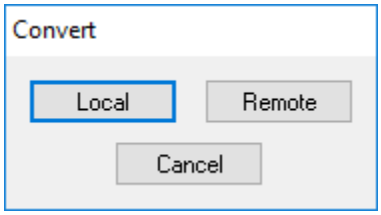
1. ［スクリプト］ペインで、インポートしたクイック関数のタイプを選択します。

   クイック関数スクリプト エディタが表示され、選択したスクリプト タイプのファイルの最初のクイック関数が表示されます。



2. 変換を選択します。［変換］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



1. タグを変換します。
   - ［**ローカル**］を選択して、プレースホルダ タグ変数をローカル タグ変数に変換します。タグ名ディクショナリで各タグを定義するよう求めるメッセージが表示されます。
   - ［**リモート**］を選択して、プレースホルダ タグ変数をリモート タグ リファレンスに変換します。［**アクセス名**］ダイアログ ボックスが表示されます。アクセス名を選択して、［**閉じる**］を選択します。
2. タグ変数が変換されたら、QuickScript エディタの［**OK**］を選択します。

インポートされたウィンドウとスクリプト用のタグ変数プレースホルダ

ウィンドウまたは QuickScript をインポートするとき、関連付けられたタグ変数を処理する方法を設定できます。

- **プレースホルダ タグ変数を使用します。**

  デフォルトで、インポートされたタグ変数は「プレースホルダ」（または「インデックス」）タグ変数に変換されます。最大 4096 のプレースホルダが許可されています。

  プレースホルダ タグ変数には、3 文字のプリフィックスが含まれます。たとえば、元のタグ変数が WaterHeater である場合、プレースホルダ タグ変数は ?d:WaterHeater です。

  30、31、または 32 文字を含むタグ変数をインポートする場合、プレースホルダのプリフィックスがタグ変数の先頭に追加され、既存のタグ変数の長さは切り捨てられません。たとえば、プレースホルダ タグ変数の場合に限り、32 文字のタグ変数は 35 文字に増加します。タグ変数の長さの増加は標準タグ変数ではサポートされていません。

  アプリケーションでプレースホルダ タグ変数を使用するには、以下のいずれかの手順に従います。

  標準の（ローカル）タグ変数に変換し、タグ変数ディクショナリで定義します。

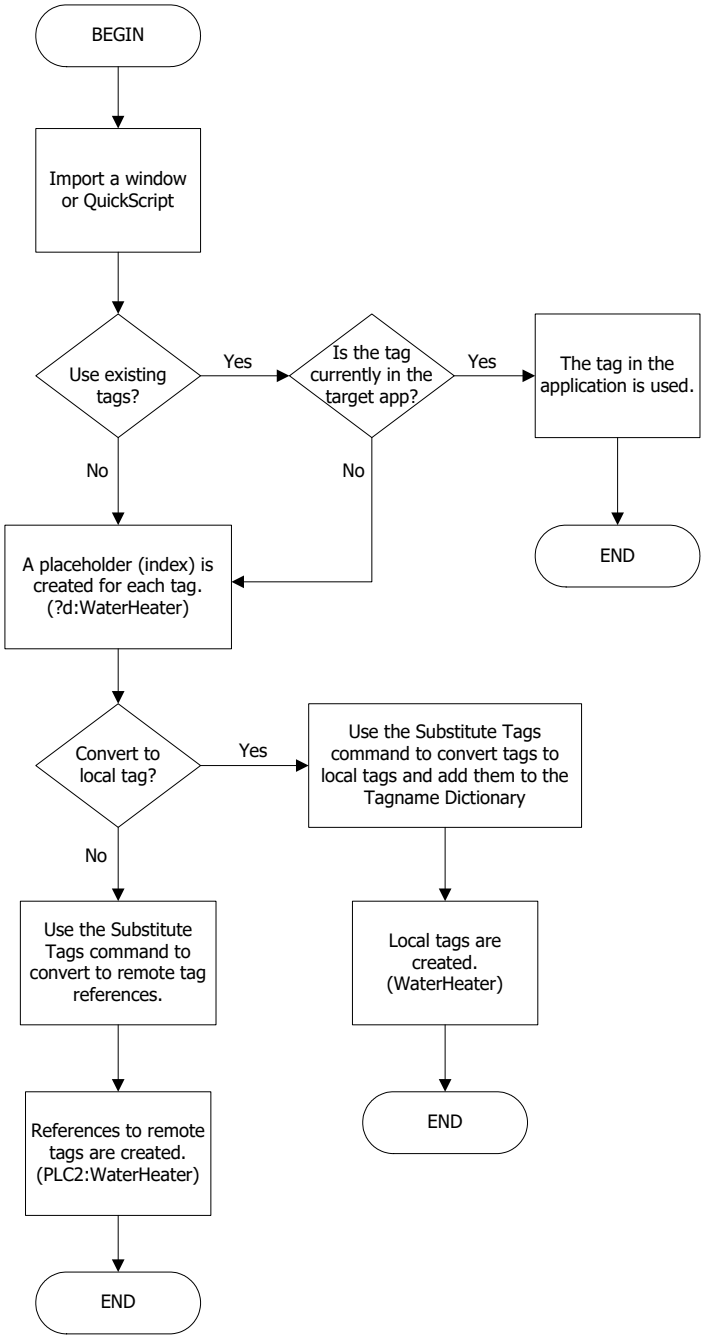
  リモート タグ リファレンスに変換します。リモート タグ変数の例は、PLC2:WaterHeater です。リモート タグ リファレンスによって、アプリケーションはリモート タグ変数サーバーからデータをすぐに受け取り、ローカル タグ変数ディクショナリで単一タグ変数を定義する必要がなくなります。

- **既存のタグ変数を使用します。**

  インポート中、既存のタグ変数の使用を選択すると、InTouch HMI はインポートされたタグ変数がタグ変数ディクショナリにすでに存在することを確認します。タグ変数がすでに存在する場合、タグ変数は完全修飾されたタグ変数としてインポートされます。このオプションを使用することによって、プレースホルダの合計数が削減され、サイズのより大きいタグ変数データベースでアプリケーションをインポートできるようになります。

以下のフローチャートは、インポートされたウィンドウと QuickScript に対してタグ変数が処理される方法を説明しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



アプリケーションからの産業用グラフィックのエクスポート

アプリケーションのすべての産業用グラフィックを aaPKG ファイルにエクスポートできます。その後、ファイルからグラフィックを同じまたは別のコンピュータ上の別の アプリケーションにインポートできます。

アプリケーションからエクスポートする産業用グラフィックを個々に選択することはできません。すべての産業用グラフィックがアプリケーションからエクスポートされます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーションからの産業用グラフィックのエクスポート

1. エクスポートする産業用グラフィックを含むアプリケーションを WindowMaker で開きます。
2. ［**ファイル**］メニューの［**エクスポート**］を選択して、［**視覚化**］グループから［**すべての産業用グラフィック**］を選択します。

   ［**産業用グラフィックのエクスポート**］ダイアログ ボックスが表示されます。このダイアログ ボックスを使用して、エクスポート先のフォルダを指定してエクスポート ファイルに名前を付けます。
3. aaPKG ファイルのエクスポート先のフォルダを選択します。
4. 必要に応じて、［**ファイル名**］フィールドにエクスポート ファイルの名前を入力します。

   エクスポート ファイルのデフォルト名は IndustrialGraphics.aaPKG です。
5. **保存**を選択します。

   エクスポート ファイルへの産業用グラフィックのロードの進捗状況を示す水平バーが表示されます。
6. エクスポート処理が終了した後、Windows エクスプローラでエクスポート先のフォルダに移動して、エクスポート ファイルが作成されていることを確認します。

アプリケーションへの産業用グラフィックのインポート

別のモダン アプリケーションで作成された産業用グラフィックを WindowMaker で実行しているアクティブなアプリケーションにインポートできます。

インポートされるのは aaPKG ファイルに含まれる産業用グラフィックだけです。WindowMaker で編集用に開いているアプリケーションのグラフィックは、インポートしたグラフィックによって上書きされます。aaPKG ファイルにサポートされないコンポーネントが含まれる場合、インポートは失敗して、エラーを示すダイアログ ボックスが表示されます。

アプリケーションへの産業用グラフィックのインポート

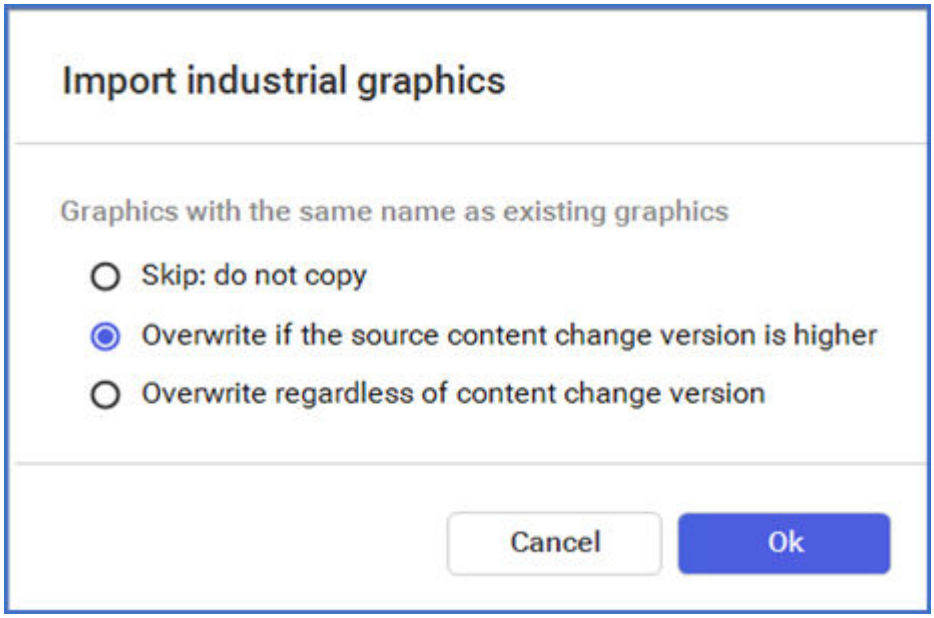
1. インポートする産業用グラフィックを含むアプリケーションを WindowMaker で開きます。
2. ［**ファイル**］メニューの［**インポート**］を選択して、［**視覚化**］グループから［**産業用グラフィック**］を選択します。

   産業用グラフィックのエクスポート ファイルを含むフォルダを指定する［**産業用グラフィックのインポート**］ダイアログ ボックスが表示されます。
3. Windows エクスプローラを使用して、エクスポートされた産業用グラフィックを含む aaPKG ファイルが保存されているフォルダに移動します。
4. インポートする aaPKG ファイルを選択します。

   選択したファイルの名前が［**ファイル名**］フィールドに表示されます。
5. **[開く]**を選択します。

   ［産業用グラフィックのインポート］ダイアログ ボックスが表示され、グラフィックの上書きに関する以下のオプションが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



- **スキップ: コピーしない** - グラフィックはインポートされません。
- **ソース コンテンツの変更バージョンが新しい場合に上書きする** - グラフィックは、インストールされているバージョンよりもインポートするファイルのバージョンの方が新しい場合にのみインポートされます。
- **コンテンツの変更バージョンに関係なく上書きする** - グラフィックがインポートされます。

6. ［**OK**］ を選択します。

   アクティブなアプリケーションへの産業用グラフィックのインポートの進行状況を示す水平バーが表示されます。処理が終了したら進捗状況を示すインジケータが消えます。

産業用グラフィック ツールボックスで選択したシンボルのエクスポート

アプリケーションの産業用グラフィック ツールボックスから選択した産業用グラフィックを aaPKG ファイルにエクスポートできます。その後、ファイルからグラフィックを同じまたは別のコンピュータ上の別の アプリケーションにインポートできます。

注意：ここでは、選択した産業用グラフィックをエクスポートする手順を示します。すべての産業用グラフィックをエクスポートする手順については、「アプリケーションからの産業用グラフィックのエクスポート」を参照してください。

選択した産業用グラフィックをアプリケーションからエクスポートする

1. エクスポートする産業用グラフィックを含むアプリケーションを WindowMaker で開きます。
2. エクスポートするシンボルを産業用グラフィック ツールボックスで選択します。
3. 選択したシンボルを右クリックしてショートカット メニューを表示します。
4. ショートカット メニューで［**エクスポート**］を選択して、［**シンボル**］を選択します。

   ［**産業用グラフィックのエクスポート**］ダイアログ ボックスが表示されます。このダイアログ ボックスを使用して、エクスポート先のフォルダを指定してエクスポート ファイルに名前を付けます。
5. aaPKG ファイルのエクスポート先のフォルダを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

6. 必要に応じて、［**ファイル名**］フィールドにエクスポート ファイルの名前を入力します。

   デフォルトのエクスポート ファイル名は、産業用グラフィック ツールボックスで最初に選択したシンボルの名前です。

7. **保存**を選択します。

   エクスポート ファイルへの産業用グラフィックのロードの進捗状況を示す水平バーが表示されます。

カスタム クライアント コントロールのインポートと埋め込み

カスタム Windows クライアント コントロールを作成して、 アプリケーションの産業用グラフィックに埋め込むことができます。最初に、クライアント コントロールを WindowMaker の産業用グラフィック ツールボックスにインポートする必要があります。このセクションでは、カスタム クライアント コントロールをインポートする手順および埋め込む手順について説明します。

カスタム クライアント コントロールをインポートする

1. アプリケーションのカスタム クライアント コントロールを作成します。
2. InTouch WindowMaker がインストールされているコンピュータからアクセスできるフォルダにクライアント コントロールを配置します。
3. ［**ファイル**］メニューの［**インポート**］を選択して、［**視覚化**］グループから［**クライアント コントロール**］を選択します。

重要：カスタム クライアント コントロールをインポートできるのは、スタンドアロン アプリケーションのみです。従来の InTouch アプリケーションまたはパブリッシュ済み InTouch HMI アプリケーションにカスタム クライアント コントロールをインポートすることはできません。

作成したカスタム クライアント コントロールの名前を入力するフィールドを含む［**クライアント コントロールのインポート**］ダイアログが表示されます。

1. Windows エクスプローラを使用して、クライアント コントロール .dll ファイルを配置したフォルダに移動します。
2. クライアント コントロール .dll ファイルを選択し、［**開く**］を選択します。

   WindowMaker は産業用グラフィック ツールボックスにインポートされたカスタム クライアント コントロールを更新し、表示します。

インポートされたクライアント コントロールを産業用グラフィック ツールボックスから削除することもできます。最初に、産業用グラフィック ツールボックス内でクライアント コントロールを選択します。次に、右クリックしてショートカット メニューを表示し、［**削除**］を選択します。

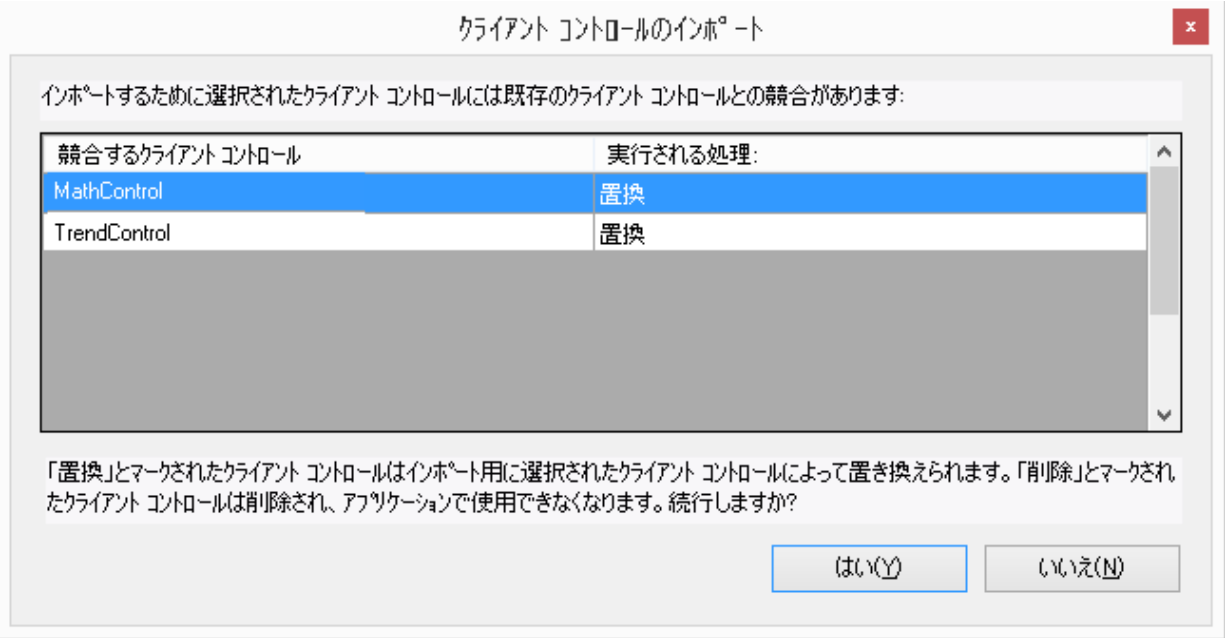
重複するクライアント コントロールをインポートする際の競合の解決

別のバージョンのクライアント コントロールをインポートして、既存のコントロールを上書きできます。既存のコントロールをホストしている .dll は、インポートするライブラリによって置換されます。競合するクライアント コントロールは、新しいクライアント コントロール .dll のインポート時に削除されます。

注意：競合検出の条件は、コントロールの名前のみです。ライブラリのファイル名やバージョンは、競合検出に影響しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

たとえば、**MathControl** と **TrendControl** という 2 つのコントロールを含むクライアント コントロール .dll をインポートする際に現在のライブラリに同じ名前のコントロールが含まれる場合、[**クライアント コントロールのインポート**] ダイアログ ボックスが表示されます。



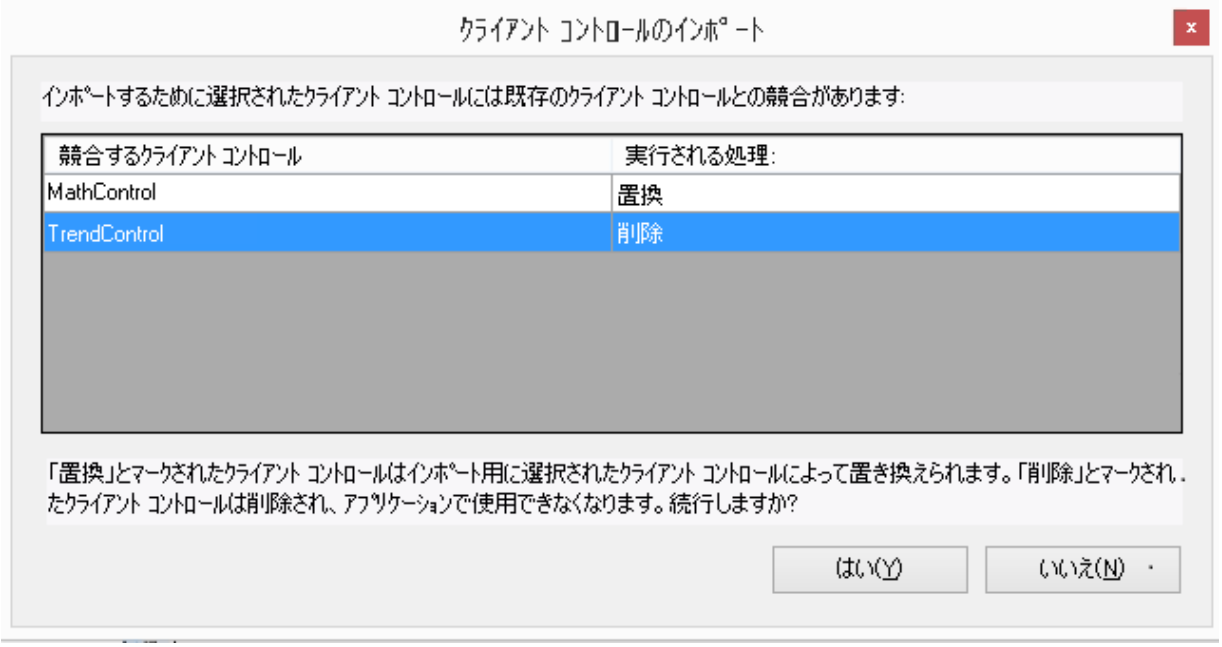
既存のクライアント コントロール .dll が置換され、新しいコントロールがライブラリで使用可能になります。

[**実行される処理**] カラムに「削除」と表示されている場合、インポートするライブラリに含まれないコントロールが現在のライブラリのコントロールにあることを意味します。競合するコントロールを解決するためには、ホストしている .dll を置き換える必要があるので、現在の .dll に含まれていて、インポートする .dll に含まれないライブラリはインポート処理中に削除されます。

たとえば、MathControl と DatabaseControl を含むクライアント コントロール .dll をインポートする際に、現在のライブラリに MathControl と TrendControl が含まれる場合、インポート時にライブラリから TrendControl が削除されます。

[**クライアント コントロールのインポート**] ダイアログ ボックスに削除を確認するプロンプトが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



ライブラリが置換され、インポートの完了時に TrendControl が削除されます。

WindowMaker を再起動して、グラフィック ツールボックスのコントロールを更新します。

注意 : シンボルに既に埋め込まれているクライアント コントロールの新しいバージョンをインポートした場合、WindowMaker を再起動してグラフィック サムネイルを最新の情報に更新してもコントロールの内容は更新されません。新しいクライアント コントロールをサムネイルに反映させるには、新しいクライアント コントロールのシンボルを編集して保存する必要があります。

産業用グラフィックへのクライアント コントロールの埋め込み

クライアント コントロールは、産業用グラフィック ツールボックスから埋め込まれます。グラフィック ツールボックスには、既にいくつかのクライアント コントロールが含まれています。これらの既存のコントロールを産業用グラフィックに埋め込むことや、カスタム コントロールをインポートして埋め込むことができます。

産業用グラフィックにクライアント コントロールを埋め込む

1. カスタム クライアント コントロールを埋め込むアプリケーションを WindowMaker で開きます。
2. カスタム クライアント コントロールを埋め込む産業用グラフィックを含むウィンドウを開きます。
3. 産業用グラフィックを選択します。
4. メニュー バーの［**産業用グラフィックの埋め込み**］アイコンを選択します。

   重要 : 産業用グラフィック ツールバーから産業用グラフィックにカスタム クライアント コントロールをドラッグ アンド ドロップすることはできません。カスタム クライアント コントロールは必ず埋め込む必要があります。

5. 必要に応じて、アプリケーション用にカスタム クライアント コントロールを設定します。

HTML5 ウィジェットのインポート

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スタンドアロン アプリケーションおよびマネージド アプリケーションのウィジェットをインポートできます。ファイル形式は、HTML5、CSS、および Javascript ファイルを含むカスタム ウィジェット パッケージ（.cwp）です。

HTML ウィジェットのインポート

1. WindowMaker を起動します。
2. ［ファイル］メニューで［インポート］を選択し、［視覚化］を強調表示してから［**HTML5 ウィジェット**］を選択します。

   ［**HTML5 ウィジェットのインポート**］ダイアログ ボックスが表示されます。
3. インポートするウィンドウを含むアプリケーションのフォルダを選択します。
4. ［**OK**］ を選択します。

   ツールボックスにウィジェットが表示されます。

ウィジェットをインポートした後、以下の操作を行います。

1. グラフィックを作成します。
2. グラフィックを編集してウィジェットを埋め込みます。
3. ［ウィジェットのプロパティ］セクションでプロパティを設定します。それぞれのウィジェットには、独自のプロパティがあります。
4. ウィジェットをウィンドウに挿入します。

ウィジェットが WindowViewer および任意の Web ブラウザで表示できるようになります。デザイン時のプロパティ設定に応じて、ウィジェットをランタイム時に操作できます。［ウィジェットのプロパティ］の下にあるカスタム プロパティを使用してウィジェットを変更するスクリプトはサポートされていません。

InTouch アプリケーションへのスクリプト関数ライブラリのインポート

InTouch アプリケーションにスクリプト関数ライブラリをインポートできます。.NET（*.dll およびその他の .NET ファイル拡張子）、スクリプト ライブラリ ファイル（*.aaSLIB）、および InTouch スクリプト拡張ファイル（*.wdf）を含むさまざまなスクリプト関数ライブラリをインポートできます。

1 つのアプリケーションにインポートしたスクリプト関数ライブラリは、別のアプリケーションを作成するためにアプリケーションをエクスポートする際、自動的にエクスポートに含まれます。スクリプト関数ライブラリは、それをインポートしたアプリケーションをパブリッシュするときにも使用できますです。

スクリプト関数ライブラリをアプリケーションにインポートする

1. スクリプト関数ライブラリをインポートする InTouch アプリケーションを開きます。
2. WindowMaker のメイン メニューで［ファイル］を選択し、［インポート］を選択して［スクリプト］を選択します。

   ［スクリプト関数ライブラリのインポート］ダイアログが開きます。
3. インポートする関数ライブラリを参照します。
4. インポートするファイルを選択して［開く］を選択すると、スクリプト関数ライブラリのインポートが開始されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

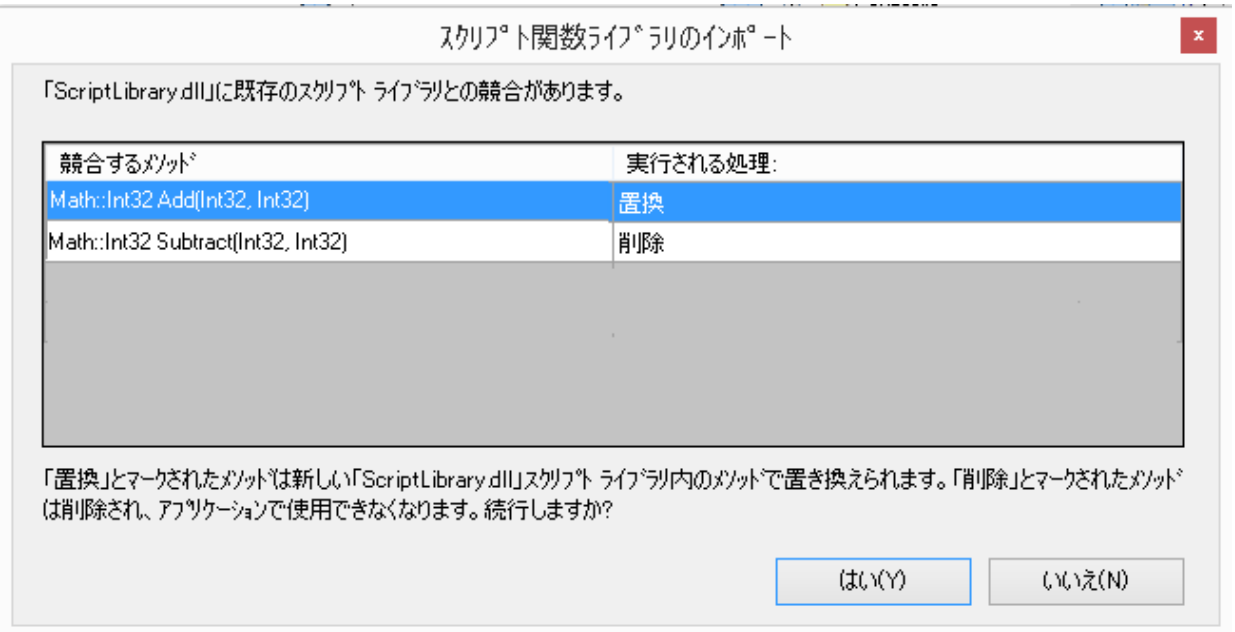
注意：インポート中は進捗状況を示すバーや進捗情報を示すウィンドウは表示されません。インポートが正常に完了すると、情報ウィンドウが開きます。

.NET スクリプト ライブラリ内の競合するメソッドのインポートの解決

.NET クラス スクリプト ライブラリをアプリケーションにインポートするとき、既存のスクリプト ライブラリはインポートするライブラリで置換されます。競合するスクリプト メソッドは、この時点で検出されます。競合検出は、名前空間、クラス名、メソッド名、およびパラメータ宣言に基づいて行われます。

注意：.dll のバージョンやファイル名はメソッド競合検出に影響しません。

インポート時に、競合するメソッドが［**スクリプト関数ライブラリのインポート**］ダイアログ ボックスに表示されます。



この例では、現在のライブラリに `Math::Int32 Add(Int32, Int32)` があり、インポートするライブラリのメソッドとして、同じクラス、メソッド名、およびパラメータが含まれています。この場合、「実行される処理」カラムに「置換」と表示されます。インポートを続行すると、インポートするライブラリによってアプリケーションのスクリプト ライブラリ全体が置換されます。

インポートするライブラリには `subtract` メソッドが含まれないので、［`Math::132 Subtract(Int32, Int32)`］は「削除」とマークされます。スクリプト メソッドの競合を解決するには、スクリプト ライブラリ全体を置換する必要があります。その結果、インポートするライブラリに含まれないメソッドが削除されることになります。

上記の例のように、既存のメソッドがライブラリから削除される結果となる場合、メソッドのインポートを個々にキャンセルできます。競合するすべてのメソッドを解決するか、インポートそのものをキャンセルする必要があります。

重要：インポート時に重複として検出できるのは、.NET クラス ライブラリ ファイルのみです。既存のライブラリ内のメソッドと競合した場合、.aaSLIB ライブラリおよび .wdf スクリプト拡張子ファイルはインポートされません。その場合、競合の通知は行われません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーションのアプリケーション スタイル ライブラリの設定

InTouch アプリケーションのグラフィックのスタイル ライブラリを設定できます。クォリティとステータス、要素スタイル、および数値フォーマット スタイルのアプリケーション スタイルを設定できます。設定の変更は、アプリケーションのリポジトリに保存されます。

- クォリティとステータス インジケータは、アプリケーション データの現在の品質、およびアプリケーション シンボルで表示される装置の状態を表すグラフィック アイコンです。
- 要素スタイルは、表示プロパティのセットを定義します。これらのプロパティは、産業グラフィックに表示されるテキスト、線、グラフィックの輪郭、および内部の塗りつぶしの外観を決定します。
- フォーマット スタイルは、産業用グラフィックで使用される一般的なタイプの数値のアプリケーション全体でのスタイルを個々に設定するオプションを提供します。

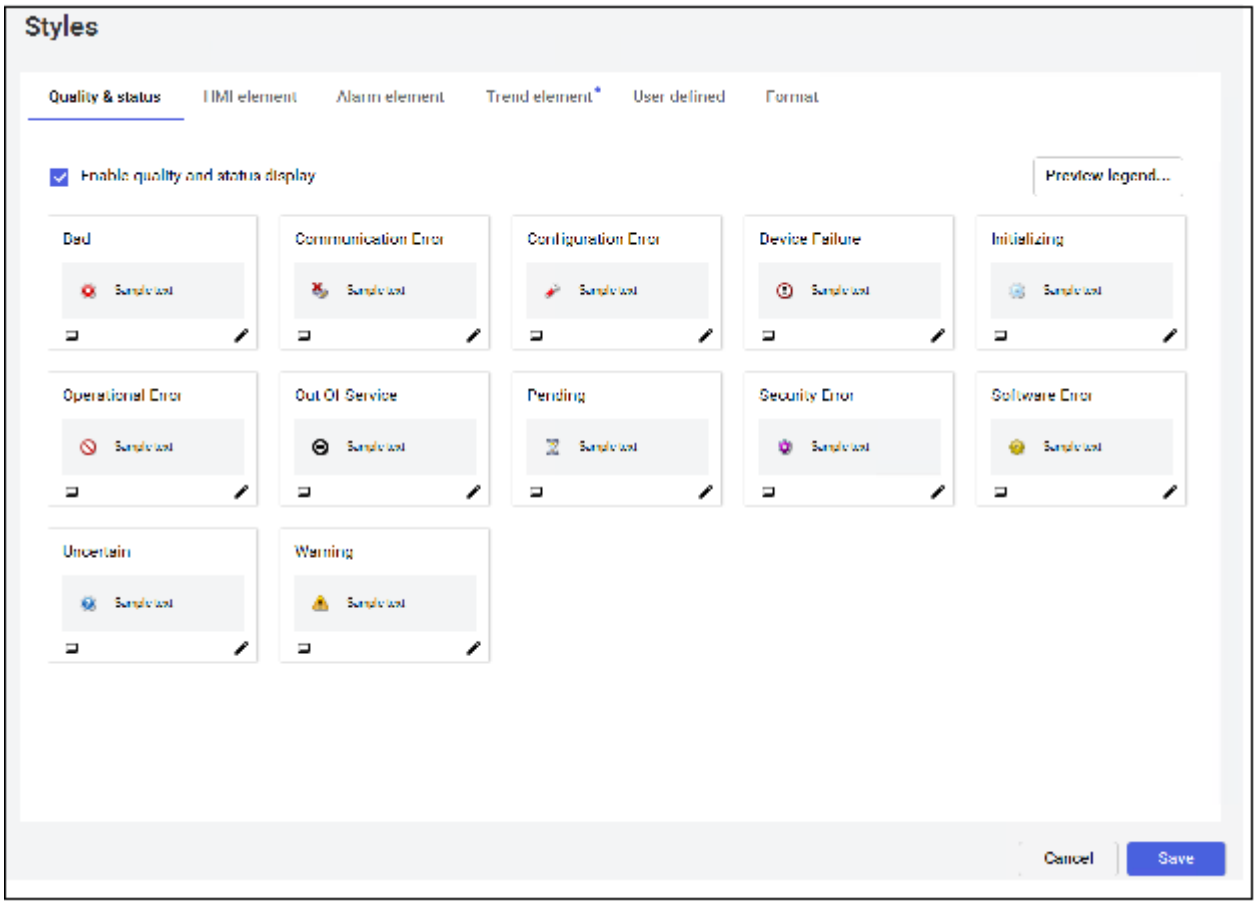
重要：このセクションでは、アプリケーションのスタイル ライブラリにアクセスするための WindowMaker 内のワークフローについて説明します。アプリケーション スタイルの編集の詳細については、WindowMaker のオンライン ヘルプまたは『*産業用グラフィック エディタ ユーザー マニュアル*』を参照してください。

アプリケーション スタイル ライブラリの設定

1. WindowMaker でアプリケーションを開きます。
2. ［**ファイル**］メニューの［**構成**］を選択し、［**スタイル**］を選択します。

   ［**スタイル**］設定画面が表示されます。この画面には、クォリティとステータス インジケータ、グラフィック要素スタイル、および数値フォーマット スタイルを設定するタブがあります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 編集するアプリケーション スタイルのタブを選択します。

注意：WindowViewer で一度に開くことのできるアプリケーションは 1 つのみです。プラットフォームがローカル ノードに配置されている場合、他のスタンドアロン アプリケーションまたはマネージド アプリケーションで設定されているスタイルよりも Galaxy で設定したスタイルが優先されます。

アプリケーション スタイル ライブラリのエクスポートとインポート

アプリケーションからアプリケーション スタイル ライブラリをエクスポートして、別のアプリケーションにインポートできます。品質、要素スタイル、および数値形式の設定が XML ファイルにエクスポートされます。

アプリケーションからアプリケーション スタイル ライブラリをエクスポートする

1. WindowMaker を開きます。
2. ［**ファイル**］メニューから［**エクスポート**］を選択し、［**スタイル**］を選択します。

   ファイル名を指定するフィールドを含む［**アプリケーション スタイル ライブラリのエクスポート**］ファイル参照画面が表示されます。
3. エクスポートされた XML ファイルを配置するフォルダを選択して、ファイルに名前を付けます。
4. **保存**を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アプリケーション スタイル ライブラリが正常にエクスポートされたことを示すダイアログ ボックスが表示されます。

アプリケーションにアプリケーション スタイル ライブラリをインポートする

1. WindowMaker を開きます。
2. ［**ファイル**］メニューから［**インポート**］を選択し、［**スタイル**］を選択します。

   ファイル名を指定するフィールドを含む［**アプリケーション スタイル ライブラリのインポート**］ファイル参照画面が表示されます。
3. C:\Program Files (x86)\ArchestrA\Framework\Bin\AdditionalElementStyles に移動します。インポートする XML ファイルが配置されているフォルダを選択します。そのファイル名が［**ファイル名**］フィールドに表示されます。
4. **[開く]**を選択します。

   アプリケーション スタイル ライブラリがインポートされたことを示すダイアログ ボックスが表示されます。
5. ［**OK**］をクリックします。

   WindowMaker を再起動するまで、WindowMaker のグラフィックは無効化されたままです。

アプリケーションのアラーム優先度マッピングの設定

InTouch アプリケーションのアラーム優先度マッピングを設定して、各アラームの重要度の優先度範囲を設定します。

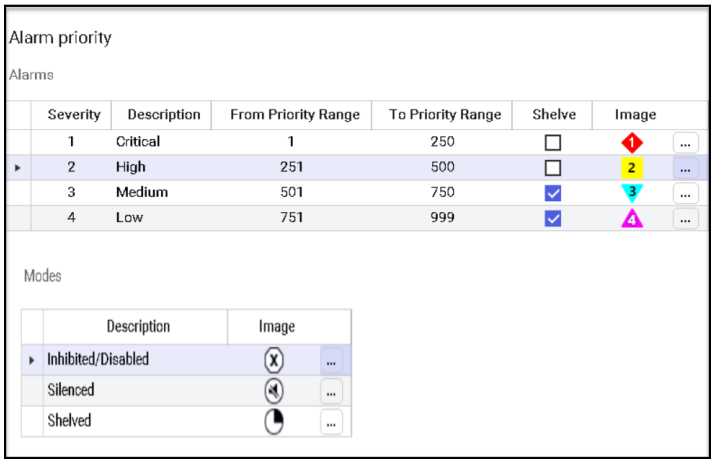
重要：このセクションでは、アラーム優先度範囲をアラーム重要度にマッピングするための WindowMaker 内のワークフローについて説明します。Application Server とは異なり、InTouch にはアラームの重要度の管理機能が組み込まれていませんが、ユーザーは InTouch タグを使用して、アラーム輪郭アニメーションを実装できます。その場合、ダイアログ ボックスでの優先度と重要度のマッピングは、優先度をアラーム輪郭の色とアラーム インジケータ アイコンに関連付けるための視覚的補助としてのみ使用されます。アラーム優先度のマッピングおよびアラームのシェルブの設定の詳細については、WindowMaker のオンライン ヘルプまたは『*産業用グラフィック エディタ ユーザー マニュアル*』を参照してください。

アプリケーションのアラーム優先度マッピングを設定する

1. WindowMaker でアプリケーションを開きます。
2. ［**ファイル**］メニューの［**構成**］を選択し、［**アラーム**］を選択します。

   優先度の範囲を各アラームの重要度にマップするフィールドを含む［**アラーム優先度**］セクションが表示されます。この画面には、アラームの重要度に基づいてアラームのシェルブを有効にするためのフィールドもあります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［**優先度始点**］フィールドと［**優先度終点**］フィールドで、1 ～ 999 の数字を選択して入力し、各アラーム重要度のアラーム優先度範囲の下限と上限を設定します。

   各優先度範囲は、優先度範囲間の重複なしで連続するものである必要があります。アラーム重要度 1 はデフォルトで優先度 1 で開始します。

4. ［**シェルブ**］カラムで、各アラーム優先度のアラームのシェルブを有効にするチェック ボックスをオンまたはオフにします。

5. ［**OK**］を選択して変更を保存します。

   変更内容はアプリケーションのアプリケーション フォルダに保存されます。

アプリケーションからの産業用グラフィック テキスト文字列のエクスポート

アプリケーションがランタイム時の言語切り替えをサポートすることを目的としている場合、その産業用グラフィックのテキスト文字列をディクショナリ ファイルにエクスポートできます。その後、テキスト エディタ、XML エディタ、または Microsoft Excel や言語アシスタントなどのスプレッドシート プログラムを使用して、ディクショナリ ファイル内の文字列を別の言語に翻訳できます。

グラフィック テキスト文字列をエクスポートするとき、ディクショナリ ファイルの出力フォルダを指定する必要があります。文字列を別の言語に変換するディクショナリ ファイルごとに個別のフォルダを作成することをお勧めします。

エクスポートしたすべてのディクショナリファイルには、<アプリケーションのフォルダ名>AA_<言語 ID>.xml. という名前が付けられます。たとえば、アプリケーションのフォルダ名が PumpStation で、エクスポートする言語がフランス語（言語 ID は 1036）の場合、ファイル名は PumpStationAA_1036.xml になります。

異なるときに異なるオブジェクトの言語文字列をエクスポートする場合、最初のエクスポートが後続のエクスポートによって上書きされるのを防止するために個別のターゲット フォルダを使用してください。

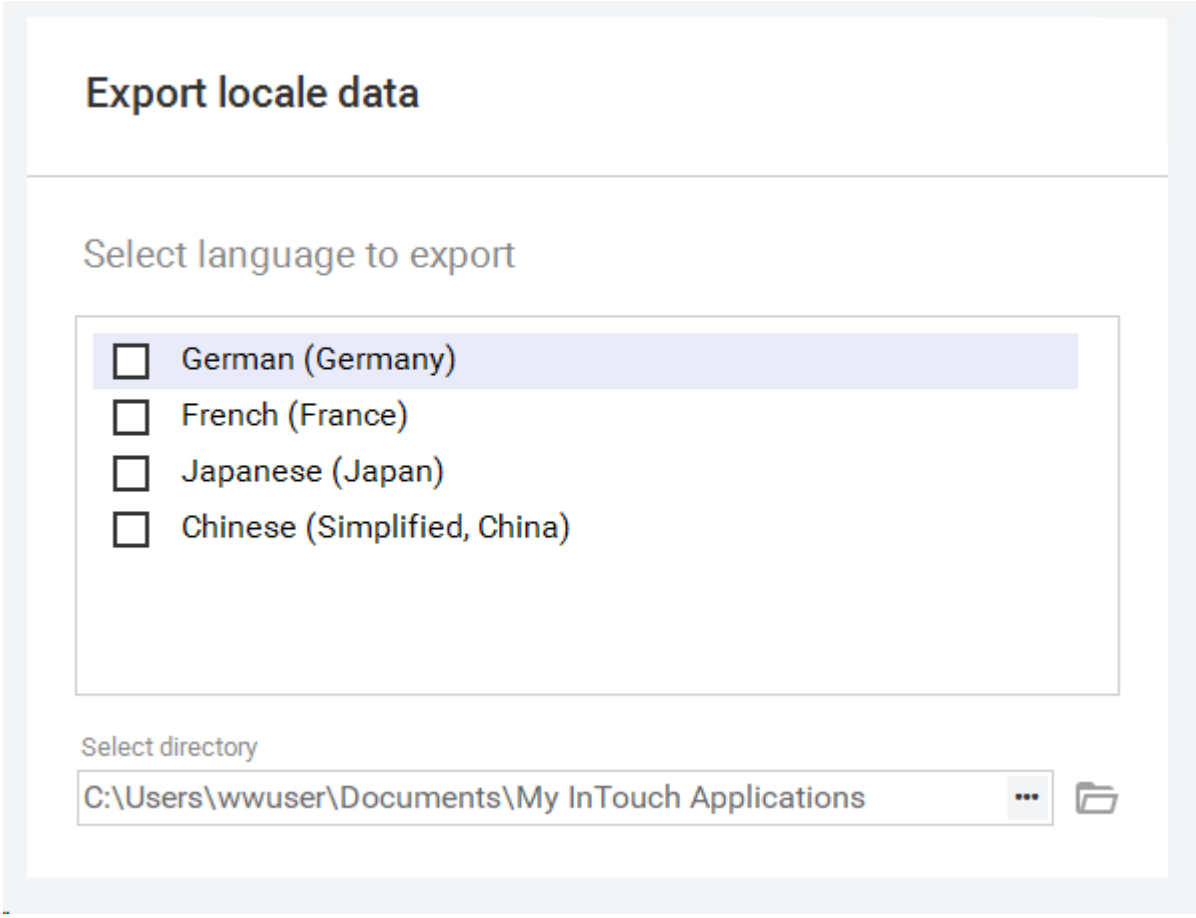
産業用グラフィック テキスト文字列をエクスポートする

1. WindowMaker でアプリケーションを開きます。

2. ［**ファイル**］メニューの［**エクスポート**］を選択し、［**ローカリゼーション**］、［**産業用グラフィックの翻訳**］の順に選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［ロケール データのエクスポート］画面が表示されます。



3. エクスポートするシンボル テキスト文字列を設定します。
   - ［**エクスポートする言語**］リストで、エクスポートする言語ディクショナリのチェック ボックスをオンにします。デフォルトの言語はリストに表示されません。
   - ［**ディレクトリの選択**］フィールドにディクショナリ ファイルのエクスポート先のフォルダを入力します。

     参照して既存のフォルダを選択することや新規フォルダを作成することもできます。
4. **[Export]** を選択します。

アプリケーションへの産業用グラフィックのテキスト文字列のインポート

シンボル テキストの場合、各言語の翻訳されたディクショナリ ファイルをインポートして、それらの言語のランタイム時の言語切り替えを有効にする必要があります。1 つの言語のすべてのディクショナリ ファイルは同じフォルダに配置する必要があります。

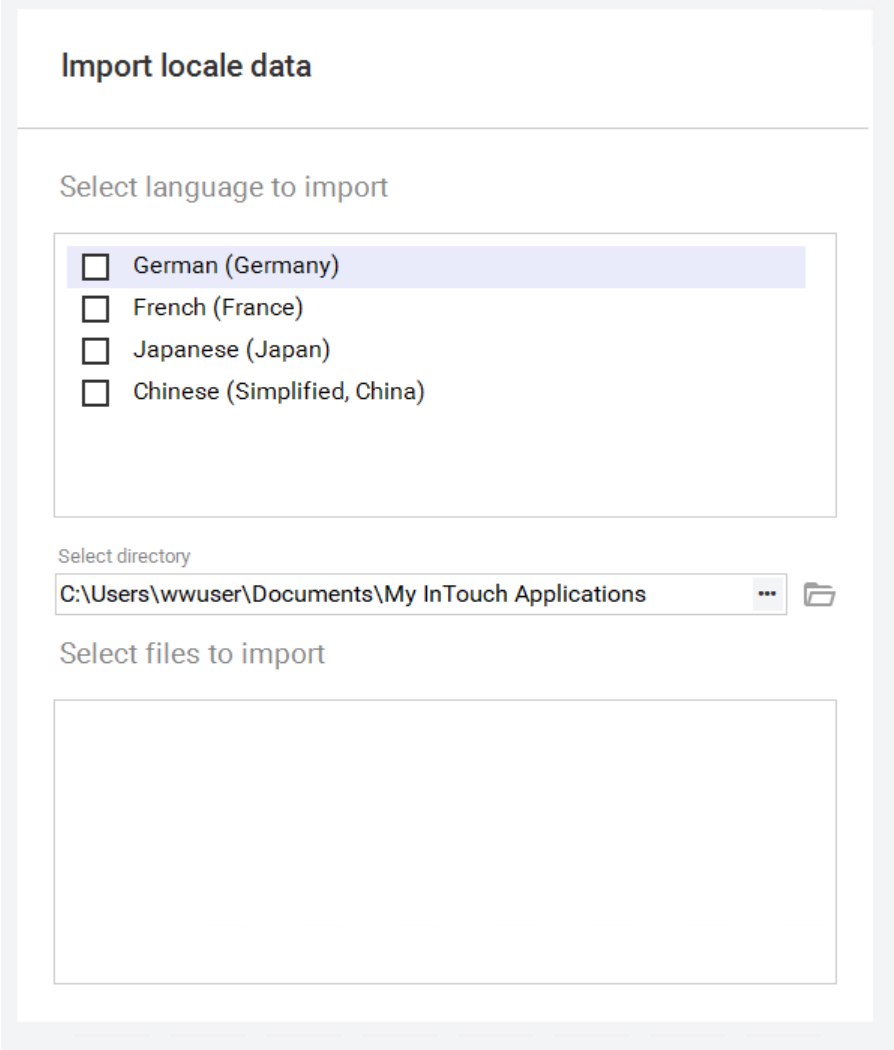
一度にインポートできるファイルは 1 つの言語だけです。インポートするとき、目的の言語を選択して、インポートするディクショナリ ファイルを指定します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

翻訳されたディクショナリファイルをインポート

1. インポートする産業用グラフィック テキストを含むアプリケーションを開きます。
1. ［ファイル］メニューの［インポート］を選択し、［ローカリゼーション］、［産業用グラフィックの翻訳］の順に選択します。

   ［ロケール データのインポート］画面が表示されます。



1. インポート設定を構成します。
   - ［インポートする言語］リストで、インポートする言語ディクショナリのチェック ボックスをオンにします。
   - ［ディレクトリの選択］ボックスで、インポートするディクショナリ ファイルを含むフォルダを指定します。
   - ［インポートするファイルの選択］ボックスで、インポートする .xml ファイルを選択します。現在のアプリケーション フォルダ名および選択した言語のロケール ID を含むファイルだけが表示されます。
2. **Import** を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

シンボルからのローカリゼーション文字列のエクスポート

アプリケーションでランタイム時の言語切り替えをサポートする場合、産業用グラフィック ツールボックスで選択した 1 つまたは複数のシンボルのテキスト文字列をエクスポートできます。その後、テキスト エディタ、XML エディタ、または Microsoft Excel などのスプレッドシート プログラムを使用して、ファイル内のエクスポート済み文字列を別の言語に翻訳できます。

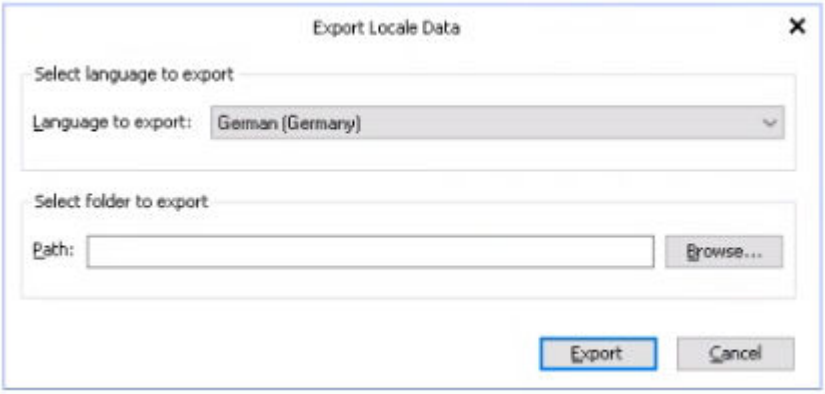
シンボルからテキスト文字列をエクスポートするとき、出力フォルダを指定する必要があります。文字列をそれぞれ異なる言語に翻訳するファイルごとに個別のフォルダを作成することが推奨されます。

エクスポートされるローカリゼーション ファイルの名前は、<アプリケーションのフォルダ名>AA_<言語 ID>.xml. になります。たとえば、アプリケーション フォルダの名前が PumpStation1 で、ローカリゼーション文字列の言語がメキシコのスペイン語（言語 ID = 2058）である場合、ファイル名は PumpStation1AA_2058.xml になります。

シンボルからのローカリゼーション文字列のエクスポート

1. WindowMaker でアプリケーションを開きます。
2. エクスポートするローカリゼーション文字列を含むシンボルを産業用グラフィック ツールボックスで選択します。
   - 単一のシンボルを選択するには、目的のシンボル名を選択します。
   - 複数のシンボルを選択するには、Ctrl キーを押しながら目的のシンボル名をクリックします。
   - 連続する範囲に含まれるすべてのシンボルを選択するには、範囲の最初または最後にあるファイルをクリックし、Shift キーを押しながら範囲の最後または最初にあるファイルをクリックします。
3. 選択したシンボルを右クリックしてショートカット メニューを表示します。
4. ［エクスポート］、［ローカリゼーション］、［選択されたシンボル］の順に選択します。

   ［ロケール データのエクスポート］ダイアログ ボックスが表示されます。



5. エクスポートするシンボル テキスト文字列を設定します。
   - ［**Languages to export（エクスポートする言語）**］リストで、シンボルからエクスポートするローカリゼーション文字列を選択します。デフォルトの言語はリストに表示されません。
   - ［パス］フィールドで、ローカリゼーション文字列をエクスポートするフォルダを入力します。［参照］をクリックして、既存のフォルダを選択するか、新しいフォルダを作成します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

6. **[Export]** を選択します。エクスポート操作の進捗状況を示すバーが表示されます。
7. ［詳細の表示］を選択して、選択した各シンボル内のローカリゼーション文字列が正常にエクスポートされたことを確認します。

産業用グラフィック ライブラリのインポート

アプリケーションを開発する際、以下の条件が満たされていれば、産業用グラフィック ライブラリおよび Situational Awareness Library を標準のアプリケーションにインポートできます。

- アプリケーションが空白のテンプレートで作成されていて、産業用グラフィックや Situational Awareness Library が含まれていない。
- 古いスタンドアロン アプリケーションまたはモダン アプリケーションが移行されていて、ライブラリはインポートされていない。

アプリケーションに産業用グラフィック ライブラリをインポートするには:

- 産業用グラフィック ツールボックスで、アプリケーション名を右クリックして［**産業用グラフィック ライブラリのインポート**］をクリックします。

  ［産業用グラフィックのインポート］ダイアログが表示されます。先に産業用グラフィック ライブラリがインポートされ、その後に Situational Awareness Library がインポートされます。

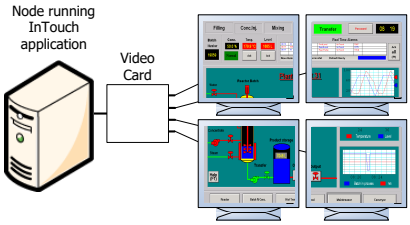
  処理が完了すると、産業用グラフィック ライブラリと Situational Awareness Library が産業用グラフィック ツールボックスに表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 28 マルチ モニタ システムの設定

マルチ モニタ システムでは、複数のモニタ上に InTouch アプリケーションが同時に表示されます。また、マルチ モニタを設定すると、InTouch アプリケーションが実行されているコンピュータに接続されているすべてのモニタから構成される複合画面が作成されます。各モニタには、画面の一部、またはキーパッドのような単一のウィンドウ コンポーネントのみが表示されます。

InTouch アプリケーションの実行中、モニタ間でマウスを移動したり、あるモニタから別のモニタにウィンドウをドラッグしたりできます。また、以下の図のように、いくつかのマルチ モニタの設定では、すべてのモニタにかけて InTouch アプリケーション ウィンドウ全体を表示できます。



マルチ モニタの設定

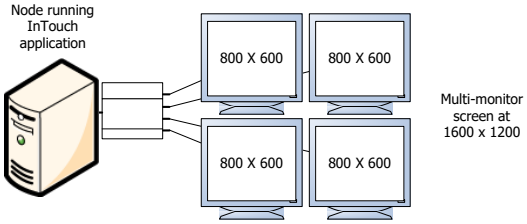
2 つの基本的なマルチ モニタの設定を使用できます。

- 単一のビデオ カード
- 複数のビデオ カード

各設定には、一意のハードウェア、ソフトウェア、および設定の必要条件があります。また、各設定では、異なるマルチ モニタ機能の設定がサポートされています。

単一のビデオ カードの設定

単一のビデオ カードの設定では、コンピュータには、モニタに接続された複数の出力ポートを持つ単一のビデオ カードがインストールされています。



複合画面の解像度は、各モニタの個々の水平および垂直の解像度の合計です。たとえば、一般的なビデオ カードでは、上下 2 台の構成のキューブとしてスタックされた 4 台の 17 インチ モニタを接続できます。上の図では、800 x 600 ピクセルの画面解像度で各モニタが実行されています。複合された仮想画面解像度は、1600 x 1200 ピクセルです。

単一のカードの設定の特性

単一のビデオ カード ドライバには、次の特性があります。

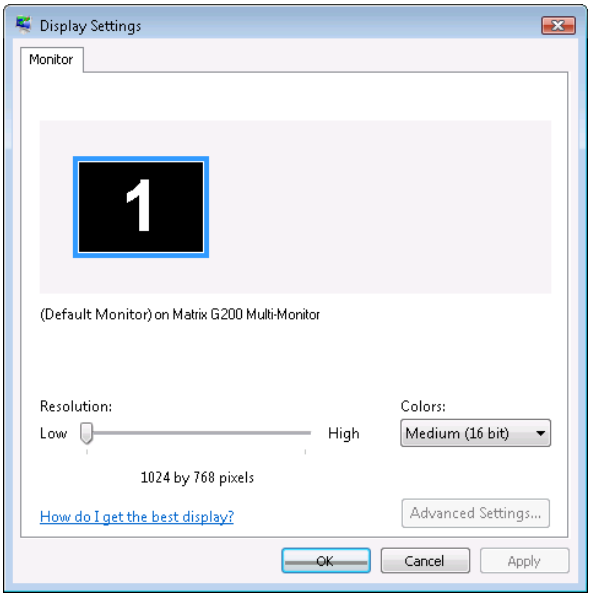
- 単一のビデオ カードにより、すべてのモニタで同時に単一の大きな画面が作成されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 接続されたすべてのモニタのプロパティは、単一の画面値を使用して設定できます。
- 複合画面には、すべてのモニタにわたって、設定の一番下の行に Windows タスクバーが表示されます。
- すべてのモニタに合わせて Windows アプリケーションを最大化できます。

単一のカード ドライバの特性

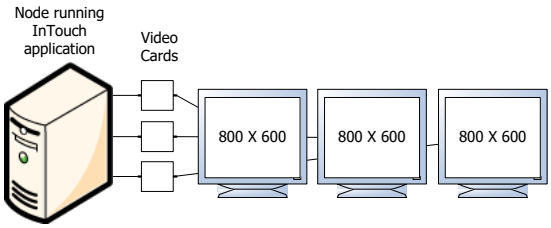
次の図は、複数の出力ポートを持つ単一のビデオ カードに接続されたすべてのモニタのドライバを設定するための、Windows の［**画面のプロパティ**］ダイアログ ボックスを示しています。



この図では、解像度の設定は、一列に並べられた 4 つのモニタ用です。各モニタの解像度は 1024 x 768 です。合計すると、複合画面の解像度は 4096 x 768 になります。設定する必要があるのは、単一のモニタの解像度、カラー深度、およびリフレッシュ レートだけです。解像度の設定は、単一のビデオ カードに接続されたすべてのモニタに適用されます。

複数のビデオ カードの設定

複数のビデオ カードの設定では、複数のビデオ カードがコンピュータにインストールされています。各ビデオ カードにより、InTouch アプリケーションが実行されているコンピュータが単一のモニタに接続されています。



複数のカードの設定の特性

ダイナミック レゾリューション機能（DRC）を他の分散機能と併用すると、画面解像度を自由に設定できるようになります。NAD アーキテクチャでは、開発ノードで InTouch アプリケーションの作成と管理

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

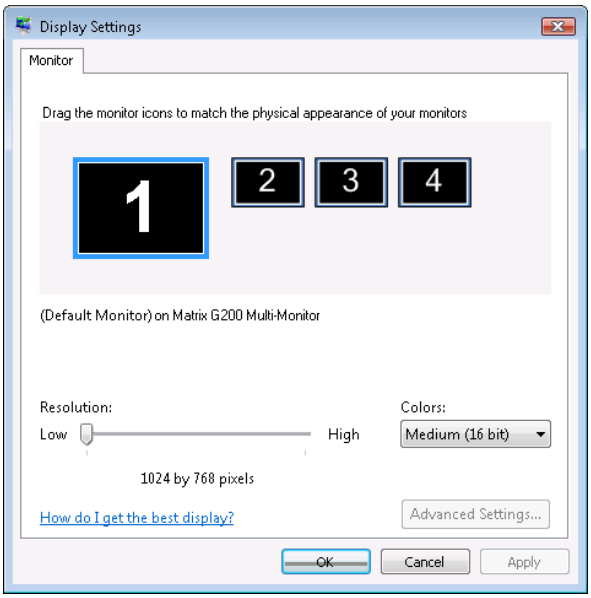
を行い、いくつかの表示ノードにコピーします。DRC を使用すると、異なる画面解像度でノードが実行されている場合でも、すべての表示ノードでアプリケーションを表示できます。

DRC を使用すれば、各表示ノードが、多くのユーザー定義オプション（解像度のカスタマイズも含む）に合わせてアプリケーションをスケーリング（拡大縮小）できます。このスケーリングは、WindowViewer がアプリケーションをコンパイルするときに行われるため、WindowMaker は必要ありません。各表示ノードで異なる DRC 設定を使用できるため、個々の表示ノードを設定する必要があります。

DRC により、マルチ モニタ システムをサポートしやすくなります。複合画面全体またはその一部に InTouch アプリケーションを表示するよう、DRC 解像度の変換オプションから選択するだけです。

複数のカード ドライバの特性

以下の図は、InTouch アプリケーションが実行されているコンピュータにインストールされた個々のビデオ カードに接続されたすべてのモニタのドライバを設定するための、Windows の［**画面のプロパティ**］ダイアログ ボックスを示しています。



［**画面のプロパティ**］ダイアログ ボックスの番号の付いた四角形を選択して、設定するモニタを選択します。モニタの物理的な場所に一致するように、番号の付いた四角形を並べます。画面解像度、カラー深度およびリフレッシュ レートは、選択したモニタにのみ適用されます。

マルチ モニタ アプリケーションの計画

アプリケーション用に複数のモニタを設定するには、次の操作を行います。

- マルチ モニタ ビデオ カードの選択
- アプリケーションの画面解像度の決定
- アプリケーションを表示するモニタの数の決定
- アプリケーション ウィンドウの場所の決定

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マルチ モニタ ビデオ カードの選択

テクニカル サポートでは､マルチ モニタ InTouch アプリケーションがサポートされている推奨ビデオ カードのリストを提供しています。

ビデオ カードを選択する前に、次の質問の回答に関する情報をテクニカル サポートから取得してください。

- どのバージョンの InTouch でビデオ カードがサポートされているか?
- カードでは、単一または複数のカードの設定がサポートされているか?
- ビデオ カードの推奨ドライバは?
- 推奨するビデオ カードの設定は?

アプリケーションの画面解像度の決定

全体の画面解像度を決定し、表示領域の正確なサイズを知っておくと、マルチ モニタ環境でアプリケーションを作成するプロセスを簡素化できます。

モニタ設定全体を表示する描画を作成します。描画には、各モニタの解像度およびすべてのモニタの解像度の合計が表示される必要があります。この描画には、各モニタの水平と垂直のピクセル範囲が表示されます。

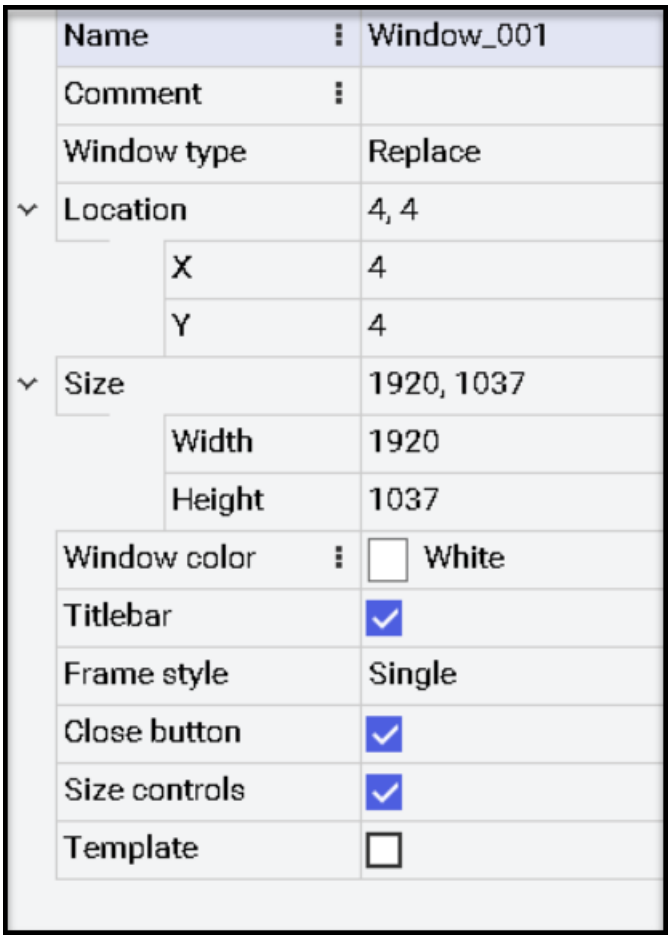
たとえば、800 x 600 の画面解像度を持つ 2 つの水平モニタから構成される複合画面がある場合、2 番目のモニタの左上のピクセルの位置はピクセル 800 x 0 になります。最初のモニタの画面のピクセル数は 0 から 799 になり、2 番目のモニタでは 800 から 1599 になります。描画を参照しながら、複合マルチ モニタ画面のアプリケーション ウィンドウの場所を決定できます。

アプリケーションを表示するモニタの数の決定

生産環境に似た開発環境を使用すると、マルチ モニタ InTouch アプリケーションの作成を簡素化できます。すべての場合で、マルチ モニタ開発環境を使用できるとは限りません。InTouch アプリケーションの開発に使用するコンピュータに接続されているモニタが単一である場合でも、ウィンドウを開発し、表示に必要なウィンドウ サイズおよび位置を設定することにより、マルチ モニタ アプリケーションを構築できます。

WindowMaker の**プロパティ** ペインを使用して、ウィンドウの特性を変更します。［**ウィンドウ**］ペインで、変更するウィンドウを選択します。右側のナビゲーション ペインにウィンドウの**プロパティ** ペインが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



［**X 座標**］および［**Y 座標**］の値により、画面上のウィンドウの左上隅の水平および垂直のピクセル位置が決まります。水平および垂直のピクセルのスケールの基点は、画面の左上隅です。

［**ウィンドウ幅**］および［**ウィンドウ高さ**］の設定により、ウィンドウ全体のサイズが決まります。たとえば、次のようにウィンドウを設定できます。

- X 座標 = 1024
- Y 座標 = 0
- ウィンドウ幅 = 1024
- ウィンドウ高さ = 768

マルチ モニタの設定は、水平に一列に並べられた 4 つのモニタから構成されます。各モニタの解像度は 1024 X 768 です。全体的な複合画面解像度は、4096 x 768 です。

ウィンドウの水平基点を 1024 に設定し、垂直基点を 0 に設定すると、ランタイム時にこのウィンドウを 2 番目のモニタに表示できます。ウィンドウが 2 番目のモニタの画面全体に表示されます。

アプリケーション ウィンドウの場所の決定

マルチ モニタ環境の InTouch ウィンドウを開発する場合、いくつかの異なる設定を使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウの表示位置の強制

ウィンドウを開発し、指定した位置に表示するように強制するのが 1 つの方法です。すべてのモニタの表示領域全体にわたって WindowViewer が最大化されるようにしてください。これにより、指定したモニタに InTouch アプリケーション ウィンドウを表示できます。

InTouch セキュリティ機能を使用すると、Windows デスクトップへのアクセスを拒否できます。

ウィンドウの手動移動

別のオプションとしては、選択したモニタにウィンドウを手動で移動し、単一のアプリケーションを異なるモニタ設定で実行できるようにアプリケーションを開発する方法があります。この場合、次の点に留意してください。

- アプリケーション内のすべてのウィンドウは、［**ポップアップ**］タイプである必要があります。
- メインの WindowViewer の親ウィンドウは、すべてのモニタを表示しないように小さくすることもできます。ただし、この設定では InTouch が最大化されていないため、InTouch セキュリティを使用して Windows デスクトップへのアクセスを拒否することはできません。

この設定では、メインの WindowViewer の親ウィンドウの位置にかかわらず、どのモニタにも簡単に移動できるポップアップ ウィンドウが使用されます。ポップアップ ウィンドウは、WindowViewer の親ウィンドウ内に残る必要はありません。メイン ウィンドウのサイズを縮小してモニタの隅に移動すると、選択したモニタにすべてのポップアップ ウィンドウを自由に移動できます。

環境に基づくウィンドウの自動配置

最後に、上記の方法に手順を追加した方法について説明します。この手順では、使用されている環境に基づいて、アプリケーションで自動的にウィンドウを配置できます。これは最も複雑な設定であり、大規模なスクリプト記述および計画が必要です。

この設定では、標軸および計算のデフォルト設定に基づいて、ShowAt() および ShowAtTopLeft() のスクリプト関数によりウィンドウが動的に配置されます。これは、アプリケーションの必要条件に応じて、さまざまな方法で設定できます。

マルチ モニタ InTouch アプリケーションの開発

マルチ モニタをサポートするには、InTouch.ini ファイルおよび Win.ini ファイル内の選択したパラメータに値を割り当てる必要があります。これらのパラメータを使用すると、InTouch システム ダイアログ ボックスおよびキーパッドを複合画面の適切な位置に配置できます。

マルチ モニタ パラメータの設定

マルチ モニタ サポートを有効にするには、一連の InTouch パラメータを Windows の Win.ini ファイルに追加します。これらのパラメータにより、InTouch アプリケーションが実行されているノードおよび各モニタの解像度のマルチ モニタ サポートが有効になります。

ノード上でマルチ モニタ設定を構成する

1. InTouch HMI ソフトウェアが実行されているコンピュータの Windows フォルダにある Win.ini ファイルを編集します。
2. Win.ini ファイル内の ［InTouch] セクションを検索し、次のパラメータを追加します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータ	説明
MultiScreen=1	値 1 の場合、マルチ モニタ モードが有効になります。値 0 の場合、マルチ モニタ モードが無効になります。
MultiScreenWidth=*nnnn*	単一の画面の幅（ピクセル）
MultiScreenHeight=*nnnn*	単一の画面の高さ（ピクセル）

たとえば、2 つの水平モニタ上に 2560 x 1024 の画面解像度で InTouch アプリケーションを表示する場合、次のように入力します。

```
[InTouch]
MultiScreen=1
MultiScreenWidth=1280
MultiScreenHeight=1024
```

画面解像度の変換の設定

異なる画面解像度で実行されているノード間で移行する場合、パラメータ値を指定して InTouch アプリケーション ウィンドウの現在の解像度を保守できます。

ScaleForResolution パラメータ値により、WindowViewer が実行されているコンピュータで表示解像度が変更された後に、アプリケーション ウィンドウ（*.win）が WindowMaker によって自動的にスケールされるかどうかが決まります。ScaleForResolution パラメータは、WindowViewer ダイアログ ボックスの解像度には影響しません。

ノード上の画面解像度の変換を設定する

1. InTouch が実行されているコンピュータの InTouch.ini ファイルを編集します。
2. ScaleForResolution パラメータをファイルに追加します。

   ```
   ScaleForResolution=1
   ```

   0 に設定すると、解像度の変換が無効になります。

   1 に設定すると、解像度の変換が有効になります。

注意：ScaleForResolution パラメータを InTouch.ini ファイルに追加しない場合、デフォルト値が有効になります（ScaleForResolution=1）。パラメータを無効にすると（ScaleForResolution=0）、解像度を変換するよう要求されます。しかし、解像度の変換は発生しません。

アプリケーションの配置およびマルチ モニタの設定の確認

マルチ モニタ システムで実行されるアプリケーションを単一のモニタ システム上で開発している場合、ScaleForResolution パラメータは特に重要になります。ScaleForResolution パラメータに割り当てられた値により、ある環境から別の環境に移動したときに、アプリケーションを調整できるかどうかが決まります。

重要：異なる環境に移動する前に、アプリケーションのコピーをバックアップしておくことをお勧めします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

たとえば、4 つモニタを一列に並べた設定を持つシステムで実行されるアプリケーション（全体の解像度は 4096 x 768）を、解像度 1024 x 768 の単一のモニタを持つコンピュータで開発した場合、アプリケーションの変換が必要です。

マルチ モニタ システムにアプリケーションを配置すると、アプリケーションの変換を要求するメッセージが表示されます。

ScaleForResolution が .ini に設定されている場合、このメッセージは表示されますが、アプリケーションは変換されずに設計したとおりに実行できます。起動を続行するには、［はい］を選択します。

.ini が設定されていない場合、アプリケーション内のすべてのグラフィックおよびウィンドウが InTouch HMI によって新しい解像度に変換され、スケールされます。そのため、すべてのウィンドウおよびグラフィックの表示が拡大され、予期しない表示になる場合があります。

重要：アプリケーションを実行する前に、移行先のコンピュータでもマルチ モニタの Win.ini パラメータが設定されていることを確認してください。Win.ini の設定は、InTouch アプリケーションと一緒に自動的に転送されません。

ランタイム中のマルチ モニタ サポートの確認

InTouch アプリケーションが実行されるローカル ノードでマルチ モニタがサポートされているかどうかを確認するために、テクニカル サポート スクリプト ライブラリから、オプションのスクリプト関数をダウンロードできます。

WWMultiMonitorNode() 関数により、ノードでマルチ モニタがサポートされるかどうか、およびノードに接続されるモニタの数が決まります。

通常、InTouch アプリケーションが実行されるノードに割り当てられたモニタの数を決定するために、QuickScript から WWMultiMonitorNode() 関数を実行します。

次の例では、InTouch の整数型タグ変数に割り当てられた WWMultiMonitorNode() 関数の値を持つ、QuickScript ステートメントの例を示しています。WindowViewer でアプリケーションが起動したときに、QuickScript が実行されるように設定できます。

```
{整数型タグ変数として定義された MultiMonitor}
MultiMonitors = WWMultiMonitorNode();
{この関数の実行後 Result = 4}
```

WWMultiMonitorNode() は、ノードの Win.ini ファイルで指定された MultiScreen パラメータを読み取ります。WWMultiMonitorNode() 関数は、0 または正の整数のどちらかを返します。

- 0 の返り値

  MultiScreen=0 の場合、または Win.ini ファイルの［InTouch］セクションで MultiScreenWidth パラメータまたは MultiScreenHeight パラメータが誤って 0 に設定されている場合、WWMultiMonitorNode() は 0 を返します。

- 正の整数の返り値

  MultiScreen=1 の場合、および MultiScreenWidth パラメータおよび MultiScreenHeight パラメータに正しい画面解像度の値が割り当てられている場合、WWMultiMonitorNode() はマルチ モニタの設定のモニタの数を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 29 タブレット PC での InTouch の使用

Windows XP Tablet PC Edition および InTouch は、一連ののポータブル タブレット PC にプレインストールされています。これらの丈夫なタブレット PC は、防水処理されており、振動耐性を持ち、多くの産業環境に適しています。InTouch アプリケーションを実行できるタブレット PC は、他のコンピュータ製造業者からも入手可能です。

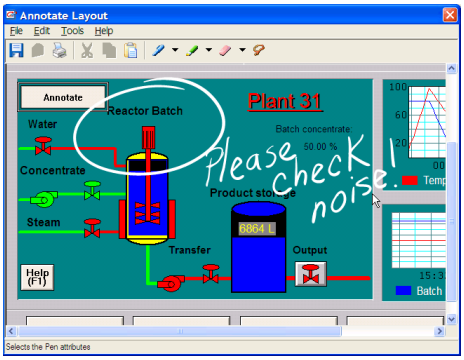
オペレータはタブレット PC を持ち運んで、工場内を移動することができます。タブレット PC では、工場での実際の処理を表す InTouch アプリケーションが実行されます。画面ポインタまたは入力デバイスとして機能するペンを使用すると、画面上の InTouch オブジェクトを選択したり、キーボードの代わりに画面上に直接メモを記述したりすることができます。

オペレータは、工場での実際の処理を直接観察し、実行中の InTouch アプリケーションにメモおよび注釈を書き込むことができます。

視覚化画面の注釈付けと電子メール メッセージによる送信

タブレット PC に表示される画面をキャプチャするには、AnnotateLayout() スクリプト関数を使用します。Windows XP Tablet PC オペレーティング システムを使用しているタブレット PC で InTouch が実行されている場合のみ、AnnotateLayout() 関数を使用できます。

AnnotateLayout() 関数は、アクティブな InTouch ウィンドウの表示されている部分の画面をキャプチャします。キャプチャされた画面は、［**Annotate Layout**］ダイアログ ボックスに表示されます。



［**Annotate Layout**］ダイアログ ボックスには、ツールバーおよびメニュー オプションが含まれています。ダイアログ ボックスでは、キャプチャされた画面がクライアント領域に表示されます。さまざまな描画ツールを使用して画像に注釈を付けたり、キャプチャされた画面を保存、印刷したり、または電子メール メッセージとして送信することができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウの注釈の作成

ウィンドウに注釈を付けるには、次のツールを使用します。

- **ペン**：描画したりコメントを記述したりします。

- **蛍光ペン**：半透明色を使用して、ウィンドウの領域をハイライトします。

- **消しゴム**：注釈の一部を削除します。

これらの各ツールには、サイズ、色または透明度などのオプションがあります。

- これらのオプションを設定するには、各ツールのアイコンの横にある下矢印を選択し、次にオプションのコマンドを選択します。
- これらのオプションをデフォルト設定に復元するには、［**ツール**］メニューで、［**既定に戻す(&S)**］を選択します。

ウィンドウの注釈の選択、コピー、および削除

ウィンドウで作成した注釈を選択、コピー、および削除できます。

注釈を選択する

1. ツールバーで［**選択ツール**］アイコンを選択します。
2. スタイラス ボタンを押したまま、選択する注釈の周りの領域を描画します。

   選択した注釈の切り取り、コピー、または削除を行うことができます。

注釈の切り取り、コピー、および貼り付け

- Windows 標準の［切り取り］、［コピー］、および［貼り付け］のコマンドを使用します。

注釈を削除する

- 以下のいずれかを実行します。
  - ウィンドウのすべての注釈を削除するには、［**編集**］メニューで［**クリア**］をポイントして、次に［**すべて**］を選択します。
  - 選択ツールを使用して選択した注釈を削除するには、［**編集**］メニューで、［**クリア**］をポイントして、次に［**選択**］をクリックします。

注釈の付いたウィンドウの保存、印刷、電子メールによる送信

ウィンドウに注釈を付けた後、画像ファイルとして保存したり、印刷したり、または電子メールの添付ファイルとして送信することができます。

電子メール サーバーを一度設定するだけで、これを行うことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注釈の付いたウィンドウを保存する

1. ［**ファイル**］メニューで、［**保存**］を選択します。Windows 標準の［**名前を付けて保存**］ダイアログ ボックスが表示されます。
2. 名前およびファイル形式を入力して、［**OK**］をクリックします。

注釈の付いたウィンドウを印刷する

1. ［**ファイル**］メニューで、［**印刷**］を選択します。Windows 標準の［印刷］ダイアログ ボックスが開きます。
2. 印刷オプションを指定して、［**OK**］を選択します。

注釈の付いたウィンドウを電子メールの添付ファイルとして送信する

1. ［**編集**］メニューで、［**電子メール設定**］を選択します。［**電子メール設定**］ダイアログ ボックスが表示されます。
2. 電子メールを送信するために使用する SMTP 電子メール サーバーのホスト名を入力します。不明な場合は、管理者に連絡してください。［**OK**］ を選択します。
3. ［**ファイル**］メニューで、［**電子メール**］を選択します。［**E-mail**］ダイアログ ボックスが表示されます。
4. 送信者および受信者のアドレスを入力して、メッセージを記述します。注釈の付いたウィンドウの画像ファイルが添付ファイルとして自動的に追加されます。
5. ［**送信**］を選択して、電子メール メッセージを送信します。

AnnotateLayout() 関数

［**Annotate Layout**］ダイアログ ボックスが表示され、このスクリプト関数が呼び出された現在の表示画面に注釈を付けることができます。この関数は、Windows XP Tablet PC Edition オペレーティング システムでのみサポートされています。

カテゴリ

システム

構文

```
AnnotateLayout()
```

備考

［**Annotate Layout**］ダイアログ ボックスが表示されると、WindowViewer の画面の画像がキャプチャされます。次の場合にこのダイアログ ボックスを使用します。

- ツール バーおよびメニュー項目の設定とペンを組み合わせて使用して、キャプチャされた画面に注釈を付ける場合
- .gif ファイルまたは .jpeg ファイルとして画像および注釈を保存する場合
- 画像および注釈を印刷する場合（プリンタが設定されている場合）
- 電子メール メッセージの添付ファイルとして画像および注釈を送信する場合（SMTP が設定されている場合）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

画面の向きの変更

タブレット PC がタブレット設定で実行されていて、画面の解像度に対してアプリケーションの解像度が動的に変更するように WindowViewer を設定している場合、横方向モードで開発された InTouch アプリケーションは縦方向モードに合うように調整されます。

アプリケーションの解像度が動的に変更するように WindowViewer を設定していない場合、横方向のアプリケーションは調整されません。この場合、InTouch ウィンドウの一部がタブレット PC 上に表示されません。

ある設定から別の設定に切り替えると、デフォルトの値を使用して画面の解像度が切り替わります。たとえば、ラップトップの設定で実行されているタブレット PC がタブレット設定に切り替わる場合、横方向モード（1024 x 768）から縦方向モード（768 x 1024）に画面の向きが切り替わります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 30 InTouch サービスの管理

サービスとは、ユーザー インターフェイスを使わずに、またはユーザーがログオンしなくても、特定のシステム機能をバックグラウンドで自動的に実行する Windows のプロセスのことです。

Windows サービスでは、以下のスタートアップ オプションを使用できます。

- **自動**。Windows が再起動すると、サービスが自動的に開始されます。ユーザーが操作する必要はありません。
- **手動**。ユーザーまたはアプリケーション プロセスによって、明確にサービスの開始が指定されなければなりません。
- **無効**。サービスは開始されません。これはトラブルシューティングの際に便利です。

注意：InTouch WindowViewer サービスのパラメータ オプションはサポートされません。

Windows セキュリティ システムを犠牲にすることなく、サービスが開始されます。

InTouch HMI には、以下の Windows サービスが含まれています。

- Alarm DB Logger
- Alarm DB Purge/Archive
- NetDDE Helper
- SuiteLinkSuiteLink
- WindowViewerWindowViewer

サービスとしての WindowViewer の実行

サービスとして実行されるように WindowViewer を設定した場合、アプリケーションがインストールされているコンピュータが起動したときに、WindowViewer が自動的に起動します。WindowViewer サービスはバックグラウンドで動作します。WindowViewer サービスが実行している場合、WindowViewer の別のインスタンスを開始することはできません。

WindowViewer をサービスとして実行すると、以下のようなメリットがあります。

- ほとんどの障害回復プランでは、停電が回復した後ですぐに主要のコンピュータ システムが起動することが要求されます。Microsoft Windows Server は、停電が回復した後、自動的に再起動できます。WindowViewer がサービスとして実行されていると、工場の自動化システムはすぐに実行を開始できます。WindowViewer で最後に開かれた InTouch アプリケーションは、コンピュータが再起動したときに自動的に起動します。
- WindowViewer は履歴データのログを続行し、アラーム情報を収集し、スクリプトを処理し、I/O サーバーとして機能し、別のオペレータがログオンおよびログオフしたときでも I/O クライアントとして値を書き込みます。

  注意：ネットワーク アプリケーションをサービスとして実行する場合、またはネットワーク パスを履歴ログ フォルダとして使用する場合、ログオン ユーザーは、ネットワークの場所への適切なアクセスが必要です。

WindowViewer が既にサービスとして実行されている場合に、ショートカット アイコンから WindowViewer を起動しようとしたり、Windows の［**スタート**］メニューから［WindowViewer］を選択

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

して起動しようとしたりすると、Operations Control Log Viewer にメッセージが記録されます。メッセージには、サービスとして実行するよう設定された場合に WindowViewer を起動する際の制限に関する説明が表示されます。

WindowViewer がサービスとして既に実行しているときにアプリケーション マネージャまたは WindowMaker を起動しようとすると、警告メッセージが Operations Control Log Viewer に記録されます。メッセージは、WindowViewer をサービスとして実行するときはアプリケーション マネージャおよび WindowMaker を開くことができない旨が示されます。

重要：WindowViewer をサービスとして実行する場合、管理者特権でサービスを実行する際に生じる潜在的なセキュリティの問題を軽減するために、ユーザー アカウントの権限は非対話型に設定されます。

サービスとして起動するための WindowViewer の設定

Windows サービスとして WindowViewer を実行すると、オペレータのログオフ後の継続稼動や、システム ブート時のオペレータを介さない自動起動などの機能を使用できます。これによって、オペレーティング システムのセキュリティを犠牲にすることなく、無人で WindowViewer を起動できます。

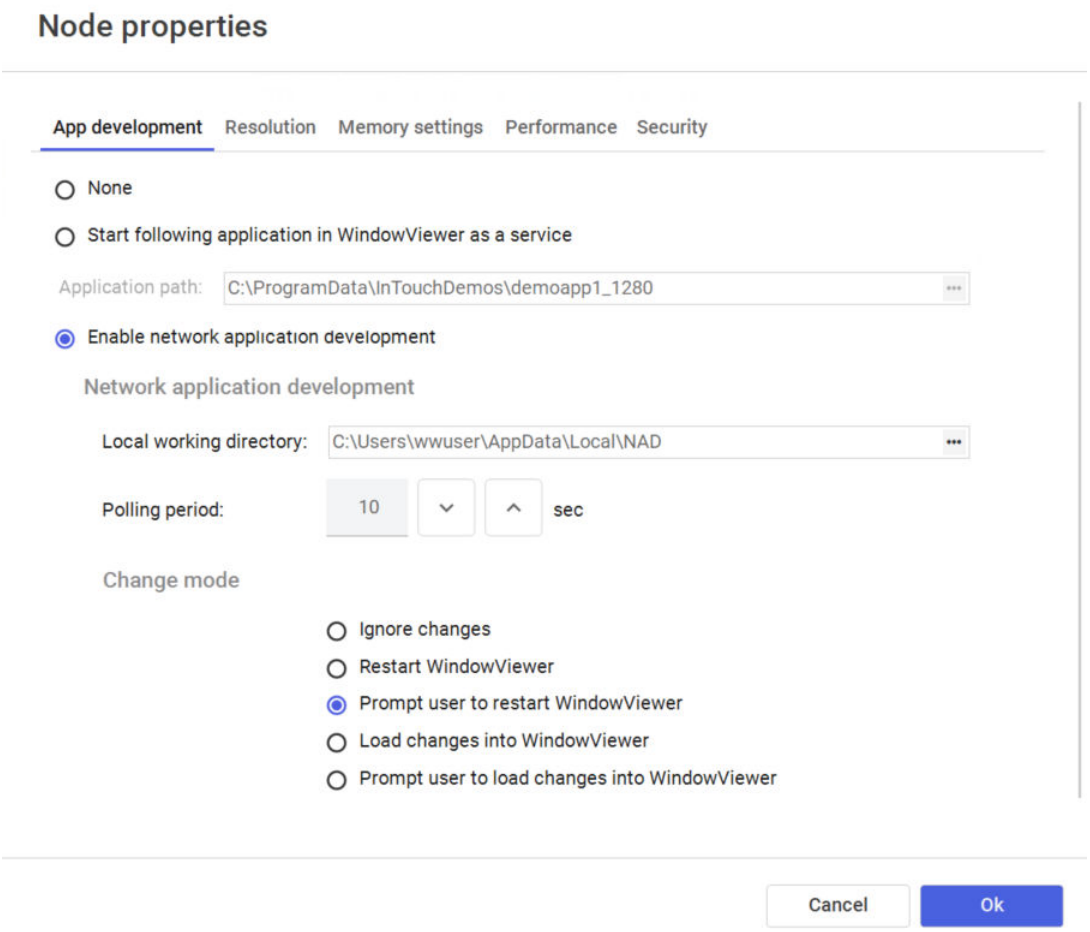
サービスとして起動するように WindowViewer が設定されている場合、InTouch アプリケーションもサービスとしての WindowViewer 内で実行するよう指定する必要があります。アプリケーション ディレクトリは、［**ノードのプロパティ**］ダイアログ ボックスで指定するか、WIN.INI ファイルに手動で入力することができます。

サービスとして起動するための WindowViewer の設定

1. InTouch アプリケーション マネージャを起動します。
2. ［**ツール**］メニューで、［**ノードのプロパティ**］を選択します。

   ［**ノードのプロパティ**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［**以下のアプリケーションをサービスとして WindowViewer で起動**］を選択して、サービスとして自動的に実行するよう WindowViewer を設定します。

   ［**アプリケーション パス**］フィールドが有効になります。

4. 省略記号ボタンを選択してファイル エクスプローラを表示し、目的の InTouch アプリケーションを選択します。

   アプリケーション ディレクトリがグループ ボックスに入力されます。

5. ［**OK**］ を選択します。

6. アプリケーション マネージャのツールボックスで［WindowViewer］アイコンを選択します。

   WindowViewer が、指定した InTouch アプリケーションのサービスとして実行するようになります。

注意：上記の手順に従って［ノードのプロパティ］を設定している場合、WindowMaker から WindowViewer にすばやく切り替えて、InTouch アプリケーション用に WindowViewer サービスを起動することもできます。この操作は、アプリケーション マネージャから WindowViewer を起動するときに行うことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowViewer でアプリケーションをサービスとして実行するための WIN.INI の編集

［ノードのプロパティ］で［**以下のアプリケーションをサービスとして WindowsViewer で起動**］オプションが有効な場合、アプリケーション ディレクトリを WIN.INI ファイルに手動で入力できます。アプリケーション マネージャでアプリケーションを選択する前に WIN.INI ファイルを更新する場合、WindowViewer は、選択した後にアプリケーションのサービスとして実行します。

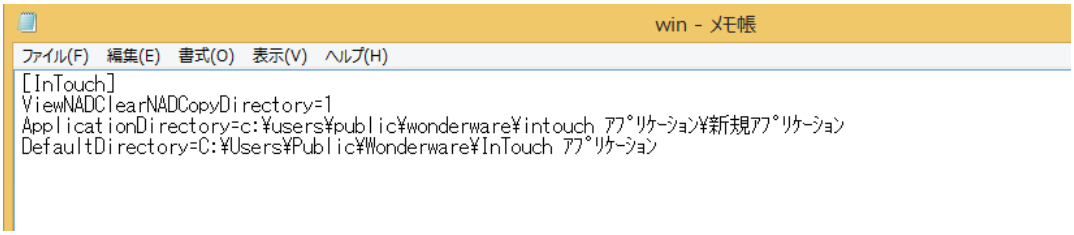
WindowMaker で開いたアプリケーションで WIN.INI を更新することもできます。その後にランタイムをすばやく切り替えると、WindowViewer はアプリケーションのサービスとして実行します。

注意：上記の機能はマネージド InTouch アプリケーションではサポートされていません。WindowMaker からマネージド アプリケーションを切り替えて WindowViewer のサービスとして実行しようとすると、警告メッセージが Logger に記録されます。

WIN.INI は次の場所にあります。

```
C:\ProgramData\Wonderware\InTouch\Service\win.ini
```

サービスとして実行するアプリケーションのディレクトリを入力します。次の例を参照してください。



手動によるサービスの開始

Windows のコントロール パネルを使用して、InTouch WindowViewer サービスを手動で起動できます。

サービスとして起動するよう設定しない限り、サービス コントロール パネルには WindowViewer が表示されません。詳細については、「サービスとして起動するための WindowViewer の設定」を参照してください。

コントロール パネルを使用して WindowViewer サービスを起動する

1. コントロール パネルを起動します。
2. ［**管理ツール**］をダブルクリックし、次に［**サービス**］をダブルクリックします。［**サービス**］ダイアログ ボックスが表示されます。
3. 詳細ペインで、［**Wonderware WindowViewer**］サービスを右クリックし、次に［**開始**］を選択します。

重要：コマンド プロンプトを使用して WindowViewer をサービスとして起動することはできません。

サービスの停止

コントロール パネルを使用して、WindowViewer サービスを手動で停止できます。

コントロール パネルを使用して WindowViewer サービスを停止する

1. コントロール パネルを起動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**管理ツール**］をダブルクリックし、次に［**サービス**］をダブルクリックします。［**サービス**］ダイアログ ボックスが表示されます。
3. 詳細ペインで［WindowViewer］を右クリックし、次に［**サービスの停止**］を選択します。

InTouch サービスのユーザー アカウントの設定

デフォルトでは、Windows サービスはローカル システム アカウントを使用して実行されます。InTouch サービスには管理特権を持つユーザー アカウントが必要ですが、このようなユーザー アカウントはローカル システム アカウントでは提供されていない場合があります。

InTouch HMI をインストールするとき、すべての AVEVA サービスを実行する管理者のアカウントを指定します（アカウントがまだ作成されていない場合）。このアカウントはプライマリ アカウントと見なされます。InTouch サービスでは、自動的に起動するためにプライマリ アカウントが使用されます。

注意 : プライマリ アカウントは偽装アカウントとも呼ばれます。偽装アカウントとは、サイトやサーバーの制限付きリソース「領域」へのアクセス権を持ったユーザーまたはグループ アカウントのことです。

プライマリ アカウントを変更するには、Change Network Account ユーティリティを使用します。

警告 : プライマリ アカウントの変更は、InTouch サービスだけではなく、すべての AVEVA サービスに影響します。

プライマリ アカウントを変更する

1. ［**スタート**］メニューの［**プログラム**］、［**AVEVA**］の順にポイントし、［**ネットワーク アカウントの変更**］を選択します。［**ネットワーク アカウントの変更**］ダイアログ ボックスが表示されます。
2. ユーザー アカウントを変更します。詳細については、ネットワーク アカウントの変更のマニュアルを参照してください。
3. ［**OK**］ を選択します。

InTouch サービスのトラブルシューティング

あるサービスの起動が他のサービスの起動に依存している場合、そのサービスを起動する前に必要なサービスが実行されているかどうかが Windows によって確認されます。

WindowViewer を実行するための必要条件に基づいて、以下の依存関係を把握する必要があります。

- Distributed Alarming または Distributed History を使用する場合、およびネットワーク DDE データにアクセスする場合、NetDDE Helper サービスが実行している必要があります。

  また、NetDDE Helper サービスを使用するには、Network DDE および Network DDE DSDM の両方のサービスが手動または自動スタートアップでインストールおよび構成されている必要があります。インストール中、NetDDE Helper サービスが［手動］スタートアップに設定されます。つまり、コンピューが起動するときに、WindowViewer でこのサービスが自動的に起動されます。

- WindowViewer を SuiteLink サーバーまたはクライアントとして使用する場合、SuiteLink サービスが実行している必要があります。

  SuiteLink サービスを使用するには、Microsoft TCP/IP がインストールされている必要があります。

- WindowViewer が実行している間にメッセージまたはエラーを保存するには、Operations Control Log Viewer サービスがインストールされていることを確認する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SuiteLink と Operations Control Log Viewer の両方のサービスは、自動スタートアップでインストールおよび設定してください。

サービスのエラー メッセージの表示

サービスに関連するエラー メッセージをトラブルシューティングするには、Windows イベント ビューアを使用します。たとえば、「1 個または複数のサービスを開始できませんでした...」というエラーが発生する場合があります。Windows イベント ビューアは Windows サービスの起動中に発生した情報提供のメッセージ、警告、またはエラーを一覧表示します。イベント ビューアの詳細については、Microsoft のマニュアルを参照してください。

InTouch サービスが起動に失敗したことによるすべての警告またはエラー メッセージを表示できます。イベント ビューアに WindowViewer サービスが起動に失敗したことが示される場合、依存関係にある必要なサービスが実行されていないことが原因であることがほとんどです。

サービス ユーザー アカウントの問題のトラブルシューティング

InTouch サービスのインストールが失敗した場合、または InTouch HMI のインストール後に起動に失敗した場合、実行中のユーザー アカウントに問題が発生する場合があります。

サービスのユーザー アカウントの問題をトラブルシューティングする

1. Windows ユーザー マネージャ ウィンドウを開き、新しいプライマリ ユーザー アカウントを作成します。

   InTouch コンポーネントをサービスとして起動するには、このユーザー アカウントがローカル コンピュータの管理者権限を持っている必要があります。ドメイン リストにコンピュータのノード名が表示されない場合は、ノード名を手動で入力します。

   詳細については、「InTouch サービスのユーザー アカウントの設定」を参照してください。

2. コンピュータのノード名の長さが 14 文字を超えないことを確認します。ノード名にアンダースコア（_）またはダッシュ（-）が含まれている場合は、削除します。
3. インストール中、ドメイン名を入力するよう求めるメッセージが表示されたら、ドメイン名ではなくコンピュータのノード名を入力します。その後、手順 1 で作成したユーザー名とパスワードを入力します。
4. InTouch HMI がすでにインストールされている場合でも、ArchestrA Change Network Account ユーティリティを実行することによって、ドメイン名、ユーザー名、およびパスワードを指定できます。
5. コンピュータを再起動します。
6. 有効なユーザー アカウントを使用して、ネットワーク ドメインにログオンします。ドメインがダウンした場合でも、ローカル コンピュータで実行している InTouch アプリケーションに影響は与えられません。

有効 I/O アイテムの使用解除

Windows オペレーティング システムを起動すると、自動的に開始するよう設定されているサービスは自動的に「バックグラウンド」で開始し、デスクトップにユーザー インターフェイスは表示されません。この状態のサービスはシステム コンテキストで実行されています。ユーザーがシステムにログオンすると、システム コンテキストで実行されていて、ユーザー インターフェイスが関連付けられているサービスは、自動的にデスクトップに表示されます。この状態では、サービスは今度はデスクトップ コンテキストで実行されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

自動的に開始するように WindowViewer サービスを設定すると、オペレーティング システムの起動時にサービスはシステム コンテキストで実行されます。その後、ユーザーがログオンすると、WindowViewer サービスは実行され続けますが、デスクトップ コンテキストで実行され、WindowViewer のユーザー インターフェイスが自動的に表示されます。

［**有効アイテムのみ要求**］オプションをオンにして定義された InTouch アクセス名および特定の InTouch アプリケーション ウィンドウでのみアクティブである I/O タグ変数（アプリケーションの他の場所では使用されないタグ変数）がある場合は、それらのタグ変数を「非アクティブ化」できます。たとえば、WindowViewer がサービスとして実行されていて、スクリプトを使用してアプリケーション ウィンドウを閉じると、ウィンドウはメモリから自動的にアンロードされ、これらのタグ変数へのリンクは解除されます。

InTouch サービスのレジストリ キー

InTouch サービスは Windows レジストリのキーとして一覧表示されます。

SuiteLink：

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SLS
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\slssvc
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SuiteLink

NetDDE Helper：

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WWNetDDE

WindowViewer：

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VIEW

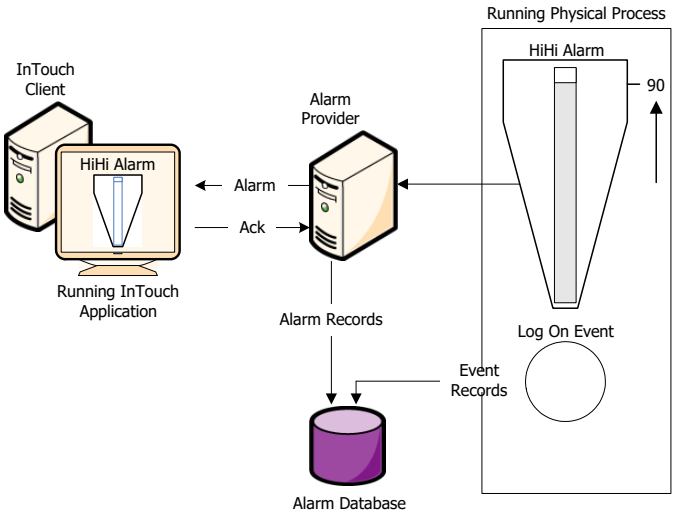
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 31 アラーム

アラームとイベントを生成してオペレータにプロセスの動作状況を通知する InTouch アプリケーションを作成できます。

- 問題を引き起こす可能性のあるプロセス状態に関する警告が、アラームによってランタイム オペレータに警告されます。通常は、プロセス値が定義済みのしきい値を超えた場合にトリガするようにアラームを設定します。オペレータは、アラームを確認する必要があります。
- イベントは、通常のシステム ステータス メッセージを表します。オペレータが InTouch アプリケーションにログオンするなど、システム条件が発生するときが典型的なイベントとなります。オペレータは、イベントを確認する必要はありません。

以下の図は、アプリケーションが実行されている間、InTouch HMI でアラームとイベントが処理される方法を示しています。アラームとイベントのデータは、アラーム データベースに保存されます。



イベント監視用のタグを設定できます。イベント メッセージは、タグの値が変更するたびにアラーム システムにログ記録されます。イベント メッセージには、値がどのように変更されたかに加えて、オペレータ、I/O、スクリプト、またはシステムのいずれによってその変更が開始されたのかという情報が含まれます。

従来のアラーム システムからの移行

標準アラーム システムまたは AlarmSuite を使用して構築されたアプリケーションを移行できます。

標準アラーム システムから分散アラーム システムへの移行

標準アラーム システムを分散アラーム システムに移行するとき、マスタ/スレーブ アプリケーションのすべての標準アラーム表示は分散アラーム表示オブジェクトに移行されます。

色、フォント、式、およびアラーム クエリーの設定は移行されません。新しい分散アラーム表示オブジェクトには、以下のデフォルト クエリーがあります。ここで、ノード名はマスタ ノードの名前です。

\\nodename\InTouch!$system

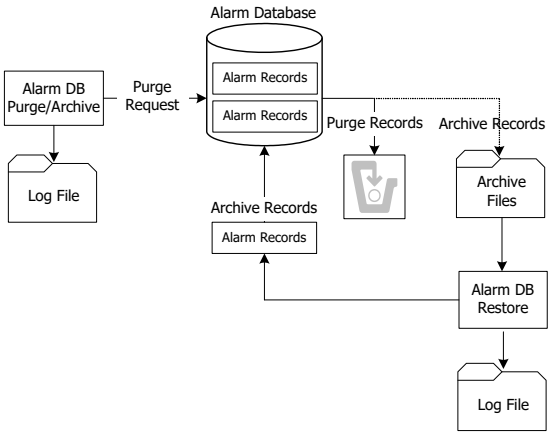
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

確認とアラーム ステータスのドットフィールドは以前と同じように機能し続けます。I/O タグ変数が NetDDE または SuiteLink のどちらに対して設定されているかによって、NetDDE を有効にする必要がある可能性があります。ただし、分散アラーム表示オブジェクトを使用してアラームが確認できるようになったため、確認を発行するために個別のコントロールは必要ないと判断する場合もあります。

アラーム データベースの維持

アラーム データベースは、2 つの InTouch ユーティリティを使用して管理します。レコードをデータベースから永久に削除するか、ファイルにアーカイブするには Alarm DB Purge-Archive ユーティリティを使用します。データベースが破壊された場合、Alarm DB Restore ユーティリティを使用して、アーカイブされたレコードを復元します。

以下の図は、両方のユーティリティがレコードをパージ／アーカイブして、データベースに復元する方法を示しています。



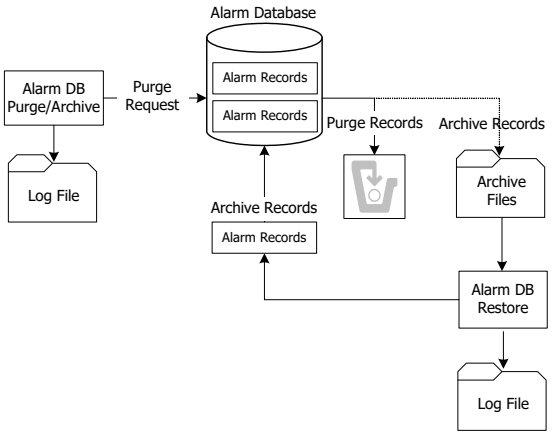
Alarm DB Purge-Archive ユーティリティを使用するには、管理者としてコンピュータにログインしている必要があります。

アラーム データベースの維持の概要

アラーム データベースは、2 つの InTouch ユーティリティを使用して管理します。レコードをデータベースから永久に削除するか、ファイルにアーカイブするには Alarm DB Purge-Archive ユーティリティを使用します。データベースが破壊された場合、Alarm DB Restore ユーティリティを使用して、アーカイブされたレコードをリストアします。

以下の図は、両方のユーティリティがレコードをパージ／アーカイブして、データベースにリストアする方法を示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



Alarm DB Purge-Archive ユーティリティを使用するには、管理者としてコンピュータにログインしている必要があります。

パージ設定またはアーカイブ設定

Alarm Purge-Archive ユーティリティは、以下の操作を実行するために使用します。

- アラーム データベースからパージするレコード タイプを選択する
- 毎日、毎週、または毎月のスケジュールでレコードを自動的にパージする
- パージされたデータベース レコードをオプションでファイルにアーカイブする
- 問題をトラブルシューティングするためにパージ操作またはアーカイブ操作のステータスを保存する
- パージ操作またはアーカイブ操作のステータスを表示する

重要 : Alarm DB Purge-Archive ユーティリティは Windows サービスのセットとして実行します。管理者権限で Alarm DB Purge-Archive ユーティリティを実行するときにセキュリティの脅威にさらされる可能性を削減するために、ユーザー アカウントの許可は非対話型に変更されます。

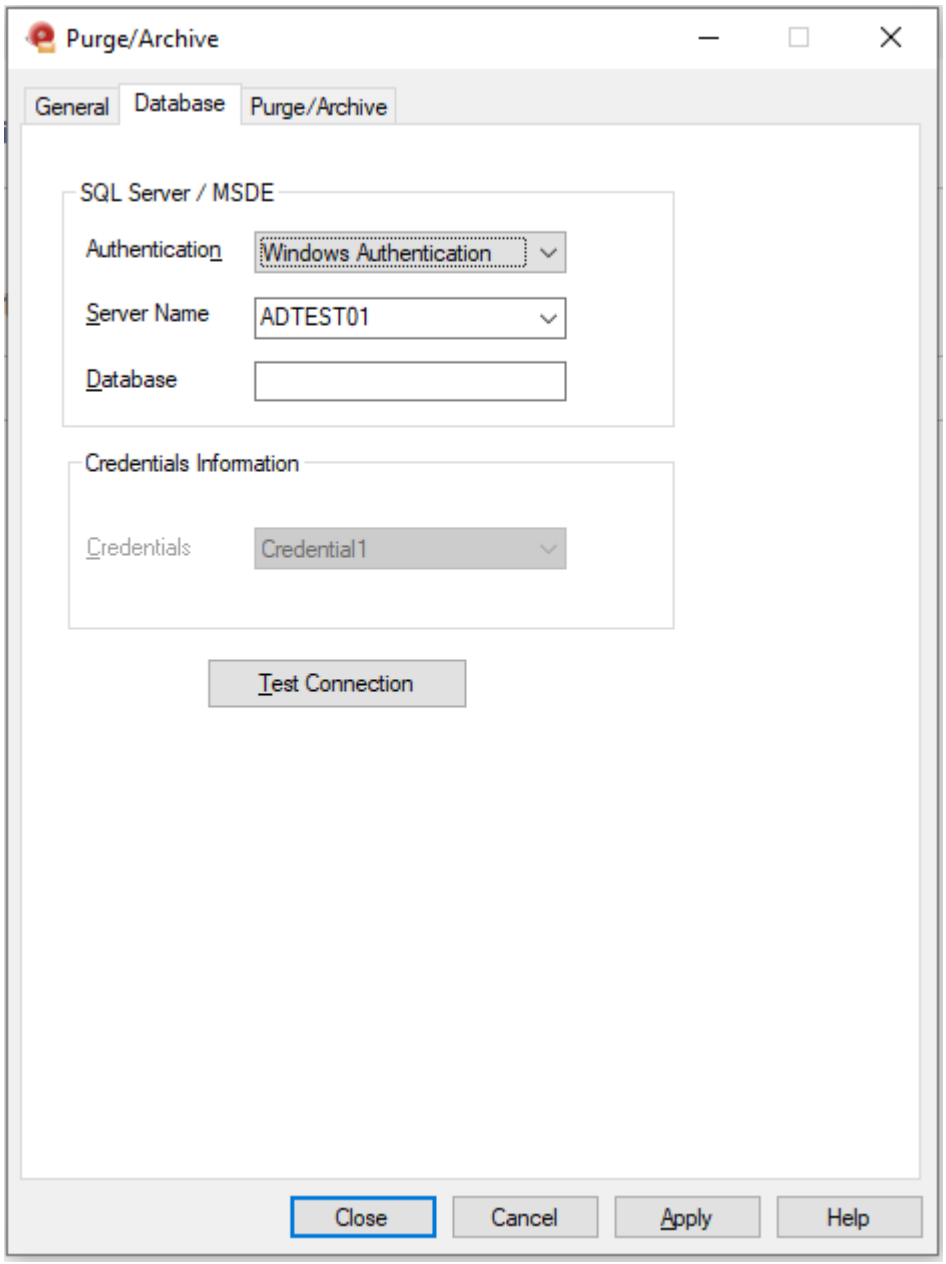
アラーム データベースへの接続

Alarm DB Purge-Archive ユーティリティを使用する前に、アラーム データベースに接続する必要があります。

データベース接続を設定する

1. **Alarm DB Purge-Archive** ユーティリティを開きます。以下の手順を実行します。
   a. ［**ツール**］ビューで［**アプリケーション**］を展開します。
   b. ［**Alarm DB Purge-Archive**］をダブルクリックします。
2. [**データベース**] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. データベース接続を設定します。以下の手順を実行します。

   a. [**認証**] リストで、認証方法(**SQL Server 認証** または **Windows 認証** (既定)) を選択します。

注：Windows 認証は、SQL 認証よりも優れたアプリケーション セキュリティを提供できます。この理由から、Windows 認証から SQL 認証に切り替えると、Windows 認証を使用することを推奨するポップアップ ダイアログが表示されます。この警告を無視して SQL 認証を使用する場合は、［OK］を選択します。同様のメッセージが Log Viewer に記録されます。

   a. ［**サーバー名**］リストで、サーバーのノード名を選択します。

   b. ［**データベース**］ボックスにアラーム データベースの名前を入力します。

   c. ［**資格情報**］領域の［**資格情報**］ドロップダウンから、認証に使用する資格情報を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：［資格情報］フィールドは、SQL Server 認証タイプを選択した場合にのみ有効になります。Windows 認証方法では、現在ログインしているユーザーの資格情報が使用され、［資格情報］フィールドは無効になります。スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「資格情報マネージャの操作」を参照してください。

4. ［**接続テスト**］を選択して、データベースへの接続をテストします。アラーム データベースへの接続が正常に終了したかどうかを示すメッセージが表示されます。
5. ［**OK**］ を選択します。
6. **[適用]** を選択します。

サーバーからパージするデータ量の設定

以下の操作を実行できます。

- アラーム データベースからパージするアラーム レコードを選択する
- オプションでアラーム データベースからファイルへパージされたレコードをアーカイブする
- パージ ログ ファイルを保存するフォルダ場所を選択する

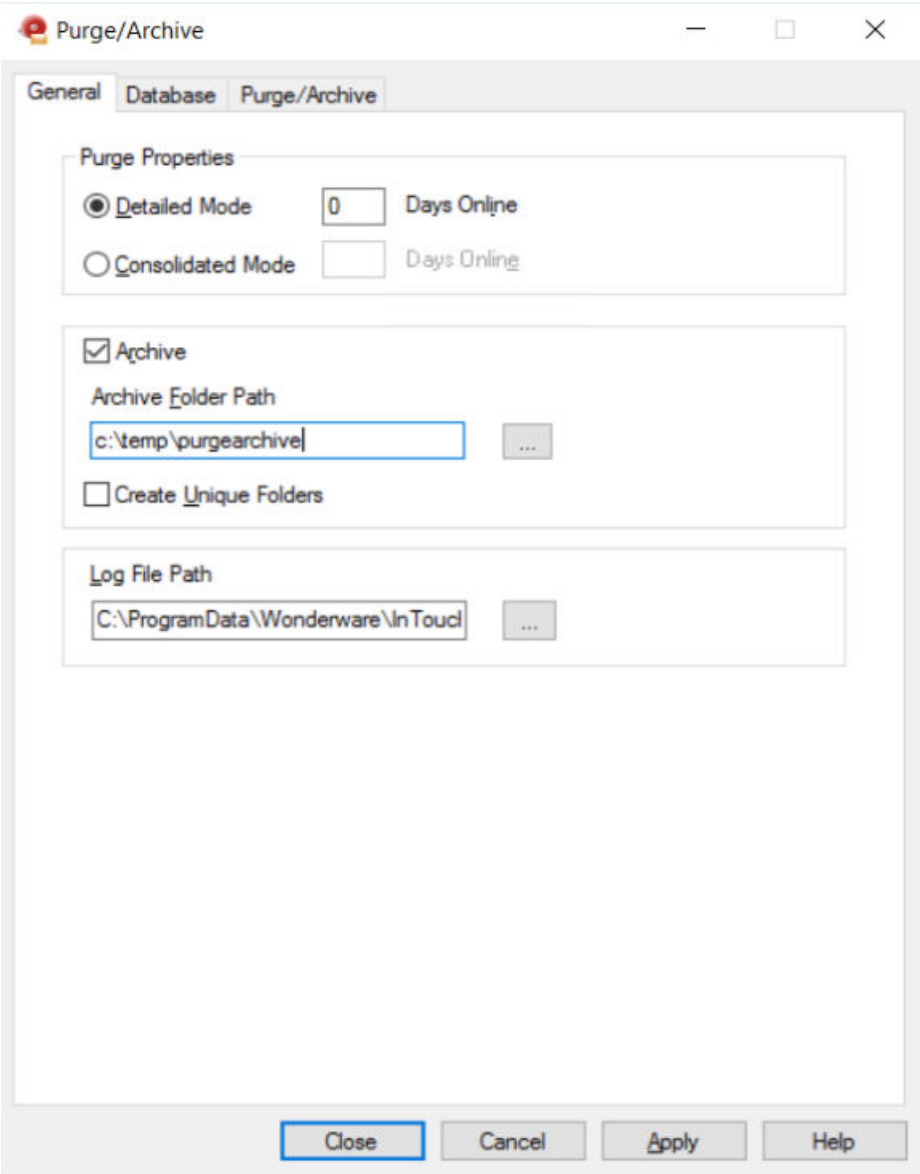
パージする必要のあるテーブルのタイプを AlarmDetail または AlarmConsolidated のいずれかのテーブルから選択します。

指定した日数の 1 日前のすべてのデータがパージされます。有効なエントリは 0 ～ 9999 です。0 を選択すると、現在の日付のレコードを除くすべてのレコードがアラーム データベースからパージされます。

パージするレコードを選択する

1. **Alarm DB Purge-Archive** ユーティリティを開きます。以下の手順を実行します。
   a. ［**ツール**］ビューで［**アプリケーション**］を展開します。
   b. ［**Alarm DB Purge-Archive**］をダブルクリックします。
2. [**全般**] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［**パージのプロパティ**］領域で、パージするレコードのタイプを設定します。以下のいずれかを実行します。
   - 詳細モードでデータベースに保存されているアラーム レコードをパージするには、［**詳細モード**］を選択します。
   - 統合モードでデータベースに保存されているアラーム レコードをパージするには、［**統合モード**］を選択します。
4. ［**オンライン日数**］ボックスに、アラーム データベースに保持するレコードの日数を入力します。
5. **[適用]** を選択します。

パージされたデータのアーカイブの設定

アラーム データベースからパージされたレコードをアーカイブし、Alarm DB Restore ユーティリティを使用して、それらのレコードを復元します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム データベースをパージする場合、Alarm DB Purge-Archive ユーティリティではパージされたアラーム データベース テーブルに対応する 9 つのアーカイブ ファイルが自動的に作成されます。各ファイルには 1 つのテーブルのパージされたレコードが含まれます。

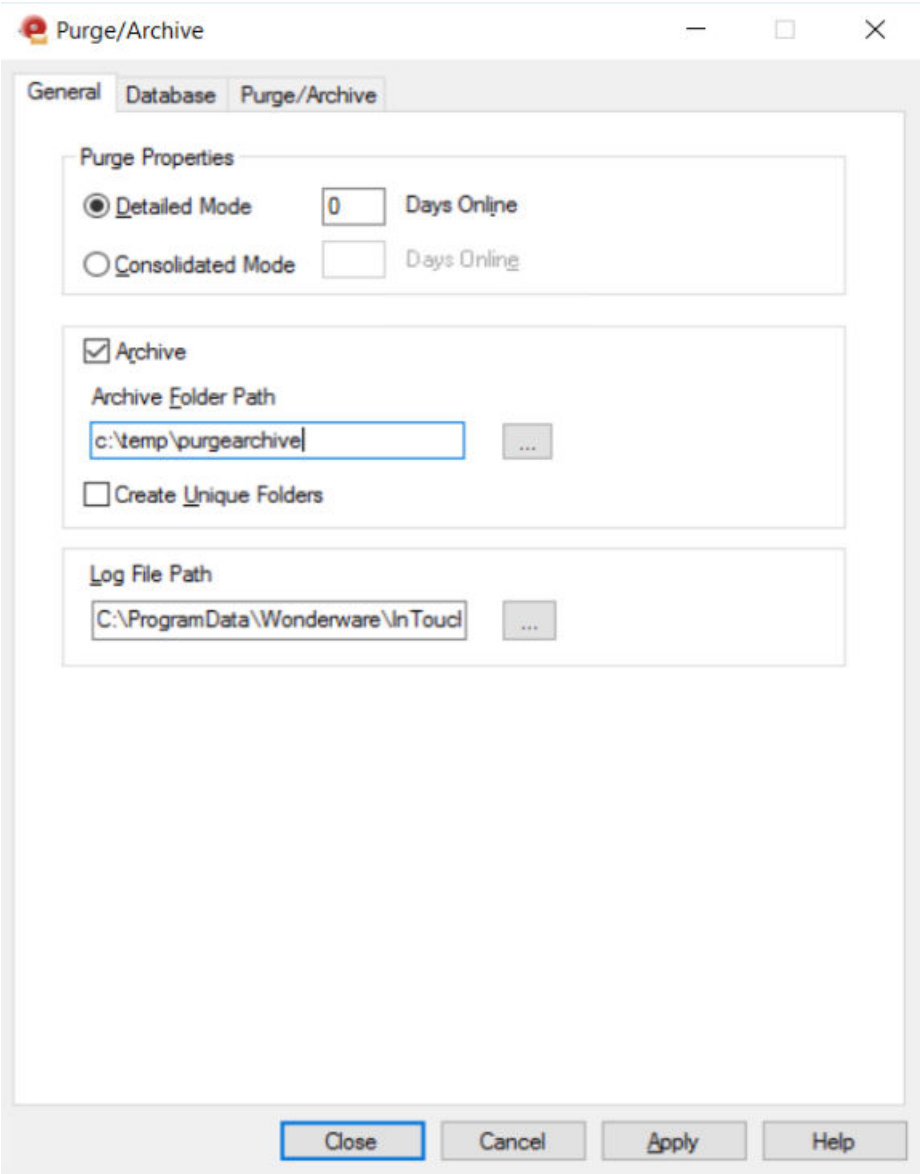
Alarm DB Purge-Archive ユーティリティでは、パージ操作が発生した際、テーブル名、日付、および時間に基づいてアーカイブされたファイルに名前が割り当てられます。たとえば、2007 年 6 月 22 日の午後 5 時 30 分にパージされた AlarmMaster テーブルに対するアーカイブ ファイルの名前の形式は、以下のようになります。

AlarmMaster_06222007_1730.txt

アーカイブを設定する

1. **Alarm DB Purge-Archive** ユーティリティを開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm DB Purge-Archive**］をダブルクリックします。
2. [全般] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［**アーカイブ**］チェック ボックスをオンにします。
4. ［**アーカイブ フォルダ パス**］ボックスに、アーカイブ ファイルが保存されるフォルダ場所を入力するか、参照ボタンを選択して場所を参照します。
5. アーカイブ ファイルをアーカイブ ファイル フォルダの下の個々のサブフォルダに配置するには、［**一意のフォルダを作成**］チェック ボックスをオンにします。
6. **[適用]** を選択します。

ログ ファイルの設定

Alarm DB Purge-Archive ユーティリティでは、パージ操作中にステータス メッセージが生成されます。これらのメッセージは、ユーティリティの［**ステータス**］ウィンドウからオンラインで表示されます。Alarm DB Purge-Archive ユーティリティではまた、パージ メッセージが WWAlmPurge.log という名前のパージ ログ ファイルに書き込まれます。

以下の例は、パージ操作が正常に終了した後でログに保存されるメッセージを示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
パージ開始時刻: 12:16:48 PM 6/22/2007
トランザクションの開始...
ProviderSession テーブルをアーカイブ中...
Query テーブルをアーカイブ中...
Casue テーブルをアーカイブ中...
Alarm Core テーブルをアーカイブ中...
OperatorDetails テーブルをアーカイブ中...
AlarmDetail テーブルをアーカイブ中...
Comment テーブルをアーカイブ中...
Events テーブルをアーカイブ中...
TagStatus テーブルをアーカイブ中...
データベースのすべてのレコードをパージ中...
実行中...
パージ完了時刻: 12:16:52 PM 6/22/2007
144 AlarmMaster から取得したレコードは、別のテーブルから取得した関連するレコードと共にパージされました。
```

デフォルトでは、パージ ログ ファイルは C:\Documents and Settings\All Users\Application Data\Wonderware\InTouch フォルダに保存されます。Microsoft Windows Vista オペレーティング システムが実行されているコンピュータでは、デフォルトのアプリケーション フォルダは、C:\Users\UserName\Documents\My InTouch Applications です。

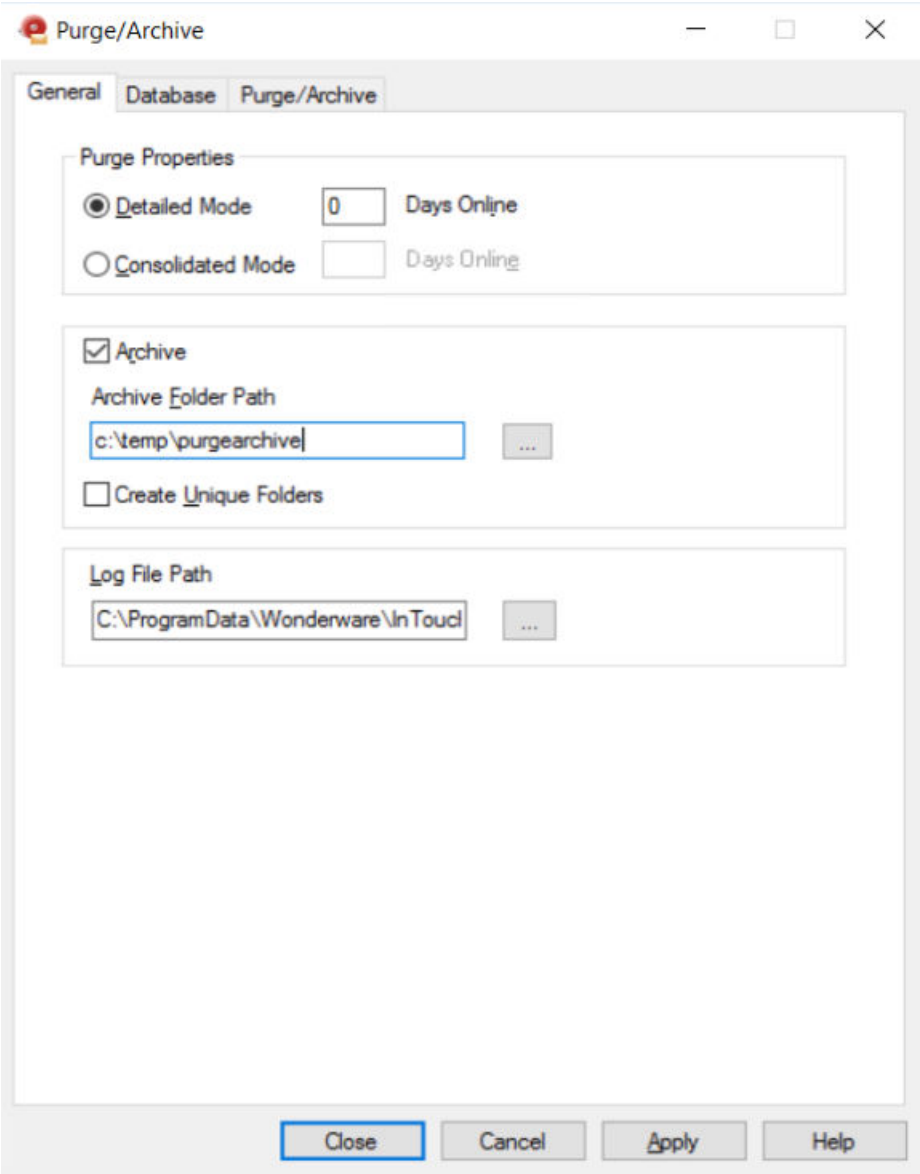
パージ ログ ファイルの保存場所は変更できます。

Alarm DB Purge-Archive ユーティリティでは、パージが発生するたびに新しいメッセージがログ ファイルに追加されます。

アーカイブ ログを設定する

1. **Alarm DB Purge-Archive** ユーティリティを開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm DB Purge-Archive**］をダブルクリックします。
2. [全般] タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［ログ フォルダ パス］ボックスに、パージ ログ ファイルが保存されるフォルダ場所を入力するか、参照ボタンを選択して場所を参照します。
4. **[適用]** を選択します。

データベースの手動によるパージとアーカイブ

アラーム データベースは手動でパージおよびアーカイブできます。これは実行日時よりも優先され、パージおよびアーカイブがすぐに開始されます。

パージ操作ではアーカイブ ファイルの存在が確認され、データが追加されます。アーカイブ ファイルが存在しない場合、ファイルは命名規則に従って作成され、アーカイブに使用されます。

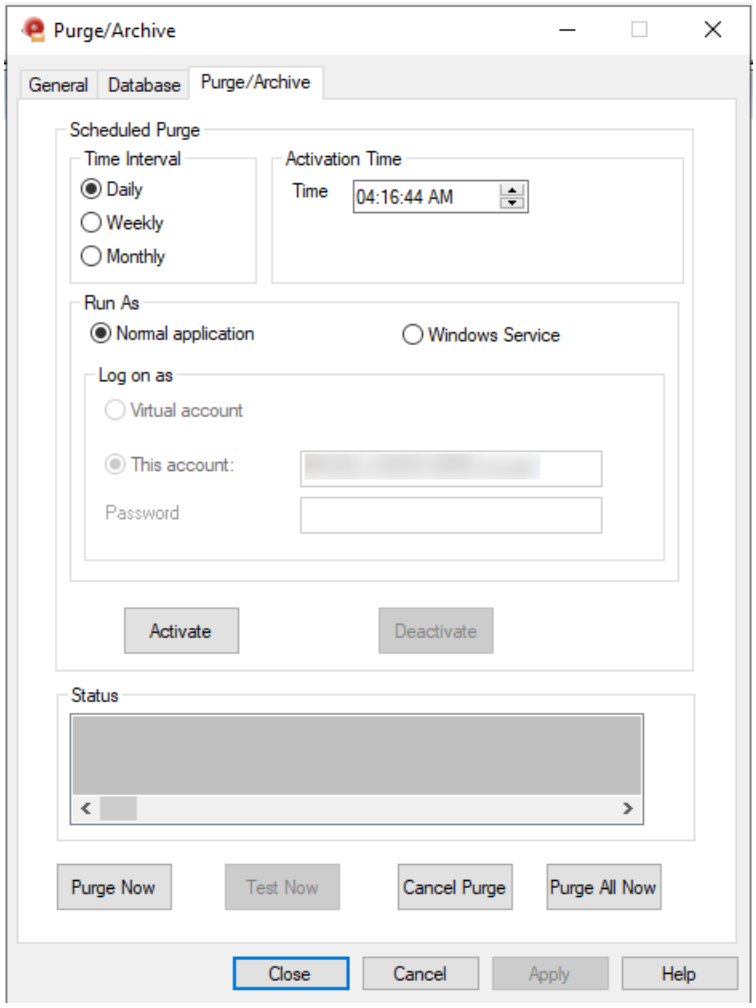
パージ操作では、外部キー制約を通じて AlarmMaster などのメイン テーブルにリンクされている ProviderSession、Query、および Cause などのテーブルのエントリは削除されません。これらのテーブルの関連レコードは、データの一貫性を維持し、データベースで保持するために記述されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

警告：すべてのレコードの手動パージ（[全てを実行] オプション）は、Alarm DB Logger サービスが停止している場合にのみ実行します。Alarm DB Logger サービスが実行している間にパージ操作が正常に実行された場合、Alarm DB Logger サービスはロギングを停止して、レコードのキャッシュを開始します。

アラーム データベースから手動でレコードをパージおよびアーカイブする

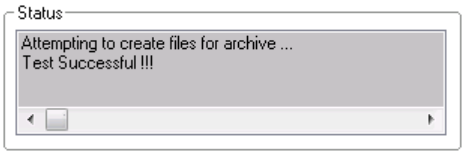
1. **Alarm DB Purge-Archive** ユーティリティを開きます。以下の手順を実行します。
   a. [ツール] ビューで [アプリケーション] を展開します。
   b. [**Alarm DB Purge-Archive**] をダブルクリックします。
2. [パージ/アーカイブ] タブを選択します。



3. [テスト] を選択して、データベースとアーカイブ場所への接続を確認するテスト パージを実行します。

   テスト パージによって、指定したアーカイブ フォルダに空のアーカイブ ファイルが作成されます。[ステータス] 領域に、テストが正常に終了したことを示すメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



［テスト］ボタンは、パージされたレコードをアーカイブすることを選択した場合のみ使用できます。［アーカイブ］オプションは［全般］タブに配置されています。

4. データベースからレコードをパージします。以下のいずれかを実行します。
   - 選択したレコードをパージするには、［今すぐ実行］を選択します。
   - すべてのレコードをパージするには、［全てを実行］を選択します。
5. パージを停止するには、［中止］を選択します。パージをキャンセルした場合、アラーム データベースは元の状態にロールバックされます。

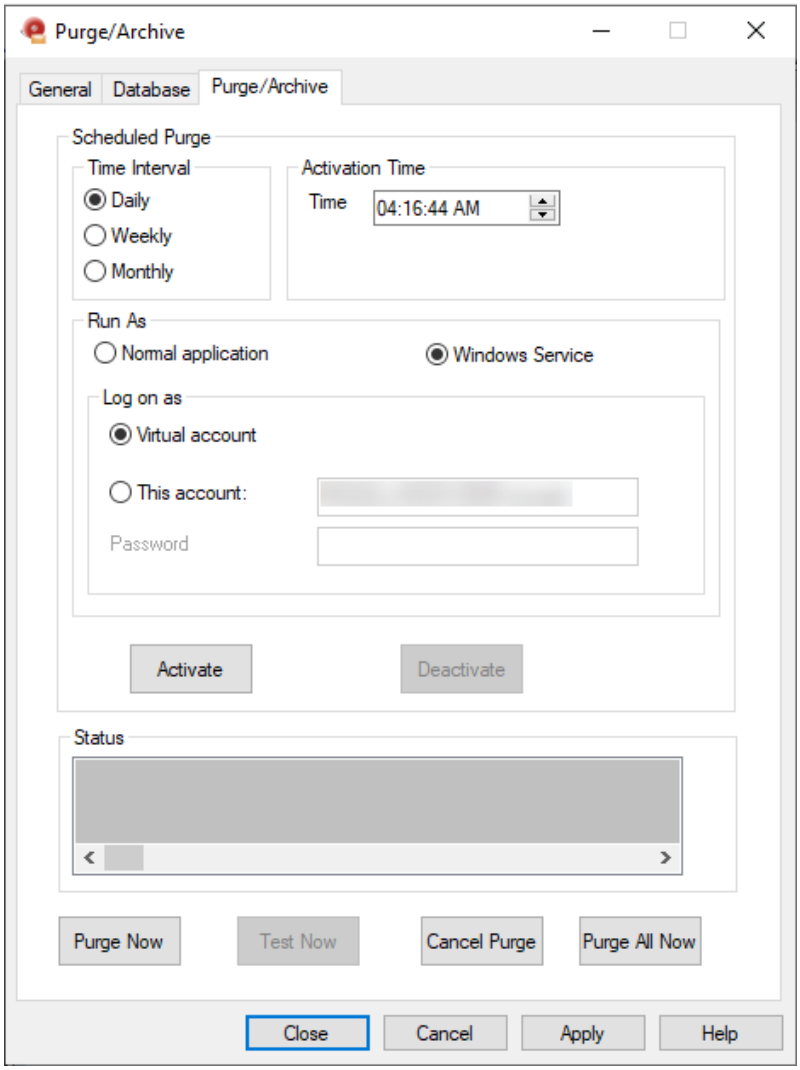
自動パージのスケジュール設定

Alarm DB Purge-Archive ユーティリティは、スケジュールされた間隔でレコードをアラーム データベースから自動的にパージまたはアーカイブできます。テスト パージを実行して、データベースやターゲット場所への接続の検証、およびパージの開始と停止を行うことができます。

自動パージのスケジュール設定

1. **Alarm DB Purge-Archive** ユーティリティを開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm DB Purge-Archive**］をダブルクリックします。
2. ［パージ/アーカイブ］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［時間間隔］領域で、毎日、毎週、毎月のいずれかからパージ間隔を選択します。

   ［**毎週**］または［**毎月**］を選択すると、［**実行日時**］領域に［**日**］ボックスが表示され、曜日または日付を指定できます。

   ［**毎日**］を選択した場合、［**時間**］ボックスで、パージ/アーカイブ操作を開始する時間を設定します。

4. ［**実行**］領域で、［**標準アプリケーション**］を選択すると、Purge-Archive ユーティリティがアプリケーションとして実行されます。［**Windows サービス**］を選択すると、ユーティリティはサービスとして実行されます。

   ［**Windows サービス**］を選択した場合、［**仮想アカウント**］を選択するか、［**指定アカウント:**］領域で別のアカウントのユーザー名とパスワードを指定します。

   注：仮想アカウントの詳細については、仮想アカウントの使用を参照してください。

5. パージとアーカイブの設定を保存するには、［**適用**］を選択します。

6. Alarm DB Purge-Archive ユーティリティを自動パージ スケジュールに設定するには、［**アクティブ化**］を選択します。

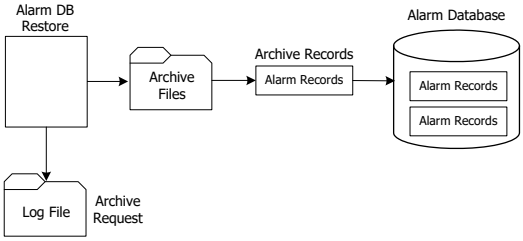
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

7. **[閉じる]**を選択します。

   自動パージ スケジュールを停止し、Windows サービスを削除するには、［**非アクティブ化**］を選択します。少し待つとサービスが Windows サービスから削除されます。

アラーム データベースの復元

Alarm DB Restore ユーティリティでは、アーカイブ ファイルのアーカイブされたアラーム レコードがアラーム データベースにリストアされます。以下の図は、アラーム レコードをデータベースにリストアする手順の概要を示しています。



データベースを復元するには、以下の操作を行う必要があります。

- アラーム データベースへの接続を設定します。
- アラーム データベースにリストアするアラームを選択します。
- アーカイブ レコードをアラーム データベースにリストアします。

Alarm DB Restore ユーティリティは最小化すると、システム トレイにアイコンで表示されます。アイコンを右クリックすると、メニューで以下のコマンドが表示されます。

コマンド	説明
リストア	リストア処理を開始します。
リストアのキャンセル	リストア処理をキャンセルします。
ステータスを消去	ステータス ウィンドウの内容をクリアします。
ウィンドウ消去	Alarm DB Restore ユーティリティを最小化し、システム トレイにアイコンで表示します。
ウィンドウ表示	Alarm DB Restore ユーティリティを開いて、最大化します。
終了	Alarm DB Restore ユーティリティを閉じます。

Alarm DB Restore ユーティリティ内を右クリックすると、同じメニューが表示されます。

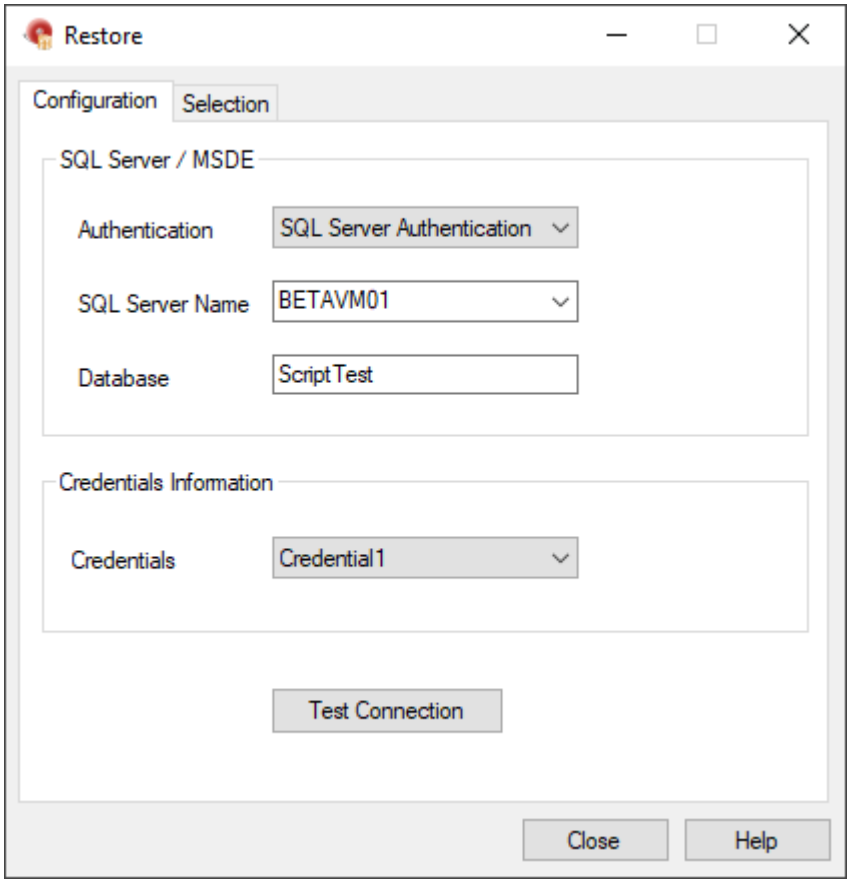
データベース接続を設定する

アーカイブ済みデータを復元するデータベースを選択する必要があります。指定したデータベースがサーバーに存在しない場合は、デフォルトのサーバー パラメータを使用して、新しいデータベースを作成するよう求めるメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アーカイブ済みデータを復元するデータベースを設定する

1. **Alarm DB Restore** ユーティリティを開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm DB Restore**］をダブルクリックします。
2. ［設定］タブを選択します。



3. アラーム データベースへの接続を設定します。以下の手順を実行します。
   a. [**認証**] リストで、認証方法(**SQL Server 認証** または **Windows 認証** (既定)) を選択します。

   注 : Windows 認証は、SQL 認証よりも優れたアプリケーション セキュリティを提供できます。この理由から、Windows 認証から SQL 認証に切り替えると、Windows 認証を使用することを推奨するポップアップ ダイアログが表示されます。この警告を無視して SQL 認証を使用する場合は、［**OK**］を選択します。同様のメッセージが Log Viewer に記録されます。

   b. ［**SQL Server 名**］リストで、アラーム データベースをホストするサーバーのノード名を選択します。
   c. ［データベース］ボックスにアラーム データベースの名前を入力します。
   d. ［資格情報］領域の［資格情報］ドロップダウンから、認証に使用するアラーム データベースの資格情報を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［資格情報］フィールドは、SQL Server 認証タイプを選択した場合にのみ有効になります。Windows 認証方法では、現在ログインしているユーザーの資格情報が使用され、［資格情報］フィールドは無効になります。

注 : スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。詳細については、『AVEVA™ InTouch HMI アプリケーション開発』の「資格情報マネージャの操作」を参照してください。

a. ［**接続テスト**］を選択して、データベースへの接続をテストします。アラーム データベースへの接続が成功したかどうかを通知するメッセージが表示されます。

b. ［**OK**］ を選択します。

4. **[閉じる]**を選択します。

復元するファイルの設定

レコードを復元する時間範囲、およびデータベース テーブルを再作成するかどうかを選択できます。

復元をキャンセルすると、データベースは元の状態にロールバックされます。

警告 : データベースに既に存在するアーカイブされたアラームを復元しようとしても、アーカイブ レコードは復元されません。この処理によって、データベースに重複したアラーム／イベント エントリが生じることが防止されます。レコードに関連付けられているアラーム GUID またはイベント GUID によって、アラームまたはイベントがデータベースに既に存在しているかどうかが判断されます。

復元するデータベース レコードを選択する

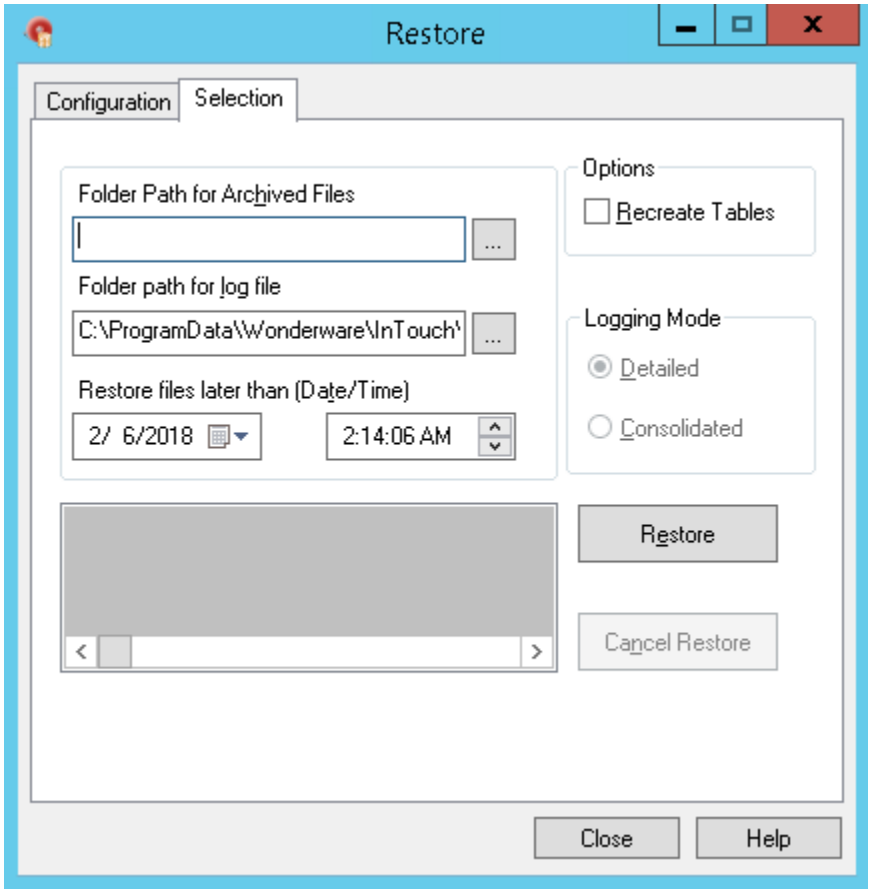
1. **Alarm DB Restore** ユーティリティを開きます。以下の手順を実行します。

   a. ［**ツール**］ビューで［**アプリケーション**］を展開します。

   b. ［**Alarm DB Restore**］をダブルクリックします。

2. ［**選択**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［**アーカイブしたファイルのフォルダ パス**］ボックスに、アーカイブされたファイルの場所への絶対パスを入力します（最大 255 文字の英数字）。または、参照ボタンを選択してアーカイブ ファイルが保存されているフォルダを検索し、選択します。
4. ［**次の日時以降のファイルを復元**］領域で、データベースへのレコードの復元を開始する日時を選択します。

   開始日時は、デフォルトで現在の日時に設定されています。
5. ［ログ ファイルのフォルダ パス］ボックスに、ログ ファイルを作成して保存する場所の絶対パスを入力します（最大 255 文字の英数字）。または、参照ボタンを選択して、フォルダを参照し、選択します。
6. ［**テーブルの再生**］チェック ボックスをオンにすると、指定したアラーム データベースのテーブルが再作成されます。
7. アーカイブ ファイルに含まれるアラーム レコードに対して選択した**ログ モード**のタイプに基づいて、以下を選択します。
   - **詳細** - 詳細モードでアラーム データベース テーブルを再作成します。
   - **統合** - 統合モードでアラーム データベース テーブルを再作成します。

   重要：テーブルを再作成すると、アラーム データベースに現在保存されているすべてのレコードが上書きされます。
8. ［**復元**］を選択します。

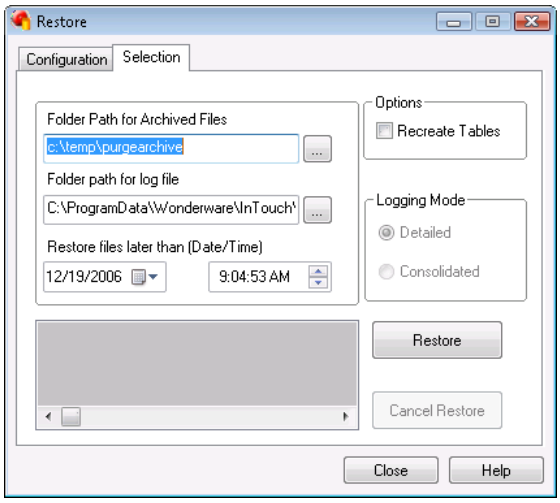
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データベース復元操作の開始

データベース接続を確立して、アーカイブされたファイル フォルダとタイム バッファを指定した後、アーカイブされたデータベース レコードを復元します。

アーカイブからデータベース レコードを復元する

1. **Alarm DB Restore** ユーティリティを開きます。以下の手順を実行します。
   a. ［ツール］ビューで［アプリケーション］を展開します。
   b. ［**Alarm DB Restore**］をダブルクリックします。
2. ［選択］タブを選択します。



3. ［復元］を選択します。復元が正常に行われたかどうか、およびデータベースに復元されたレコードの数を示すメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

付録 AF INTOUCH.ini ファイルからのアプリケーション設定のカスタマイズ

InTouch アプリケーションを初めて実行するとき、アプリケーション フォルダに INTOUCH.ini ファイルが作成されます。INTOUCH.ini ファイルが作成されるときに、個々の InTouch アプリケーションの動作特性を決定する一連のパラメータに値が割り当てられます。

WindowMaker または WindowViewer からアプリケーションの設定を続行すると、新しい INTOUCH.ini パラメータが作成されるか、または既存のパラメータが変更されます。たとえば、WindowMaker の［**履歴ログの設定**］ダイアログ ボックスからログを設定すると、INTOUCH.ini ファイルにログのパラメータが追加されます。

その他の設定パラメータは、INTOUCH.ini ファイルに手動で追加する必要があります。

アプリケーションのカスタマイズ後、INTOUCH.ini ファイルを別のアプリケーションのフォルダにコピーできます。このようにすると、カスタマイズに必要なすべての手順を繰り返さなくても、一貫したアプリケーションの動作特性を保つことができます。

カスタム INTOUCH.ini パラメータ

以下の表には、カスタム プロパティを InTouch アプリケーションに追加するために、INTOUCH.ini ファイルに手動で入力できる一連のパラメータが一覧表示されています。

INTOUCH.ini パラメータ	**説明**
16PenTrendDrawMode16PenTrendDrawMode	16 ペン トレンドで、データ値が平均化モードまたは最小 - 最大モードのどちらで表示されるのかを決定します。
ApplicationThumbnail	アプリケーション サムネイル ファイルの名前を設定します。
AllowPubAppEdit	アプリケーション フラグを設定し、パブリッシュ済みアプリケーションを編集できるようにします。値が 1 の場合、パブリッシュ済みの InTouch ファイルを編集できます。
CommentRetentiveCommentRetentive	［アラーム コメント］フィールドへのランタイムの変更が保存されるかどうかを決定します。
CompileAllWindows	CompileAllWindows = 1 に変更して WindowViewer を起動すると、すべてのウィンドウがコンパイルされて .wvw ファイルが作成されます。 CompileAllWindows = 0 の状態で WindowViewer を起動すると、ホーム ウィンドウのみがコンパイルされて .wvw ファイルが作成されます。その後ウィンドウを開くたびに、.wvw ファイルが作成されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

INTOUCH.ini パラメータ	説明
ForceLogCurrentValueForceLogCurrentValue	ForceLogging パラメータによって設定された間隔で、ログ済みタグの現在の値が履歴ログ ファイルに書き込まれるかどうかを決定します。
ForceLoggingForceLogging	現在の値にかかわらず、タグの値が履歴ログ ファイルに定期的に書き込まれるときの間隔の長さを設定します。
LoopTimeOut	InTouch スクリプトでの FOR-NEXT ループ処理のタイム アウト期間を設定します。
MarkAppReadOnlyNonRDS	このパラメータが非 RDS ノードで 1 に設定されている場合、そのノードは読み取り専用ノードとみなされ、InTouchView アプリケーションの読み取り専用ライセンスが使用されます。
MouseMustBeOnObjectForOnKeyUpMouseMustBeOnObjectForOnKeyUp	[離した時] アクションをトリガするためにマウスがオブジェクト上にある必要があるかどうかを決定します。デフォルトでは、この値は '1' です。
NoKeyboardResizeNoKeyboardResize	数値キーボードのサイズが WindowViewer の画面解像度用に変更されるかどうかを決定します。
OldRightMouseBehavior	WindowMaker で右マウス ボタンを有効にするかどうかを決定します。
PrintScreenWaitPrintScreenWait	WindowViewer から画面を印刷する前の待機時間を設定します。
PrintWindowWait	WindowViewer から InTouch ウィンドウを印刷する前の待機時間を設定します。
RemoteTagsLogEvents	InTouch アプリケーションが、リモート参照されたタグ アラームおよびイベントをログ記録するかどうかを決定します。
RemoteTagsNoIOEvents	InTouch アプリケーションが、リモート参照されたタグ アラームをログ記録するかどうかを決定します。
ScaleForResolutionScaleForResolution	異なる画面解像度を持つノードに変更する場合に、InTouch アプリケーションのウィンドウが自動的にサイズ変更されるかどうかを決定します。
ViewLicenseRetryCount	セットアップ時および使用可能なライセンスがないときに WindowViewer がバックグラウンドでライセンス取得を試行する回数を決定します。
WindowNameWithSpecialCharacters	このパラメータを 1 に設定すると、新しいウィンドウを作成する際にウィンドウ名に特殊文字を使用できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カスタム ログ プロパティの設定

タグ変数の値がどのように InTouch の履歴ログ ファイルに保存されるのかを指定する INTOUCH.ini ファイルに、一連のパラメータを追加できます。これらのパラメータに割り当てられられた値により、ログの頻度、およびリモート参照されたタグ変数の値がログ済みかどうかが決まります。

ログ記録の頻度の設定

次の 2 つの条件に基づいて、InTouch HMI によって履歴ログ ファイルにエントリが書き込まれます。

- タグ変数値がログ デッドバンド値より大きい工学単位値によって変更されると、InTouch HMI によってすぐにログ エントリが書き込まれます。
- InTouch HMI によって、すべてのログ済みのタグ変数の現在の値が固定間隔で書き込まれます。デフォルトの固定間隔は 60 分です。

周期を変更するには、次の 2 つのパラメータを INTOUCH.ini ファイルに追加します。

- ForceLoggingForceLogging

  ForceLogging は、ログ記録間隔の長さを分単位で指定します。ForceLogging の値は、5 ～ 120 分の間で設定できます。デフォルトは、ForceLogging=60 です。

- ForceLogCurrentValueForceLogCurrentValue

  ForceLogCurrentValue を使用すると、現在の値がログのデッドバンドの範囲以下であっても、InTouch HMI によってすべてのログ済みのタグ変数のログ エントリが書き込まれます。デフォルトは、ForceLogging=0 です。

次の例では、15 分の周期で、またはタグ変数の値が変更されたときに、現在のタグ変数の値が履歴ログ ファイルに書き込まれます。

```
ForceLogging=15
ForceLogCurrentValue=1
```

リモート参照されたタグ変数のログ記録

デフォルトでは、リモート参照されたタグ変数はイベント ログ ファイルにログ記録されません。リモート参照されたタグ変数をログ記録するには、イベント ログを有効にし、RemoteTagsLogEvents パラメータを INTOUCH.ini ファイルに追加する必要があります。

```
RemoteTagsLogEvents=1
```

I/O タグ変数がログ記録されないようにするには、RemoteTagsNoIOEvents パラメータを INTOUCH.ini ファイルに追加します。RemoteTagsLogEvents パラメータが 1 に設定されている場合のみ、RemoteTagsNoIOEvents パラメータが適用されます。

```
RemoteTagsNoIOEvents=1
```

WindowMaker ショートカット メニューの無効化

デフォルトでは、選択したオブジェクト上でマウスを右クリックしたときに、WindowMaker にショートカット メニューが表示されます。以前のバージョンの InTouch HMI と同じマウスの動作を使用してアプリケーションを開発する場合、INTOUCH.ini ファイルの oldrightmousebehavior パラメータを 1 に設定することにより、WindowMaker の右クリックの動作をオフにできます。

```
oldrightmousebehavior=1
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カスタム WindowViewer プロパティの設定

次を行うために、WindowViewer の動作を設定する一連の INTOUCH.ini ファイル パラメータを追加できます。

- スクリプト ループの処理
- 異なる画面解像度用に InTouch ウィンドウを調整
- ウィンドウまたは画面を印刷するための待ち時間の設定
- アラーム コメントへのランタイムでの変更のログ記録
- 16 ペン トレンドの描画モードの設定
- 数値キーパッドのサイズ変更
- アナログおよび文字列のデータ入力リンクの入力フィールドのサイズ変更

スクリプト ループ タイマーの追加

デフォルトでは、InTouch スクリプト内の FOR-NEXT ループは 5 秒以内に完了する必要があります。FOR-NEXT ループ処理がタイムアウト制限までに終了していない場合、WindowViewer でスクリプトが自動的に停止します。このタイムアウト制限により、スクリプト エラーによる無限ループを防ぐことができます。

FOR-NEXT ループ コード処理が 5 秒のタイムアウト制限を超えるスクリプトを記述する必要がある場合もあります。LoopTimeout パラメータを INTOUCH.ini ファイルに追加することにより、タイムアウト制限の長さを変更できます。

この例では、ループ処理が最大 20 秒間続行します。

```
LoopTimeout=20
```

異なる画面解像度への InTouch ウィンドウの調整

異なる画面解像度で実行されている他のノードにアプリケーションを移行する場合、INTOUCH.ini ファイルにパラメータを追加して、InTouch ウィンドウの現在の解像度を保持できます。

ScaleForResolution パラメータ値により、WindowViewer が実行されているコンピュータで表示解像度が変更された後に、アプリケーション ウィンドウが WindowMaker によって自動的にスケールされるかどうかが決まります。ScaleForResolution パラメータは、WindowViewer ダイアログ ボックスの解像度には影響しません。ScaleForResolution パラメータが 1 に設定されている場合、解像度の変換が有効になります。.

```
ScaleForResolution=1
```

印刷の待ち時間の長さの設定

印刷するウィンドウまたは画面を選択する場合、選択したウィンドウまたは画面が WindowViewer によってメモリにロードされます。次に WindowViewer は、ウィンドウまたは画面に表示されたすべての DDE 変数が更新されるまで 10 秒間待機します。待ち時間が終了した後、WindowViewer によってウィンドウまたは画面がプリンタに送信されます。

WindowViewer の印刷の待ち時間は、PrintWindowWait パラメータまたは PrintScreenWait パラメータを INTOUCH.ini ファイルに追加することにより変更できます。両者のパラメータの待機時間は、ミリ秒で表されます。

```
PrintWindowWait=15000
PrintScreenWait=20000
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

アラーム コメントのログ記録

アラームを確認するとき、オペレータはコメントを追加できます。タグ変数データベースの［アラーム コメント］フィールドにランタイムでの変更を書き込むには、現在のアプリケーションの INTOUCH.ini ファイルに次の行を追加します。

```
CommentRetentive=1
```

16 ペン トレンドの描画モードの設定

16PenTrendDrawMode パラメータの値に基づいて、16 ペン トレンドの線描画モードを選択できます。

- 平均化モード：`16PenTrendDrawMode=0`

  16 ペン トレンドの時間枠およびバッファ サイズにより、トレンド上の各ピクセルは数秒相当のデータを表しています。各周期には、異なる値を持つ複数のサンプルが含まれている場合もあります。結果として、トレンドのデータ ポイントは、その周期内に監視された最大値と最小値の間の垂直線として表示されます。

  最小から最大までの垂直線が描画された後、計算された周期の平均値にトレンド ペンが移動します。平均値からトレンドの次の周期に線を描画すると、次の周期が開始します。垂直の最小から最大までの線が描画され、その周期に対して計算された平均値でペンが止まります。このプロセスは各サンプリング間隔に対して繰り返されます。

  INTOUCH.ini ファイルで 16PenTrendDrawMode が指定されていない場合、平均化は 16 ペン トレンドのデフォルトの描画モードです。

- 最小 - 最大モード：`16PenTrendDrawMode=1`

  最小 - 最大の描画モードでは、各データの収集期間のエンドポイントを直接繋ぐことによってトレンドの線が描画されます。

数値キーパッドのサイズ変更

InTouch アプリケーションの数値キーパッドのサイズを変更できるかどうを決定する INTOUCH.ini ファイルにパラメータを追加できます。キーパッドのサイズをより高い画面解像度（1280 x 1024）にすると、キーパッドに表示されるテキストが読みやすくなります。しかし、アプリケーションで画面スペースが制限されていると、キーパッドのサイズに現実的な制限が設定されている場合があります。

NoKeyboardResize パラメータを INTOUCH.ini ファイルに追加できます。デフォルトでは、パラメータは含まれていません。デフォルト値は次のとおりです。

```
NoKeyboardResize=0
```

デフォルト値を使用すると、画面解像度に応じて数値キーパッドのサイズを変更できます。

パラメータに割り当てることができる別の値は次のとおりです。

```
NoKeyboardResize=1
```

この場合、画面解像度に基づいてキーパッドのサイズは変更されず、数値キーパッドのサイズは固定されたままです。

アナログおよび文字列のデータ入力リンクの入力フィールドのサイズ変更

Resizable InputLink パラメータを INTOUCH.ini ファイルに追加すると、アナログまたは文字列のデータ入力リンクの入力ボックスをマウスでサイズ変更できます。Resizable InputLink パラメータは、ゼロ以外の値に設定する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

入力フィールドを初めてサイズ変更した後、WindowViewer によって INTOUCH.ini ファイルに Resizable InputLink Width パラメータおよび Resizable InputLink Height パラメータが追加されます。これらのパラメータにより、入力ボックスの幅および高さがピクセルで指定されます。

例:

```
Resizable InputLink = 1
Resizable InputLink Width=300
Resizable InputLink Height=50
```

また、これらのパラメータに割り当てられた値を手動で変更して INTOUCH.ini ファイルを編集できます。

反応しなくなったアプリケーション ボタンまたは表示値の問題の解決

パラメータを InTouch.ini ファイルに追加して、InTouch アプリケーション ボタンが押されたままになる問題や、表示値が変化しなくなる問題を解決できます。この問題が発生すると、マウスを繰り返しクリックしてもボタンと値が反応しなくなります。

この問題は、OnKeyDown スクリプトを含むグラフィック要素によってウィンドウが非表示になるため、OnKeyUp スクリプトが実行しなくなることが原因で発生することがあります。また、反応しなくなったボタンの問題は、ビットを設定する OnKeyDown とビットをクリアする OnKeyUp の 2 つのスクリプトがある場合に発生することがあります。この場合、オペレータがボタンをクリックしたとき、マウス ボタンを離す前にボタンを含むウィンドウが閉じます。

これらの問題を解決するには、次の処理を行います。

- `UseLegacyOnKeyUp=1` パラメータを Intouch.ini ファイルに挿入します。
- ［**WindowViewer のプロパティ**］ダイアログ ボックスの［**メモリ内ウィンドウのキャッシュを使用する**］チェック ボックスをオンにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Diagnose

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 33 QuickScript のトラブルシューティング

Log Viewer を使用してタグ変数のランタイム値を表示することによって、QuickScript をトラブルシューティングできます。

Log Viewer へのメッセージのログ記録

Log Viewer は、QuickScript のデバッグに役立ちます。Operations Control Log Viewer は InTouch HMI をインストールする際にインストールされ、System Management Console にあります。

QuickScript をデバッグする 1 つの方法を以下に示します。

1. QuickScript でチェック ポイントを設定して、Log Viewer に値をログ記録します。
2. Operations Control Log Viewer を開いて、値を表示します。

QuickScripts をデバッグする別の方法として、タグ値を Log Viewer に記録するキー スクリプトを作成する方法があります。

QuickScript でチェック ポイントを設定する

1. エラーの原因と考えられる QuickScript を開きます。
2. チェック ポイントを設定する行を検索します。
3. その行の後に、以下のいずれかのコードの一部を挿入します。
   - LogMessage(messagetag);
     このスクリプトでは、messagetag は、値をログ記録するメッセージ型タグの名前です。
   - LogMessage(StringFromIntg(inttag,10));
     このスクリプトでは、inttag は、値をログ記録する整数型タグの名前です。
   - LogMessage(Text(realtag,"#.#######"));
     このスクリプトでは、realtag は、値をログ記録する実数型タグの名前です。
   - LogMessage(DText(disctag,"TRUE","FALSE"));
     このスクリプトでは、disctag は、値をログ記録する論理型タグの名前です。
   - 識別子やタグ名など、チェックポイントで LogViewer に詳細情報をログ記録します。例:
     ```
     LogMessage("DEBUG tag:"+ind.name+" value:"+Text(ind,"#.####"));
     ```
     このスクリプトでは、ind は、アナログ型間接タグです。

LogMessage() 関数 LogMessage()かんすう

ユーザー定義のメッセージを Log Viewer に書き込みます。

カテゴリ

その他

構文

```
LogMessage("Message_Tag");
```

パラメータ

messagetag

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Log Viewer にログ記録する文字列です。実際の文字列またはメッセージ型タグです。

備考

これは、InTouch スクリプトのトラブルシューティングに非常に役立つ関数です。LogMessage() 関数をスクリプトに戦略的に配置することで、QuickScript の実行順序やスクリプトのパフォーマンスを判断し、タグが変更される前と QuickScript によって影響が与えられた後のタグの値を識別できます。Log Viewer に記録される各メッセージには、正確な日付および時刻が示されます。

重要：パーセント (%) 文字は、スクリプトのデバッグ時に Log Viewer に表示される診断メッセージをフォーマットします。パーセント文字がログ文字列または関数パラメータ内にある場合、WindowViewer が応答しなくなることがあります。パーセント文字によって生じるエラーを排除するには、2 つのパーセント文字 (%%) を使用します。

例

```
LogMessage("レポート スクリプトが実行中です");
```

上記のステートメントは、以下の情報を Log Viewer に出力します。

94/01/14 15:21:14 WWSCRIPT Message: レポート スクリプトが実行中です。

```
LogMessage("MyTag の値：" + Text(MyTag, "#"));
MyTag = MyTag + 10;
LogMessage("MyTag の値：" + Text(MyTag, "#"));
```

Log Viewer メッセージの表示

Log Viewer は InTouch HMI をインストールする際にインストールされ、Operations Control Management Console にあります。

Log Viewer にログ記録された値を表示する

1. ［スタート］を選択し、［プログラム］をポイントしてから［**AVEVA**］をポイントし、［**System Platform Management Console**］を選択します。System Management Console が表示されます。
2. 左ペインで、［**Log Viewer**］、［**Default Group**］の順に展開して、［**Local**］を選択します。Log Viewer メッセージが詳細ペインに表示されます。
3. LogMessage() 関数からログ記録された値を検索します。

注意：リモート InTouch HMI ノードでスクリプトをデバッグしている場合、ノード名を Log Viewer のノード グループに追加して、そのノードの Log Viewer メッセージを表示する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 34 エラー メッセージの理解

Tag Viewer では、要求の実行に失敗するとエラー メッセージが表示されます。このエラー メッセージは、エラーの説明を示すダイアログ ボックスに表示されます。エラー メッセージの詳細については、以下を参照してください。

メイン ウィンドウ

タグ リファレンスの追加

タグ値の変更

監視ウィンドウの名前の変更

メイン ウィンドウ

機能	説明	理由
クイック検索	検索されたタグ名／アラーム グループが見つかりませんでした。	システムは一致するタグ名/アラーム グループを見つけることができませんでした。
監視ウィンドウ	監視ウィンドウに追加できるリファレンスの最大数は 2000 です。	監視ウィンドウに 2000 個以上のアイテムを追加しようとしました。
	監視ウィンドウの最大数は 50 です。	50 個以上の監視ウィンドウを追加しようとしました。
	<File1> は有効な監視リスト ファイルではありません。	監視ウィンドウを開く際に、無効なファイル名を指定しました。
	<File1> がすでに存在します。置き換えますか?	監視ウィンドウを保存する際に、既存のファイル名を指定しました。
	現在の InTouch アプリケーションには以下のリファレンスは存在しません。またこれらのリファレンスは監視ウィンドウには追加されていませんでした。 <リファレンス リスト>	監視ウィンドウに読み込まれたリファレンスには、現在の InTouch アプリケーションで無効なものや存在しないものがあります。

タグ リファレンスの追加

機能	説明	理由
タグ リファレンスを追加	<abcd> に <f> が見つかりません	[変更] ボックスに文字列 <f> を入力しましたが、この文字列はタグ リファレンスの文字列 <abcd> にはありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

機能	説明	理由
	<始点> の値は <終点> の値以下である必要があります	［始点］ボックスの値よりも小さい値を、［終点］ボックスに入力しました。
	<始点> に数値を入力してください または <終点> に数値を入力してください	［変更］ボックスに文字列を入力しましたが、［始点］ボックスまたは［終点］ボックスのどちらかが空白です。
	現在の InTouch アプリケーションには以下のリファレンスは存在しません。またこれらのリファレンスは監視ウィンドウには追加されていませんでした。 <リファレンス リスト>	無効なリファレンスを 1 つ入力したか、または入力した複数のリファレンスのうちのいくつかが無効です。
	監視ウィンドウに追加できるリファレンスの最大数は 2000 です。	監視ウィンドウに 2000 個以上のアイテムを追加しようとしました。

タグ値の変更

機能	説明	理由
整数値を変更	<1a> は有効な数値ではありません。	［値］ボックスに無効なエントリである <1a> を入力して、［適用］を選択しました。
	入力値は、整数型のデータ タイプの有効な範囲外です。	許容範囲外の数値を入力しました（有効な数値は -2147483648 ～ 2147483647）。
実数値を変更	入力値は、実数型データ タイプの有効な範囲外です。	許容範囲外の数値を入力しました（有効な数値は -3.402823466e38 ～ 3.402823466e38）。
［タグ リファレンス］ダイアログ ボックス	このアプリケーションにタグ リファレンス <InvalidReference> は存在しません。	［タグ リファレンス］ダイアログ ボックスに <InvalidReference> を入力して、［OK］を選択しました。
	設定するはデータは、誤ったデータ タイプのものです。	リファレンスは読み取り専用のタグですが、値を変更しようとしました。 または タグ リファレンスの選択したタイプに対して無効な値を入力しました。 たとえば、整数型タグに対して文字列値を入力した場合などです。

監視ウィンドウの名前の変更

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

機能	説明	理由
監視ウィンドウ	<Watch List 1> という名前の監視ウィンドウは、すでに存在します。	監視ウィンドウの名前を変更する際に、既存の名前を入力しました。
	「ReadonlyReference」は読み取り専用であり、変更できません。	監視ウィンドウに読み取り専用タグを追加して、そのタグをダブルクリックして値を変更しようとしました。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 35 InTouch Web Client の管理

このセクションには、以下の内容が含まれます。

InTouch Web Client でのエラー処理

ブラウザとモバイルのサポート

Web ブラウザでのグラフィックの表示に関するトラブルシューティング

無効な証明書エラー

Chrome Web ブラウザへの拡張機能の追加

InTouch Web Client でのエラー処理

InTouch Web Client は、予期されるエラーを表示する一貫したページを提供します。エラー ページは、以下のシナリオで表示されます。

- **アプリケーションがない** - WindowViewer で最後に実行したアプリケーションの必要なサポート ファイルが Web Client で検出されませんでした。必要なアプリケーションを WindowMaker で開きます。
- **権限が拒否された** - 現在ログインしているユーザーに Web ページを表示するための適切な権限がありません。ユーザーを aaInTouchUsers または aaInTouchRWUsers ユーザー グループに追加します。
- **使用可能なライセンスがない** - 現在の Web セッションの有効なライセンス ステータスが取得されませんでした。有効な Web Client ライセンスが使用可能であることを確認します。
- **高速切り替えモード** - Web Client アプリケーションは高速切り替えモードですが、ユーザーが http://localhost/InTouch 以外の URL を使用して Web Client にアクセスしようとしています。

ブラウザとモバイルのサポート

ブラウザのサポート:

注意 : InTouch Web Client が正しく機能するためには、常に最新バージョンのブラウザーをインストールするか、使用できる最新バージョンにブラウザーをアップグレードすることをお勧めします。

InTouch Web クライアントは、以下の Web ブラウザでテストされています。

- Google Chrome
  - デスクトップ用 Chrome バージョン 64.0.3282.186 以降
  - Android スマートフォンおよびタブレット用 Chrome バージョン 64.0.3282.137 以降
- Microsoft Internet Explorer 11.0.9600.18861 以降
- Microsoft Edge
  - Microsoft Edge（Windows 10）バージョン 41.16299.15.0 以降
  - Microsoft Edge（Surface）バージョン 41.16299.248.0 以降
- Apple Safari for iOS バージョン 11.2.6

サポートされていないブラウザの場合はエラー メッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

モバイル デバイスのサポート

InTouch Web Client は、以下のモバイル デバイスおよびそのデフォルト ブラウザをサポートします。

- Apple iPhone
- Apple iPad
- Android スマートフォンおよびタブレット
- Microsoft Surface

Web ブラウザでのグラフィックの表示に関するトラブルシューティング

InTouch Web Client が表示されない場合、以下の設定を確認してください。

- *InTouch Web サービス*がノード上で実行している。
- InTouch Web ページへのアクセスに使用するユーザー アカウントがいずれかのユーザー グループに含まれている。
- 同じマシン上で IIS が実行している場合、InTouch Web Server と競合する可能性があります。
  - IIS が実行していて使用されていない場合、IIS を停止してサービスを無効にします。IIS を実行する必要がある場合は、競合を防ぐためにポート 80 以外のポートを使用するよう設定します。

HTTP Error 404.0 - Not Found エラーが表示された場合：

- 選択したノードで Web Client が有効化されていることを確認してください。管理者に連絡してください。

グラフィック プリミティブまたはアニメーションが表示されない場合、または予期した通りに動作しない場合は、以下に従ってください。

- サポートされていない機能のリストをチェックして、Web Client でサポートされているかどうかを確認してください。
- http://localhost/InTouch から Web Client をローカルで実行し、シンボルを参照します。次に、互換性アイコンを探して、グラフィック内のサポートされていないアイテムのリストを表示します。
- サポートされていないグラフィックで使用されているプリミティブまたはアニメーションは削除され、メッセージがロガーに記録されます。記録されたメッセージには、生成された CSV レポート ファイルのパスが含まれます。詳細については、ロガー メッセージにアクセスしてレポートを参照してください。

選択したグラフィックが表示されていてもデータが表示されない場合やアニメーションが更新されない場合は、以下に従ってください。

- グラフィックが InTouch タグをポイントする場合は、InTouch WindowViewer が実行していることを確認してください。
- グラフィックがアプリケーション オブジェクト参照をポイントする場合は、Application Server が実行していることを確認してください。
- InTouch iData サービスが実行していることを確認してください。

無効な証明書エラー

エラー：証明書エラー: このサイトへの接続は安全ではありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch Web Client にアクセスするときに有効な証明書がクライアントで使用できない場合、このエラーがブラウザに表示されます。

解決方法

1. サーバー マシンに接続し、System Platform **コンフィグレータ**を起動します。
2. System Management Server ページで［**詳細**］を選択します。

   ［**証明書**］ダイアログ ボックスが表示されます。
3. ［**詳細**］タブで［**ファイルにコピー**］を選択します。
4. ［**証明書のエクスポート ウィザード**］の指示に従ってファイルを保存します。
5. 証明書ファイルをクライアント マシンにコピーします。
6. クライアント マシンで、**Microsoft 管理コンソール**（mmc）を起動します。
7. 証明書をインポートします。
8. 証明書をインポートした後にブラウザを起動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Manage

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 36 AVEVA Operations Control 接続エクスペリエンス

Operations Control 接続エクスペリエンスについて

Operations Control 接続エクスペリエンスオプションでは、CONNECT クラウド機能を介して、特定のノードにおけるすべての Operations Control 製品間でのシングルサインオン（SSO）エクスペリエンスが有効になります。Operations Control 接続エクスペリエンスには、有効な Operations Control サブスクリプションおよびユーザー管理機能のある CONNECT アカウントが必要です。

- 統合ユーザー管理：ノード上で Operations Control 接続エクスペリエンスを有効にすると、そのノードのすべての対象 Operations Control 製品の動作が変更され、ノード上で最初の製品を起動するときに CONNECT でのログイン認証が必要になります。以降にノード上で起動する製品では CONNECT を使用した認証が行われ、認証済みユーザーはシングルサインオン（SSO）が可能になります。CONNECT ベースの承認は、Operations Control 接続エクスペリエンスで使用可能な唯一のセキュリティモードです。
- ノード間の互換性：Operations Control 接続エクスペリエンスは、システムのすべてのノードで有効にする必要があります。Operations Control 接続エクスペリエンスが有効化されていないノードで以前に構築されたアプリケーションは、Operations Control 接続エクスペリエンス環境で機能するように再設定する必要があります。

Operations Control 接続エクスペリエンスはいつでも無効にできますが、Operations Control 接続エクスペリエンスはシステムのすべてのノードで無効にする必要があります。Operations Control 接続エクスペリエンスの下で構築されたアプリケーションは、認証方法と製品ライセンスなど、Operations Control 接続エクスペリエンスが有効でない環境で機能するように再設定する必要があります。

Operations Control 接続エクスペリエンスの設定

AVEVA Operations Control の下で InTouch およびその他の System Platform 製品を（製品サブスクリプションのみとして、または Operations Control 接続エクスペリエンスを使用して）運用するには、インストールされたライセンス済み製品を設定する必要があります。この場合でも、ヘッドレス サーバーが適切に機能するにはライセンス サーバーを設定する必要があります。

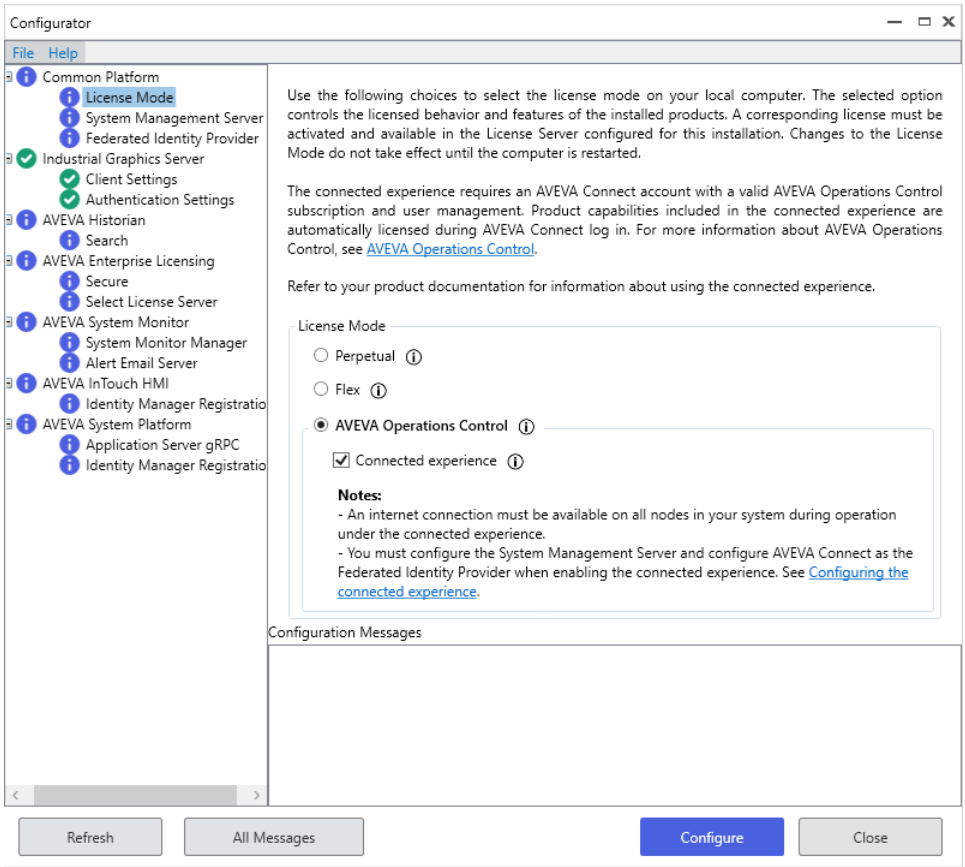
詳細については、以下を参照してください。

- コンフィグレータ ヘルプ
- AVEVA Identity Manager ヘルプ

Operations Control 接続エクスペリエンスを設定するには

1. ［スタート］>［**AVEVA**］>［**Configurator**］からコンフィグレータを開きます。
2. 左側のペインで［**ライセンス モード**］を選択してライセンス モードを選択します。
   a. 2 つの AVEVA Operations Control パッケージ（Edge および Supervisory）の少なくとも 1 つをサブスクライブするには、［**AVEVA Operations Control**］ラジオ ボタンを選択します。これには、定義済みのユーザーのセットに対する製品パッケージのすべての製品の無制限の使用が含まれます。
   b. CONNECT クラウド機能（製品で既定で使用可能）を備えたノード上ですべての Operations Control 製品でシングルサインオン（SSO）エクスペリエンスを有効にするには、**[接続エクスペリエンス]**チェックボックスをオンにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. System Management Server を設定するには、左側のペインの［**共通プラットフォーム**］で［**System Management Server**］を選択します。

注：System Management Server のユーザー資格情報の入力が求められた場合、DomainName\UserName の形式でユーザー名を入力します。ドメイン管理特権があってもローカル マシンの管理者でない場合は、ユーザー資格情報が要求されることがあります。System Management Server を設定するには、Administrators または aaAdministrators OS グループのメンバーである必要があります。

---

a. 既存の System Management Server に接続するには、［**既存の System Management Server に接続**］を選択します。使用可能な System Management Server のリストから目的の System Management Server を選択します。リストには、ネットワーク上で System Management Server として機能するように設定されたすべてのマシンが表示されます。ほとんどの場合、1 つの System Management Server だけが表示されます。

b. マシンを冗長 SSO（RSSO）サーバーとして設定するには、［**既存の System Management Server に接続**］で［**このマシンを Redundant SSO サーバーとして設定**］を選択します。

または

c. 現在のマシンを System Management Server として設定するには、［**このマシンを System Management Server にする**］オプションを選択します。

d. ［**設定**］をクリックします。

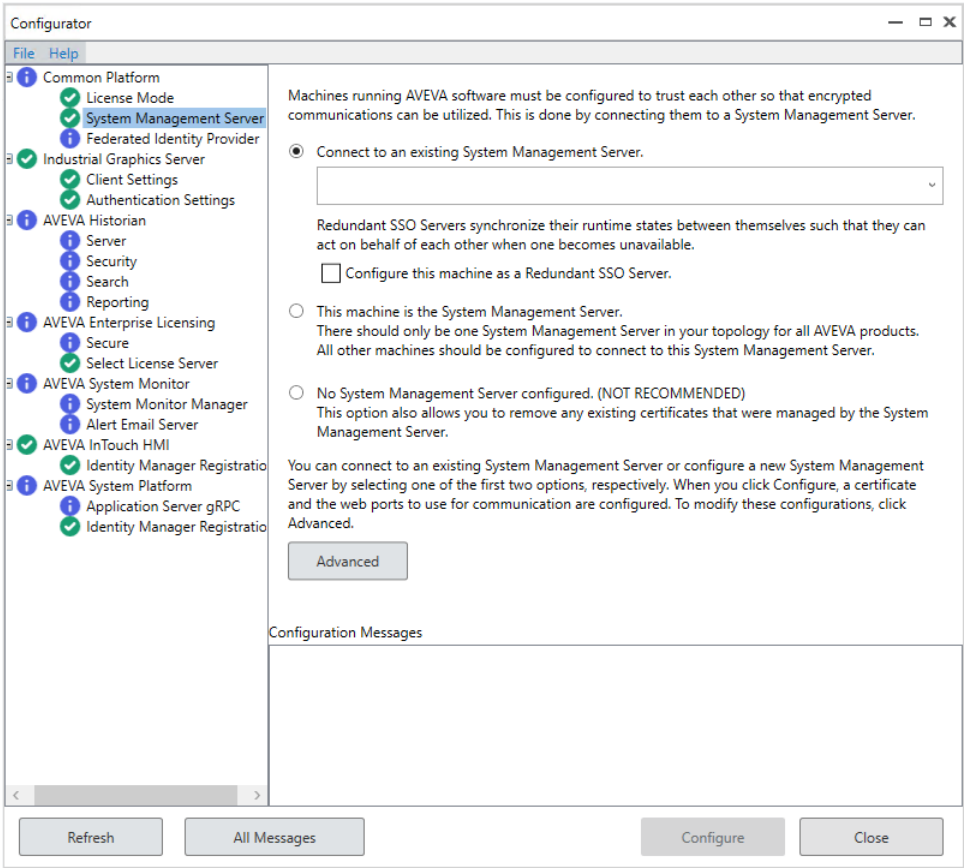
注：

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- Operations Control 接続エクスペリエンスでは［**System Management Server は設定されていない**］オプションはサポートされません。Operations Control 接続エクスペリエンスを使用するには System Management Server を設定する必要があります。

- ネットワークで System Management Server として識別および設定できるのは 1 つのマシンだけです。AVEVA 製品を実行するマシンは、その System Management Server に接続して信頼関係を確立し、暗号化された通信を構成できます。

- 冗長 SSO を設定しても別の System Management Server は設定されません。



4. フェデレーション ID プロバイダを設定するには、左側のペインの［**共通プラットフォーム**］で［**フェデレーション ID プロバイダ**］を選択します。

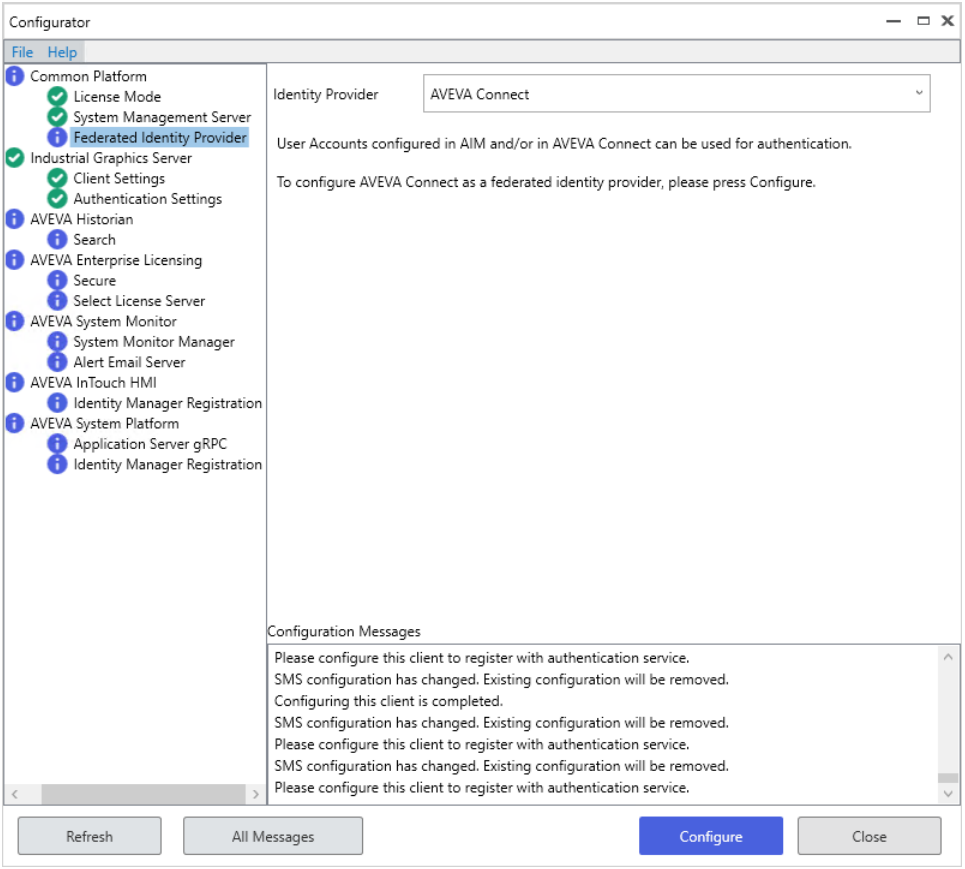
   System Management Server（SMS）および冗長 SSO（RSSO）マシンで使用可能な Platform Common Services（PCS）Framework の Identity Manager コンポーネントは、CONNECT との「フェデレーション」ログイン用に設定できます。したがって、ユーザーが CONNECT アカウントに登録されている E メールアドレスを Identity Manager のログインフォームに入力すると、ログインするために CONNECT にリダイレクトされます。現在、Operations Control 接続エクスペリエンスでは、CONNECT とのフェデレーションだけがサポートされています。Azure AD はサポートされていません。

   注：

   - フェデレーション ID 接続の場合、有効な CONNECT アカウントが必要です。また、ユーザーはそのアカウントの管理者である必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

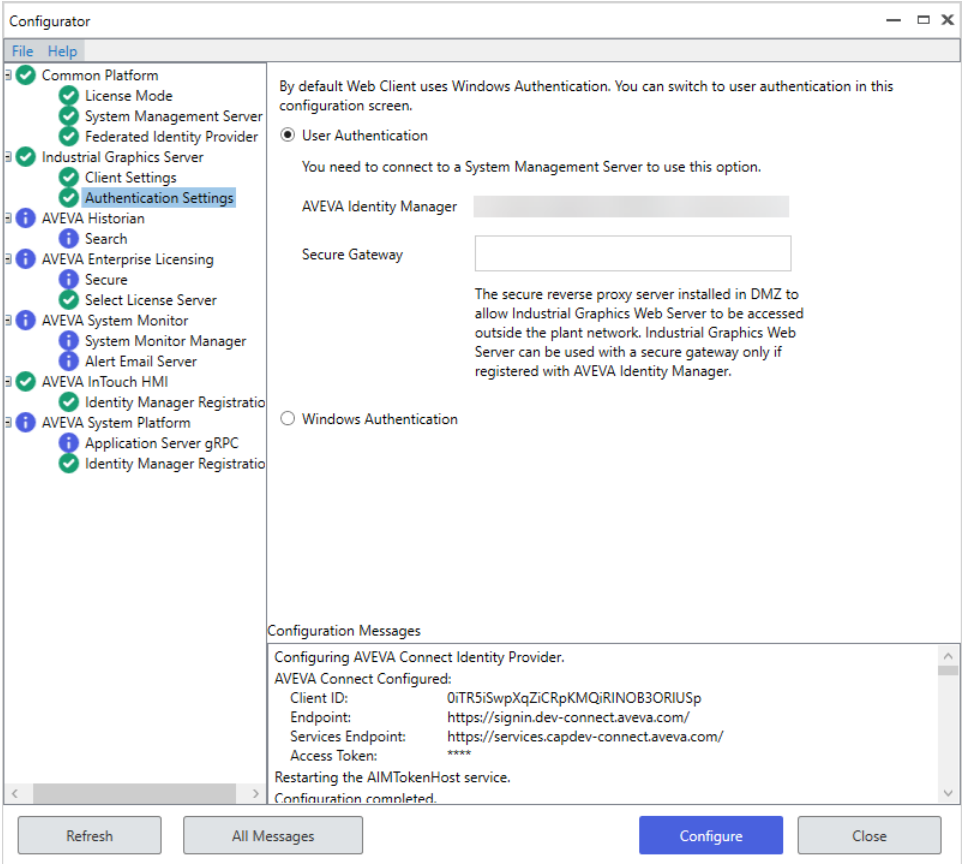
- この手順は、現在のマシンが System Management Server として設定されているノードにのみ適用されます。ノードが既存の System Management Server に接続する場合、フェデレーション ID プロバイダを設定する必要はありません。



複数の CONNECT アカウントがある場合、認証後、ユーザーが属する複数の CONNECT アカウントの名前の一覧を含むアカウント選択ダイアログが表示されます。フェデレーションを設定するアカウントを選択すると、それ以降は、マシンとすべての Operations Control 製品は、その選択したアカウントのサブスクリプションに対して認証または検証されます。CONNECT アカウントが 1 つだけの場合、アカウント選択ダイアログは表示されません。

5. ユーザー認証を使用するように InTouch Web Client を設定するには、左側のペインの［**産業用グラフィック サーバー**］>［**認証設定**］から［**ユーザー認証**］を選択し、［**設定**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

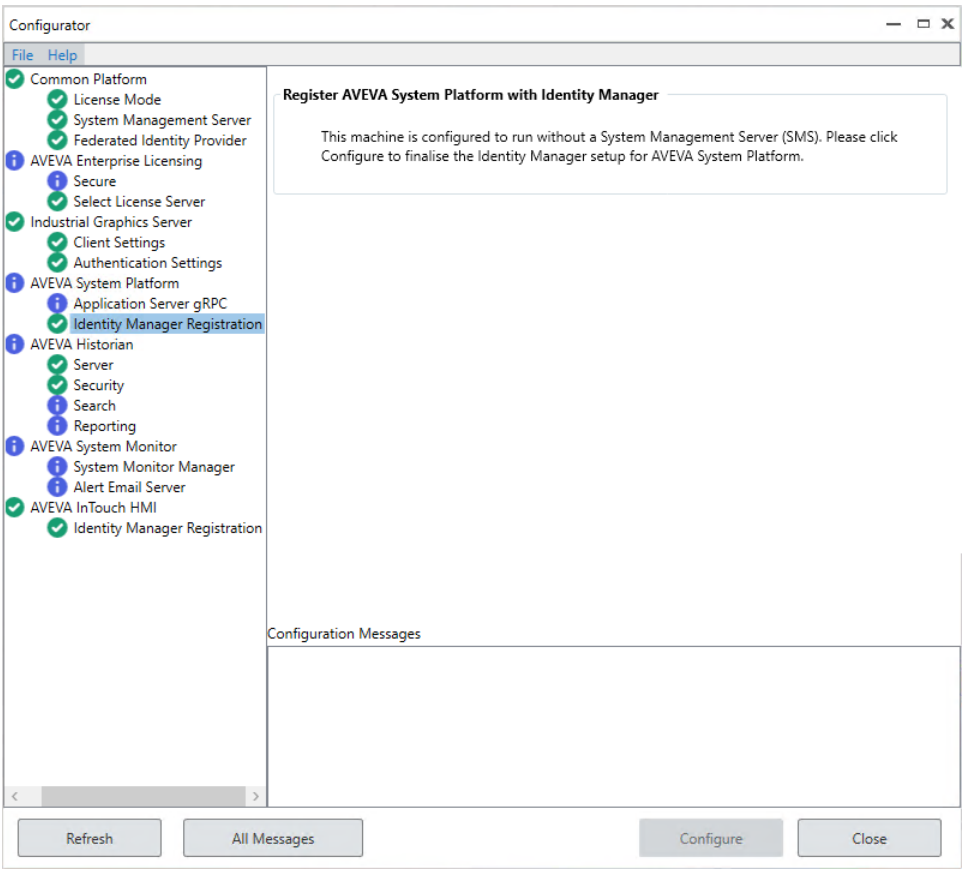


6. System Platform を AVEVA Identity Manager に登録するには、左側のペインの［**AVEVA System Platform**］で［**Identity Manager 登録**］を選択します。

   Identity Manager への AVEVA System Platform の登録を行う前に、必ず System Management Server の設定を完了してください。

   ［設定］をクリックして Identity Manager に登録します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



7. InTouch with AVEVA Identity Manager を登録するには、左側のペインの［**AVEVA InTouch HMI**］で［**Identity Manager 登録**］を選択します。

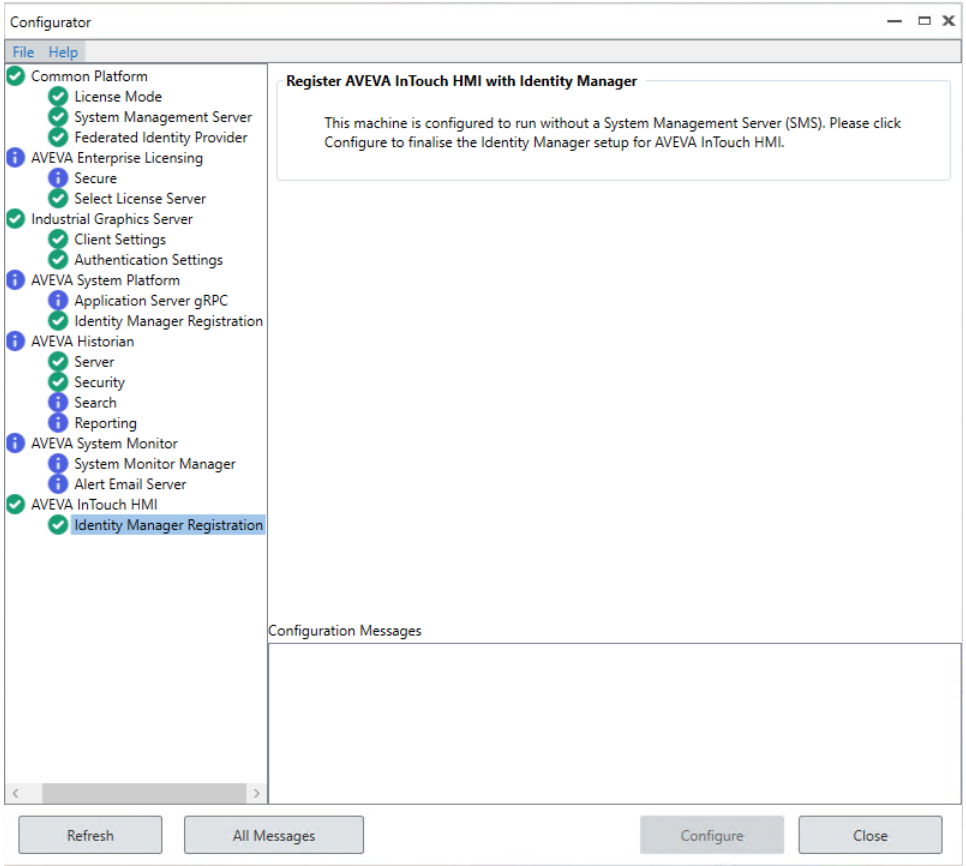
   Identity Manager への InTouch HMI の登録を行う前に、必ず前の手順（下記参照）を完了してください。

   - ライセンス モードとして AVEVA Operations Control 接続エクスペリエンスを有効化します。
   - System Management Server を設定します。
   - フェデレーション ID プロバイダで CONNECT を選択します。

   ［設定］をクリックして Identity Manager に登録します。

   注：System Management Server を設定していない（非推奨）場合でも InTouch は設定できますが、Operations Control 接続エクスペリエンスは使用できません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



認証とエンタイトルメント

認証、エンタイトルメント、承認は、組織で使用する AVEVA 製品へのユーザーアクセスを制御するためのセキュリティエレメントです。

- *認証*とは、一意の資格情報のセット（ユーザー名とパスワード）を使用してユーザーの本人確認を行うことです。
- *エンタイトルメント*とは、組織がサブスクリプションを持っており、認証されたユーザーがアクセスを許可されている製品です。ユーザーが有効な Operations Control サブスクリプションを持っていない場合、ユーザーは認証されていても特定製品の使用を許可されない場合があります。
- *承認*とは、認証されたユーザーに関連付けられている権限のレベルのことです。通常、これはユーザーをユーザーグループに割り当てるか、フェデレーション ID 管理を介して行われます。

  フェデレーション ID は*フェデレーション ID プロバイダ*（FIDP）を介して管理され、AVEVA 製品のポートフォリオ全体のユーザーに対する一元的なユーザー管理とシングルサインオンを可能にします。これはコンフィグレータを通じて有効になります。通常はインストール時に完了しますが、いつでも再設定できます。

選択したライセンスモードは、System Platform 環境における認証、エンタイトルメント、承認の処理において重要な役割を果たします。詳細については、「ライセンスモード」を参照してください。

Operations Control 接続エクスペリエンスが有効になっている場合、ユーザーが IDE、Historian、OMI ViewApp などの System Platform コンポーネントにログインするたびに、ログインが CONNECT 監査ログにイベントとして保存されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データ ログの表示

監査ログにより、組織と AVEVA 両方から承認されたユーザーが、アカウントで発生したイベントまたは操作のログを CONNECT ポータルに表示できます。ログには、選択した時間範囲に組織のアカウントで行われたすべての操作が表示されます。監査ログを表示するには、アカウント管理者、または Report Viewer の役割を持つユーザーである必要があります。

注 : 監査ログには、アカウントで監査機能が使用可能になった時点以降のデータが記録されます。

監査ログを表示するには、AVEVA Connect の Web サイトに移動します。サイト ナビゲーション メニューで［監査］を選択します。［監査ログ］ページが表示されます。

監査ログには以下の情報が表示されます。

- 操作が実行されたときのタイムスタンプ（日付と時刻）。
- ユーザー名、マシン名、アプリケーション名（例 : AVEVA OMI や System Platform IDE）など、操作の詳細。ログに記録される情報には、どの認証されたユーザーがどの製品にアクセスしたかに加え、そのユーザーの使用可能なエンタイトルメントアクセスが含まれます。
- 監査ログデータは CONNECT の**エンタイトルメントチェック**機能を使用してフィルタリングできます。この機能は、接続エクスペリエンスを介して製品の使用状況に関連するすべてのデータが表示します。

注 : データをユーザーインターフェイスに表示する際はローカル時間が使用されます。ただし、操作の詳細およびデータがエクスポートされる際は UTC 時間が表示されます。

監査ログの詳細については、CONNECT のヘルプを参照してください。

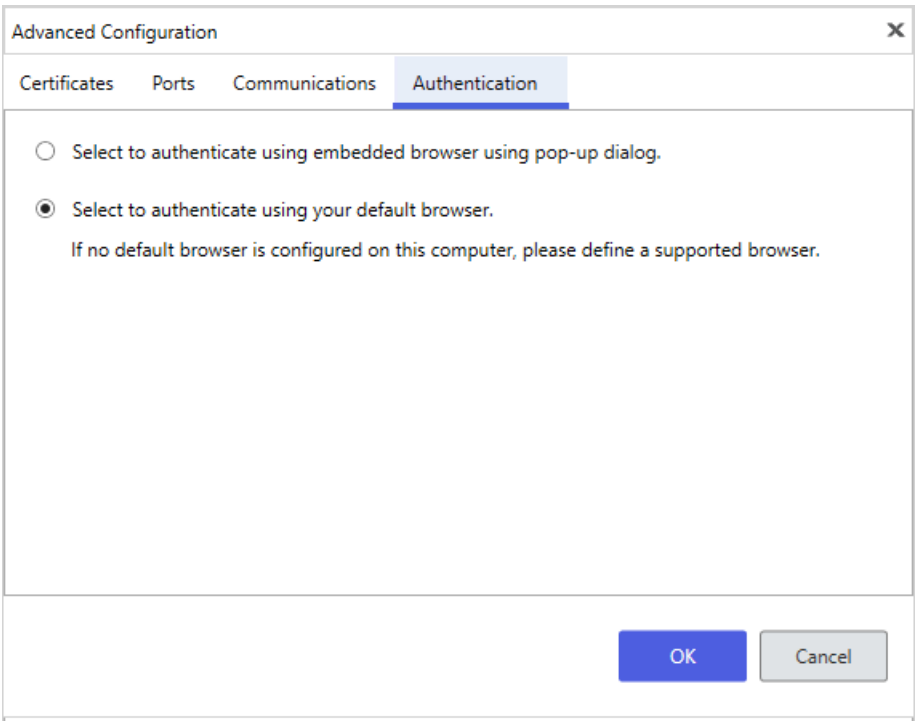
AVEVA Identity Manager を使用した認証

コンフィグレータで Operations Control 接続エクスペリエンスが有効化されている場合、アプリケーション マネージャ、WindowMaker、または WindowViewer の初回起動時に AVEVA Identity Manager での認証を受けるよう求めるメッセージが表示されます。その後の起動では、SSO での認証が行われます。

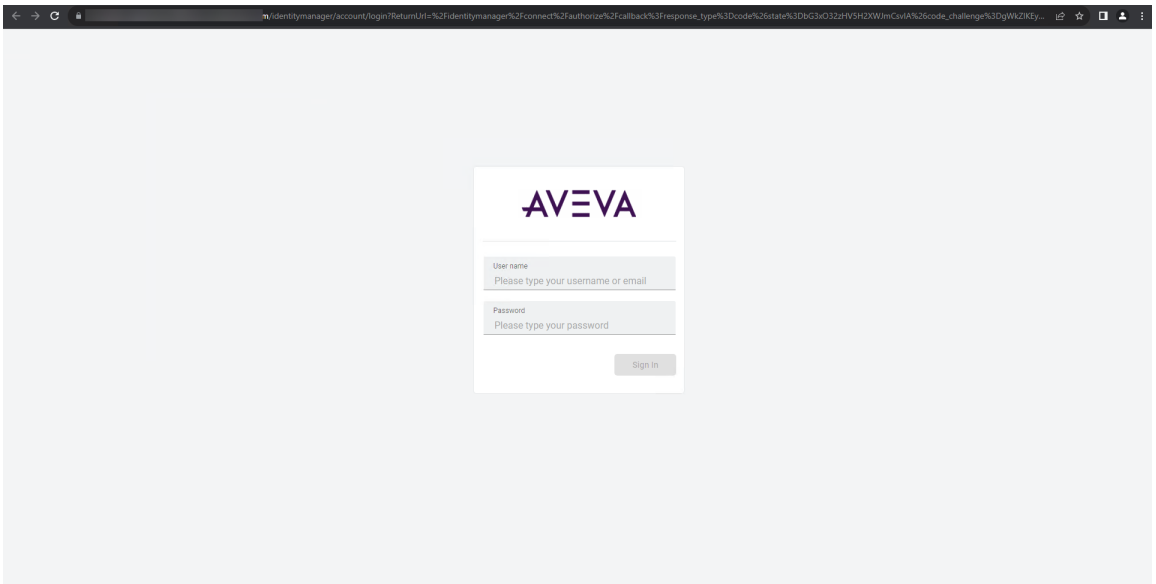
Web ブラウザで AVEVA Identity Manager を使用して認証 :

1. Web ブラウザで AVEVA Identity Manager サインイン画面を開くには、コンフィグレータで［**共通プラットフォーム**］>［**System Management Server**］>［**詳細**］>［**認証**］の［**デフォルト ブラウザを使用して認証する**］を選択する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. いずれかの AVEVA Operation Control アプリケーションを起動すると、システムのデフォルト ブラウザで AVEVA Identity Manager ログイン ページが開きます。
3. AVEVA Identity Manager へのログイン用に CONNECT アカウントに登録されている E メールアドレスを AVEVA Identity Manager ログインページの **[ユーザー名]** フィールドに入力します。
4. ［パスワード］フィールドに**パスワード**を入力します。
5. ［サイン イン］をクリックします。

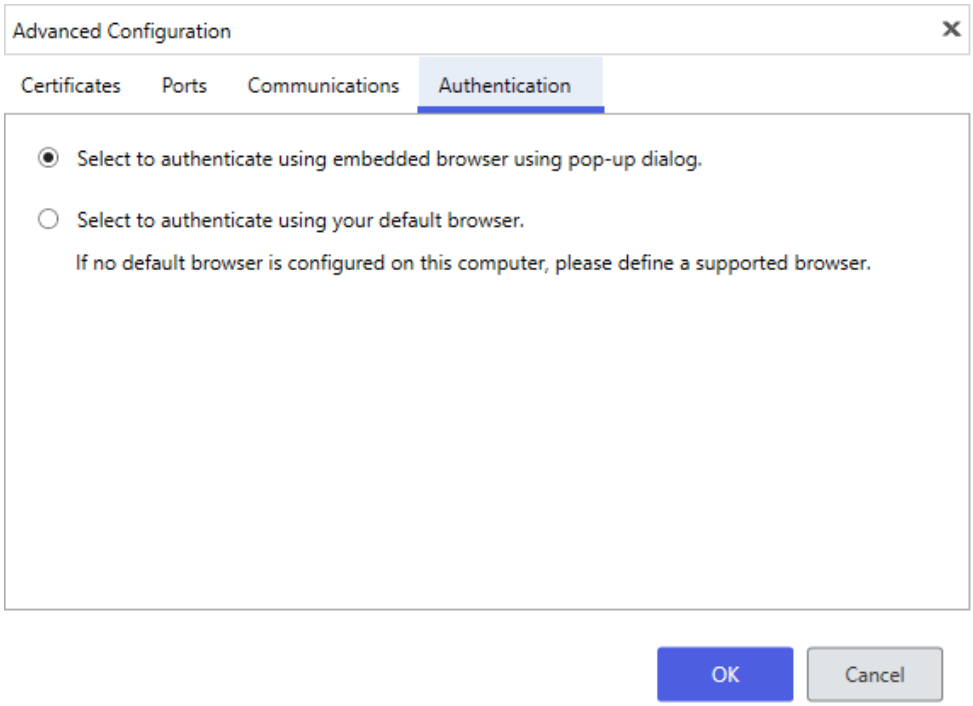


© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

埋め込まれたブラウザで **AVEVA Identity Manager** を使用して認証するには

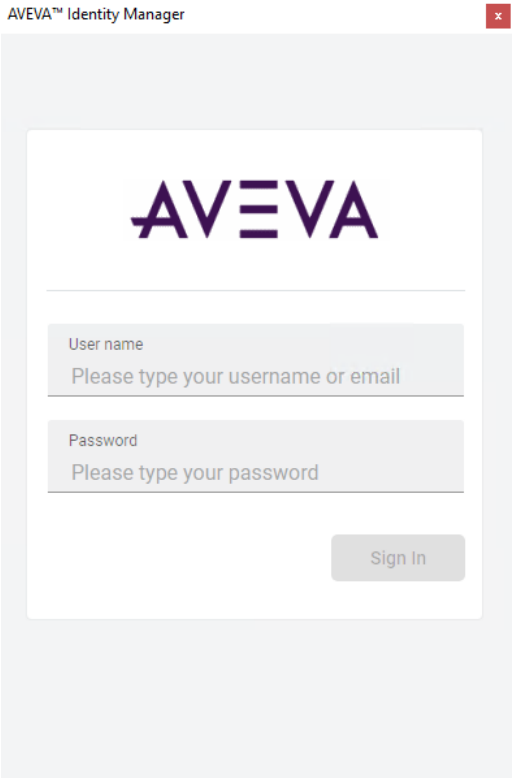
1. 埋め込まれたブラウザで AVEVA Identity Manager サインイン画面を開くには、コンフィグレータで［**共通プラットフォーム**］>［**System Management Server**］>［**詳細**］>［**認証**］の［**ポップアップを使用して、埋め込まれたブラウザを使用して認証する**］オプションを選択する必要があります。既定ではこのオプションが選択されます。

   InTouch Access Anywhere の場合、セキュリティの理由から埋め込まれたブラウザを使用することをお勧めします。



2. いずれかの AVEVA Operation Control アプリケーションを起動すると、埋め込まれたブラウザで AVEVA Identity Manager ログイン ページが開きます。
3. AVEVA Identity Manager へのログイン用に CONNECT アカウントに登録されている E メールアドレスを AVEVA Identity Manager ログインページの **[ユーザー名]** フィールドに入力します。
4. ［**パスワード**］フィールドに**パスワード**を入力します。
5. ［**サイン イン**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



接続損失時の動作

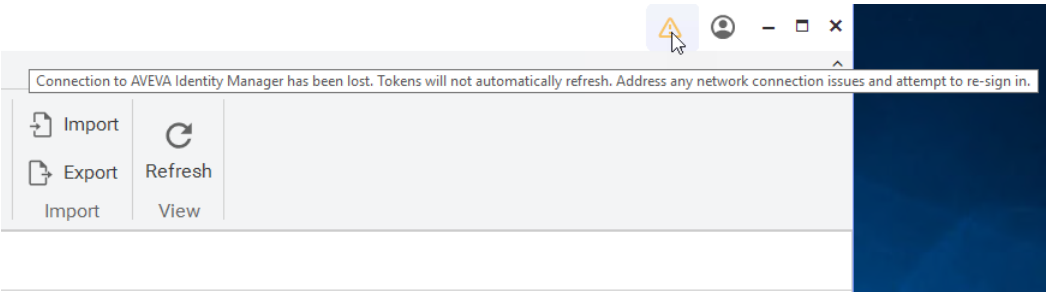
接続が失われた場合:

- 認証が正常に行われ、アプリケーションが正常に実行した後に AVEVA Identity Manager と CONNECT の間の接続が失われた場合でも、オンプレミス機能は引き続き機能します。
- 認証が正常に行われ、アプリケーションが正常に実行しているときに InTouch アプリケーションと AVEVA Identity Manager の間の接続が失われた場合、Operations Control 接続エクスペリエンス機能が停止します。［**サイン イン**］オプションを使用して再度サインインする必要があります。
- 認証が正常に行われた後に接続が失われた場合、WindowMaker でアプリケーションを開こうとすると WindowMaker はデモ モードで実行します。
- 認証が正常に行われた後に接続が失われた場合、WindowViewer でアプリケーションを開こうとすると WindowViewer は読み取り専用モードで実行します。
- AVEVA Identity Manager にサインインしたことがない状態で接続が失われると、WindowMaker または WindowViewer を起動しようとすると、WindowMaker は実行せず、WindowViewer はデモ モードで実行します。

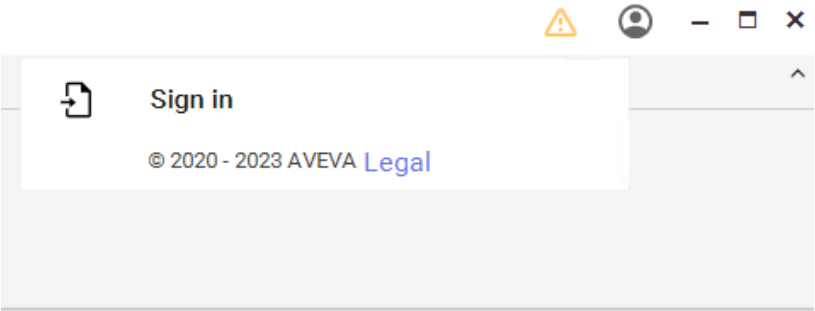
接続損失の問題に対処する

1. CONNECT または AVEVA Identity Manager への接続の問題がある場合、アプリケーションマネージャと WindowMaker に警告アイコンが表示され、接続状態が通知されます。警告アイコンの上にカーソルを置くと、接続損失に関するメッセージを含むツールヒントが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ネットワーク接続の問題に対処してプロファイル アイコンをクリックし、[**サイン イン**] をクリックして再度サインインします。



接続が回復してアプリケーションが再起動すると、ツールチップは表示されなくなります。

ライセンスとエンタイトルメント

Operations Control 接続エクスペリエンスでの InTouch のライセンス取得は、マネージドまたはスタンドアロンアプリケーションの場合と同じです。2 種類のサブスクリプションが使用可能です。

AVEVA Operations Control Edge サブスクリプション

InTouch アプリケーションが ViewApp として設定されておらず、コンフィグレータでライセンスモードが Operations Control 接続エクスペリエンスに設定されている場合、WindowViewer の起動にはこのサブスクリプションが必要です。

AVEVA Operations Control Supervisory サブスクリプション

InTouch アプリケーションが ViewApp として設定されており、コンフィグレータでライセンスモードが Operations Control 接続エクスペリエンスに設定されている場合、WindowViewer の起動にはこのサブスクリプションが必要です。

どちらのサブスクリプションでも、無制限の数のセッションが許可され、WindowMaker の完全な機能が有効になります。

データ ログの表示

監査ログには、アカウントのイベントまたは操作のログが含まれています。ログには、選択した時間帯にアカウントで行われたすべての操作が表示されます。監査ログを表示できるのは、アカウント管理者、または Report Viewer の役割に割り当てられたユーザーだけです。

注：監査ログには、アカウントで監査機能が使用可能になった時点以降のデータが記録されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

監査ログを表示：

- サイト ナビゲーション メニューで［監査］を選択します。［監査ログ］ページが表示されます。

監査ログの以下の情報が表示されます。

- 日付 – 操作が実行された日
- メッセージ – 操作の詳細（どのユーザーが何の操作を実行したかなど）
- 機能 – 操作によって影響を受ける機能

注 : データをユーザーインターフェイスに表示する際はブラウザのローカル時間が使用されます。しかし、.csv ファイルにデータをエクスポートするときには UTC タイム ゾーンが使用されます。

詳細については、 のヘルプを参照してください。

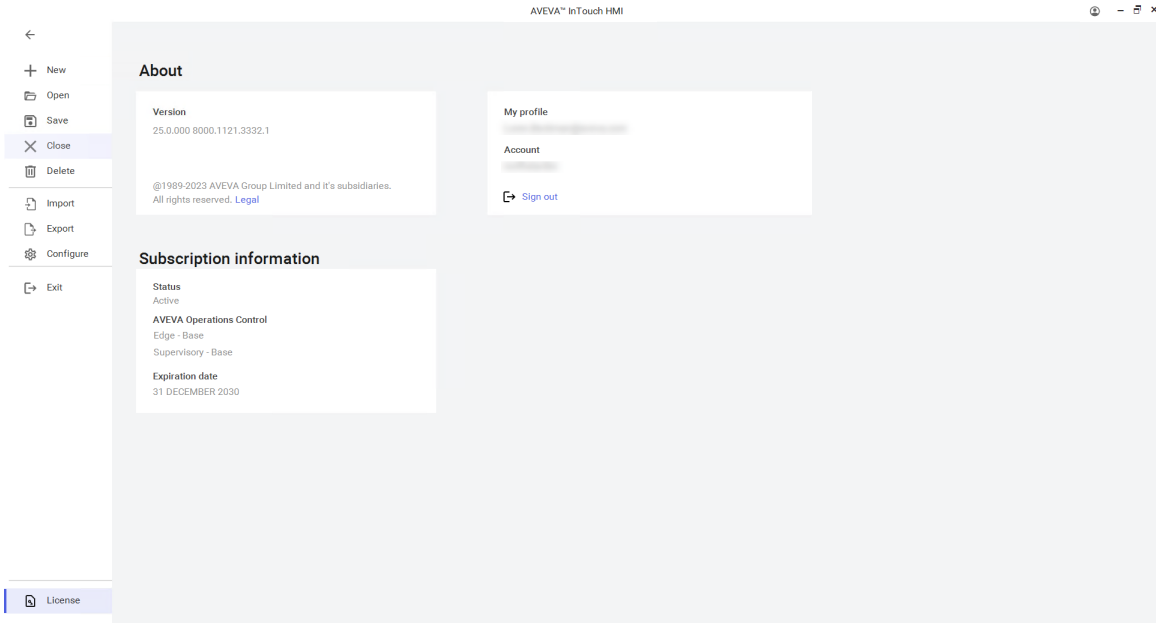
サブスクリプション情報の表示

WindowMaker または WindowViewer によって使用されている現在のサブスクリプションの具体的な情報を表示できます。

WindowMaker のサブスクリプション情報を表示する

1. WindowMaker を開きます。
2. 画面左下隅にある［**ファイル**］メニューの［**ライセンス**］を選択します。

   ［**バージョン情報**］画面が表示されます。［**バージョン情報**］画面には、WindowMaker の**バージョン**および**サブスクリプション情報**が表示されます。

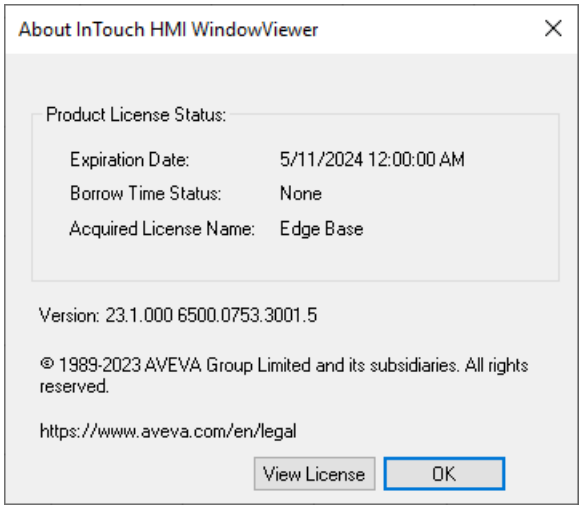


WindowViewer のサブスクリプション情報を表示:

1. WindowViewer を開きます。
2. ［ファイル］、［**WindowViewer のバージョン情報**］の順に選択します。

   ［**WindowViewer のバージョン情報**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. ［ライセンスを表示］をクリックして、サブスクリプションの詳細を表示します。

WindowMaker の起動と実行

このセクションでは、Operations Control 接続エクスペリエンスでの WindowMaker の動作について説明します。

Operations Control 接続エクスペリエンスで WindowMaker を操作する

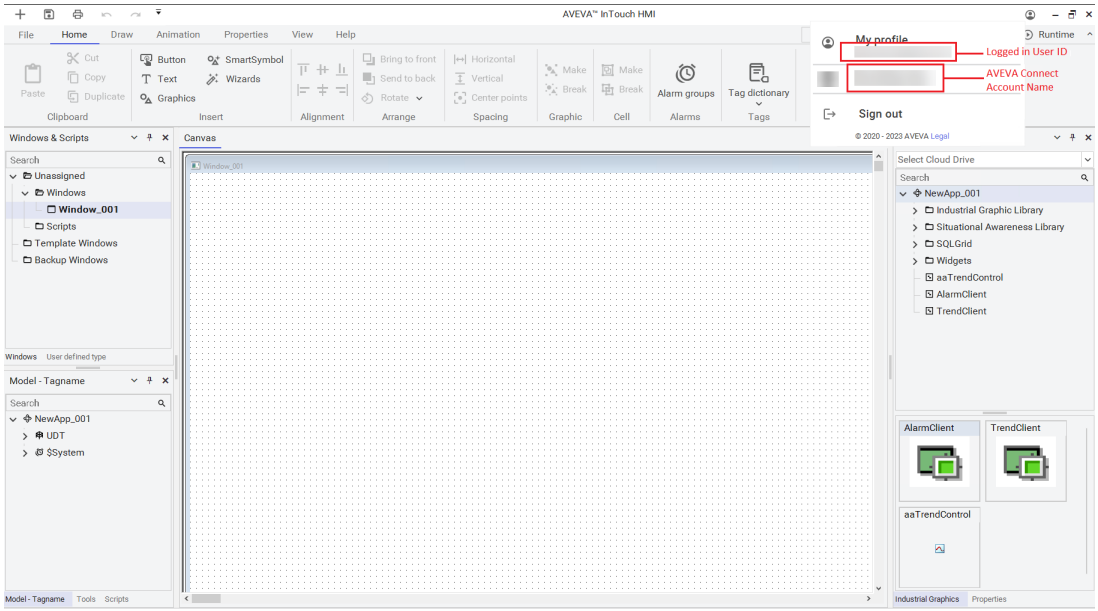
1. WindowMaker を起動します。

   以前に他の Operations Control 製品が起動していない状態で、特定のノードで認証を受けていない場合に同じノードで WindowMaker を初めて起動すると、（コンフィグレータで System Management Server の［認証］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。このノードで認証済みの WindowMaker を既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

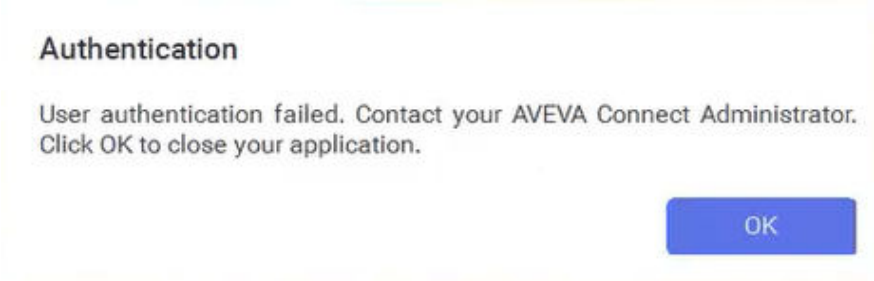
2. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

   正常にサインインした後、Edge または Supervisory サブスクリプションのエンタイトルメントがある場合、WindowMaker のタイトル バーにユーザー プロファイルのサインイン情報が表示されます。マイプロファイルには、ログインユーザーのユーザー ID と CONNECT アカウントの名前が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



3. 有効な資格情報が入力されない場合、または認証がキャンセルされた場合、エラー メッセージが表示されます。CONNECT 管理者に資格情報を確認してサインインをもう一度試してください。[**OK**] をクリックすると WindowMaker が閉じます。



- AVEVA Identity Manager での認証で起動する埋め込まれたブラウザのログイン ページを閉じると、認証失敗のダイアログが直ちに表示されます。
- 資格情報が入力されない場合、または AVEVA Identity Manager での認証で起動した Web ブラウザを閉じた場合、3 分後に認証失敗のダイアログが表示されます。

4. WindowMaker が正常に実行している場合、AIM および CONNECT の更新トークンの有効性がチェックされます。

   有効な AIM トークンが確認された場合、AVEVA Identity Manager と CONNECT の接続が失われるとオンプレミス機能が機能します。CONNECT 産業用グラフィックドライブにはアクセスできません。CONNECT の産業用ドライブにアクセスするには、AVEVA Identity Manager にサインインする必要があります。

WindowViewer の起動と実行

このセクションでは、Operations Control 接続エクスペリエンスでの WindowViewer の動作について説明します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

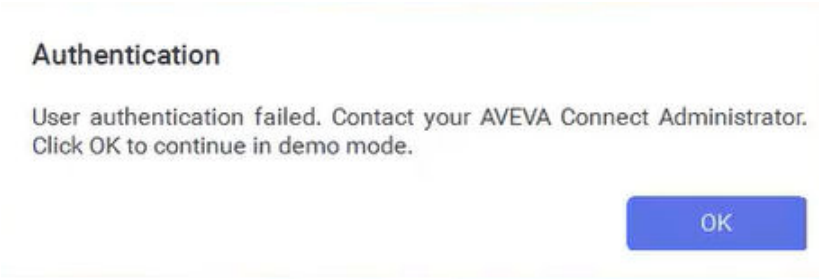
Operations Control 接続エクスペリエンスで WindowViewer を操作：

1. WindowViewer を起動します。

   以前に他の Operations Control 製品が起動しておらず、特定のノードで認証を受けていない場合に同じノードで WindowViewer を初めて起動すると、（コンフィグレータで System Management Server の［認証］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。このノードで認証済みの WindowViewer を既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

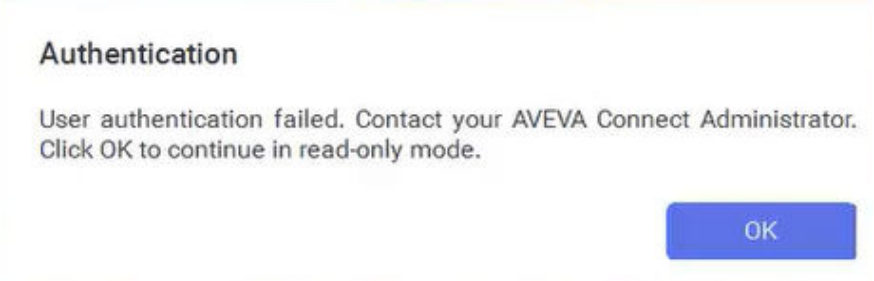
2. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

3. 認証に失敗した場合、または以前にサインインしたことがない場合、WindowViewer はデモ モードで起動します。



4. 認証に失敗した場合、または認証をキャンセルした場合でも、以前にそのノードで少なくとも 1 回は正常に認証およびログインしたことがあれば WindowViewer は読み取り専用* モードで起動します。

   - AVEVA Identity Manager での認証で起動する埋め込まれたブラウザのログイン ページを閉じると、認証失敗のダイアログが直ちに表示されます。
   - 資格情報が入力されない場合、または AVEVA Identity Manager での認証で起動した Web ブラウザを閉じた場合、3 分後に認証失敗のダイアログが表示されます。



5. AVEVA Identity Manager との接続が失われた場合、WindowViewer は読み取り専用* モードで実行します。

6. 正常にサインインした後、Edge または Supervisory サブスクリプションがある場合、WindowViewer のタイトル バーにサインイン情報が表示されます。

7. 正常にサインインしても Edge または Supervisory サブスクリプションがない場合、以下の処理が行われます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アプリケーションに設定されているタグが 64 以下の場合、WindowViewer はデモ モードで起動できます。アプリケーションで 64 以上のタグが設定されている場合、WindowViewer はデモ モードで開かずに終了します。

8. WindowViewer が正常に実行している場合、AVEVA Identity Manager および CONNECT の更新トークンの有効性がチェックされます。有効な AVEVA Identity Manager トークンだけを受け取り、AVEVA Identity Manager が CONNECT から切断された場合は次のようになります。
   - WindowViewer は読み取り専用モードで実行します。警告アイコンは表示されません。接続のステータスを確認するには、GetTokenConnectionstatus() 関数スクリプトを使用する必要があります。スクリプト メソッドを使用して、ユーザー インターフェイスおよびその適切なステータスを表すグラフィックを設定できます。接続が失われると、設定されているスクリプトが実行し、WindowViewer ユーザーには、現在の接続状態に関連付けられている適切なユーザー インターフェイスが表示されます。GetTokenConnectionstatus() スクリプトの詳細については、AVEVA™ InTouch HMI アプリケーション開発ガイド（ITBuild.pdf）の「GetTokenConnectionStatus() 関数」セクションを参照してください。

注：

- Operations Control 接続エクスペリエンスでは、WindowMaker で［**ファイル**］>［**設定**］>［**WindowViewer**］>［**ウィンドウ**］の［**ユーザーの設定**］および［**パスワードの変更**］オプションは無効化されます。これらのオプションは Operations Control 接続エクスペリエンスで適用されないので、これらのオプションを WindowMaker で設定することできません。これらのオプションはデフォルトでオフになり無効化され、WindowViewer で使用することはできません。

- サービスとしての WindowViewer は CONNECT のサブスクリプションを使用せず、従来の永続ライセンスが使用されます。

***読み取り専用モード:**

読み取り専用とは I/O タグへの書き込みができない状態を指しますが、メモリタグへの書き込みは引き続き可能です。読み取り専用モードでは画面を表示できますが、変更を行うことはできません。このモードでは、認証が失敗した場合でもアプリケーションを WindowViewer で実行できます。この読み取り専用モードは WindowViewer 上の実装で、市場でサポートされている通常の読み取り専用モードとは異なります。この読み取り専用モードは、InTouch 製品の設定を有効にする読み取り専用は提供せず、読み取り専用ライセンスにも関連しません。

以前に少なくとも 1 回はサインインしたことがあり、サブスクリプションの有効期限が切れていないとき、以下のいずれかの条件に一致する場合、WindowViewer は読み取り専用モードで実行します。

- 認証に失敗した
- 認証を望まない
- ネットワーク障害またはタイムアウトが発生した

サブスクリプションの有効期限が切れると、WindowViewer はデモ モードで実行します。

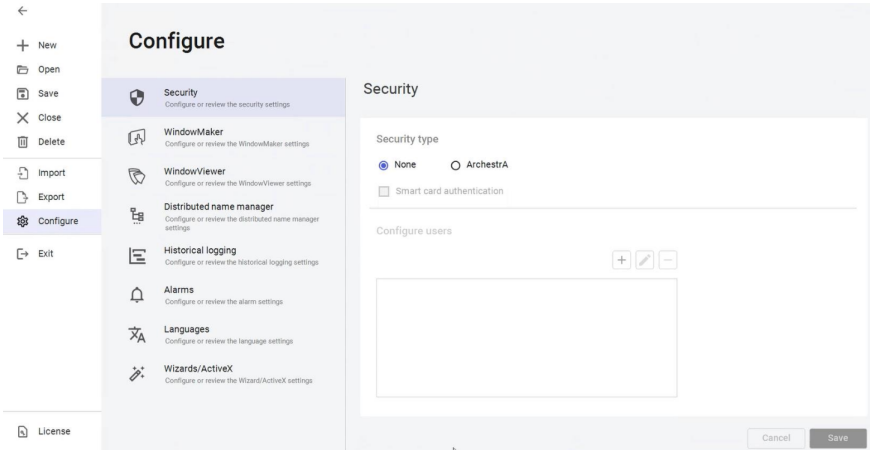
Operations Control 接続エクスペリエンスでのアプリケーション セキュリティの設定

WindowMaker が Operations Control 接続エクスペリエンス モードで機能するには、以下のいずれかのセキュリティ タイプを設定する必要があります。

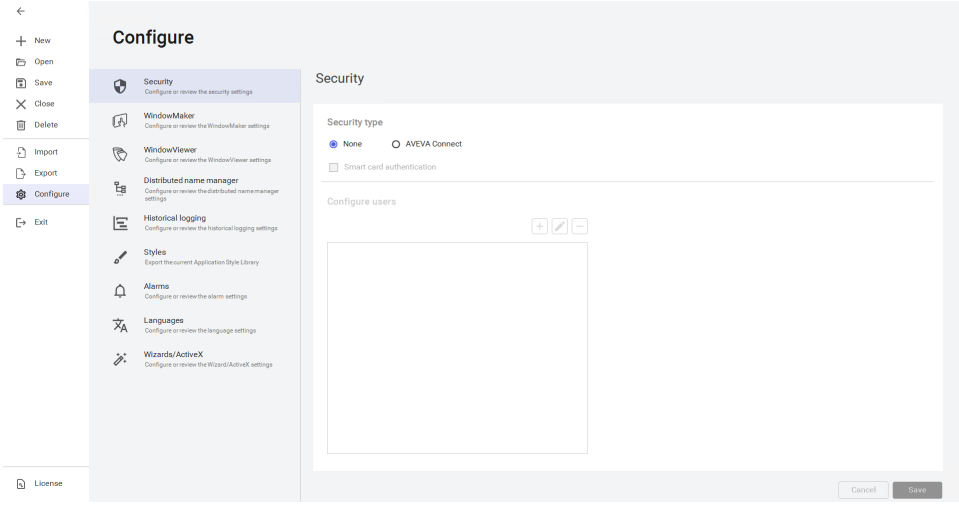
- マネージド アプリケーションでのセキュリティ設定

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- なし
- ArchestrA（**[認証モード]**タブの CONNECT ベースのセキュリティグループ設定を System Platform IDE で設定する必要があります。詳細については、「マネージド アプリケーションでのセキュリティ設定」を参照してください。



- スタンドアロン アプリケーションでのセキュリティ設定
  - なし
  - CONNECT



Operations Control 接続エクスペリエンス モードが有効で、互換性のないセキュリティ タイプが設定されている場合、WindowViewer または WindowMaker を開くと、上記のサポートされているセキュリティ タイプのいずれかで WindowMaker のセキュリティを再設定するよう求めるメッセージが表示されます。［**OK**］をクリックすると WindowViewer が閉じ、セキュリティ タイプを編集するオプションが WindowMaker に表示されます。

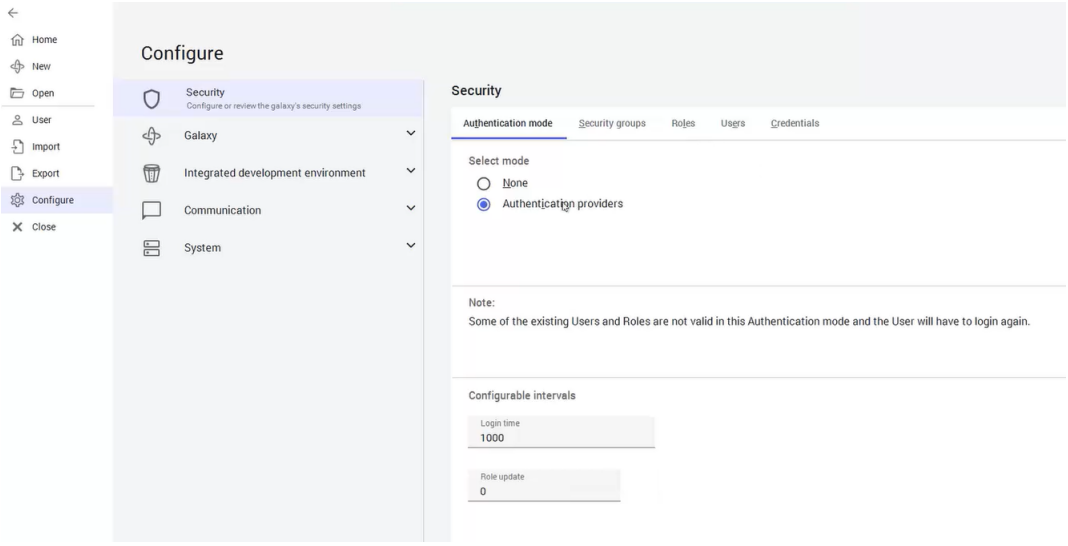
マネージド アプリケーションでのセキュリティ設定

マネージド アプリケーションでセキュリティを ArchestrA として設定：

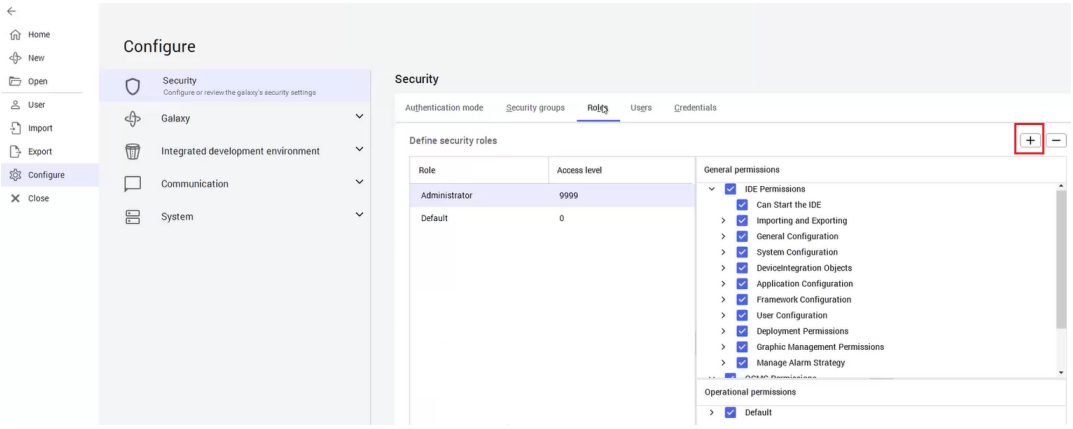
1. System Platform IDE を起動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. ［**Galaxy**］>［設定］>［セキュリティ］に移動して［セキュリティ］ページを開きます。［**認証モード**］タブで［**認証プロバイダ**］オプションを選択します。

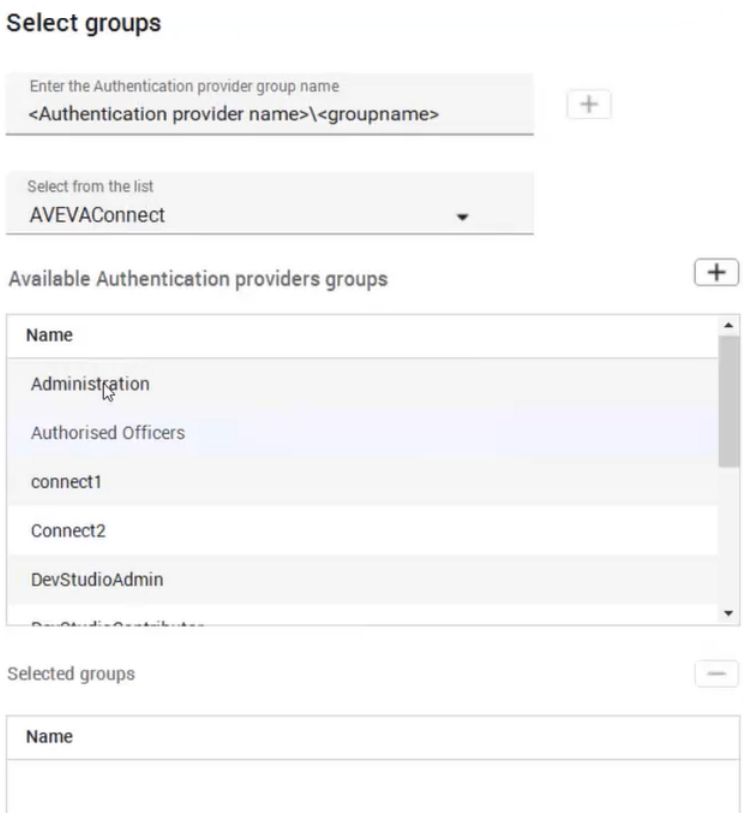


3. アクセス レベルを CONNECT グループにマップするには、**[設定]>[セキュリティ]>[役割]**に移動してプラスアイコンをクリックします。

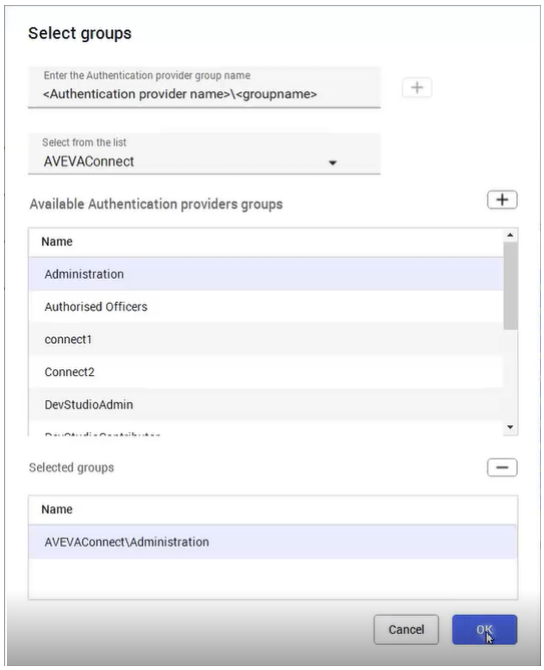


4. ［リストから選択］ドロップダウンで ［**CONNECTProduct**］ が選択されていることを確認します。CONNECT のアカウントで使用できるグループのリストが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. 使用可能なリストから目的のグループを追加すると、［選択されたグループ］リストに表示されます。終了したら［**OK**］をクリックします。



6. グループに対して適切な**アクセス レベル**、**一般権限**、および**操作権限**を設定し、［**保存**］をクリックします。
7. ViewApp を選択して WindowMaker を起動します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

8. WindowMaker の ［ファイル］>［設定］>［セキュリティ］でセキュリティ オプションとして［**ArchestrA**］を選択し、［保存］をクリックします。

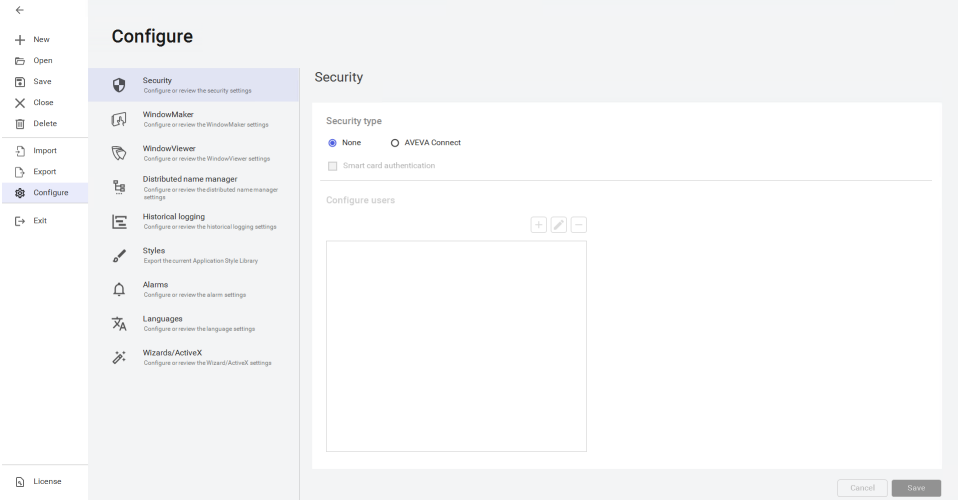
マネージド アプリケーションでセキュリティを「なし」として設定するには

1. System Platform IDE を起動します。
2. ViewApp を選択して WindowMaker を起動します。
3. ［ファイル］>［設定］>［セキュリティ］でセキュリティ オプションとして［なし］を選択し、［保存］をクリックします。

スタンドアロン アプリケーションでのセキュリティ設定

スタンドアロン アプリケーションでのセキュリティ設定：

1. WindowMaker の［ファイル］>［設定］>［セキュリティ］でセキュリティオプションとして［なし］または［**CONNECTProduct**］を選択し、［保存］をクリックします。



Operations Control 接続エクスペリエンスが有効で、互換性のないセキュリティ タイプが設定されている場合、WindowViewer または WindowMaker を開くと、上記のサポートされているセキュリティタイプのいずれかで WindowMaker のセキュリティを再設定するよう求めるメッセージが表示されます。**[OK]**をクリックすると WindowViewer が閉じ、セキュリティタイプを編集するオプションが WindowMaker に表示されます。**[InTouch]**と**[オペレーティングシステム]**の各セキュリティタイプは Operation Control 接続エクスペリエンスではサポートされていません。また、非接続エクスペリエンスでは CONNECT セキュリティタイプを使用できません。

2. スタンドアロン InTouch でグループにアクセス レベルにマップするには、スクリプト関数 **AddPermission()** を使用します。

   AddPermission() メソッドが Operations Control 接続エクスペリエンスで受け入れるのは 2 つのパラメータだけです。

   - CONNECT グループ
   - アクセス レベル

   Operations Control 接続エクスペリエンスの AddPermission() のスクリプト関数を以下に示します。

```
DiscreteTag=AddPermission("", "CONNECT group", AccessLevel);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイムユーザーが CONNECT の複数のグループのメンバーである場合、アクセスレベルは最上位のアクセスレベルを持つグループによって決定されます。

混合モードでの操作

このセクションでは、セキュリティ タイプに Operations Control 接続エクスペリエンスとの互換性がない場合の WindowMaker および WindowViewer の動作について説明します。

互換性のないセキュリティタイプで WindowMaker を起動：

1. WindowMaker を起動します。

   以前に他の Operations Control 製品が起動していない状態で、特定のノードで認証を受けていない場合に同じノードで WindowMaker を初めて起動すると、（コンフィグレータで System Management Server の［**認証**］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。このノードで認証済みの WindowMaker を既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

2. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

3. 正常にサインインして有効なエンタイトルメントがある場合、セキュリティ タイプに互換性がなければ、セキュリティを設定するよう求めるメッセージが表示されます。

   注 : [InTouch]および**[OS]**の各セキュリティタイプは Operation Control 接続エクスペリエンスではサポートされていません。

4. ［**はい**］をクリックして続行し、セキュリティ タイプを変更します。［**いいえ**］をクリックすると終了します。

   セキュリティ タイプを変更するには、WindowMaker でセキュリティを設定します。セキュリティの設定方法の詳細については、Operations Control 接続エクスペリエンスでのアプリケーション セキュリティの設定」を参照してください。

   注 : スクリプト関数に AVEVA Operations Control 接続エクスペリエンスとの互換性があることを確認する必要があります。詳細については、「サポートされている InTouch HMI 機能」セクションを参照してください。

互換性のないセキュリティ タイプでの WindowViewer の起動:

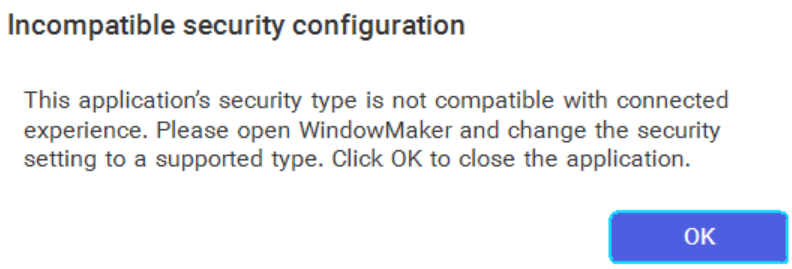
1. WindowViewer を起動します。

2. Web ブラウザを使用して AVEVA Identity Manager で認証を受けるよう求めるメッセージが表示されます。

3. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

4. 正常にサインインして有効なエンタイトルメントがある場合、セキュリティ タイプに互換性がなければ、WindowMaker でセキュリティを設定するよう求めるメッセージが表示されます。［**OK**］をクリックして WindowViewer を終了します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



注 : **[InTouch]**および**[OS]**の各セキュリティタイプは Operation Control 接続エクスペリエンスではサポートされていません。

5. WindowMaker でセキュリティ タイプを設定します。セキュリティの設定方法の詳細については、Operations Control 接続エクスペリエンスでのアプリケーション セキュリティの設定」を参照してください。。
6. WindowViewer を起動します。

Application Manager の起動と実行

このセクションでは、Operations Control 接続エクスペリエンスでのアプリケーション マネージャの動作について説明します。

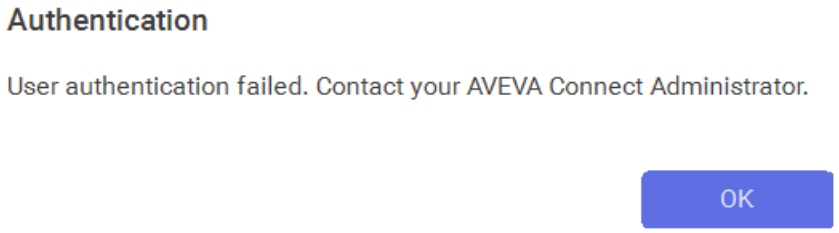
Operations Control 接続エクスペリエンスでアプリケーション マネージャを操作する

1. アプリケーション マネージャを起動します。

   以前に他の Operations Control 製品が起動しておらず、特定のノードで認証を受けていない場合に同じノードでアプリケーション マネージャを初めて起動すると、（コンフィグレータで System Management Server の［**認証**］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。このノードで認証済みのアプリケーション マネージャを既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

2. AVEVA Identity Manager へのサインイン用にアカウントに登録されている電子メール アドレスとパスワードを入力します。
   - 認証が正常に行われた場合、以下の処理が行われます。
     - 以降のログインはシングル サインオンで認証されます。
     - アプリケーション マネージャのリボンにログイン ユーザーの情報が表示されます。
     - タグのパブリッシュ、AVEVA 産業用グラフィック ドライブと資格情報マネージャへのアクセス、読み取り専用または読み取り/書き込みモードでの View アプリケーションの実行、および DBLoad の実行が可能です。
   - 認証に失敗した場合、または認証がキャンセルされた場合、以下の処理が行われます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



  - アプリケーション マネージャはログイン ユーザーなしの状態で引き続き表示されます。
  - WindowMaker または WindowViewer を起動するには、AVEVA Identity Manager で認証を受ける必要があります。
  - WindowViewer は読み取り専用モードでのみ実行できます。読み取り専用モードではアプリケーションを表示できますが、変更を行うことはできません。WindowMaker は終了します。
- AVEVA Identity Manager での認証で起動する埋め込まれたブラウザのログイン ページを閉じると、認証失敗のダイアログが直ちに表示されます。
- 資格情報が入力されない場合、または AVEVA Identity Manager での認証で起動した Web ブラウザを閉じた場合、3 分後に認証失敗のダイアログが表示されます。

アプリケーション マネージャからのアプリケーションの選択と実行

このセクションでは、Operations Control 接続エクスペリエンスでアプリケーション マネージャを使用してアプリケーションを開く動作について説明します。

Operations Control 接続エクスペリエンスでアプリケーションマネージャを使用してアプリケーションを開く：

1. アプリケーション マネージャを起動します。アプリケーション マネージャを起動するには、［**スタート**］をクリックし、［**プログラム**］をポイントして、［**AVEVA InTouch HMI**］をポイントし、次に［**InTouch HMI アプリケーション マネージャ**］をクリックします。

   以前に他の Operations Control 製品が起動しておらず、特定のノードで認証を受けていない場合に同じノードでアプリケーション マネージャを初めて起動すると、（コンフィグレータで System Management Server の［**認証**］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。このノードで認証済みのアプリケーション マネージャを既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

   AVEVA アプリケーション マネージャが開きます。

2. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。認証が正常に完了するか認証に失敗するとアプリケーションマネージャが起動します。

3. 実行するアプリケーションを選択します。

   前の手順で認証に失敗した場合、AVEVA Identity Manager で認証を受けるよう求めるメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

認証が成功し、アプリケーションのセキュリティ タイプに Operations Control 接続エクスペリエンスとの互換性がない場合、セキュリティ タイプに互換性がないことを示すメッセージが表示されます。このメッセージでセキュリティ タイプを変更するか、終了するかを選択する必要があります。

4. ［はい］をクリックすると、アプリケーションが WindowMaker で開きます。
5. ［セキュリティ タイプ］を設定します。

セキュリティ タイプの詳細については、「Operations Control 接続エクスペリエンスでのアプリケーション セキュリティの設定」を参照してください。

InTouch Access Anywhere の起動と実行

このセクションでは、Operations Control 接続エクスペリエンスでの InTouch Access Anywhere の動作について説明します。

Operations Control 接続エクスペリエンスで InTouch Access Anywhere を操作する

1. Web ブラウザを使用して InTouch Access Anywhere を起動します。以下のアドレスにアクセスします。

   `http://ITAA_Server_Node_Name:8080/`

   または

   `http://ITAA_Server_IP_Address:8080/`

1. ログオン ページでユーザー名とパスワードを入力し、表示する InTouch アプリケーションを［アプリケーション名］フィールドのドロップダウン リストから選択します。
2. ［接続］をクリックして InTouch Access Anywhere ブラウザ セッションを開始します。

   以前に他の Operations Control 製品が起動しておらず、特定のノードで認証を受けていない場合に同じノードで InTouch Access Anywhere を初めて起動すると、（コンフィグレータで System Management Server の［認証］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。セキュリティの理由から埋め込まれたブラウザを使用することをお勧めします。このノードで認証済みの InTouch Access Anywhere を既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

3. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。認証が正常に完了すると、WindowViewer が起動します。
4. 認証が正常に行われた場合:
   - 以降のサインインはシングル サインオンで認証されます。
   - WindowViewer では、Edge エンタイトルメント アクセスが使用可能であることが検証されます。
   - エンタイトルメントが確認された場合、WindowViewer では Edge の使用が CONNECT の Web サイトの［監査ログ］ページに記録されます。詳細については、 のヘルプを参照してください。

ネットワーク アプリケーション開発（NAD）アプリケーションの起動

ネットワーク アプリケーション開発（NAD）は Operations Control 接続エクスペリエンスをサポートします。NAD が Operations Control 接続エクスペリエンスで機能するには、システムのすべてのノードで Operations Control 接続エクスペリエンスを有効にする必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Operations Control 接続エクスペリエンスで NAD アプリケーションを操作：

1. NAD クライアントを起動します。

   以前に他の Operations Control 製品が起動しておらず、特定のノードで認証を受けていない場合に同じノードで NAD を初めて起動すると、（コンフィグレータで System Management Server の［**認証**］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。このノードで認証済みの NAD を既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。

2. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

   正常にサインインし、Edge または Supervisory サブスクリプションがある場合、シングル サインオン（SSO）によるサインインが行われ、アプリケーションが実行します。以降のログインでは資格情報を再度入力する必要はありません。

注：

- サインインが正常に行われ、有効なエンタイトルメントがない場合、管理者に有効なエンタイトルメントを要求してください。サインインに失敗した場合、エラー メッセージが表示されます。管理者に有効な資格情報を問い合わせてください。

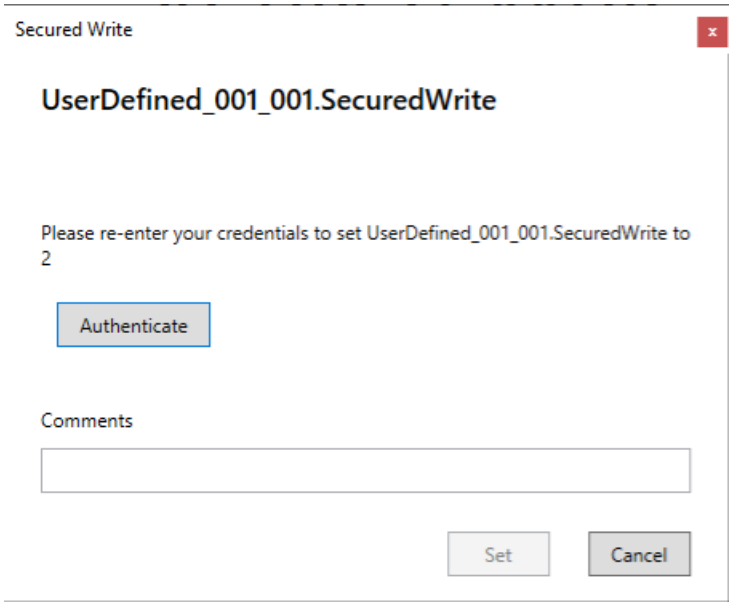
- マシンの接続が失われた場合、再認証を行うことができなくなるので、アプリケーションを実行できなくなります。

セキュリティ書き込みおよび確認書き込みの実行

実行セキュリティ書き込みの実行:

1. 値を変更します。
2. ［セキュリティ書き込み］ダイアログが表示されます。
3. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

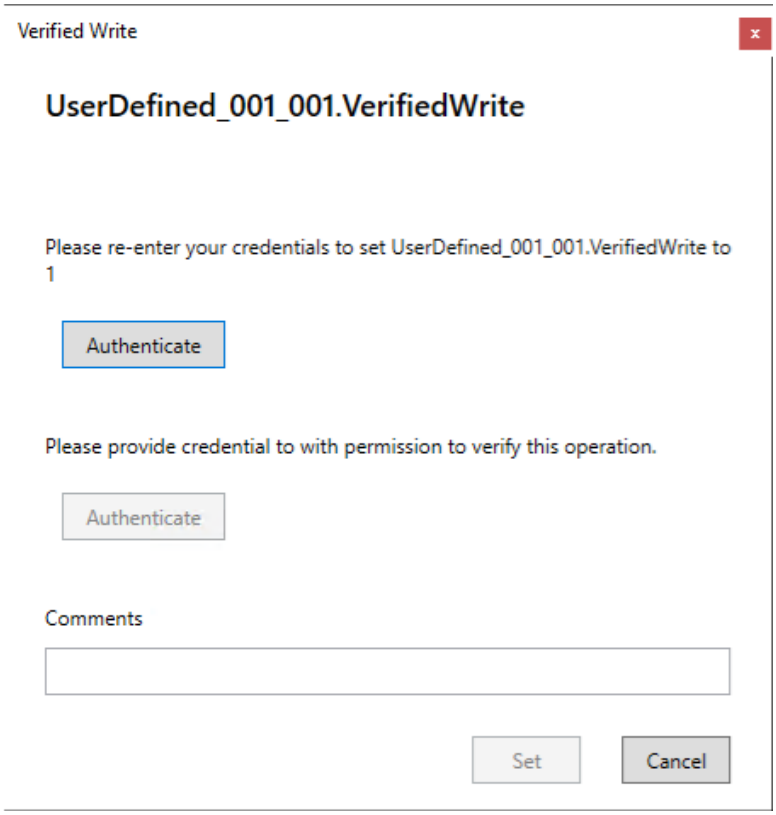


注 : セキュリティ書き込み操作では、シングルサインオンはサポートされていません。したがって、オペレータはセキュリティ書き込み操作で視覚情報を明示的に入力する必要があります。

確認書き込みの実行:

1. 値を変更します。
2. ［確認書き込み］ダイアログが表示されます。
3. オペレータは［**認証**］をクリックし、AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。
4. 確認者は［**認証**］をクリックし、AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



注：確認書き込み操作では、シングルサインオンはサポートされていません。したがって、オペレータと確認者の両方は確認書き込み操作で視覚情報を明示的に入力する確認必要があり両方ます。

Web Client（オンプレミス）の起動と実行

コンフィグレータで Operations Control 接続エクスペリエンスが有効化されている場合、Web Client の初回起動時に AVEVA Identity Manager での認証を受けるよう求めるメッセージが表示されます。サインインには AVEVA ID とパスワードを使用する必要があります。Web Client のその後の起動では、シングルサインオン（SSO）で認証されます。その他の AVEVA アプリケーション（アプリケーション マネージャや WindowMaker など）を起動したときに AVEVA Identity Manager にサインインしていて、有効なエンタイトルメントと権限がある場合、SSO を使用したサインインが行われます。デスクトップ アプリケーション と Web Client の間の SSO は、デスクトップ アプリケーションで認証にシステム ブラウザが使用されている場合にのみ機能します。SSO はブラウザ固有です。したがって、たとえば、1 つのブラウザ（Google Chrome）に資格情報を入力した後に別のブラウザ（Microsoft Edge）で Web Client を開いた場合、または Web Client をシークレット モードで開いた場合、資格情報をもう一度入力する必要があります。

注：System Platform 2023 から System Platform 2023 R2 にアップグレードする際、System Platform 2023 で AVEVA Identity Manager を既に設定している場合、Web Client が Operations Control 接続エクスペリエンスで機能するようにするには、アップグレードした後に Windows 認証に切り替え、次にユーザー認証に切り替えて、ユーザー認証を再設定する必要があります。最初に［**産業用グラフィック サーバー**］>［**認証設定**］で［**Windows 認証**］を選択し、［**設定**］をクリックします。次に、［**産業用グラフィック サーバー**］>［**認証設定**］で［**ユーザー認証**］を選択し、［**設定**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Operations Control 接続エクスペリエンスで Web Client（オンプレミス）を操作：

1. Operations Control 接続エクスペリエンスで Web Client を有効にするには、コンフィグレータの左側のペインで［**産業用グラフィック サーバー**］>［**認証設定**］に移動し、［**ユーザー認証**］を選択して［**設定**］をクリックします。
2. ブラウザで Web Client を起動します。

   以前に他の Operations Control 製品が起動しておらず、特定のノードで認証を受けていない場合に同じノードで Web Client を初めて起動すると、（コンフィグレータで System Management Server の［認証］タグを設定する際に行った選択に基づいて）Web ブラウザまたは埋め込まれたブラウザを使用して AVEVA Identity Manager で認証するよう求めるメッセージが表示されます。このノードで認証済みの Web Client を既に起動している場合、または以前に同じノードで他の Operations Control 製品を起動して認証したことがある場合、シングル サインオンを使用した認証が行われます。詳細については、「AVEVA Identity Manager を使用した認証」セクションを参照してください。
3. AVEVA Identity Manager へのサインイン用に CONNECT アカウントに登録されている E メールアドレスとパスワードを入力します。

   認証が正常に行われ、有効なエンタイトルメントがある場合、Web Client が実行します。以降のログインではシングル サインオン（SSO）が使用されるので、資格情報を入力する必要はありません。

   注：資格情報が正しくない場合、エラーメッセージが表示され、入力を求められます。管理者に資格情報を問い合わせて、正しい資格情報を入力してください。

無効なエンタイトルメントのシナリオ

このセクションではエンタイトルメントが有効でない場合に発生する可能性のあるさまざまなシナリオについて説明します。

- エンタイトルメントへのアクセスがない場合、WindowViewer はデモ モードで起動します。

  注：アプリケーションに設定されているタグが 64 個以下の場合、デモモードで起動できます。アプリケーションで 64 以上のタグが設定されている場合、デモ モードで起動できません。

- 現在の認証が失敗しても、ノードで以前に少なくとも 1 回は正常な認証とログインが行われたことがある場合、WindowViewer は読み取り専用モードで起動します。読み取り専用モードは、ローカルに保存された有効期限まで機能します。有効期限に達すると、アプリケーションはデモ モードで実行します。
- WindowViewer を起動して認証をキャンセルした場合、およびログイン タイムアウトがある場合、WindowViewer はデモ モードまたは読み取り専用モードで起動します。
- 認証を試みたときに有効な AVEVA Identity Manager トークンだけを受け取り、CONNECT から切断された場合、オンプレミス機能は機能しますが、クラウドアクセスは失敗します。クラウド アクセスを取得するにはログインする必要があります。
- 認証を試みたときに有効な AVEVA Identity Manager トークンを受け取らなかった場合、および AVEVA Identity Manager から切断された場合、WindowViewer はデモ モードまたは読み取り専用モードで起動します。
- WindowViewer が正常に実行されている場合、AVEVA Identity Manager および CONNECT の更新トークンの有効性がチェックされます。有効な AVEVA Identity Manager トークンだけを受け取り、AVEVA Identity Manager と CONNECT から切断された場合、オンプレミスの機能は機能しますが、クラウド アクセスは失敗します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- AVEVA Identity Manager との接続が失われている場合、WindowViewer はデモ モードまたは読み取り専用モードで起動します。

サポートされていない機能

以下の機能は、Web Client ではサポートされていません。

- Windows 認証は Operations Control 接続エクスペリエンスでサポートされていません。
- 共有機能は Operations Control 接続エクスペリエンスでサポートされていません。
- Web Client では、冗長シングルサインオン機能はいずれのライセンスモードでもサポートされていません。

Operations Control 接続エクスペリエンスでの Web Client のライセンスとエンタイトルメント

他のライセンス モードから Operations Control 接続エクスペリエンスに切り替えた場合、［**認証**］設定を再構成する必要があります。

Operations Control 接続エクスペリエンスで［**InTouchView アプリケーション**］チェック ボックスがオンの場合、Web Client を起動するには、アカウントに Supervisory サブスクリプションのエンタイトルメントが必要です。Web Server を起動した後に**[InTouchView アプリケーション]**チェックボックスの設定を変更した場合、CONNECT 産業用グラフィック Web Server を再起動する必要があります。

トレンドコントロールの起動と実行

トレンド コントロールはシングル サインオン機能を活用するので、WindowViewer からのシームレスなシングル サインオンが実現します。このセクションでは、Operations Control 接続エクスペリエンスでのトレンド コントロールの動作について説明します。

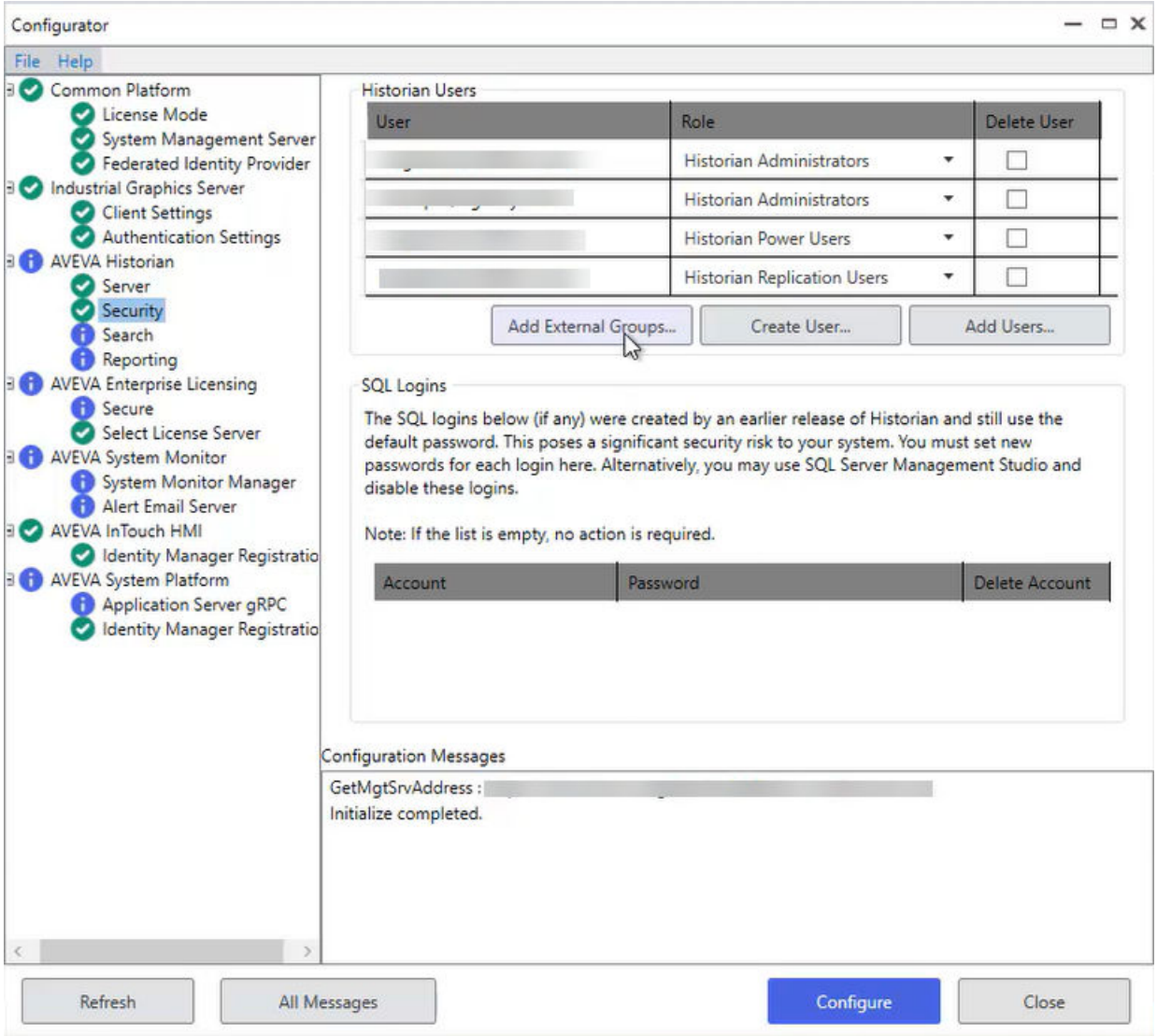
Operations Control 接続エクスペリエンスでトレンド コントロールを操作

コンフィグレータの設定:

「Operations Control 接続エクスペリエンスの設定」セクションに示されているその他のコンフィグレータ設定とは別に、以下を設定する必要があります。

1. AVEVA Historian Server を設定します。詳細については、コンフィグレータ のヘルプを参照してください。
2. コンフィグレータで［**AVEVA Historian**］>［**セキュリティ**］に移動します。
3. ［**外部グループを追加**］を選択します。

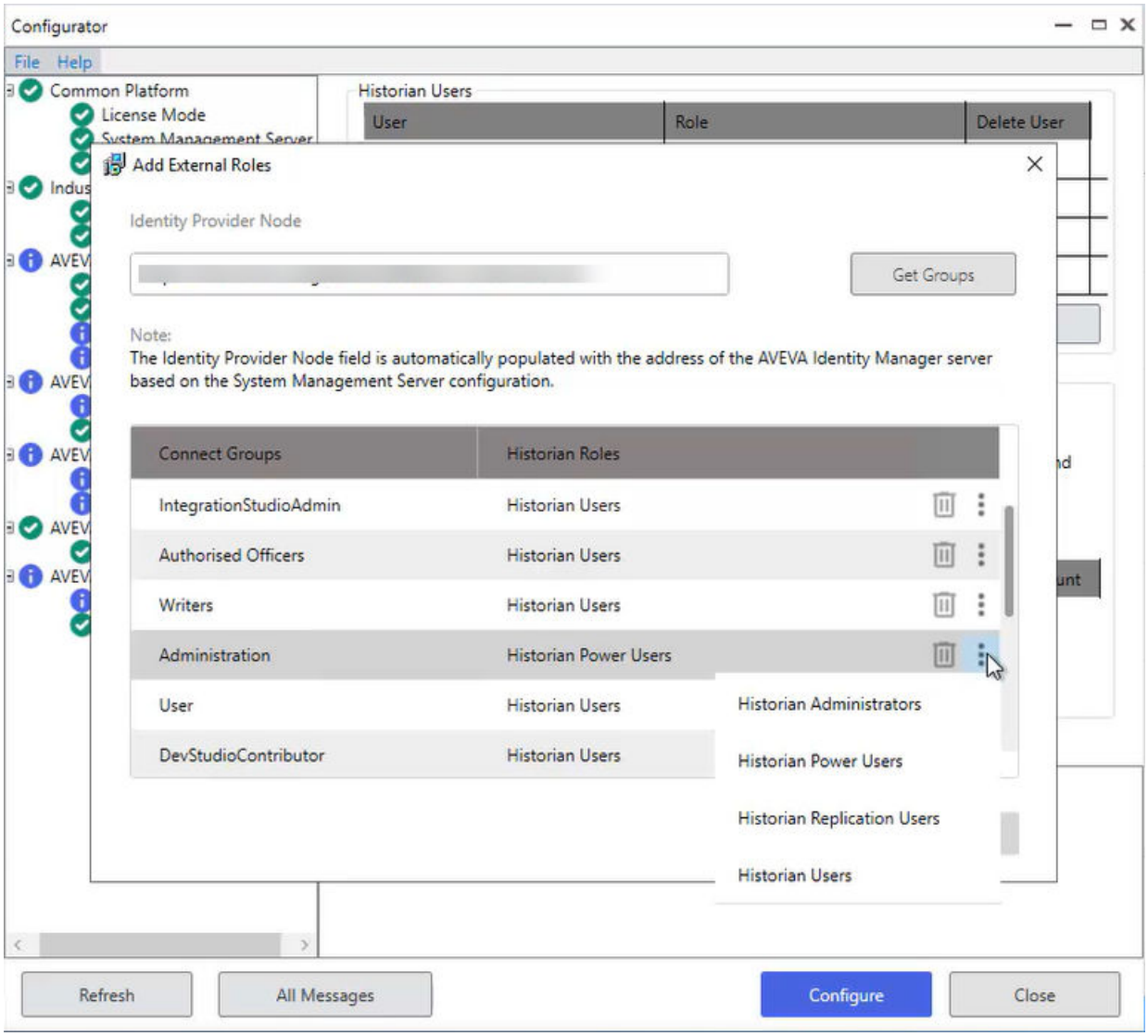
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. **[外部役割を追加]**ウィンドウに CONNECT に含まれるグループが一覧表示されます。**[Connect グループ]**列でトレンドコントロールユーザーを追加する CONNECT グループを選択し、垂直の省略記号（3 つのドット）をクリックして**[Historian 管理者]**または**[Historian パワーユーザー]**のいずれかを選択します。

注：トレンドコントロールが Operations Control 接続エクスペリエンスで機能するには、トレンドコントロールのユーザーは CONNECT グループに属していること、そのグループが **Historian 管理者** または **Historian パワーユーザー**としての **Historian の役割**を持っている必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. ［保存］を選択し、［設定］を選択します。

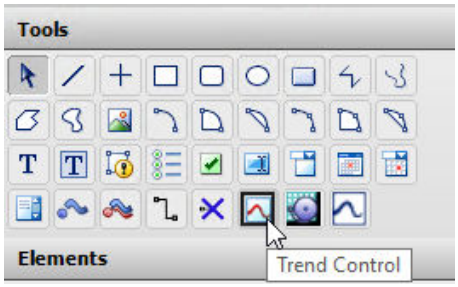
履歴ログの設定

コンフィグレータを設定した後、WindowMaker で**履歴ログ**を設定する必要があります。詳細については、AVEVA™ InTouch HMI アプリケーション開発ガイド（ITBuild.pdf）の「一般的なログ プロパティの設定 - Historian への保存」セクションを参照してください。

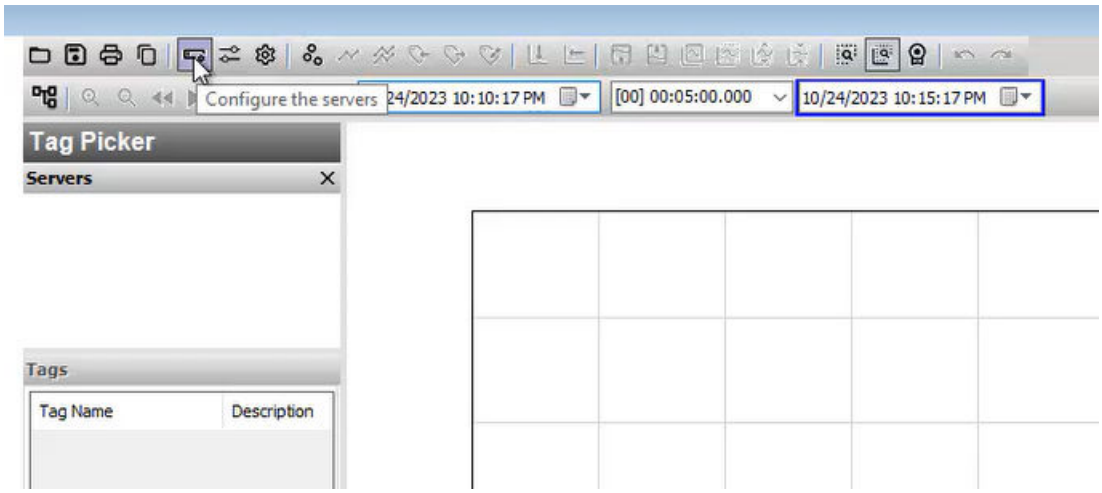
サーバー リスト設定でログインをシングル サインオンとして設定します。

1. WindowMaker の［産業用グラフィック］ペインで右クリックし、［新規グラフィック］を選択してグラフィックを作成します。
2. 新しく作成したグラフィックを右クリックして［開く］を選択するか、グラフィックをダブルクリックして産業用グラフィック エディタで開きます。
3. ［ツール］ペインから［トレンド コントロール］を選択してキャンバスに追加します。グラフィックを保存して閉じます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

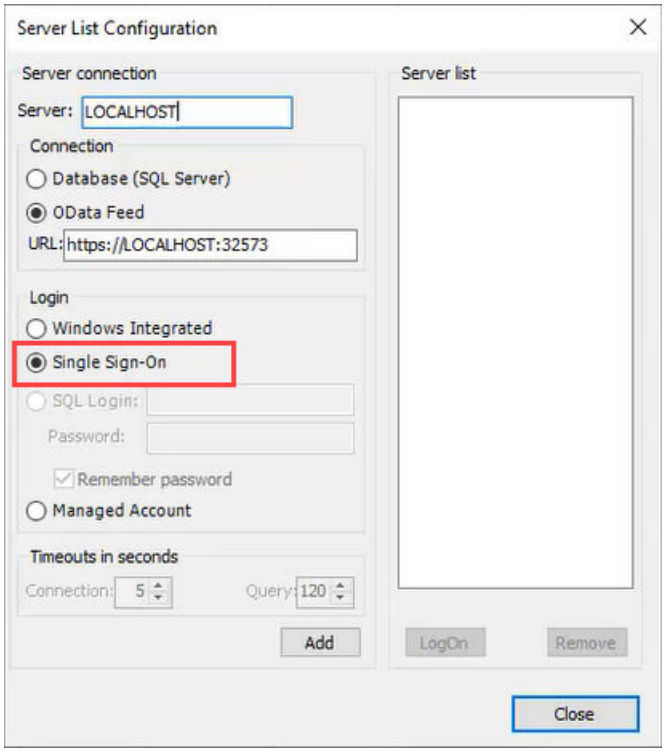


4. WindowMaker で、新しく作成したグラフィックをキャンバスに追加して保存します。
5. WindowMaker の右上隅にある［**ランタイム**］をクリックして、WindowViewer でウィンドウを開きます。
6. ［**サーバーを設定**］をクリックします。



7. ［**サーバー リストの設定**］ダイアログ ボックスの［**サーバー**］フィールドに入力すると、入力内容にしたがって［**OData フィールド**］の［**URL**］が更新されます。
8. ［**ログイン**］セクションで［**シングル サインオン**］を選択し、［**追加**］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



サーバーが［**サーバー リスト**］に追加されます。

9. ［**ログオン**］をクリックします。

AVEVA アプリケーションからのサインアウト

［**プロファイル**］アイコンの［**サイン アウト**］オプションをクリックして、InTouch、ViewApp、またはその他の Operations Control 接続エクスペリエンス アプリケーションからサインアウトできます。アプリケーションからサインアウトすると、以下の処理が行われます。

- CONNECT セッションが閉じるとサインアウトします。
- デスクトップ アプリケーションからサインアウトすると、その特定のデスクトップ アプリケーションからのみサインアウトします。その他のデスクトップ アプリケーションは、現在のサインイン ユーザーで引き続き実行します。
  - 後でデスクトップ アプリケーションを起動すると、再度認証を受けるよう求めるメッセージが表示されます。
- 埋め込まれたブラウザを使用してデスクトップ アプリケーションからサインアウトすると、Web アプリケーションは引き続き実行します。
- サインアウトせずにアプリケーションを閉じると、セッションには影響しません。閉じるのは、そのアプリケーション インスタンスだけです。セッションは CONNECT サーバー上で存続します。そのアプリケーションを再度開くと、セッションは利用可能で、ユーザーはシングル サインオンで認証されます。
  - セッションの有効期限が切れている場合、アプリケーションを起動すると、再度認証を受けるよう求められます。
- WindowMaker からサインアウトすると、セッションが閉じて WindowMaker が終了します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- WindowViewer からサインアウトすると、WindowViewer は読み取り専用モードで実行します。

サポートされている InTouch HMI 機能

サポートされているクライアント コントロール

以下の InTouch HMI クライアント コントロールが AVEVA Identity Manager 認証 を使用した Operations Control 接続エクスペリエンスでサポートされています。

- **アラームクライアントコントロール：**アラームクライアントコントロールは SSO 機能を活用するので、履歴ブロックのビューアプリケーションからシングルサインオンをシームレスに利用できます。エンタイトルメントで認証されたユーザーは、WindowViewer で Historian アラームを表示できます。

  注：アラームクライアントコントロールが Operations Control 接続エクスペリエンスで機能するには、CONNECT から外部役割を設定する必要があります。

- **トレンドコントロール：**トレンドコントロールは SSO 機能を活用するので、ビューアプリケーションからシングルサインオンをシームレスに利用できます。
- **トレンドペン：**トレンドペンは SSO 機能を活用するので、ビューアプリケーションからシングルサインオンをシームレスに利用できます。エンタイトルメントで認証されたユーザーは、WindowViewer で履歴トレンドを表示できます。

サポートされているシステム タグ

以下の InTouch HMI システム タグが AVEVA ID を使用した Operations Control 接続エクスペリエンスでサポートされています。

- **$AccessLevel：**$AccessLevel システムタグが更新され、AddPermission()メソッドとともに使用されるようになりました。ランタイムユーザーが CONNECT の複数のグループのメンバーである場合、アクセスレベルは最上位のアクセスレベルを持つグループによって決定されます。
- **$AccessTokenChanged：**$AccessTokenChanged システム論理型タグは読み取り専用タグです。
- **$Operator：**$Operator システムタグは CONNECT のサインインユーザーを表示します。
- **$OperatorName：**$OperatorName システムタグは CONNECT のサインインユーザーを表示します。

スクリプト メソッド

InTouch HMI は、Operations Control 接続エクスペリエンス モードでスクリプト メソッドもサポートします。Operations Control 接続エクスペリエンスをサポートするために以下のスクリプト メソッドが追加または変更されています。

- GetAccessToken()
- GetAccessSecureToken()
- AddPermission()

  AddPermission() メソッドが Operations Control 接続エクスペリエンスで受け入れるのは 2 つのパラメータだけです。

  - CONNECT グループ
  - アクセス レベル

  Operations Control 接続エクスペリエンスの AddPermission() のスクリプト関数を以下に示します。

  ```
  DiscreteTag=AddPermission("", "CONNECT group", AccessLevel);
  ```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイムユーザーが CONNECT の複数のグループのメンバーである場合、アクセスレベルは最上位のアクセスレベルを持つグループによって決定されます。

- GetTokenConnectionstatus()
- PostLogonDialog() 関数
- SignedWrite()
- IsAssignedRole()
- Logoff()

注：ChangePassword()スクリプト関数は Operations Control 接続エクスペリエンスでサポートされていないため、無効になります。

CONNECT 産業用グラフィックドライブ

Operations Control 接続エクスペリエンスは、WindowMaker で CONNECT 産業用グラフィックドライブのシングルサインオン機能を提供します。接続エクスペリエンスの CONNECT で認証されている場合は、WindowMaker の起動時に産業用グラフィックドライブを表示できます。その場合、再認証を行う必要なく CONNECT 産業用グラフィックドライブを WindowMaker で使用できます。したがって、Operations Control 接続エクスペリエンスでは、WindowMaker の**[ファイル]**メニューに **CONNECT** タブは表示されません。

- Operations Control 接続エクスペリエンスで WindowMaker を起動すると、資格情報を使用した CONNECT での認証またはシングルサインオンでの認証が正常に行われ、かつ有効なサブスクリプションがある場合、WindowMaker が開きます。
  - グラフィックツールボックスで使用できる CONNECT 産業用グラフィックドライブを表示できます。ドロップダウン メニューを使用して、アカウント内の他のテナントに移動することもできます。
  - CONNECT 産業用グラフィックドライブに対して設定されている権限に基づいて、産業用グラフィッククラウドドライブを操作できます。

注：CONNECT 産業用グラフィックドライブにアクセスするには、コンテンツ投稿者の役割が必要です。これは、Operations Control 接続エクスペリエンスと非接続エクスペリエンスの両方で使用する場合に必要です。

セキュリティ書き込み/確認書き込み

- WindowViewer は、AVEVA ID を介した認証でセキュリティ書き込み/確認書き込みをサポートします。
- セキュリティ書き込みアクションまたは確認書き込みアクションを実行するには、CONNECT で認証を受ける必要があります。
- セキュリティ書き込み/確認書き込み操作では、SSO はサポートされていません。

ネットワーク アプリケーション開発（NAD）

NAD は、した Operations Control 接続エクスペリエンスおよび AVEVA Identity Manager での認証でサポートされています。

InTouch Access Anywhere

InTouch Access Anywhere は、Operations Control 接続エクスペリエンスおよび AVEVA Identity Manager での認証でサポートされています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Web Client

- Web Client の URL にアクセスすると、CONNECT での認証を受けるよう求めるメッセージが表示されます。
- 正常にログインした後、有効なエンタイトルメントと権限がある場合、Web Client が実行します。

サポートされていない InTouch HMI 機能

このリリースの時点では、以下の InTouch 製品と機能は Operations Control 接続エクスペリエンスでサポートされていません。

- インサイトパブリッシャ
- ウィジェット

以下の InTouch 製品と機能は Operations Control 接続エクスペリエンスでサポートされていません。

- Alarm DB Logger
- Alarm DB Purge-Archive
- Alarm DB の復元
- Alarm Hot Backup Manager
- スマート カード
- アプリケーション マネージャ資格情報マネージャ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

章 37 Operation Control モード以外での InTouch の使用

このセクションでは、AVEVA Operations Control 接続エクスペリエンスの認証およびエンタイトルメントによって制御されない機能について説明します。

InTouch HMI でのライセンス

InTouch HMI では、InTouch でライセンスを使用するために AVEVA Enterprise License Server を使用します。AVEVA Enterprise License Manager は、1 つまたは複数の License Server を管理します。

InTouch HMI でライセンスを使用できるようにするには、以下の手順を実行します。

1. ライセンスの購入時に提供されたエンタイトルメント XML ファイルをインポートします。
2. **License Manager** のインターフェイスを使用して、License Server 上でアクティブ化するエンタイトルメントのライセンスを選択します。
3. アクティブ化されたライセンスは、WindowMaker または WindowViewer の起動時に使用可能になります。

アクティブ化されたライセンスは、License Manager の License Grid に表示されます。

InTouch は、以下の場合に使用中のライセンスを解放して License Server に返します。

- InTouch を実行するマシンがシャットダウンした場合
- InTouch アプリケーションがシャットダウンした場合

注記：InTouch HMI が異常シャットダウンした場合、ライセンスは返却されません。ライセンスを返すには、InTouch HMI を再起動して、手動でシャットダウンする必要があります。

License Manager と License Server は、InTouch HMI と共にインストールされます。デフォルトでは、InTouch HMI はローカルの License Server をポイントします。この設定は、インストール後のコンフィグレータで変更できます。詳細な手順については、『*AVEVA Enterprise Licensing Guide*』を参照してください。

InTouch ライセンスは、アプリケーションを実行するときに使用できるさまざまな数のタグ変数に基づいています。InTouch のライセンス体系でタグ変数がどのように数えられるのかを理解する必要があります。一連の関数を使用して、アプリケーション内のリモート参照タグの数を予期して計算できます。

重要：ライセンス数は随時変わる場合があります。

ライセンスタグの数について

InTouch アプリケーションの実行中は、タグのハンドルがメモリ データベースに保存されます。各タグには、ハンドルを割り当てる必要があります。タグのハンドルが初期化され、WindowViewer によって使用されますが、アプリケーションの停止後はディスクに永続的に保存されません。

InTouch HMI は、無制限の数のタグをサポートするようになりました。したがって、ランタイム データベースには理論的に無制限の数のライセンス タグを保存できます。これには、ローカル タグとリモート タグ ソースを参照するタグの両方が含まれます。ただし、テストされているのは最大 300000 のタグハンドル（300K）です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アプリケーションは、このランタイム データベース内で利用可能なメモリ ハンドルを超える数のアクティブなタグを同時に持つことができません。InTouch ライセンスによって、ランタイム データベース内でハンドルを割り当てることができる、ローカル タグおよびリモート タグの数が決まります。

また、アプリケーション内のアクティブなタグの最大数は、ランタイム データベースの機能的な制限によって制限されます。実際のタグの最大数は、タグ ハンドルの理論上の最大数より少なくなります。考えられるタグの最大数から、一連の定数が減算されます。

- WindowViewer ランタイム データベース内で無効なハンドル値が発生したかどうかを示すために、無効なタグのハンドル ビットが予約されます。
- InTouch バージョン 10 には、ユーザー定義のタグによって置き換えることができない、34 個のシステム タグが含まれています。バージョン 7.11 以前のアプリケーションを現在のバージョンの InTouch に移行する場合、システム タグの数は 37 個です。
- 設定時に、プレースホルダ タグに保存するために 4096 個のデータベース ハンドルが予約されます。設定時にウィンドウ、スクリプト、またはシンボルをインポートすると、このメモリ セグメントにプレースホルダ タグが割り当てられます。ランタイム中に、リモート参照タグに割り当てるために、4096 個のすべてのプレースホルダ ハンドルを利用できます。

InTouch ライセンス オプションは、アプリケーションで使用できるローカルおよびリモート参照タグの最大数に基づきます。

InTouch タグ ライセンスが 60K タグ未満用である場合、スティッキー タグ ライセンス スキーマが適用されます。スティッキー タグは、WindowViewer がリモート参照のデータ変更の通知を受信したときに、ランタイム時にバインドされるリモート タグ リファレンスです。WindowViewer によって、ランタイム時にリモート タグ リファレンスが InTouch ライセンスの上限まで更新されます。WindowViewer では、ライセンス制限を超えた追加のリモート タグ リファレンスは更新されません。WindowViewer では、ウィンドウが閉じているときは、リモート参照タグの数は削減されません。各リモート参照タグの数は、アプリケーションの実行中に固定されます。

InTouch ライセンスのリモート参照タグの最大数を超えると、単一のメッセージが表示されます。ライセンスの最大数に達した後は、無効なリモート参照タグに関連付けられている値はアプリケーションで更新されません。ライセンス制限に対してカウントされるリモート タグ リファレンスにまだ関連付けられていない 1 つまたは複数のリモート タグ リファレンスが含まれている他のウィンドウを開く前に、アプリケーションを停止して再起動する必要があります。

60K 以上のタグを許可するライセンスでは、スティッキー タグは適用されず、タグの数や、InTouch アプリケーション内で使用できるローカル タグとリモート タグの組み合わせの制限はありません。

たとえば、60K ライセンスを使用した場合のローカル タグおよびリモート参照タグの使用制限を以下に示します。

- 可能なローカル タグの合計

  61404 = 65535 – (4096 + 37 + 1)

- 可能なリモート参照タグの合計

  最大 = 65535 – (37 + 1 + ローカル タグの数)

したがって、60K ライセンスでアプリケーションを実行すると、ローカル タグ データベース内のユーザー定義のすべてのタグの潜在的なリモート参照と効率的にやり取りできます。ランタイム設定では、常に 4,096 個以上の可能なリモート参照タグを利用できます。100000 個のタグを使用できるライセンスの場合、最大数 = 100000 - (37 + 1 + ローカル タグの数) です。

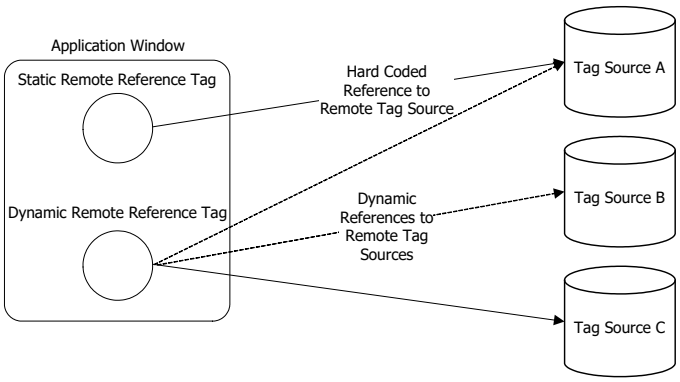
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch リモート参照の制限とは

InTouch リモート参照タグには、2 つのタイプがあります。*静的*リモート参照は、タグ名ディクショナリからタグを定義するときに、固定リモート アドレスにハード コード化されます。静的リモート参照は、アプリケーションが実行を開始するときに、タグ データベース内のタグのハンドルに割り当てられます。静的リモート参照タグの数は、アプリケーションの実行中に固定されます。

*動的*リモート参照によって、アプリケーションの実行中に対象アドレスが解決されます。動的リモート参照タグがデータベース ハンドルに割り当てられた場合、.Reference ドットフィールドまたは IOSetRemoteReference() 関数をスクリプト内で使用することにより、ランタイム中に対象アドレスを変更できます。

以下の図は、300K ライセンス下で実行されている、スティッキー リモート参照タグを持たない InTouch アプリケーションの例を示しています。静的リモート参照タグの数は、アプリケーションの実行中に固定されます。しかし、動的リモート参照タグの数は、アクティブなタグ ソース用のみとなります。リモート タグ ソースへの以前の接続は固定されず、リモート参照またはタグの合計数には数えられません。



InTouch 60K ライセンスでは、動的リモート タグ参照の数に制限を課すスティッキー タグの数は使用されません。これにより、アプリケーションはアプリケーションの実行中に 60K を超えるタグに動的にアクセスできるようになります。動的リモート参照のタグの使用数は、リモート参照が使用されているウィンドウを開いたり閉じたりするたびに変動します。しかし、アプリケーションは、ランタイム タグ データベースの使用制限を超える数のアクティブなタグを同時に持つことができません。

動的参照タグの数は、WindowViewer がディスク保存領域を使用して実行中のアプリケーションの内容を保存する際にのみ増減します。WindowViewer が InTouch ウィンドウまたは産業用グラフィックをキャッシュするよう設定されている場合、ウィンドウがキャッシュから削除された場合以外、リモート参照タグの数は減少しません。

ウィンドウが表示されていない場合でも、リモート タグ参照がそのソースにバインドされていないわけではありません。しかし、ウィンドウ キャッシュが完全に無効化され、高優先度ウィンドウが指定されていない場合は、WindowViewer は従来の「常にディスクからロード」シナリオと同様に機能します。その場合、ウィンドウが閉じられ、動的リモート タグ参照が 60K ライセンス環境で再要求されたとき、すべてのウィンドウがメモリから削除されます。

I/O タグのリモート参照は、InTouch ライセンスのスティッキー リモート参照数には含まれません。I/O タグのリモート参照は、スティッキー リモート参照の制限数を超えることなく変更できます。回数の制限はありません。

セカンダリ タグ ソースへのフェイルオーバーが発生した場合、アプリケーションは、ライセンスされたタグの数を増やさなくても、セカンダリ ソースから同じアイテムにアクセスできます。フェイルオー

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

バー後、セカンダリ タグ ソースから新しいアイテムにアクセスすると、タグの数が増えます。これらのアイテムは、メイン タグ ソースにフェイルバック後にアクセス可能になります。

タグの数がライセンスされた最大数に達すると、メインまたはセカンダリ タグ ソースからアイテムにアクセスできるとしても、それ以上アイテムをアクティブにすることはできません。

InTouch HMI で使用できるライセンス

InTouch HMI は、さまざまなシナリオを管理するための各種ライセンスを提供します。ライセンスは、以下のようなさまざまなパラメータに基づいて決定されます。

1. **コンソール タイプ**: コンソール タイプ（RDS/TSE または非 RDS ノード）を指定します。RDS/TSE は、ターミナル サーバーで設定されるマシン上で実行するコンソールです。非 RDS は、ターミナル サーバーで設定されないマシン上で実行するコンソールです。詳細については、「ターミナル サービスとリモート デスクトップ サービスの配置と操作」を参照してください。
2. **アクセス タイプ**: ノードが設定されているアクセス タイプ（読み取り専用または読み取り/書き込み）を指定します。詳細については、「リモート セッションで実行するアプリケーションへのユーザー アクセスを設定する」を参照してください。
3. **データ ソース**: アプリケーションが使用するデータ ソース（Galaxy または InTouch Tag Server）を指定します。詳細については、「InTouchView アプリケーション」を参照してください。

InTouch HMI の無制限 RDS ライセンス

InTouch HMI の無制限 RDS ライセンスは、WindowViewer における無制限のクライアント セッションで使用されます（RDS 対応のマシンのみ）。WindowViewer がライセンスを取得すると、アプリケーションがライセンスされます。読み取り/書き込みアクセスは、リモート アクセス設定に応じて異なります。同一の無制限ライセンスでは、読み取り専用と読み取り/書き込みアクセスの両方が提供されます。ライセンスが取得されていない場合は、ライセンスは既存の RDS 処理およびリモート アクセス設定に依存します。

RDS と非 RDS 環境の InTouch ライセンス

InTouch アプリケーションが リモート デスクトップ サービス（RDS）で有効化されたサーバー ノード上で実行している場合、コンソールは RDS クライアント セッションと同様に動作します。各セッションでは 1 つのライセンスが使用されます。セッションごとに 1 つの InTouch 開発ライセンスも別途使用されます。

この場合、InTouch アプリケーションの ReadWrite（読み取り／書き込み）機能はリモート アクセス設定によって拒否され、使用されたライセンスによって確認されます。たとえば、RDS クライアント セッションで ReadOnly（読み取り専用）リモート アクセス設定が WindowViewer で起動された場合、ReadOnly InTouch ライセンスが検索されます。License Server で RDS ReadOnly ライセンスが使用できない場合、起動時のライセンス検証は失敗します。

RDS が有効でないノードでは、オペレーティング システムで許可されている RDS クライアント セッションでログインすることもできます。この非 RDS 環境で InTouch アプリケーションが実行している場合、クライアント セッションはコンソールと同様に動作します。この場合、アプリケーションのリモート アクセス設定では、ReadWrite アクセスは決定されません。ReadWrite アクセスは、非 RDS 環境のライセンスによってのみ決定されます。

RDS 以外の環境では、アプリケーションがフリーモードで実行されている場合、常に読み取り/書き込みモードで実行されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouchView アプリケーション ライセンスについて

InTouchView アプリケーションには、Application Server 環境で使用するために特に設計された視覚的なインターフェイスが表示されます。このアプリケーション タイプの詳細については、「InTouchView アプリケーション」を参照してください。

InTouchView アプリケーションで使用されるライセンスのタイプは、その IntouchView アプリケーションが RDS 環境で実行しているかどうかに依存します。

InTouchView アプリケーションが RDS クライアント セッションで実行している場合、アプリケーションのリモート アクセス設定に応じて、ReadOnly または ReadWrite クライアント接続ライセンスが検索されます。

1 つの RDS セッションで使用されるのは、1 つの接続ライセンスだけです。

注：Galaxy データソースで設定されている場合、InTouchView アプリケーションは OMI の ViewApp アプリケーションと同じライセンスを使用します。

InTouchView アプリケーション ライセンス

InTouch HMI アプリケーションは InTouchView アプリケーションとして設定し、InTouch タグ変数サーバーまたは Galaxy に対するクライアントとして使用できます。

InTouch タグ変数サーバーのデータにアクセスするように InTouchView アプリケーションを設定した場合に使用可能なライセンスを以下に示します。

	InTouch HMI クライアント読み取り/書き込みライセンス	**InTouch HMI クライアント読み取り専用ライセンス**	**InTouch HMI の無制限クライアント ライセンス***
リモート アクセス	読み取り/書き込み	読み取り専用	読み取り/書き込み & 読み取り専用
RDS/TSE	可	Yes	可
非 RDS	可	可	不可
MarkAppReadOnlyNonRDS のサポート	可	可	該当なし

* このライセンスは、無制限の数の RDS クライアントで使用できます。

Galaxy のデータにアクセスするように InTouchView アプリケーションを設定した場合に使用可能なライセンスを以下に示します。

	監視クライアント読み取り/書き込みライセンス	**監視クライアント読み取り専用ライセンス**	**監視クライアント サーバー ライセンス**
リモート アクセス	読み取り/書き込み	読み取り専用	読み取り/書き込み &

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

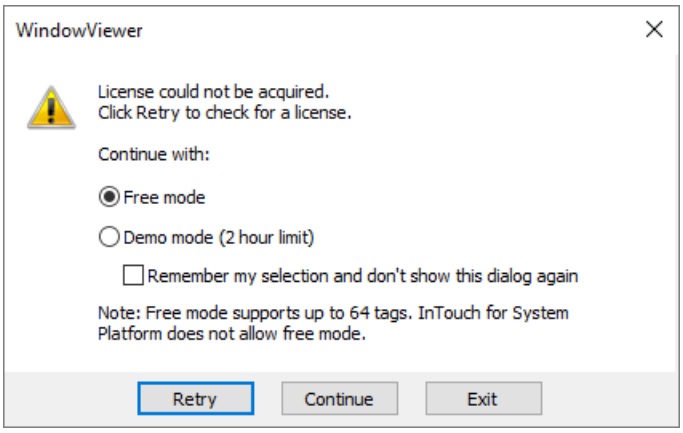
			読み取り専用
RDS/TSE	可	Yes	可
非 RDS	可	可	不可
OMI との共有	可	Yes	可
MarkAppReadOnlyNonRDS のサポート	可	可	該当なし

WindowMaker ライセンス

WindowMaker にはライセンスは必要ありません。WindowMaker にはライセンスは不要で、制限なく実行されます。

WindowViewer ライセンス

WindowViewer を開くと、ライセンスチェックが行われます。適切なライセンスを取得すると、WindowViewer は通常通り開きます。ライセンス取得に失敗した場合は、以下のダイアログが表示されます。



次のいずれかの操作を実行できます。

- 有効なライセンスを取得して［**再試行**］を選択します。
- **フリーモード**を選択し、**続行**を選択して Free モードを続行します。詳細については、「フリーモードとデモモードでの操作を参照してください。

  手記： InTouch アプリケーションが InTouchView アプリケーションとして設定されている場合、またはタグ数が 64 より大きい場合、 **Free モード** オプションは無効になります。

- ［**デモモード**］を選択し、**続行**を選択して 120 分間のデモモードで続行します。
- ［**終了**］を選択して WindowViewer を閉めます。

[選択内容を **忘れずにこのダイアログを二度と表示しない**] チェックボックスをオンにすると、以降の WindowViewer の起動でこのモードが自動的に選択され、このダイアログは二度と表示されません。このチェックボックスをリセットする場合は、INTOUCH を編集できます。以下のエントリのアプリケーションフォルダ内の INI ファイルを 0 に設定します。

```
LicenseRememberOption=0
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

サーバー上の InTouch ライセンスの管理に関するベスト プラクティス

InTouch ライセンスを管理する際、InTouch ライセンスの使用を予測できるようにするためのいくつかのベスト プラクティスがあります。ライセンスの使用を予測することにより、特定のシステムで適切なライセンスをオンデマンドで使用できます。このようなライセンス使用を行うと、サーバー ベースの AVEVA Enterprise Licensing システムを使用した InTouch ライセンス管理が容易になります。

予測的なライセンス使用に関する 2 つのベスト プラクティスは、ライセンス予約と浮動ライセンスです。詳細については、以下のセクションを参照してください。

ライセンスの予約

License Manager で特定のデバイスにライセンスを予約できます。特定のデバイスにライセンスを予約すると、別の InTouch アプリケーションではライセンスを取得できなくなり、アプリケーションが中断されることや実行できなくなることを防止できます。

ユーザー ベースのライセンス予約

AVEVA Enterprise License Manager のライセンス予約ページで、特定のユーザー向けにライセンスを予約できます。予約ページでは、この特定の構成が可能ですが、InTouch OMI と InTouch HMI ViewApp のいずれでもユーザー ベースのライセンス予約がサポートされないことを理解しておくことが重要です。最終的な結果として、ソフトウェアは予約されたライセンスを取得できません。したがって、Supervisory クライアント ライセンスにはデバイス ベースの予約を使用してください。

デバイス ベースのライセンス予約

特定のデバイス用に Supervisory クライアント ライセンスを予約する場合、デバイス名は InTouch HMI/OMI ViewApp を実行するコンピュータの名前にする必要があります。ViewApp が RDS またはターミナル サーバー内で実行している場合、デバイス名は以下の命名規則に従う必要があります。

<RDS ホスト名>-<RDP クライアント名>-<インデックス>

ここで、<RDS ホスト名> は RDS またはターミナル サーバーの名前を示し、<RDP クライアント名> は RDP クライアント ソフトウェアを実行する PC を指します。<インデックス> は、単一のクライアント マシンから複数の RDP セッションを実行する場合以外は 1 です。単一のクライアント マシンから複数の RDP セッションを実行する場合、<インデックス> は、それぞれの RDP クライアントの予約ごとに 1 から特定の RDP クライアントから RDP セッションの総数まで増分します。

例 1: "ControlRoomA" というホスト名のコンピュータで InTouch OMI が実行している場合

デバイス名: 「ControlRoomA」

例 2： "ControlRoomB" という名前のコンピュータが単一のリモート デスクトップ クライアント（RDP）を実行し、"PrimaryRDS" というホスト名のリモート デスクトップ サーバー（通称ターミナル サーバー）に接続しています。

デバイス名: "PrimaryRDS-ControlRoomB-1"

例 3： 2 台のコンピュータ（"SupervisorPC1" および "LineMgrA"）がそれぞれ単一のリモート デスクトップ クライアント（RDP）を実行し、、"PrimaryRDS" というホスト名のリモート デスクトップ サーバー（通称ターミナル サーバー）に接続しています。

デバイス名:

ライセンス予約 1："PrimaryRDS-SupervisorPC1-1"

ライセンス予約 2："PrimaryRDS-LineMgrA-1"

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例 4： "ExecutiveDesktop" というホスト名のコンピュータが 4 つのリモート デスクトップ クライアント（RDP）を実行し、、"PrimaryRDS" というホスト名のリモート デスクトップ サーバー（通称ターミナル サーバー）に接続しています。

デバイス名:

ライセンス予約 1："PrimaryRDS-ExecutiveDesktop-1"

ライセンス予約 2: "PrimaryRDS-ExecutiveDesktop-2"

ライセンス予約 3: "PrimaryRDS-ExecutiveDesktop-3"

ライセンス予約 4: "PrimaryRDS-ExecutiveDesktop-4"

RDS 負荷分散をサポートする場合、複数の RDS クライアント セッションがポイントできる単一の License Server 上ですべての RDS ライセンスをアクティブ化することができます。サーバー上のライセンスは、各 RDS クライアント セッション間で共有できるように同じ機能である必要があります。内部パラメータの値が同じである場合、ライセンスは同じ機能であるとみなされます。このシナリオでは予約は必要ありません。RDS クライアント セッションごとに異なる種類のライセンスが必要な場合、各 RDS サーバーに License Server をインストールする必要があります。

ライセンス予約の詳細な手順については、『*AVEVA Enterprise Licensing Guide*』を参照してください。

浮動ライセンス

浮動ライセンスは、いずれのマシンに対しても予約されていないライセンスです。製品名と機能が同じ浮動ライセンスを単一の License Server 上に配置することが推奨されます。たとえば、同じ容量のいくつかの InTouch 2017 ランタイム 60K タグ ライセンスが有効化された License Server を設定できます。同様に、同じ容量の複数の InTouch 2023 ランタイム無制限（Unlimited）タグ ライセンスが有効化された License Server を設定できます。これは、計画的なライセンス使用を行うために推奨される方法です。

しかし、製品名が同じで機能が異なるライセンスを同一の License Server で有効にすることは推奨されません。たとえば、InTouch 2023 ランタイム無制限（Unlimited）タグが有効なライセンスと InTouch 2017 ランタイム 500 タグが有効なライセンスを同一のライセンス サーバーに混在させることができます。このシナリオでは、どのインスタンス WindowViewer でタグの数が多い方のライセンスが使用されるかを把握することはできません。

ライセンス情報を表示する

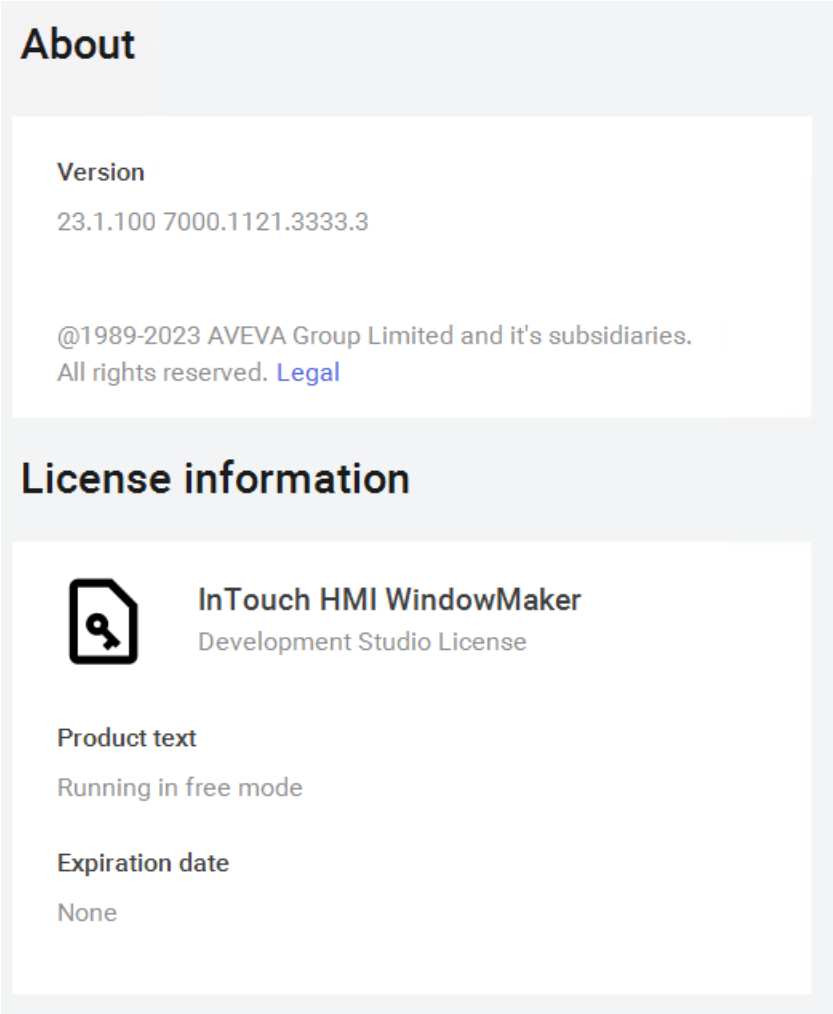
WindowMaker または WindowViewer によって使用されている現在のライセンスの特定の情報を表示できます。

WindowMaker ライセンス情報を表示する

1. WindowMaker を開きます。
2. 画面左下隅にある［**ファイル**］メニューの［**ライセンス**］を選択します。

   ［**バージョン情報**］画面が表示されます。［**バージョン情報**］画面には、WindowMaker の**バージョン**および**ライセンス情報**が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



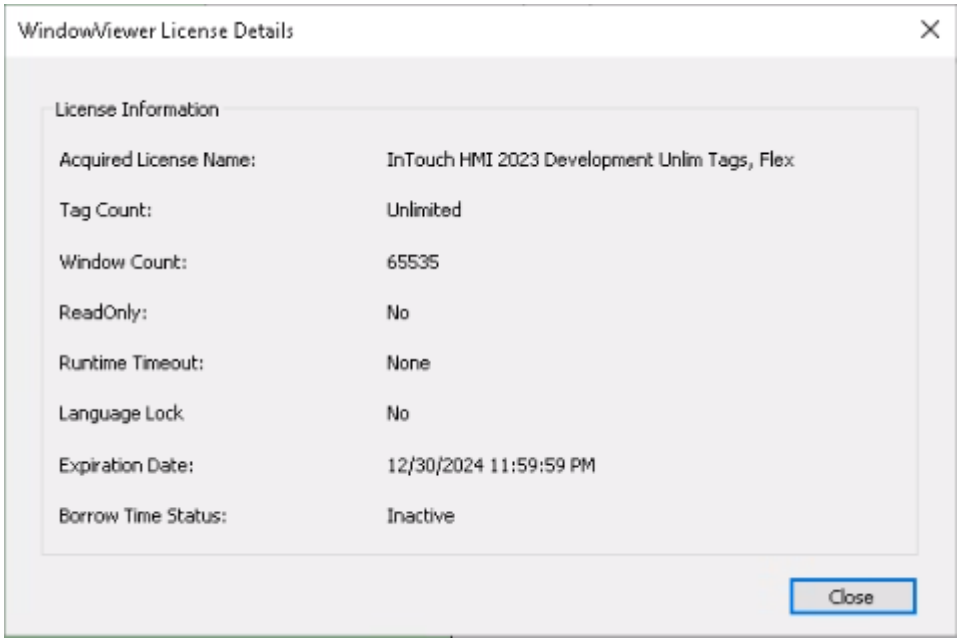
このダイアログには、会社名とライセンスのシリアル番号は表示されません。これらの情報は、AVEVA Enterprise License Manager のインターフェイスに表示されます。

WindowViewer ライセンス情報を表示するには

1. WindowViewer を開きます。
2. ［ファイル］、［**WindowViewer のバージョン情報**］の順に選択します。

   ［**WindowViewer のバージョン情報**］ダイアログ ボックスが表示されます。
3. ［**ライセンス契約書を表示**］を選択してエンドユーザーライセンス契約書を表示します。
4. ［**ライセンスを表示**］をクリックして、ライセンスの詳細を表示します。［ライセンス情報］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



ライセンス情報に表示されるパラメータを以下に示します。

- **取得済みライセンスの名前:** 製品が License Server から使用しているライセンスの完全な名前です。
- **タグ カウント:** 使用中のライセンスで許可されているタグの数。
- **ウィンドウ カウント:** 使用中のライセンスで許可されているウィンドウの数。
- **読み取り専用:** ライセンスで許可されている I/O 読み取り/書き込み権限が表示されます。アプリケーションが I/O タグに書き込めることを示すものではありません。
- **ランタイム タイムアウト:** ライセンスによって割り当てられているアプリケーション ランタイム。タイムアウト時間に達すると InTouch セッションが終了します。
- **言語ロック:** 中国語のオペレーティング システム上の InTouch のライセンスにのみ適用されます。中国語のオペレーティングシステム上で実行するには、中国語か英語のライセンスが必要です。

  言語ロックの制限は、接続ライセンスには適用されません。
- **有効期限:** 使用中のライセンスの有効期限日。
- **猶予時間のステータス:** 割り当てられた猶予期間に関係なく、ライセンスが 50% の状態のときユーザーに通知することを意図します。ステータスは、50%に達すると**アクティブ**に変わります。猶予期間が完全に過ぎてもライセンスが更新されない場合、InTouch のライセンスが切れます。

注：InTouch アプリケーションマネージャでは**[View License]**オプションは無効になっています。アプリケーション マネージャはライセンスを使用しないので、表示できるのは EULA（エンド ユーザー ライセンス契約書）だけです。［**アプリケーション マネージャ**］ダイアログ ボックスには、ライセンス関連の情報は表示されません。

起動後における別のライセンスの使用の管理

標準アップグレードとメンテナンス アクティビティの一環として、License Manager で個々のライセンスをアクティブ化/非アクティブ化することができます。起動後に有効なライセンスを使用した

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WindowMaker または WindowViewer でライセンスを更新できない場合、別のライセンス（当初割り当てられたライセンスや消費されたライセンス）を使用できます。その場合、InTouch は、新しいライセンスの機能が適切かどうかを検証する必要があります。最後の有効なライセンスと現在使用されているライセンスのライセンス機能の比較によって、猶予期間に入ることなく WindowMaker または WindowViewer の実行を続けることができるかどうかが決定されます。猶予期間を終了する方法については、「猶予期間での操作」を参照してください。

以下のセクションでは、起動後に別のライセンスを使用する 2 つのシナリオを示します。

シナリオ 1：起動後に機能の低いライセンスが使用された

このシナリオでは、起動後に使用されたライセンスの機能が最後の有効なライセンスよりも低くなります。これは、ライセンスのダウングレードとみなされ、WindowMaker または WindowViewer が猶予期間に入ります。

ライセンスは、新しいライセンスのパラメータが以下の条件に当てはまる場合にダウングレードしたとみなされます。

猶予期間をトリガするパラメータの変更を以下に示します。

- タグ カウント: タグ カウントが減少した場合、猶予期間がトリガされます。
- ウィンドウ カウント: ウィンドウ カウントが減少した場合、猶予期間がトリガされます。
- ランタイム タイムアウト: タイムアウトの値が［**なし**］からその他の値に変わった場合、猶予期間がトリガされます。
- 言語ロック: 言語ロックの値が［**いいえ**］から［**はい**］に変わった場合（またはその逆の場合）、猶予期間がトリガされます。
- 読み取り専用: 読み取り専用パラメータが［**いいえ**］から［**はい**］に変わった場合（またはその逆の場合）、猶予期間がトリガされます。

ライセンスのダウングレードが発生すると、ダウングレード（「違反」）したパラメータが［**ライセンスを表示**］ダイアログ ボックスに表示されます。たとえば、最初に WindowMaker でタグ カウントが 6000 のライセンスを使用し、3000 タグ カウントのライセンスにダウングレードした場合、タグ カウント パラメータは次のように表示されます。

重要：アプリケーションは、最後の有効なライセンスの機能で実行し続けます。この例では、猶予期間に入ったとき、元の 6000 タグ カウントが維持されます。

シナリオ 2：起動後に機能の高いライセンスが使用された

このシナリオでは、起動後に使用されたライセンスの機能が最後の有効なライセンスと同じか、それよりも高くなります。これはライセンスのアップグレードで、WindowMaker または WindowViewer が猶予期間に入りません。

ライセンスのアップグレードが発生した場合、［**ライセンスを表示**］ダイアログ ボックスに表示されるすべてのパラメータが更新され、高い機能の値が表示されます。

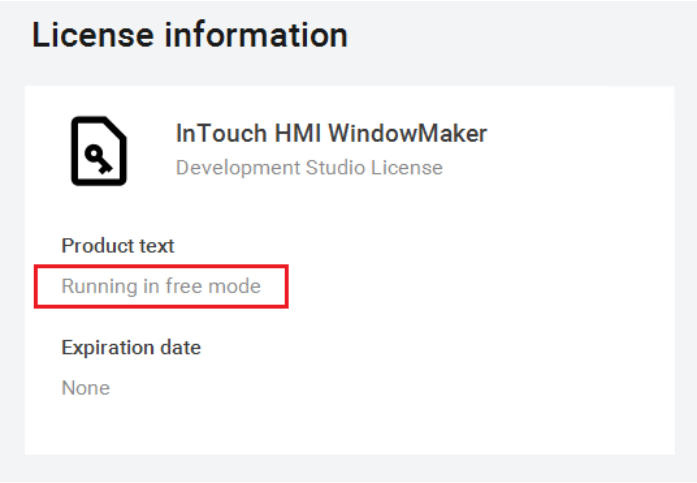
フリーモードとデモモードでの操作

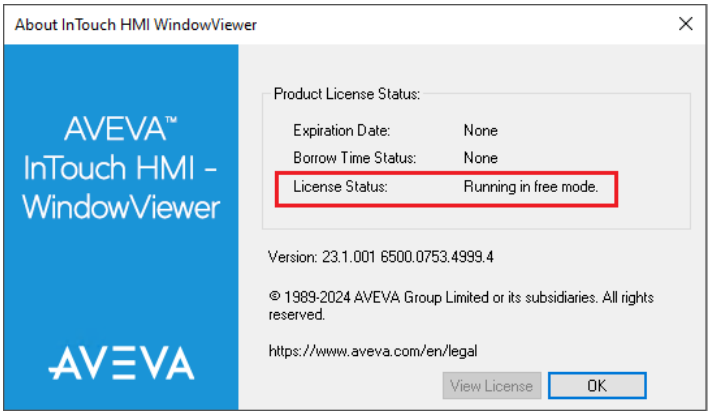
WindowViewer がフリーモードで実行されている間は、Galaxy アクセス名がランタイムから削除されます。以下の場合、フリーモードは利用できません。

- タグが 64 より大きいか、
- アプリケーションが InTouchView アプリケーションとして設定されています

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注意：フリーモードは Operations Control 接続エクスペリエンスでサポートされていません。





デモモードの実行中はタグカウントとウィンドウカウントの上限はありません。ただし、タイムアウトまでに WindowViewer をデモモードで実行できるのは 2 時間のみです。デモモードのタイムアウトに達すると、終了するよう求めるメッセージが表示されます。

注：デモモードでは、有効なライセンスをアクティブ化した場合でも、有効なライセンスを使用するには WindowViewer を終了して WindowViewer を再起動する必要があります。

InTouch.ini ファイルで ViewLicenseRetryCount キーを設定すると、WindowViewer は、パラメータで指定された回数だけバックグラウンドでライセンス取得の試行を続けます。ライセンスが取得された場合、ダイアログが閉じて WindowViewer が起動します。

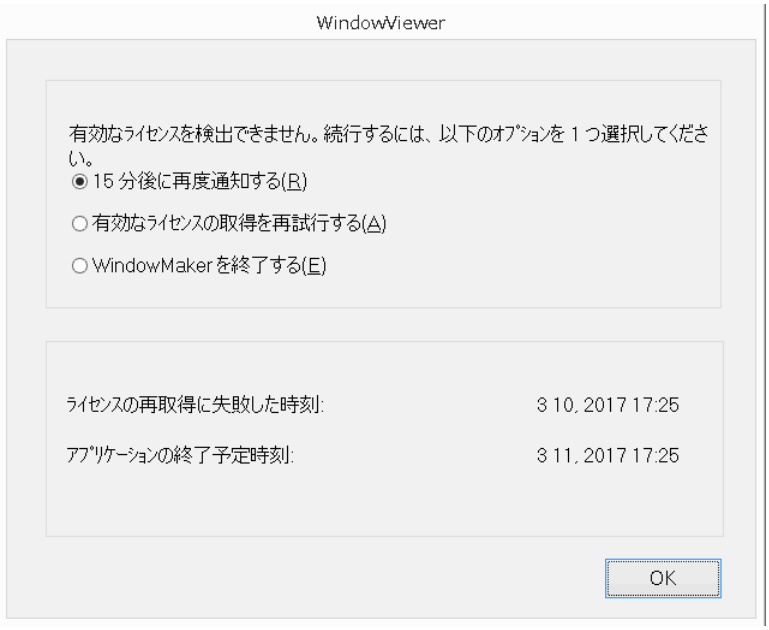
猶予期間での操作

猶予期間は、特定の条件が発生した後に最後の有効なライセンスの機能で InTouch の実行を継続できる期間（24 時間）です。猶予期間は、WindowMaker と WindowViewer の両方で使用できます。猶予期間が終了すると、割り当てられた時間内に有効なライセンスが取得されない場合、InTouch が終了します。

猶予期間は、以下のシナリオによってトリガされます。24 時間以内に有効なライセンスが取得されない場合、WindowMaker または WindowViewer は終了します。

WindowMaker または WindowViewer が猶予期間に入ると、次のダイアログが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



通知を後で再度表示するかライセンスの取得を試みることができます。アプリケーションを終了することもできます。

シナリオ 1：使用していたライセンスが失われた

License Server から取得した有効なライセンスが製品の実行中に無効になった場合、WindowMaker または WindowViewer は猶予期間に入ります。

猶予期間を終了して通常の操作を再開するには、License Server で有効なライセンスを有効にします。ライセンスを使用すると、WindowMaker または WindowViewer の猶予期間が終了し、通常の操作が再開されます。

シナリオ 2：ライセンスが失効された

License Server から取得した有効なライセンスの有効期間が切れた場合、WindowMaker または WindowViewer は猶予期間に入ります。猶予期間を終了して通常の操作を再開するには、License Server で有効なライセンスを有効にします。

WindowMaker または WindowViewer でライセンスを取得できなかった場合、最初のダイアログが再度表示されます。

シナリオ 3：ライセンスがダウングレードされた

AVEVA Enterprise License Manager はサーバー ベースのライセンス システムなので、ライセンスは定期的に更新する必要があります。低い機能のライセンスへの更新中に WindowMaker または WindowViewer ライセンスがダウングレードされた場合、WindowMaker または WindowViewer は猶予期間に入ります。ダウングレード シナリオの詳細については、「起動後における別のライセンスの使用の管理」を参照してください。

猶予期間を狩猟して通常の操作を再開するには、最後の有効なライセンスまたはそれ以上のライセンスを取得してください。

重要：最後の有効なライセンスによって有効化された機能は、猶予期間でも維持されます。

上記のすべてのシナリオにおいて、適切なライセンスが正常に取得された場合、ライセンスが正常に取得されたことを示すメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch でライセンスを取得できなかった場合、最初のダイアログが再度表示されます。

注記： InTouch.ini ファイルで ViewLicenseRetryCount キーを設定すると、WindowViewer は、パラメータで指定された回数だけバックグラウンドでライセンス取得の試行を続けます。ライセンスが取得された場合、ダイアログが閉じて WindowViewer が起動します。

リモートタグを数える関数

InTouch HMI には、アプリケーションがライセンスのリモート タグの要件に準拠しているかどうかを確認する一連の関数が含まれています。生産環境に配置する前に、アプリケーションで考えられるライセンスの問題をテストおよび識別するための関数が含まれている一時スクリプトを記述できます。アプリケーションにライセンス上の問題がないことを確認したら、スクリプトを削除します。

IORRGetSystemInfo() 関数 IORRGetSystemInfo()かんすう

IORRGetSystemInfo() 関数は、実行中の InTouch アプリケーションのタグの数を返します。引数値に基づいて、IORRGetSystemInfo() 関数は以下のような数値を返します。

- InTouch ライセンスによって指定された、リモート タグ アドレスの最大数。
- InTouch アプリケーションの実行中、ライセンスに対してカウントされたリモート タグ アドレスの数。
- InTouch アプリケーションで現在アクティブ化されているリモート タグの数。
- 実行中の InTouch アプリケーションで利用可能なリモート タグの数。
- 現在無効になっているリモート参照タグの数。
- 実行中の InTouch アプリケーション内のローカル タグの数。
- 関数呼び出し中にエラーが発生した場合、または *Option* 引数に無効な値が割り当てられている場合は -1。

カテゴリ

その他

構文

```
IORRGetSystemInfo(Option);
```

引数

オプション
返すリモート参照タグの数のタイプを指定する、整数型タグまたは整数の定数。可能な値は以下のとおりです。

1	InTouch ライセンスに基づいて、許可された最大数のリモート タグ アドレスを返します。ローカル I/O タグは、リモート タグの数には数えられません。 この数は、InTouch アプリケーションの実行中は一定しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2	InTouch アプリケーションの実行中にアクティブ化されている、ライセンス制限を超えた数の独自のリモート タグ アドレスの数を返します。ローカル I/O タグは、リモート タグの数には数えられません。 60K を超えるリモート参照タグがライセンスによって許可された場合、アクティブ化されているリモート タグ アドレスの数にかかわらず、この数は 0 になる場合があります。無制限のライセンス下で実行されている場合、WindowViewer はアクティブ化されているリモート タグ アドレスを数えません。 リモート タグの制限を持つライセンス下でアプリケーションを実行している場合、この数はリモート タグのライセンス数の上限まで増加します。リモート タグの制限に達した後には、追加的なリモート タグ アドレスをさらにアクティブ化することはできません。現在アクティブ化されているアドレスのみ再度アクティブ化できます。ライセンス制限を超えた数のリモート参照タグのリストを取得するには、*Option* 引数を 3 に設定して IORRWriteState を使用します。
3	InTouch アプリケーション内で現在アクティブ化されているリモート参照タグの数を返します。
4	リモート タグのハンドルを使い切ることなく InTouch アプリケーションでアクティブ化できるリモート参照タグの数を返します。この数は通常アプリケーションの実行中に変わります。リモート参照タグが含まれているスクリプトを停止および開始したり、ウィンドウを開いたり閉じたりすると、リモート参照タグの数に影響します。 この数は、特にライセンスが無制限である場合、ライセンスに残っている数より少なくなる場合があります。これは、同時にアクティブ化できるリモート参照タグの数に内部制限があるため発生します。
5	InTouch アプリケーション内で現在無効な状態にあるリモート タグの数を返します。
6	現在 InTouch アプリケーション内にあるローカル タグの数を返します。

例

以下の例は、InTouch アプリケーションの実行中にライセンス制限を超えた数のリモート参照タグの数を返します。返されたリモート参照タグの数は、RRTagCount 整数型タグの値として割り当てられます。

```
RRTagCount = IORRGetSystemInfo(2);
```

IORRWriteState() 関数 IORRWriteState()かんすう

IORRWriteState() 関数により、アプリケーションのリモート タグの現在の状態に関する情報がテキスト ファイルに保存されます。ファイルが存在しない場合、関数によって作成されます。関数が含まれているスクリプトが実行されるたびに、新しい情報がファイルに連結されます。

どのようなリモート タグ情報がファイルに保存されるのかを指定できます。また、関数の戻り値は、情報がファイルに正常に追加されたかどうかを示します。

カテゴリ

その他

構文

```
IORRWriteState(FilePath, Option, " ");
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

FilePath
アプリケーションのリモート タグに関する情報が含まれているテキスト ファイルへの完全フォルダ パス。*FilePath* 引数は、文字列定数またはメッセージ型タグ変数になります。

オプション
リモート タグの数の情報を指定する整数型タグ変数または整数の定数が、ファイルに書き込まれます。可能な値は以下のとおりです。

1	現在のリモート タグ アドレスのリスト。情報には、各リモート タグの状態、アクティブ化された時間、非アクティブ化された時間も含まれます。
2	現在アクティブなすべてのアドレスのリストと、それらがアクティブ化された時間。
3	アクティブ化され、ライセンス制限を超えた数の、すべてのリモート リファレンス タグのアドレスのリスト。 新しいリモート タグ アドレスがアクティブ化されると、新しいアイテムがリストに追加されます。ライセンス制限に達すると、アイテムをリストに追加することができなくなります。 ただし、リモート タグのライセンス制限が無制限である場合、このリストにはアドレスは追加されません。
4	リモート リファレンス タグの数がライセンス制限を超えるため、または内部のタグ変数のハンドル制限に達したため、現在のアドレスのリストがアクティブ化されません。 リモート タグのライセンス制限が無制限である場合、ライセンスに関連するアドレスを返しません。 ライセンスが無制限である場合、使用制限のため、リストには現在アクティブでないすべてのアイテムが含まれます。このリスト内のアイテムがアクティブ化されると、リストから削除されます。ライセンス制限のためにアイテムが非アクティブ化されると、リストから削除されます。このリストは、InTouch アプリケーションの実行中に更新されます。

空の文字列 " "
この引数は今後使用するために予約されていますが、スクリプトでは IORRWriteState() 関数に含める必要があります。

結果

IORRWriteState() 関数呼び出しによってファイルに保存された情報を解釈するためのいくつかの例を以下に示します。

現在のアドレスのリスト

出力ファイルの以下の行では、完全にアクティブ化された、更新可能なリモート リファレンス タグの例を示しています。以下の行は、TestProt:di000 リモート リファレンス タグにアドレス 65535 が割り当てられていることを示しています。

```
65535 <TestProt:di000> (RAA) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

以下のリモート リファレンス タグ名は、括弧で囲まれた 3 つのフラグです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 最初のフラグ R は、タグ変数のリモート リファレンスが正常に解決されたことを示します。タグ変数のリモート リファレンスがまだ保留である場合に、最初のフラグに値 X が割り当てられます。
- 2 番目のフラグは、リモート タグが現在アクティブである場合は A、アクティブでない場合は D を示します。
- 3 番目のフラグは、アドレスが許可されている場合は A、ライセンス制限のために許可されていない場合は D を示します。

C の後ろの日付は、リモート タグが作成された日付を示します。A の後ろの日付は、タグ変数が最近アクティブ化された日付を示します。InTouch アプリケーションの実行中にリモート リファレンス タグが非アクティブ化された場合、非アクティブ化された時間が行の後ろに含まれます。

出力ファイルの以下の行では、InTouch ライセンスのタグ変数の数の制限を超える、アクティブなリモート リファレンス タグの例を示しています。タグ変数のリモート リファレンスが正常に解決され、タグ変数が現在アクティブになっています。

```
65414 <TestProt:di121> (RAD) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

しかし、リモート リファレンス タグに割り当てられたアドレスが、InTouch ライセンスのタグ変数の数の制限を超えるため、 アプリケーション内でタグ変数の値の更新は発生しません。

アクティブなアドレスのリスト

出力ファイルの以下の行では、割り当てられた値によって InTouch アプリケーションが更新される、完全にアクティブ化されたリモート タグの例を示しています。

```
65429 <TestProt:di106> (A) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

最初の数はリモート タグのハンドルであり、その後ろに、アドレス、フラグ、A （許可されている場合）または D （許可されていない場合）が表示されます。出力行のフラグの後ろには、作成日時、最近アクティブ化された日時、および最近非アクティブ化された日時が表示されます。アプリケーションの実行中にリモート タグが非アクティブ化されていない場合、非アクティブ化された時間は表示されません。

出力ファイルの以下の行では、ライセンス制限を超えるアクティブなリモート タグの例を示しています。

```
65342 <TestProt:di193> (D) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

ライセンスされたアドレス

リストの以下の行は、リモート リファレンス タグに割り当てられたアドレス、およびリストに追加された日時を示しています。

```
<testprot:di000> {C:5/23/2007 9:58:36 AM}
```

拒否されたアドレス

使用制限のため、またはタグ変数の数がライセンスの最大数を超えるため、拒否されたアドレスがリストに表示されます。

この例では、ライセンス制限を超えるリモート タグ アドレスを示しています。

```
testprot:di125 [1] (L) {F:5/23/2007 9:58:39 AM} {R:5/23/2007 9:58:39 AM}
```

アドレス、およびアイテムを参照するために試行された回数を示す数がリストに表示されます。フラグはアドレスがリスト内にあるかどうかを示します。ライセンスを超えることが原因である場合は L、または内部の使用制限が原因である場合は I が表示されます。2 つの時間は、リストに最初に追加された時間、および最新のアクセス時間を表します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例では、現在のアクティブ化されているリモート タグ アドレスが、c:\intouch\data フォルダにあるファイルに書き込まれます。ReturnValue タグ変数に整数が割り当てられます。これは、関数呼び出しによってリモート タグ情報がファイルに正常に書き込まれたことを示します。

```
ReturnValue = IORRWriteState("c:\intouch\data", 2, "");
```

IORRGetItemActiveState() 関数 IORRGetItemActiveState()かんすう

IORRGetItemActiveState() 関数は、指定したリモート タグ アドレスのステータスを返します。

カテゴリ

その他

構文

```
IORRGetItemActiveState(ItemPath, Option);
```

引数

ItemPath
ItemPath は対象となるアドレスを表す文字列です。*ItemPath* には、文字列定数またはメッセージ型タグ変数を指定できます。

オプション
返すリモート参照タグの数のタイプを指定する、整数型タグまたは整数の定数。可能な値は以下のとおりです。

1	現在のリモート タグ アドレスが現在アクティブであるかどうかが判断されます。 アドレスが現在のものであり、アクティブな場合、戻り値は 1 です。アドレスが現在のものでない場合、戻り値は -1 です。 アドレスが現在のものであるけれどもアクティブでない場合、戻り値は 0 です。
2	アプリケーションの実行中に、現在のリモート タグ アドレスがアクティブ化されたことがあるかどうかが判断されます。 アドレスが現在のものであり、少なくとも一度アクティブ化されたことがある場合、戻り値は 1 です。アドレスが現在のものでない場合、戻り値は -1 です。アドレスが現在のものであるけれども、アクティブ化されたことがない場合、戻り値は 0 です。
3	現在のリモート タグ アドレスが非アクティブ化されているかどうかが判断されます。 アドレスが現在のものでない場合、戻り値は -1 です。 アドレスが現在のものであるけれども、非アクティブ化されたことがない場合、戻り値は 0 です。
4	現在のリモート タグ アドレスが無効化されているかどうかが判断されます。 アドレスが現在のものであり、少なくとも一度非アクティブ化されたことがある場合、戻り値は 1 です。アドレスが現在のものでない場合、戻り値は -1 です。アドレスが無効化されていない場合、戻り値は 0 です。アドレスが現在のものであり、無効化されている場合、戻り値は 1 です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5	アドレスが許可リストにあるかどうかが判断されます。 アドレスがリストにない場合、戻り値は 0 です。 アドレスがリストにある場合、戻り値は 1 です。
6	アドレスが否認リストにあるかどうかが判断されます。 アドレスがリストにない場合、戻り値は 0 です。 アドレスがリストにある場合、戻り値は 1 です。

例

この例では、TestProt:di000 リモート タグ アドレスが現在アクティブであるかどうかが判断されます。

```
ReturnValue = IORRGetItemActiveState("TestProt:di000", 1);
```

この例では、TestProt:di121 リモート タグ アドレスが現在無効化されているかどうかが判断されます。

```
ReturnValue = IORRGetItemActiveState("TestProt:di121", 4);
```

この例では、TestProt:di001 リモート タグ アドレスが現在ライセンス制限を超えた数であるかどうかが判断されます。

```
ReturnValue = IORRGetItemActiveState("TestProt:di001", 5);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch Web Client のライセンス付与

ログインしてアプリケーション グラフィックを Web ブラウザで表示するには、有効な Web Server ライセンスが必要です。ライセンスは、外部 Web サイトでのアプリケーション グラフィックのホスティングにも適用されます。ライセンスの詳細については、*AVEVA Enterprise Licensing* のヘルプを参照してください。Web Server は、無制限の数のセッションに接続して Web ブラウザでアプリケーション グラフィックの表示および操作を行うことができる次のようなさまざまなタイプのライセンスを提供します。

- InTouch Web Server *Flex ライセンス*
- InTouch Web Server *接続の読み取り/書き込みライセンス*
- InTouch Web Server *無制限の読み取り/書き込みライセンス*
- InTouch Web Server *無制限の読み取り専用ライセンス*

ライセンス ステータスの変更

有効なライセンスを取得した Web Server がライセンスの更新に失敗した場合、Web Client は猶予期間モードになります。通知メッセージが通知ページに表示されます。メッセージは、Web Client によるライセンス更新と失敗が発生するたびにログに記録されます。

InTouch HMI アプリケーションの場合、Web Client WindowViewer に接続して InTouch タグ データを受信するために WindowViewer は読み取り/書き込みライセンスで実行する必要があります。

InTouch アプリケーショングラフィックスのライセンスを取得する

ライセンスの取得は 2 段階のプロセスです。Web Server が起動すると、最初にユーザー認証が行われ、次にユーザーの承認が決定されます。匿名アクセスが有効になっている場合、認証ステップがバイパスされ、InTouch Web Client は「ゲスト」ユーザー用の読み取り専用モードで起動します。

認証

1. InTouch Web Client は Windows 認証をサポートしています。Web Server では、Web Client の使用の認証に使用される "aaInTouchUsers" または "aaInTouchRWUsers" のいずれかにユーザーが含まれているかどうかが検証されます。両方のユーザー グループはインストール時に作成されます。リモート認証サーバーの場合、サーバーにドメイン ユーザー グループを作成する必要があります。
2. Web Client のインストール時のログイン ユーザーは、両方のグループに自動的に追加されます。後でその他のユーザーを追加できます。新しいユーザーをグループに追加した後、変更を適用するには、新しいユーザーはログオフした後に再度ログインする必要があります。

取得

起動時、およびユーザー認証後、Web Server は以下のワークフローでライセンスを取得します。

1. Flex オプションがコンフィグレータで有効化されているかどうかが Web Server によってチェックされます。

   Flex オプションが有効化されていない場合、Web Server は無制限の読み取り/書き込みライセンスの取得を試みます。（ステップ 4 に進む）
2. Flex および Enterprise オプションがコンフィグレータで有効化されている場合、Web Server は Enterprise ライセンスの取得を試みます。

   コンフィグレータで Flex オプションが有効化されていて、Enterprise オプションが有効化されていない場合、Web Server は Flex ライセンスの取得を試みます。（ステップ 3 に進む）

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ライセンスのない Web Server は Flex ライセンスの取得を試みます。

   Flex ライセンスが使用可能な場合、Web Server は Flex ライセンスを取得します。

   Flex ライセンスが使用できない場合、Web Server はライセンスのない状態になります。Web Server にライセンスがないときに Web Client が接続を試みると、ライセンス エラー ページが表示されます。

4. ライセンスのない Web Server は無制限の読み取り/書き込みライセンスの取得を試みます。

   無制限の読み取り/書き込みライセンスが使用可能な場合、Web Server は無制限の読み取り/書き込みライセンスを取得します。

   無制限の読み取り/書き込みライセンスが使用できない場合、Web Server は接続の読み取り/書き込みライセンスの取得を試みます。

5. 接続の読み取り/書き込みライセンスが使用可能な場合、Web Server は接続の読み取り/書き込みライセンスを取得します。

   接続の読み取り/書き込みライセンスが使用できない場合、Web Server は無制限の読み取り専用ライセンスの取得を試みます。

6. 無制限の読み取り専用ライセンスが使用可能な場合、Web Server は無制限の読み取り専用ライセンスを取得します。

   無制限の読み取り専用ライセンスが使用できない場合、Web Server は読み取り専用のシングル セッション モードになります。

ライセンス機能

InTouch Web Client は以下のライセンス機能をサポートしています。

- Flex ライセンス
- 読み取り/書き込みライセンス
- 無制限の読み取り専用ライセンス

各モードの特徴を以下に示します。

取得されたライセンス	サポートされている機能	条件

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Flex ライセンス	• 読み取り/書き込みライセンスでは無制限の接続が許可されます • InTouch と AppServer ネームスペースの両方の個人用作業領域	以下の「注記」を参照してください。
読み取り/書き込みライセンス	• AppServer 属性や InTouch タグなどの外部参照への書き込み • アラームの確認とそのユーザー資格情報は、アラームを確認したオペレータとして記録されます。 • 接続の読み取り/書き込みライセンスでは、Web Server に対する複数の InTouch Web Client 接続（5、10、15 などのセット）が許可されています。	以下の条件が満たされている場合、書き込み機能およびアラームの確認機能にアクセスできます。 • Web Server が Flex、Brand Neutral 読み取り/書き込みライセンス、無制限の読み取り/書き込みライセンス、または接続の読み取り/書き込みライセンスを取得している。 • 許可されている読み取り/書き込み接続内で InTouch Web Client セッションが行われている。 • Web Client 読み取り/書き込みグループに属する InTouch Web Client セッションにユーザーがログインしている。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

読み取り専用ライセンス	• クライアント セッションは外部参照への書き込みやアラームの確認を行うことができません。	ユーザーが読み取り専用モードにアクセスできるのは、次のいずれかの場合だけです。 • Web Server が Flex、Brand Neutral 読み取り/書き込み、無制限の読み取り/書き込みライセンス、または接続の読み取り/書き込みライセンスを取得していて、ログインユーザーが InTouch Web Client aaInTouchRWUsers ユーザーグループに属していない場合 • Web Server で無制限の読み取り専用ライセンスが取得されている場合 • Web Server でライセンスが取得されず、その Web Client セッションが最初のクライアント セッションである場合。

注：Flex ライセンスではフォールバックは許可されていません。Flex ライセンスを取得できない場合、Web Server はライセンスのない状態になります。Web Server にライセンスがない状態で InTouch Web Client が接続を試みると、ライセンスエラーページが表示されます。

シングル セッション モード

起動時にライセンスが取得できない場合、Web Server のはシングル Web セッション モードで動作し、順番に割り当てられます。

- 有効なライセンスがなく、最初のセッション ライセンスが既に取得されている場合、ライセンスが使用できないことを示すメッセージが InTouch Web クライアントに表示されます。
- Web Server で無制限のライセンスが取得された後にライセンスが失われた場合、セッションの猶予期間が設けられます。

Application Server データのサブスクリプションには、Supervisory ライセンスが必要です。Supervisory ライセンスがない場合、Application Server データのサブスクリプションは 2 時間後に終了します。

猶予期間

以下の条件が満たされる場合、Web Server は猶予期間に入ります。

- 1 つの有効なライセンスが取得された後にライセンスの有効期間が終了した。
- 1 つの有効なライセンスが取得された後、そのライセンスを取得できなくなった。

猶予期間中、接続済みおよび新規セッションですべての InTouch Web Client の機能を使用できます。Web Server が猶予期間モードであることを示す通知が Web Client ページに表示されます。猶予期間は 14 日間です。猶予期間が終了した後、Web Server は 1 つのセッション用の単一ライセンスに直ちに切り替わって機能し続けます。この場合、ライセンスは要求順に割り当てられます。同じラインセンスが取得された場合、Web Server の猶予期間モードが終了します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

定期的な更新

有効なライセンスが取得された後、定期的なライセンス更新が行われます。定期的なライセンス更新操作では、同じライセンスを更新できるかどうかが確認されます。同じライセンスを更新できない場合、Web Server は猶予期間モードになります。

更新の頻度は、以下に示すライセンス タイプに依存します。

ライセンス タイプ	更新の頻度
Flex ライセンス 無制限の読み取り/書き込みライセンス 無制限の読み取り専用ライセンス	7 分ごと
接続の読み取り/書き込みライセンス	24 時間ごと

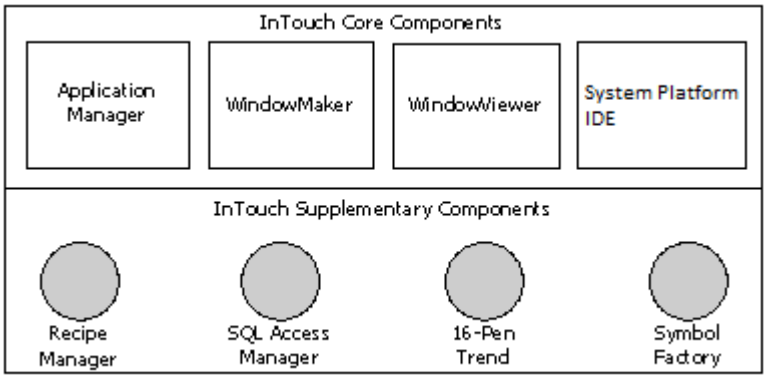
更新中、ライセンス ステータスのアップグレードやダウングレードは行われません。別のライセンスを取得するには、Web Server を再起動して、ライセンス取得の起動シーケンスを開始します。

ライセンスの解放

Web Server がシャットダウンされた場合、クライアント セッションの数や Web Server ライセンスの現在の状態に関係なく、Web Server で取得されたすべてのライセンスが解放されます。すべての Web Client も直ちに終了します。

補足コンポーネントについて

オプションで、一連の 4 つの補足コンポーネントを InTouch HMI コア コンポーネントと共にインストールすることもできます。これらの補足コンポーネントにより、InTouch HMI アプリケーションの追加的な機能が提供されます。



- レシピ マネージャには、製造レシピを作成するための一連のスプレッドシートおよびスクリプト関数が含まれています。
- SQL アクセス マネージャは、データベースに InTouch データを保存するためのプログラムと一連の SQL 関数で構成されます。

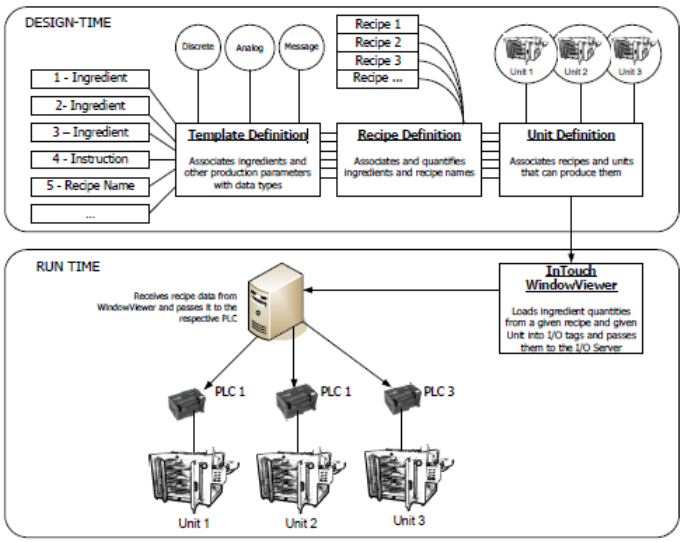
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 16 ペン トレンドには、リアルタイム トレンドおよび履歴トレンドを作成するための、トレンド ウィザードおよびスクリプト関数が含まれています。
- Symbol Factory では、一連の産業用シンボルが提供されており、プロセス コンポーネントを表すために InTouch アプリケーションに配置することができます。

レシピ マネージャの使用

製造業者は、標準の数量の原料を使用する繰り返し可能なプロシージャに従って製品を製造します。つまり、製品はレシピに従って製造されます。レシピは原料、その数量、およびそれらをどのように組み合わせて最終的な製品を製造するかについて説明しています。最もわかりやすい例として、たとえばベーカリーではクッキーを作るためのすべての原料と手順を一覧表示した基本レシピに従って製造を行っています。

レシピ マネージャは、製造レシピを作成するプロセスを簡略化するために使用できる InTouch HMI 用の補足コンポーネントです。以下の図は、製品を作成するプロセスを管理するためにレシピ マネージャがレシピ テンプレートから情報を取得する方法を示しています。



レシピ マネージャの概要

レシピ マネージャは InTouch と共にオプション コンポーネントとしてインストールできます。レシピ マネージャは、レシピ マネージャ ユーティリティと InTouch レシピ スクリプト関数のセットで構成されます。

レシピ マネージャ ユーティリティには、WindowMaker から起動するか、C:\Program Files (x86)\Wonderware\InTouch\ から Recipe.exe を起動することによってアクセスできます。レシピ マネージャ ユーティリティにはレシピ テンプレートを作成および編集するためのインターフェイスが含まれています。レシピ マネージャはテンプレートをレシピ ファイルに保存します。

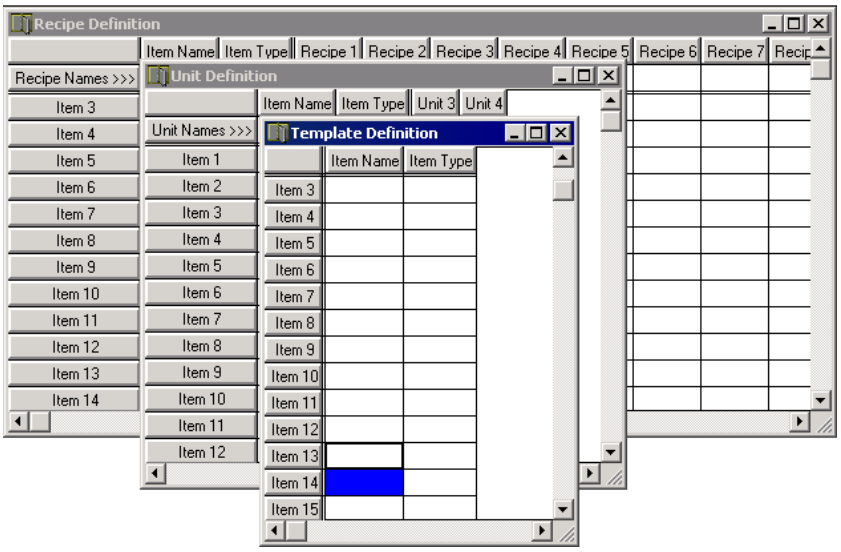
一般的に、製造プロセスに関連付けられているタグ変数は QuickScript を使用してレシピ テンプレート ファイル内のデータにアクセスします。レシピ マネージャは、テンプレート ファイルに含まれている製造レシピを選択、ロード、変更、作成、および削除するための QuickScript 関数のセットを含みます。

レシピ マネージャ ユーティリティ

レシピ マネージャ ユーティリティは、レシピ テンプレート ファイルを作成および維持するためのスプレッドシートに似たユーザー インターフェイスを提供します。ファイルは 3 つのテンプレートで構成

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

されています。これらのテンプレートは、各テンプレートのスプレッドシートのセルにあるデータを追加または変更することによって作成または編集できます。



これらのテンプレートは、カンマ区切り値（CSV）ファイルに保存されます。レシピ テンプレート定義は、メモ帳または Excel など .csv ファイル形式をサポートする任意のプログラムで作成または編集できます。ただし、レシピ マネージャでは、テンプレートを確実に簡単に作成および維持するための書式設定済みスプレッドシートおよび編集ツールのセットを提供しています。

レシピ テンプレート ファイル

レシピ マネージャ テンプレート ファイルには、以下の情報が含まれています。

- レシピで使用される原料の名前およびそのデータ タイプ。
- InTouch タグ変数をレシピ原料値と関連付けるユニット名。
- レシピ インスタンスで使用される各原料の数量または値を含むレシピ名。

テンプレート定義

テンプレート定義テンプレートは、すべてのレシピ原料を定義します。データ タイプは各レシピ原料と関連付けられています。すべての原料データ タイプは、アナログ型、論理型、またはメッセージ型のいずれかです。原料名は InTouch タグ変数である必要はありません。

Unit Definition

ユニット定義テンプレートは、InTouch タグ変数をレシピ原料に関連付けます。多くの異なるロード定義を作成できます。これらの定義はユニットと呼ばれます。RecipeLoad() 関数を使用すると、レシピの特定のインスタンスを関連付けられている InTouch タグ変数にロードできます。ユニット定義は、ファイルに定義されているすべての原料またはこれらの原料のサブセットで構成されます。

注意：ユニット タグ変数は、InTouch ウィンドウで表示、編集できるメモリ型であるか、ＰＬＣ に直接ロードできる I/O タグ変数のいずれかです。

Recipe Definition

レシピ定義テンプレートは、各レシピの名前およびレシピで使用される原料数量を指定します。レシピ インスタンスは、ランタイムでレシピ関数を使用して変更、作成、または削除できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

レシピ マネージャでのレシピ データの編集

製造レシピは、一連の連続タスクを完了することによって作成します。以下のリストは、レシピを作成するレシピ マネージャのタスクおよびタスクが完了する順番を示しています。

- レシピ マネージャの編集グリッドの設定
- テンプレートのデータの編集
- テンプレート定義テンプレートへの原料名およびユニットタイプの割り当て
- ユニット定義テンプレートの原料への InTouch タグ変数のマッピング
- レシピ定義テンプレート内のレシピ原料への値の割り当て

レシピ マネージャの編集グリッドの設定

製造レシピを作成する前に、レシピ マネージャを設定する必要があります。レシピ マネージャ編集関数を設定するには、2 つのタスクがあります。

- テンプレート アイテムの最大制限を設定します。
- Enter キー スクロール関数を設定します。

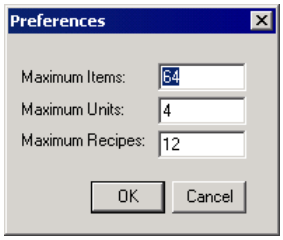
レシピを作成する前に、レシピ テンプレートに入力できるアイテムの最大数を設定する必要があります。アイテム、ユニット、およびレシピ名の最大制限セットを割り当てる必要があります。

テンプレートは最大で 9999 のアイテム、ユニット、およびレシピ名を含むことができます。ただし、最大制限数が大きくなると、システムのパフォーマンスに影響が与えられる可能性があります。また、設定した最大制限がコンピュータの使用可能メモリよりも多くのメモリを必要とした場合にはエラーメッセージが表示されます。

レシピ テンプレートの最大制限を設定する

1. 以下のいずれかの方法で、レシピ マネージャを起動します。
   - WindowMaker を起動する。［**ツール**］ビューで、［**アプリケーション**］を展開し、［**レシピ マネージャ**］を選択します。

   ［レシピ マネージャ］ ダイアログ ボックスが表示されます。
2. ［**オプション**］メニューで、［**システム設定**］を選択します。［環境設定］ダイアログ ボックスが表示されます。



3. ［**最大アイテム数**］ボックスに、テンプレート定義テンプレートで許容されているアイテム名の最大数を入力します。
4. ［**最大ユニット数**］ボックスに、ユニット定義テンプレートで許容されているユニットの最大数を入力します。
5. ［**最大レシピ数**］ボックスに、レシピ定義テンプレートで許容されているレシピ名の最大数を入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

警告 :［**システム設定**］ダイアログ ボックスに設定した値は、作成するすべてのレシピ テンプレート ファイルに適用されます。これらの値を変更すると、既存のレシピ テンプレート ファイルもすべて変更されます。

1. ［**OK**］ を選択します。

レシピ マネージャには、レシピ テンプレートへのデータ入力を簡潔化するオプションが含まれています。［**入力後セル移動**］オプションを選択した場合、Enter キーを押すとカーソルがカラム内の次のセルに移動します。

Enter キー テンプレート スクロールを設定する

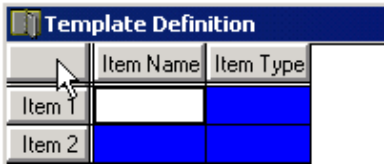
1. レシピ マネージャを開きます。

   デフォルトでは、レシピ マネージャはテンプレート スプレッドシート内の次のセルには自動的にはスクロールしません。

2. ［**オプション**］メニューで、［**入力後セル移動**］を選択して、セル スクロールを設定します。
3. セル スクロールをオフにする場合は、［**入力後セル移動**］を再び選択します。

編集グリッドの操作

レシピ マネージャには、テンプレートからのデータを追加、変更、または削除する編集コマンドのセットが含まれます。一般に、テンプレートで編集するデータを選択して、編集作業を行います。以下の表には、テンプレート データを入力して選択するためのレシピ テンプレートの一般的な機能が説明されています。

機能	ディスクリプション
入力ボックス	選択したテンプレート セル用のデータを入力するために使用するテキスト入力ボックス。テンプレート セルが選択されると、その内容が［レシピ マネージャ］ダイアログ ボックスの上部付近のテキスト入力ボックスに表示されます。
すべてのセルの選択	すべてのセルを選択するには、テンプレートの左上のセルを選択します。 
すべての行の選択	行内のすべてのセルを選択するには、テンプレートの行名を選択します。
すべてのカラムの選択	カラム内のすべてのセルを選択するには、テンプレートのカラム見出しを選択します。
すべてのカラムの自動サイズ調節	テンプレートのすべてのカラムを最も長いエントリの幅に自動的にサイズ変更するには、テンプレートをダブルクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

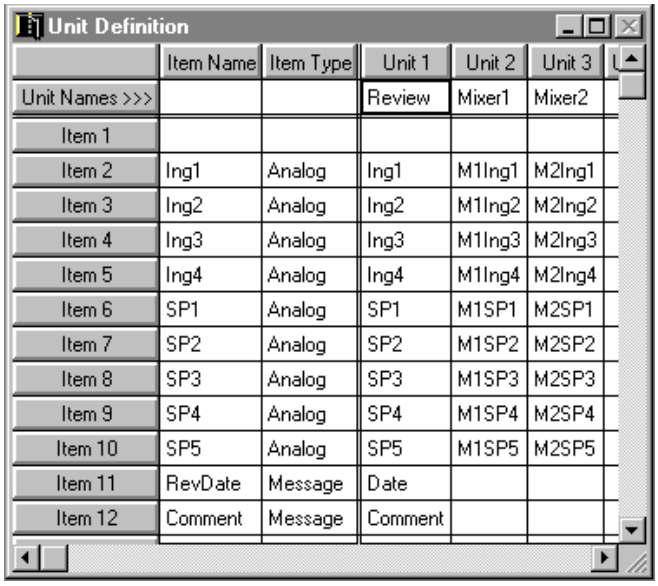
機能	ディスクリプション
カラムの自動サイズ調節	カラムを最も長いエントリの幅にサイズ変更するには、見出しをダブルクリックします。 **注記:**テンプレート定義テンプレート内のデータ タイプ カラムは自動サイズ変更できません。

テンプレートを編集するときには、以下の手順を実行します。

- 一定範囲のセルからデータを消去する
- 一定範囲のセルからのデータを近接する選択した一定範囲のセルへコピーする
- **テンプレート定義**テンプレートに行を挿入する
- テンプレート内にカラムを挿入する
- 行を**テンプレート定義**テンプレートから削除する
- テンプレートからカラムを削除する

一定範囲のセルからデータを消去する

1. テンプレートから一定範囲のセルを選択します。

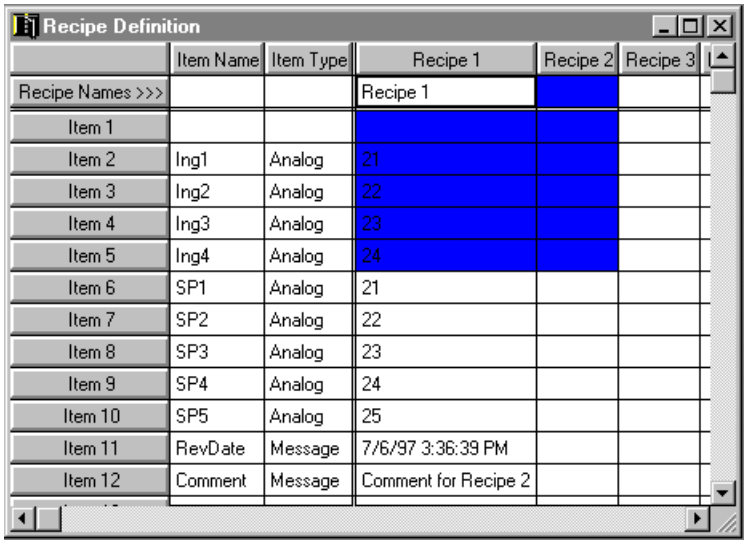


2. ［**編集**］メニューの［**クリア**］を選択します。選択した一定範囲のセルがクリアされることを確認するよう促すメッセージが表示されます。
3. ［**はい**］を選択します。テンプレートによって選択した一定範囲のセルからデータがクリアされます。

一定範囲のセルを近接の選択範囲へコピーする

1. コピーするセルまたは一定範囲のセルを選択します。
2. データをコピーする近接の一定範囲のセルを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



	Item Name	Item Type	Recipe 1	Recipe 2	Recipe 3
Recipe Names >>>			Recipe 1		
Item 1					
Item 2	Ing1	Analog	21		
Item 3	Ing2	Analog	22		
Item 4	Ing3	Analog	23		
Item 5	Ing4	Analog	24		
Item 6	SP1	Analog	21		
Item 7	SP2	Analog	22		
Item 8	SP3	Analog	23		
Item 9	SP4	Analog	24		
Item 10	SP5	Analog	25		
Item 11	RevDate	Message	7/6/97 3:36:39 PM		
Item 12	Comment	Message	Comment for Recipe 2		

選択した範囲は同じサイズであり、選択したオリジナルのセルの範囲の上、下、右、または左である必要があります。

3. ［編集］メニューで、適切な塗りつぶしコマンドを選択します。データが選択した範囲のセルにコピーされます。

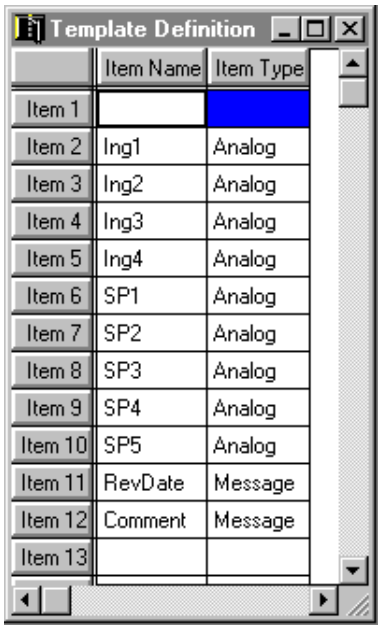
注記：データがコピーされる新しいカラムが最大エントリを含むのに十分な大きさではない場合、カラムの見出しをダブルクリックして、幅を最も長いエントリに合わせて変更します。

テンプレート定義テンプレートに行を挿入する

1. **テンプレート定義**テンプレートを選択します。
2. **アイテム番号**を選択して、テンプレート定義テンプレートの行を選択して、新しい行をその上に挿入します。

   レシピ定義テンプレートまたは**ユニット定義**テンプレートでは、直接行を挿入することはできません。その代わり、**テンプレート定義**へのすべての変更は**レシピ定義**テンプレートおよび**ユニット定義**テンプレートによって自動的に継承されます。
3. ［編集］メニューで、［挿入］を選択します。新しい行が選択した行のすぐ上に挿入され、後続のすべての行の番号は自動的に変更されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



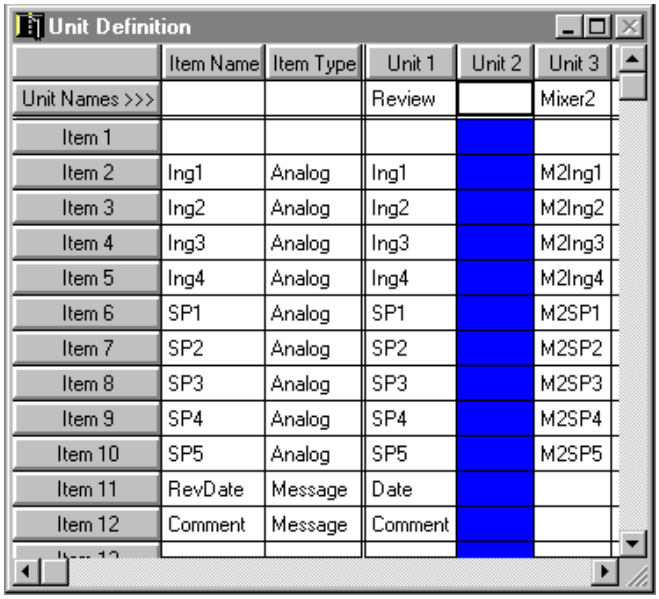
注記：レシピ マネージャの［環境設定］に設定された最大値に達した場合、［**挿入**］コマンドは非アクティブになります。レシピ テンプレートにアイテム／ユニット／レシピを追加するには、指定した数値を増加させる必要があります。
システム設定を変更すると、変更は既存のすべてのレシピ テンプレート ファイルに適用されます。

列の挿入

1. **ユニット番号**または**レシピ番号**を選択して、挿入したカラムの右側に配置するカラムを選択します。

   カラムはレシピ定義テンプレートまたはユニット定義テンプレートに挿入できます。

2. ［**編集**］メニューで、［**挿入**］を選択します。新しいカラムが選択したカラムの左側に挿入されます。



この例では、**Mixer2** データが［**ユニット 3**］カラムに移動され、新しいカラムが［**ユニット 2**］として挿入されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カラムを削除する

1. **ユニット番号**または**レシピ番号**カラム見出しを選択して、削除するカラムを選択します。

   レシピ定義テンプレートまたはユニット定義テンプレートからカラムを削除できます。

2. ［**編集**］メニューで、［**削除**］を選択します。削除を確認する確認メッセージ ダイアログ ボックスが表示されます。
3. ［**はい**］を選択します。カラムがテンプレートから削除され、残りのカラムの番号が変更されます。

行の削除

1. **テンプレート定義**テンプレートを選択します。

   テンプレート定義テンプレートからは行を削除できますが、**レシピ定義**テンプレートまたは**ユニット定義**テンプレートからは行を削除することはできません。

2. ［**アイテム番号**］行見出しを選択して、削除する行を選択します。
3. ［**編集**］メニューで、［**削除**］を選択します。削除を確認する確認メッセージ ダイアログ ボックスが表示されます。
4. ［**はい**］を選択します。テンプレートから行が削除されます。

原料名とデータ タイプの定義

テンプレート定義テンプレートには、各原料に関連付けられているレシピ原料およびアイテム タイプが一覧表示されています。データをその他のレシピ テンプレートに追加する前に、テンプレート定義テンプレートを完了する必要があります。

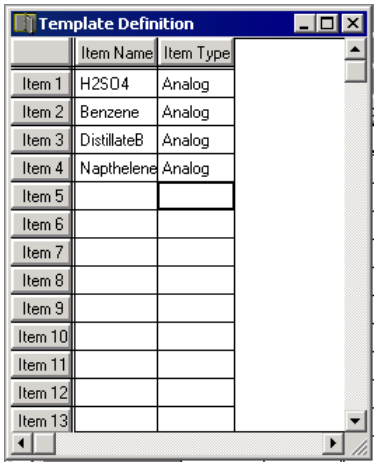
テンプレート定義テンプレートを定義するには

1. レシピ マネージャを起動します。
2. ［ファイル］メニューで、［**新規**］を選択します。3 つのレシピ マネージャ テンプレートが表示されます。
3. ［**テンプレート定義**］タイトル バーを選択して、テンプレート ウィンドウを選択します。
4. ［**アイテム名**］カラム セルに、レシピ原料に選択した名前を入力します。

   各セルには 1 つの原料のみを入力できます。

5. ［**アイテム タイプ**］カラム セルに、それぞれのレシピ原料に対する有効なアイテム タイプを入力します。

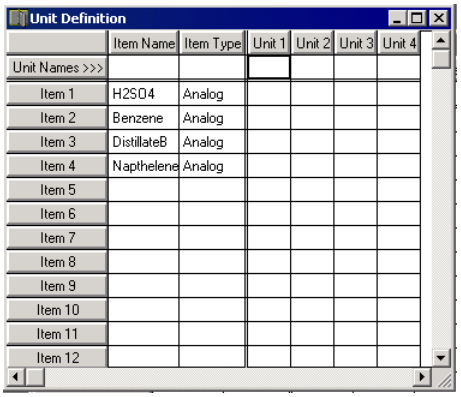
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



有効なアイテム タイプは、アナログ型、論理型、またはメッセージ型です。アナログ型の場合は A、論理型の場合は D、メッセージ型の場合は M を入力します。**Enter** キーを押すと、レシピ マネージャによって残りのアイテム タイプが自動的に完了されます。

原料への InTouch タグのマッピング

ユニット定義テンプレートは、InTouch タグを特定のユニットのレシピ原料に関連付けます。次の図に示すように、**ユニット定義**テンプレートの最初の 2 つのカラムには、**テンプレート定義**テンプレートからのアイテム名およびアイテム タイプが一覧表示されています。



ユニットに定義されているタグは、PLC データを I/O サーバーから取得するメモリ型タグまたはリモート タグとなります。

RecipeLoad() 関数を InTouch QuickScript で使用する場合、ユニット名を指定する必要があります。レシピ名定義に含まれる値は、QuickScript が実行されると、ユニット名に指定されているタグにロードされます。

ユニット定義テンプレートを定義するには

1. ［**ユニット定義**］テンプレートのタイトル バーを選択して、テンプレート ウィンドウを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Unit Definition

	Item Name	Item Type	Unit 1	Unit 2	Unit 3
Unit Names >>>			Review	Mixer 1	Mixer 2
Item 1	Ing1	Analog	Ing1	M1Ing1	M2Ing1
Item 2	Ing2	Analog	Ing2	M1Ing2	M2Ing2
Item 3	Ing3	Analog	Ing3	M1Ing3	M2Ing3
Item 4	Ing4	Analog	Ing4	M1Ing4	M2Ing4
Item 5	SP1	Analog	SP1	M1SP1	M2SP1
Item 6	SP2	Analog	SP2	M1SP2	M2SP2
Item 7	SP3	Analog	SP3	M1SP3	M2SP3
Item 8	SP4	Analog	SP4	M1SP4	M2SP4
Item 9	SP5	Analog	SP5	M1SP5	M2SP5
Item 10	RevDate	Message	Date		
Item 11	Comment	Message	Comment		

2. ［**ユニット名**］行に、定義する各ユニットの名前を入力します。
3. ［**ユニット番号**］カラム セルで、以下のいずれかの方法を使用して、それぞれ対応するレシピ原料に InTouch タグの名前を入力します。
   - タグ名を入力します。
   - WindowMaker を実行している場合は、セルをダブルクリックして、［**タグの選択**］ダイアログ ボックスを表示します。次に、セルに挿入する目的のタグ変数をダブルクリックするか、選択して、［**OK**］を選択します。
4. それぞれのユニット／レシピの組み合せに対しても同じ手順を繰り返します。

異なるレシピ内の原料の値を定義

レシピ定義テンプレートは、各レシピの名前およびレシピで使用される原料数量を指定します。レシピ定義テンプレートは、以前定義されたテンプレート定義テンプレートからのアイテム名およびデータ タイプ情報を表示します。

原料値は、**RecipeLoad()** 関数が InTouch QuickScript 内で実行されると InTouch タグ変数内にロードされます。

レシピ定義テンプレートを定義する

1. ［**レシピ定義**］テンプレートのタイトル バーを選択して、テンプレート ウィンドウを選択します。
2. ［レシピ名］行に、定義する各レシピの名前を入力します。

Recipe Definition

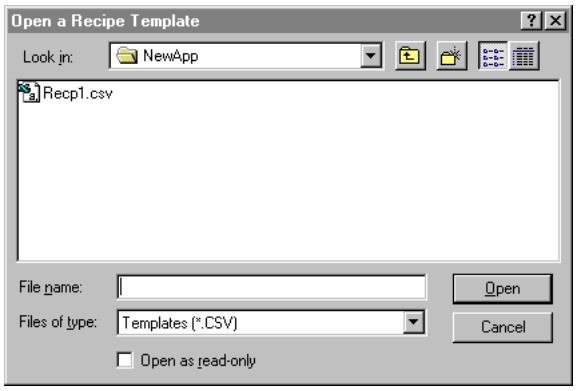
	Item Name	Item Type	Recipe 1	Recipe 2
Recipe Names >>>			Recipe 1	Recipe 2
Item 1	Ing1	Analog	11	21
Item 2	Ing2	Analog	12	22
Item 3	Ing3	Analog	13	23
Item 4	Ing4	Analog	14	24
Item 5	SP1	Analog	11	21
Item 6	SP2	Analog	12	22
Item 7	SP3	Analog	13	23
Item 8	SP4	Analog	14	24
Item 9	SP5	Analog	15	25
Item 10	RevDate	Message	7/15/97 3:19:56 PM	7/6/97 3:36:39 PM
Item 11	Comment	Message	A Comment	Comment for Recipe 2

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［**レシピ番号**］カラム セルに、［アイテム名］カラム内の対応する各レシピ原料の値を入力します。
4. ［**ファイル**］メニューで、［**上書き保存**］を選択して、レシピ テンプレート ファイルを保存します。

既存のレシピ テンプレート ファイルを開く

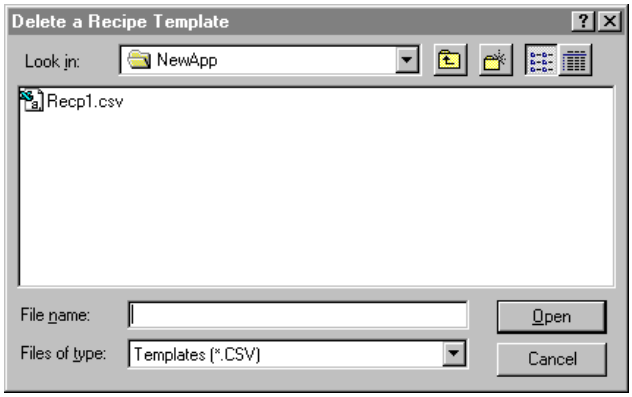
1. **レシピ マネージャ**を開きます。
2. ［**ファイル**］メニューで、［**開く**］を選択します。［**レシピ テンプレートを開く**］ダイアログ ボックスが表示されます。



3. レシピ ファイルを検索して選択し、［**開く**］を選択します。ファイルの 3 つのレシピ テンプレートがレシピ マネージャに表示され、編集できます。

レシピ テンプレート ファイルを削除する

1. ［**ファイル**］メニューで、［**削除**］を選択します。［**レシピ テンプレートを削除**］ダイアログ ボックスが表示されます。



2. レシピ ファイルを検索して選択し、［**開く**］を選択するか、またはファイル名をダブルクリックします。削除を確認するメッセージが表示されます。

注記：開いているレシピ テンプレート ファイルは削除できません。

3. ［**はい**］を選択して、ファイルを削除します。

別のアプリケーションでのレシピ データの編集

レシピ テンプレート定義は、カンマ区切りデータをサポートする任意のプログラムで作成および編集できます。Microsoft Excel またはメモ帳を使用して、レシピ マネージャ テンプレート ファイルを作成および編集できます。

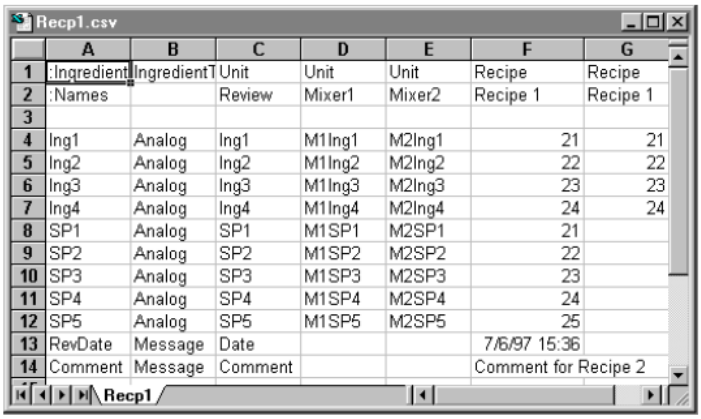
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Excel でのレシピ テンプレート ファイルの使用

レシピ マネージャ ユーティリティを使用しない場合は、Excel を使用してレシピ テンプレートを作成または編集できます。Excel で作成または編集したレシピ マネージャ テンプレートは、.csv ファイル名拡張子を付けたファイルに保存する必要があります。

Microsoft Excel で既存のレシピ テンプレート ファイルを開く

1. Excel を起動します。
2. ［ファイル］メニューで、［開く］を選択します。［開く］ダイアログ ボックスが表示されます。
3. .csv ファイルを検索して選択し、［開く］を選択するか、ファイル名をダブルクリックします。Excel にファイルの内容が表示されます。

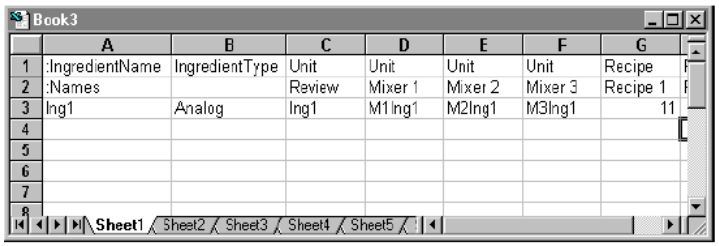


	A	B	C	D	E	F	G
1	:Ingredient	IngredientT	Unit	Unit	Unit	Recipe	Recipe
2	:Names		Review	Mixer1	Mixer2	Recipe 1	Recipe 1
3							
4	Ing1	Analog	Ing1	M1Ing1	M2Ing1	21	21
5	Ing2	Analog	Ing2	M1Ing2	M2Ing2	22	22
6	Ing3	Analog	Ing3	M1Ing3	M2Ing3	23	23
7	Ing4	Analog	Ing4	M1Ing4	M2Ing4	24	24
8	SP1	Analog	SP1	M1SP1	M2SP1	21	
9	SP2	Analog	SP2	M1SP2	M2SP2	22	
10	SP3	Analog	SP3	M1SP3	M2SP3	23	
11	SP4	Analog	SP4	M1SP4	M2SP4	24	
12	SP5	Analog	SP5	M1SP5	M2SP5	25	
13	RevDate	Message	Date			7/6/97 15:36	
14	Comment	Message	Comment			Comment for Recipe 2	

4. レシピ ファイルの内容を編集して、変更内容を保存します。

Excel で新しいレシピ テンプレート ファイルを作成する

1. Excel を起動します。
2. 新しいワークブックを作成します。
3. 次の図に示すように、スプレッドシートにレシピ データを入力します。



	A	B	C	D	E	F	G
1	:IngredientName	IngredientType	Unit	Unit	Unit	Unit	Recipe
2	:Names		Review	Mixer 1	Mixer 2	Mixer 3	Recipe 1
3	Ing1	Analog	Ing1	M1Ing1	M2Ing1	M3Ing1	11
4							
5							
6							
7							
8							

エントリは、図に示されている順番で作成する必要があります。ユニット名はレシピ名を含むカラムの左にあるカラムで定義する必要があります。

4. スプレッドシートを .csv ファイル名拡張子を付けて保存します。

メモ帳でのレシピ テンプレート ファイルの使用

レシピ マネージャ ユーティリティを使用しない場合は、メモ帳を使用してレシピ テンプレートを作成または編集できます。メモ帳で作成または編集したレシピ マネージャ テンプレートは、.csv ファイル名拡張子を付けたファイルに保存する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

メモ帳で既存のレシピ テンプレート ファイルを開く

1. メモ帳を起動します。
2. ［ファイル］メニューで、［開く］を選択します。［開く］ダイアログ ボックスが表示されます。
3. レシピ ファイルを検索して選択し、［開く］を選択するか、またはファイル名をダブルクリックします。
4. レシピ ファイルの内容を編集して、変更内容を保存します。

メモ帳で新しいレシピ テンプレート ファイルを作成する

1. メモ帳を起動します。
2. ［ファイル］メニューで、［新規］を選択します。
3. 次のデータをこの形式で入力します。

```
:IngredientName,IngredientType[,Unit]...[,Recipe]...
:Names,,[,UnitName]...[,RecipeName]...
IngredientName,{Analog,Discrete,Message},[,tag]...[,value]
```

注記：すべてのユニット名は、レシピ名が定義される前にファイル内に定義する必要があります。

4. ファイルを .csv ファイル名拡張子を付けて保存します。

複雑な構造を作成するためのレシピのネスト

複数のレシピ テンプレート ファイルを InTouch QuickScript を使用して相互にリンクして、複雑なアプリケーションを作成できます。レシピ テンプレート ファイルは、ユニット名テンプレート内のメッセージ型タグ変数に関連付けられている原料名を定義することによってリンクします。次に、レシピの名前を持つメッセージ型タグ変数をロードします。

レシピ テンプレート ファイルをリンクすると、異なるレシピ ファイルの様々なレシピによって使用されるマシン設定パラメータを定義するプライマリ レシピ テンプレート ファイルを作成できます。このようなタイプの情報を 1 つの中央ファイルに保持すると、変更するたびにデータを維持および更新するための時間を大幅に削減できます。

次の図では、アイテム名 Setup タグ変数がメッセージ型として定義され、ユニットにはこのアイテム用の Setup メッセージ型タグ変数が含まれます。各レシピには、レシピが選択されたときに Setup タグ変数にロードされる別のレシピ ファイルに 2 番目のレシピ名が定義されます。

RECFILEA.CSV

	1	2	3	4	5	6	7	8	9
1	:Item Name	Item Type	Unit	Unit	Unit	Unit	Recipe	Recipe	Recipe
2	:Names		**Review**	Mixer 1	Mixer 2	Mixer 3	Recipe 1	**Recipe 2**	Recipe 3
3	Ing1	Analog	Ing1	M1Ing1	M2Ing1	M3Ing1	11	21	31
4	Ing2	Analog	Ing2	M1Ing2	M2Ing2	M3Ing2	12	22	99
5	Ing3	Analog	Ing3	M1Ing3	M2Ing3	M3Ing3	13	23	66
6	Ing4	Analog	Ing4	M1Ing4	M2Ing4	M3Ing4	14	24	34
7	SP1	Analog	SP1	M1SP1	M2SP1	M3SP1	11	21	31
8	SP2	Analog	SP2	M1SP2	M2SP2	M3SP2	12	22	32
9	SP3	Analog	SP3	M1SP3	M2SP3	M3SP3	13	23	33
10	SP4	Analog	SP4	M1SP4	M2SP4	M3SP4	14	24	34
11	SP5	Analog	SP5	M1SP5	M2SP5	M3SP5	15	25	35
12	**Setup**	Message	**Setup**	LinkFile	LinkFile	LinkFile	Setup2A	**Setup3A**	Setup1A
13									

この処理を実行するには、次のスクリプトを入力します。

```
RecipeName="Recipe2";
RecipeLoad("c:\recipe\recfilea.csv", "Review", RecipeName);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

このスクリプトを実行すると、**Setup** タグ変数の値が Setup3A となり、Review ユニットにロードされます。次のスクリプトを実行することによって、Setup タグ変数の値が、マシン セットアップ パラメータを PLC1 ユニットに対して定義されているタグ変数にロードする次のレシピ ロードのレシピ名として使用されます。

```
RecipeLoad("c:\recipe\machine.csv", "PLC1", Setup);
```

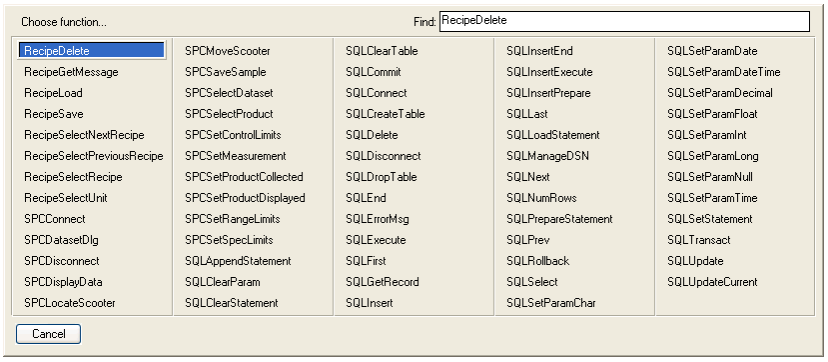
MACHINE.CSV

	1	2	3	4	5	6	7	8
1	:Item Name	Item Type	Unit	Recipe	Recipe	Recipe		
2	:Names		PLC1	Setup1A	Setup2A	Setup3A		
3	PARM1	Analog	PARM1	11	21	31		
4	PARM2	Analog	PARM2	12	22	99		
5	PARM3	Analog	PARM3	13	23	66		
6	PARM4	Analog	PARM4	14	24	34		
7	PARM5	Analog	PARM5	11	21	31		
8	PARM6	Analog	PARM6	12	22	32		
9	PARM7	Analog	PARM7	13	23	33		
10	PARM8	Analog	PARM8	14	24	34		
11	PARM9	Analog	PARM9	15	25	35		
12								

InTouch でのレシピの使用

InTouch QuickScript を使用して、レシピ テンプレート ファイルと対話します。レシピ マネージャには、QuickScript に挿入できる一組のスクリプト関数が含まれます。これらの関数を含むスクリプトを使用すると、レシピ テンプレート ファイルのレコードを選択、変更、挿入、または削除できます。

レシピ関数は、InTouch スクリプト エディタを使用して、任意のタイプのスクリプトに追加できます。次の図は、レシピ関数を一覧表示する InTouch スクリプト エディタのウィンドウを示しています。すべてのレシピ関数は、関数名の一部として Recipe プリフィックスによって識別されます。



InTouch レシピ関数は、レシピ テンプレート ファイルに直接読み込みしたり、書き込みしたりします。したがって、レシピ関数が InTouch QuickScript で適切に実行されるようにするために、レシピ マネージャ プログラムを実行する必要はありません。

レシピ テンプレート ファイルが InTouch HMI によって使用されている場合、作成する新しいレシピまたは既存のレシピへの変更はレシピ テンプレート ファイルには書き込まれません。レシピ マネージャは、レシピ テンプレート ファイルのみを作成します。レシピ テンプレート ファイルを作成した後は、レシピ マネージャを閉じます。

レシピ関数をスクリプトに自動的に挿入する

1. InTouch スクリプト エディタを開きます。
2. スクリプト内でレシピ関数を挿入する場所にカーソルを置きます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［関数］で、［アドオン］を選択します。 ［関数の選択］ダイアログ ボックスが表示されます。
4. QuickScript に挿入するレシピ関数を選択します。ダイアログ ボックスが閉じ、関数がカーソル位置に挿入されます。

レシピ ファイルに対するレシピ データのロードと保存

レシピ マネージャは、レシピ ファイル内のレシピ データをロードおよび保存する別の InTouch QuickScript 関数を提供しています。

RecipeLoad() 関数

RecipeLoad() 関数は、レシピからのデータを InTouch アプリケーションの特定のタグ変数ユニットにロードします。

カテゴリ

レシピ

構文

```
RecipeLoad("Filename","UnitName","RecipeName");
```

引数

FileName
レシピ テンプレート ファイルの名前。FileName に関連付けられる値は、文字列定数またはレシピ テンプレート ファイルの名前を含むメッセージ型タグ変数となります。

UnitName
指定したレシピ テンプレート ファイル内の特定ユニットの名前。RecipeSelectUnit() 関数は、このパラメータに値を返します。UnitName に関連付けられる値は、文字列定数またはユニットの名前を含むメッセージ型タグ変数です。

RecipeName
指定したレシピ テンプレート ファイルの特定レシピの名前。*RecipeName* に関連付けられる値は、文字列定数またはレシピの名前を含むメッセージ型タグ変数です。

例

次のステートメントは、Recipe1 レシピの値を Unit に定義されているタグ変数にロードします。

```
RecipeLoad("c:\recipe\recfile.csv", "Unit1", "Recipe1");
```

RecipeSave() 関数

RecipeSave() 関数は、新しいレシピまたは既存のレシピに行った変更を指定したレシピ テンプレート ファイルに保存します。

カテゴリ

レシピ

構文

```
RecipeSave("Filename","UnitName","RecipeName");
```

引数

FileName
レシピ テンプレート ファイルの名前。FileName に関連付けられる値は、文字列定数またはレシピ テンプレート ファイルの名前を含むメッセージ型タグ変数となります。

UnitName

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

関数によって使用される指定したレシピ テンプレート ファイルの特定ユニットの名前。RecipeSelectUnit() 関数は、このパラメータに値を返します。UnitName に関連付けられる値は、文字列定数またはユニットの名前を含むメッセージ型タグ変数です。

RecipeName
指定したレシピ テンプレート ファイルの特定レシピの名前。*RecipeName* に関連付けられる値は、文字列定数またはレシピの名前を含むメッセージ型タグ変数です。

例

次の例は、Recipe3 レシピに行われた変更を recfile.csv ファイルに保存します。Recipe3 が現在 recfile.csv ファイルに存在しない場合は、作成されます。値は、Unit2 に対して定義されたタグ変数の値を設定します。

```
RecipeSave("c:\recipe\recfile.csv","Unit2", "Recipe3");
```

レシピ ファイルからのレシピの削除

RecipeDelete 関数を使用して、指定したレシピ テンプレート ファイルからレシピを削除します。

RecipeDelete() 関数

RecipeDelete 関数は、指定したレシピ テンプレート ファイルからレシピを削除します。

カテゴリ

レシピ

構文

```
RecipeDelete("Filename","RecipeName");
```

引数

FileName
レシピ テンプレート ファイルの名前。FileName に関連付けられる値は、文字列定数またはレシピ テンプレート ファイルの名前を含むメッセージ型タグ変数となります。

RecipeName
指定したレシピ テンプレート ファイルの特定レシピの名前。*RecipeName* に関連付けられる値は、文字列定数またはレシピの名前を含むメッセージ型タグ変数です。

例

次のステートメントは、Distlt1 レシピを recfile.csv ファイルから削除します。

```
RecipeDelete("c:\recipe\recfile.csv", "Distlt1");
```

ユニットの選択（タグ変数原料マッピング）

RecipeSelectUnit() 関数を使用して、現在のレシピ値がロードされるタグ変数のユニットを選択します。

RecipeSelectUnit() 関数

RecipeSelectUnit() 関数は、ランタイム ユーザーがユニットを選択できるように、［**ユニットの選択**］ダイアログ ボックスを開きます。選択したユニット名はメッセージ型タグに返されます。RecipeSelectRecipe() 関数および RecipeSelectUnit() 関数はどちらも RecipeLoad() 関数と共に使用されます。

カテゴリ

レシピ

構文

```
RecipeSelectUnit("Filename","UnitName",Number);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

FileName
レシピ テンプレート ファイルの名前。FileName 引数は、文字列定数またはレシピ テンプレート ファイルの名前を含むメッセージ型タグとなります。

UnitName
選択したユニットの名前を書き込む先のメッセージ型タグです。引用符または文字列リテラルのない実際のメッセージ型タグ。

数値
引数に返される最大文字列長。InTouch では、文字列（メッセージ型）タグ変数の最大文字数は 131 文字です。InTouch タグ変数の最大文字列長を削減していない限り、この引数には 131 を使用します。数値または整数型タグ変数。

例

以下のステートメントを実行すると、［**ユニットの選択**］ダイアログ ボックスが表示されます。

```
RecipeSelectUnit("c:\recipe\recfile.csv", UnitName, 131);
```

ユニットが選択された後に、その名前が UnitName タグ変数に返されます。

レシピ ファイルからの個別レシピの選択

レシピ マネージャには、個別のレシピをレシピ ファイルから選択するための関数セットが含まれています。これらの関数をスクリプトで使用する場合、ファイルから名前によって、またはファイル内のシーケンス内の次のレシピまたは前のレシピによって、レシピを選択できます。

RecipeSelectRecipe() 関数

RecipeSelectRecipe() 関数は、ランタイム ユーザーがレシピを選択できるように、［**レシピの選択**］ダイアログ ボックスを開きます。選択したレシピ名はメッセージ型タグに返されます。

カテゴリ

レシピ

構文

```
RecipeSelectRecipe("Filename", RecipeName, Number);
```

引数

FileName
レシピ テンプレート ファイルの名前。FileName 引数は、文字列定数またはレシピ テンプレート ファイルの名前を含むメッセージ型タグとなります。

RecipeName
選択したレシピの名前を書き込む先のメッセージ型タグです。引用符なしまたは文字列リテラルのない実際のメッセージ型タグ。

数値
引数に返される最大文字列長。InTouch メッセージ型タグの最大文字数は 131 文字です。InTouch タグの最大文字列長を削減していない限り、このパラメータには 131 を使用します。数値または整数型タグ。

例

以下のステートメントを実行すると、［レシピの選択］ダイアログ ボックスが表示されます。

```
RecipeSelectRecipe("c:\recipe\recfile.csv", RecipeName, 131);
```

ダイアログ ボックスからレシピが選択すると、名前が **RecipeName** タグ変数に返されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

RecipeSelectNextRecipe() 関数

RecipeSelectNextRecipe() 関数は、レシピ テンプレート ファイルの次のレシピを選択します。

カテゴリ

レシピ

構文

```
RecipeSelectNextRecipe("Filename", "RecipeName", Number);
```

引数

FileName
レシピ テンプレート ファイルの名前。FileName 引数は、文字列定数またはレシピ テンプレート ファイルの名前を含むメッセージ型タグとなります。

RecipeName
開始点として使用するレシピ名（関数の実行前）、および選択したレシピ名（関数の実行後）を格納するメッセージ型タグです。引用符なしまたは文字列リテラルのない実際のメッセージ型タグ。

数値
引数に返される最大文字列長。InTouch では、文字列（メッセージ型）タグ変数の最大文字数は 131 文字です。InTouch タグの最大文字列長を削減していない限り、このパラメータには 131 を使用します。数値または整数型タグ変数。

例

次のステートメントは、タグ変数 RecipeName の現在の値を読み取って、ファイルの次のレシピを返します。RecipeName の値が空白であるか、見つからない場合は、ファイルの最初のレシピが返されます。RecipeName にファイルの最後のレシピ名が含まれている場合は、変更されないまま返されますレシピは作成された順番でレシピ テンプレート ファイルに保存されます。

```
RecipeSelectNextRecipe("c:\recipe\recfile.csv",
RecipeName, 131);
```

RecipeSelectPreviousRecipe() 関数

RecipeSelectPreviousRecipe() 関数は、レシピ テンプレート ファイルに定義されている前のレシピを選択します。

カテゴリ

レシピ

構文

```
RecipeSelectPreviousRecipe("Filename","RecipeName",Number);
```

引数

FileName
レシピ テンプレート ファイルの名前。FileName 引数は、文字列定数またはレシピ テンプレート ファイルの名前を含むメッセージ型タグとなります。

RecipeName
開始点として使用するレシピ名（関数の実行前）、および選択したレシピ名（関数の実行後）を格納するメッセージ型タグです。引用符なしまたは文字列リテラルのない実際のメッセージ型タグ。

数値

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

パラメータに返される最大文字列長。InTouch では、メッセージ型タグの最大文字数は 131 文字です。InTouch タグの最大文字列長を削減していない限り、このパラメータには 131 を使用します。数値または整数型タグ。

例

以下のステートメントを実行すると、システムがタグ RecipeName の現在の値を読み取り、ファイルの前のレシピ名を返します。返されたこの文字列は、RecipeName に保存され、現在の値を上書きします。RecipeName の値が空白か、または見つからない場合は、ファイルの最後のレシピが返されます。RecipeName に現在ファイルの最初のレシピ名が含まれている場合は、変更されないまま返されますレシピは作成された順番で保存されます。

```
RecipeSelectPreviousRecipe("c:\recipe\recfile.csv", RecipeName, 131);
```

レシピ スクリプト関数によって返されるエラー メッセージについて

レシピ アプリケーションは、レシピ関数によって返される診断エラー コードを使用してトラブルシューティングします。このセクションでは、レシピ関数エラー コードのリストを記載し、RecipeGetMessage() 関数を使用してエラー コードと関連付けられているメッセージを表示する方法について説明しています。

RecipeLoad() 関数は、正常に行われた場合はアナログ型 ErrorCode タグ変数の値を 0 に設定します。RecipeLoad() が失敗した場合は、ErrorCode タグ変数に特定のエラー状態の番号を設定します。

レシピ関数のエラー コードを取得するには、InTouch アナログ型タグ変数と同等にする必要があります。次の例は、レシピ関数エラー コードを返すためのスクリプト ステートメントを示しています。

```
ErrorCode = RecipeLoad(FileName, UnitName, RecipeName);
```

エラー コード メッセージの表示

各レシピ関数は、関数のエラー状態を表す番号を返します。RecipeGetMessage() 関数を InTouch データ変化スクリプトで使用することによって、対応するエラー コードがアナログ型タグに書き込まれ、関連付けられているエラー コード メッセージがメッセージ型タグに書き込まれます。

次のコード例は、データ変化スクリプトを示しています。

```
RecipeGetMessage(ErrorCode, ErrorMessage, 131);
```

このスクリプトは、ErrorCode タグの値が変更されるたびに自動的に実行されます。このスクリプトが実行されると、RecipeGetMessage() 関数が ErrorCode タグの現在の数値を読み取り、その値に関連付けられているメッセージを ErrorMessage タグに返します。

以下の表には、発生する可能性のあるエラー コードおよびそれに対応するエラー メッセージとその説明が一覧表示されています。

値	エラーメッセージ	説明
0	成功	呼び出されたレシピ関数は正常に実行されました。
-1	そのようなレシピ テンプレートはありません	指定したレシピ テンプレート ファイルが存在しません。
-2	WindowViewer が起動されていません	別のプログラムによって呼び出されたレシピ関数は、WindowViewer が実行されていないため、実行できません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値	エラーメッセージ	説明
-3	メモリ不足	現在のアクティビティを完了するために十分なメモリがありません。
-4	レシピ内の行が長すぎます	レシピ テンプレート ファイルの行が最大許容長を超えています。
-5	レシピ ファイル内に切り取られた行があります	レシピ テンプレート ファイル内の行が切り取られています。
-6	このレシピ テンプレート ファイルは無効です	指定したファイルが有効なレシピ テンプレート ファイルではありません。
-7	ユニット名またはレシピ名を入力してください	レシピ テンプレート ファイルからユニット名またはレシピ名が見つかりません。
-8	レシピ テンプレート ファイルでユニットが定義されていません	レシピ ファイルのユニット定義テンプレートにユニットが定義されていません。
-9	レシピ テンプレート ファイル内にレシピ名が見つかりません	指定したレシピがレシピ テンプレート ファイルで定義されていません。
-10	レシピ テンプレート ファイル内にユニット名が見つかりません	指定したユニット名がユニット定義テンプレート ファイルで定義されていません。
-12	アナログ／論理値／メッセージの型を入力してください	レシピ テンプレート ファイルのアイテムに間違ったタイプが入力されました。有効なタイプは、アナログ型、論理型、またはメッセージ型です。
-13	タグのタイプがアナログ／論理値／メッセージと照合しません	指定したタグがアイテム タイプと一致しません。たとえば、レシピ アイテムはアナログ型として定義されていて、メッセージ型タグがそのユニットとして定義されています。
-14	不正な論理値です。0 または 1 を入力してください	レシピ テンプレート ファイルの論理型タグに対して間違った値が入力されました。論理型タグの有効な値は 0 または 1 のみです。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

値	エラーメッセージ	説明
-15	一時ファイルを開くことができません	空きディスク容量が不十分であるために、一時ファイルを開くことができません。
-16	レシピ テンプレート ファイルを保存中に書き込みエラーが発生しました	レシピ テンプレート ファイルを保存中にエラーが発生しました。
-17	ユーザーを選択できませんでした	ユーザーが［レシピの選択］ダイアログ ボックスでレシピ名でなく、［キャンセル］を選択しました。
-19 ページの「SmartSymbol テンプレートの名前の変更」を参照してください。	別のアプリケーションでレシピ テンプレートが使用されているか、ビューを実行しているユーザーにファイルを開くための十分なセキュリティ権限がありません。	指定されたレシピ テンプレート ファイルが開いているか、ユーザーに十分な権限がないため、WindowViewer ではアクセスできません。

RecipeGetMessage() 関数

RecipeGetMessage() 関数は（他のレシピ関数によって返された）エラー番号を取り、そのエラー番号のプレーン テキストのエラー メッセージを返します。

カテゴリ

レシピ

構文

```
RecipeGetMessage(Analog_Tag,Message_Tag,Number);
```

引数

Analog_Tag
エラー メッセージを取得するエラー番号です。

messagetag
引用符または文字列リテラルのない実際のメッセージ型タグ。

数値
Number 引数は、*Message_Tag* 引数で返される文字列の最大長さを設定します。デフォルトでは、InTouch メッセージ型タグの最大文字数は 131 文字に設定されます。InTouch タグ変数ディクショナリの *Message_Tag* の最大文字列長を削減していない限り、このパラメータには 131 を使用します。*Number* 引数は、定数または数字を含む整数型タグ変数です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

RecipeGetMessage() 関数を InTouch データ変化 QuickScript で使用することによって、エラー コードが ErrorCode タグに書き込まれ、関連付けられているエラー コード メッセージが ErrorMessage タグに書き込まれます。

```
Data Change Script Tagname[.field]:ErrorCode
Script body:RecipeGetMessage(ErrorCode, ErrorMessage,131);
```

この QuickScript は、ErrorCode タグ変数の値が変更されるたびに自動的に実行されます。この QuickScript が実行されると、RecipeGetMessage() 関数が **ErrorCode** タグ変数の現在の数値を読み取り、その値に関連付けられているメッセージを **ErrorMessage** タグ変数に返します。

```
ErrorCode = RecipeLoad ("c:\App\recipe.csv","Unit1","cookies");
RecipeGetMessage(ErrorCode, ErrorMessageTag, 131);
```

レシピへのセキュリティの適用

レシピへのアクセスは、レシピをロード、保存、または削除するために必要な最小セキュリティ アクセス レベルを設定するレシピ テンプレート ファイルにアイテム名を定義することによって管理できます。

次のファイル サンプルでは、SecurityLevel アイテム名がアナログ型タグ変数として定義されています。Review ユニットには、このアイテムの SecurityLevel アナログ型タグ変数が含まれます。各レシピは、レシピが Review ユニットにロードされるときに SecurityLevel タグ変数にロードされる値を定義します。

MACHINE.CSV

	1	2	3	4	5	6	7
1	:Item Name	Item Type	Unit	Unit	Recipe	Recipe	Recipe
2	:Names		**Review**	PLC1	Setup1A	Setup2A	**Setup3A**
3	PARM1	Analog		PARM1	11	21	31
4	PARM2	Analog		PARM2	12	22	99
5	PARM3	Analog		PARM3	13	23	66
6	PARM4	Analog		PARM4	14	24	34
7	PARM5	Analog		PARM5	11	21	31
8	PARM6	Analog		PARM6	12	22	32
9	PARM7	Analog		PARM7	13	23	33
10	PARM8	Analog		PARM8	14	24	34
11	PARM9	Analog		PARM9	15	25	35
12	**SecurityLevel**	Analog	**SecurityLevel**		2000	5000	**7000**

選択したレシピに対してセキュリティ アクセス レベルが無効であるときに表示されるアクセス拒否メッセージを含むウィンドウを作成できます。この操作を実行するには、選択したレシピのセキュリティ レベル値が検証用にロードされるアナログ型タグ変数のみを含むユニットに、選択したレシピがロードされる必要があります。

以下に例を示します。

```
RecipeSelectRecipe("c:\recipe\machine.csv",MyRecipe, "131");
```

［レシピの選択］ダイアログ ボックスが表示されます。レシピ名を選択すると、RecipeName タグ変数に返され、スクリプトが実行を続行します。

```
RecipeLoad( "c:\Recipe\Machine.csv", "Review", MyRecipe );
IF SecurityLevel <= $AccessLevel THEN
   Status =RecipeLoad( "c:\Recipe\Machine.csv", "PLC1", MyRecipe );
   ELSE
   Show "アクセスが拒否されました";
ENDIF;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

このスクリプトが実行されると、アクセス レベルが 7000 に等しいかそれより大きい場合、選択したレシピの値は PLC1 ユニットのタグ変数にロードされます。それ以外の場合は、［アクセスが拒否されました］ウィンドウが表示され、レシピは PLC1 にロードされません。

InTouch からの SQL データベースの操作

データベースは、共通の属性またはフィールドを共有するテーブルに情報を保存します。SQL（Structured Query Language）は、クエリー形式の情報にアクセスするために使用する言語です。SQL アクセス マネージャを使用すると、クエリーを含むデータベース テーブルへのアクセス、変更、作成、および削除を行うことができます。

SQL アクセス マネージャは、InTouch と共にインストールできるオプションのプログラムです。SQL アクセス マネージャでは以下のことを行うことができます。

- 複雑なクエリーを作成および実行するこれらのクエリーは、動的に構築することや、外部ファイルに保存することができます。また、これらのクエリーにはランタイム時でクエリーに渡されるパラメータを含めることができます。
- データベースでサポートされている SQL ステートメントを実行して、クエリーから結果を取得する。SQL アクセス マネージャでストアド プロシージャを使用することもできますが、すべてのストアド プロシージャ関数が完全にサポートされているわけではありません。

SQL 関数は、QuickScript エディタ ダイアログ ボックスの［アドオン］ボタンを選択することによって、自動的に InTouch QuickScript に挿入できます。SQL 関数はスクリプトの現在のカーソル位置に自動的に挿入されます。

SQL アクセス マネージャを使用して、バッチ レシピなどのデータを SQL データベースから InTouch アプリケーションへ転送できます。SQL アクセス マネージャは、ランタイム データ、アラーム ステータス、または履歴データを InTouch からデータベースに転送するために使用することもできます。たとえば、マシン サイクルが完了した後で、会社は異なるアプリケーション用に複数のデータ セットを保存しておきたい場合があります。SQL データベースは、情報を 1 つまたは複数のサードパーティ アプリケーション間で簡単に転送できるようにする機能を提供しています。SQL アクセス マネージャは、このデータにアクセスして、任意の InTouch アプリケーションで表示できるようにします。

SQL アクセス マネージャは、プログラムおよび一組の SQL 関数で構成されます。SQL アクセス マネージャ プログラムは、データベース カラムを作成して、InTouch タグ変数に関連付けます。データベース カラムと InTouch データベース タグ変数の関連付けプロセスは、バインディングと呼ばれます。InTouch データベース タグ変数をデータベース カラムにバインドすると、SQL アクセス マネージャがデータベースに保存されている InTouch データを直接操作できます。

SQL アクセス マネージャは、データベース フィールド名とその関連付けを SQL.DEF と呼ばれるカンマ区切り変数ファイルに保存します。このファイルは InTouch アプリケーション フォルダに常駐し、SQL アクセス マネージャまたはメモ帳などの任意のテキスト エディタを使用して、表示および修正できます。SQL アクセス マネージャは、InTouch で使用されるデータベースの構造および形式を定義するテーブル テンプレートも作成します。

SQL 関数はスクリプトで使用して、オペレータの入力、タグ変数値の変更に基づいて、または特定の条件セットが存在する場合に自動的に実行できます。これらの関数を使用すると、アクセスするために選択したテーブルのレコードを選択、変更、挿入、または削除できます。たとえば、アラーム条件が存在する場合、アプリケーションはすべての適用可能なデータ ポイントとアラームの状態を保存する SQLInsert() 関数または SQLUpdate() 関数を含むスクリプトを実行できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データ ソースの設定

SQL アクセス マネージャは、ODBC 互換アプリケーションであり、利用できる ODBC ドライバまたは OLE DB プロバイダをサポートする任意のデータベースと通信します。

データベースへの通信文字列は以下の方法で設定できます。

- Microsoft ODBC アドミニストレータ プログラムを使用して、ODBC ドライバを SQL アクセス マネージャで使用するように設定します。
- SQLConnect() 関数を実行して、OLE DB プロバイダに引数値を指定します。詳細については、「SQL Server データベース アプリケーション」を参照してください。
- データベース接続文字列を外部 UDL ファイルに設定します。

ODBC ドライバを設定する

1. Microsoft ODBC アドミニストレータ プログラムを実行します。
2. ドライバまたはデータ ソースを選択して、［**新しい名前の追加**］、［**既定値に設定**］または［**設定**］を選択します。［**ODBC ドライバ セットアップ**］ダイアログ ボックスが表示されます。

オプション	説明
データ ソース名	データ ソースを識別するユーザー定義の名前
説明	データ ソースのユーザー定義の説明
データベース ディレクトリ	データベース ファイルを含むフォルダを識別します。指定されていない場合は、現在の作業フォルダが使用されます。

3. ［**OK**］ を選択します。

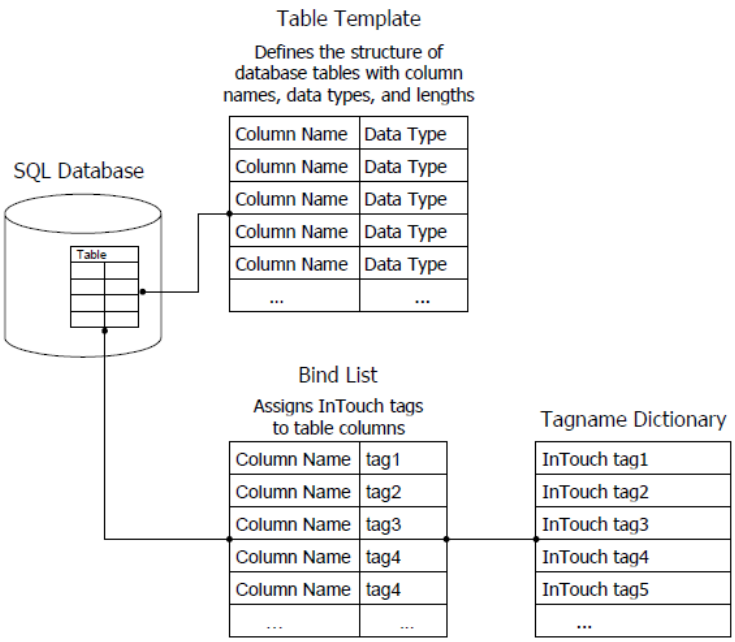
注記：ODBC データ ソースを作成すると、ODBC.INI ファイルが Windows ディレクトリに作成されます。ODBC.INI ファイルは手動で編集できます。

データベース カラムへの InTouch タグのマッピング

InTouch タグをデータベース カラムにマップできます。これはバインド リストを使用して実行できます。ほとんどの SQL Access 関数ではバインド リストを使用して、InTouch タグを有効にして SQL データベース テーブルのデータにアクセスできるようにします。

バインド リストはデータベース テーブル カラムを InTouch タグ名ディクショナリのタグに関連付けます。バインド リストには、データベース テーブルの形式を説明するテーブル テンプレートも含まれます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



SQLInsert()、SQLSelect()、または SQLUpdate() 関数を含むスクリプトを実行すると、バインド リストが更新されて、どの InTouch タグが使用されるか、どのテーブル カラムがこれらのタグと関連付けられるかが指定されます。

新しいバインド リストを作成する

1. ［ツール］ペインで［**SQL アクセス マネージャ**］を展開して［**バインド リスト**］を選択します。

   SQL.DEF ファイルの作成を確認するメッセージが表示されます。

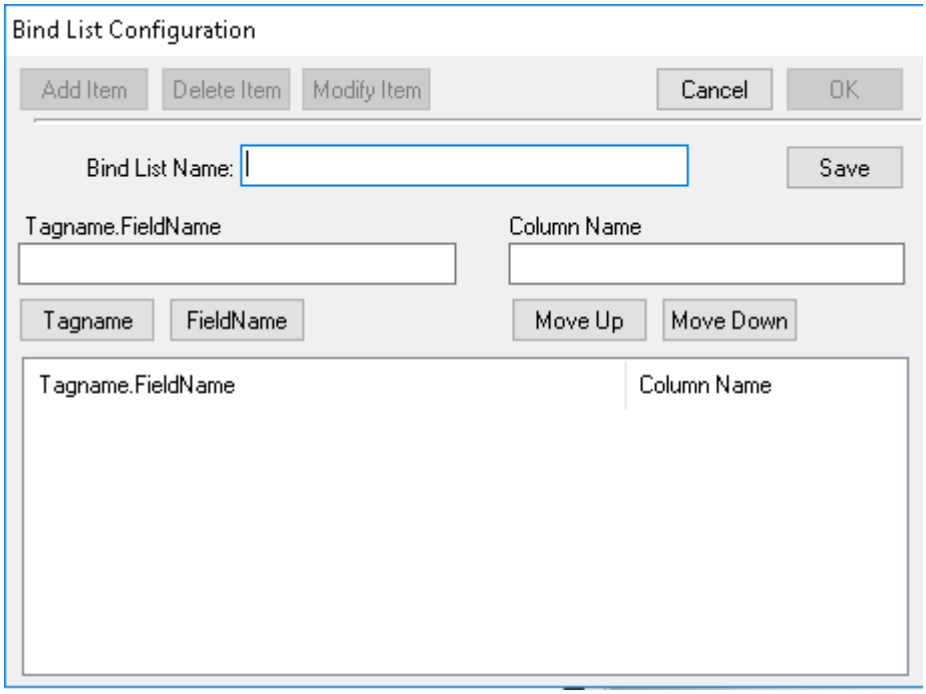
2. ［はい］を選択して、SQL.DEF ファイルを作成します。

   ［**バインド リストを選択してください**］ダイアログ ボックスが表示されます。

3. ［**新規作成**］を選択します。

   ［**バインド リストの定義**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**バインド リスト名**］ボックスに、バインド リスト名を入力します。

   バインド リスト名は最大 32 文字まで使用できます。

5. バインド リストのタグを定義するには、以下のいずれかを実行します。
   - ［**タグ名.フィールド名**］ボックスに、InTouch タグ名を入力します。オプションのタグ ドットフィールドを *tag_name.dotfield_name* の形式で追加できます。
   - ［**タグ名**］をダブル選択して、既存のタグ変数を選択します。［**タグの選択**］ダイアログ ボックスが表示されます。リストからタグを選択します。

   注記：アプリケーションで使用されていないものの、SQL アクセス バインド リストに指定されている I/O タイプ タグは、WindowViewer が起動するとすぐに有効になります（DAServer に通知）。

6. 以下のいずれかを実行して、タグに追加するドットフィールドを選択します。
   - ［**タグ名.フィールド名**］ボックスに、タグ名の後にピリオドとドットフィールド名を入力します。
   - 「**フィールド名**］を選択します。［フィールド名の選択］ダイアログ ボックスが表示されます。タグに追加するドットフィールドを選択します。

7. ［**カラム名**］ボックスに、カラムの名前を入力します。

   カラム名の長さは、最大 30 文字までです。カラム名にスペースがある場合は、バインド リスト内およびスクリプトで使用される場合にはカラム名を角括弧で囲みます。以下に例を示します。

```
WHERE EXPR= "[Valve ID] = " + text (tagname,"#");
```

8. バインド リストにタグを配置するには、以下のいずれかを実行します。
   - 選択したタグ変数をリストで 1 つ上のレベルに移動するには、［**上へ移動**］を選択します。
   - 選択したタグ変数をリスト内で 1 つ下のレベルに移動するには、［**下へ移動**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

9. ［**アイテムの追加**］を選択して、新しい［タグ名.フィールド名］および［カラム名］をバインド リストに追加します。
10. ［**OK**］を選択して、新しいバインド リスト設定を保存して、ダイアログ ボックスを閉じます。

バインド リストの SQL Server 文字列区切り文字の設定

SQLInsert() 関数および SQLUpdate() 関数は、メッセージ文字列を一重引用符で囲んだデフォルト形式を使用します。ただし、SQL データベースの中には、別のタイプの区切り文字で囲まれたメッセージ文字列を予期しているものがあります。たとえば、Oracle 8.0 では括弧で囲まれた日付文字列を受け取ることを予期しています。この場合は、Delim() 関数を使用する必要があります。

［バインド リストの定義］ダイアログ ボックスの［カラム名］フィールドに、カラム名の後で delim() 関数を使用します。キーワード "delim" の後には、以下のように入力する必要があります。

- 左括弧
- 左区切り文字
- システム［地域と言語のオプション］で定義されているのリスト区切り文字
- 右区切り文字
- 右括弧

英語システムの例: `datestring delim (',')`

ドイツ語システムの例: `datestring delim (';')`

左と右の両方に同じ区切り文字を使用するには、以下の例のように区切り文字を区切り文字なしで括弧内に指定します。

```
日付文字列デリム('')
```

バインド リストを変更する

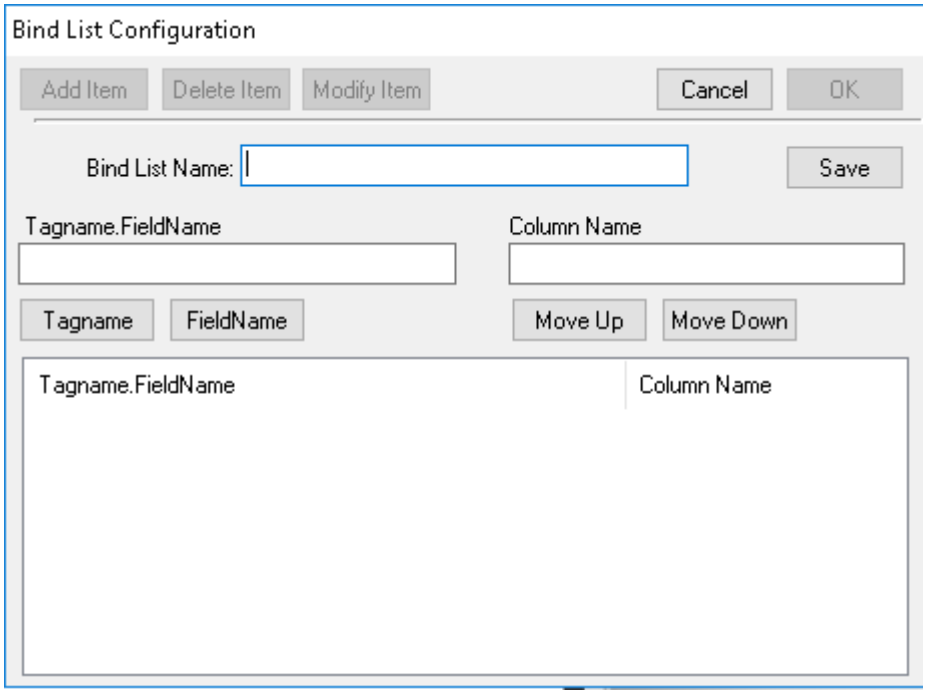
1. ［**ツール**］ペインで［**SQL アクセス マネージャ**］を展開して［**バインド リスト**］を選択します。

   SQL.DEF ファイルの作成を確認するメッセージが表示されます。
2. ［**はい**］を選択して、SQL.DEF ファイルを作成します。

   ［**バインド リストを選択してください**］ダイアログ ボックスが表示されます。
3. 変更するバインド リスト名を選択して、［**修正**］を選択します。

   ［**バインド リストの定義**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. 必要なアイテムを変更します。
5. ［**OK**］を選択して変更を保存し、ダイアログ ボックスを閉じます。

バインド リストを Excel で変更する

SQL アクセス マネージャは､バインド リストの設定情報およびテーブル テンプレートを SQL.DEF ファイルに保存します。このファイルは、カンマ区切り値（CSV）ファイルの形式となっています。

SQL.DEF ファイルは、Excel などのカンマ区切り値ファイルをサポートする任意の製品を使用して変更できます。

データはファイルで、以下のように表示されます。

```
:BindListName, BindListName
Tagname1.FieldName,ColumnName1
Tagname2.FieldName,ColumnName2
Tagname3.FieldName,ColumnName3
:TableTemplateName, TableTemplateName
ColumnName1,ColumnType,[ColumnLength],Null,Index
ColumnName2,ColumnType,[ColumnLength],Null,Index
ColumnName3,ColumnType,[ColumnLength],Null,Index
```

バインド リストを削除する

1. ［ツール］ペインで［**SQL アクセス マネージャ**］を展開して［**バインド リスト**］を選択します。

   SQL.DEF ファイルの作成を確認するメッセージが表示されます。
2. ［はい］を選択して、SQL.DEF ファイルを作成します。

   ［**バインド リストを選択してください**］ダイアログ ボックスが表示されます。
3. 削除するバインド リスト名を選択します。
4. ［**削除**］を選択します。バインド リストの削除を確認するよう促すメッセージが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. 選択したバインド リストを削除するには、［はい］を選択します。

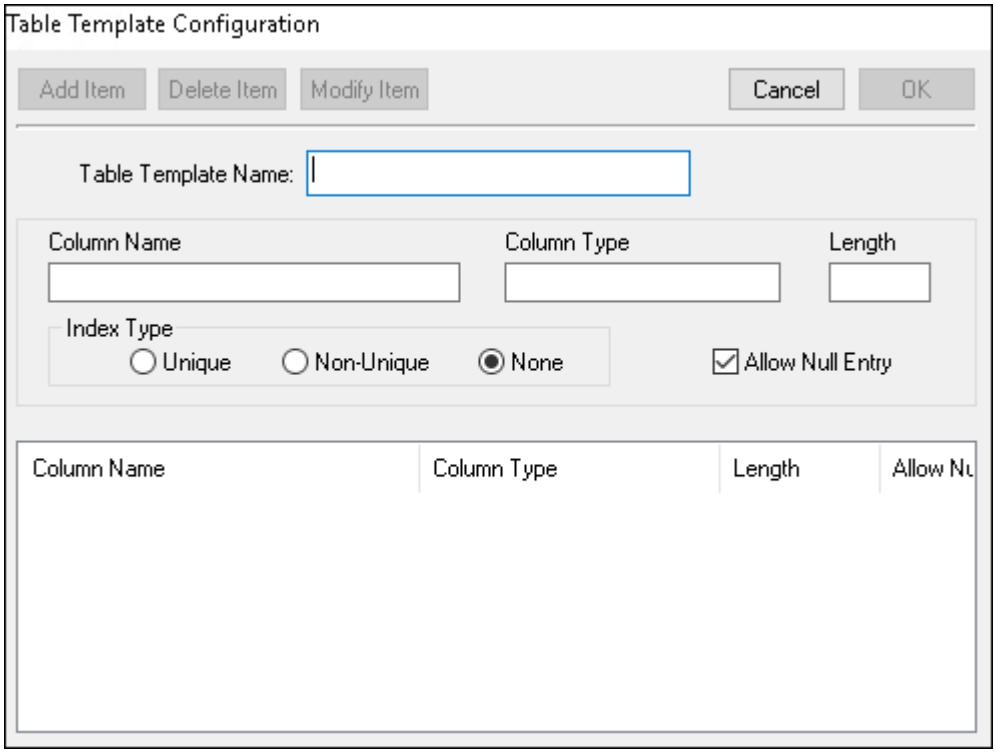
新しいテーブルの構造の定義

テーブル テンプレートは、データベース内に作成する新しいテーブルの構造と形式を定義します。テーブル テンプレートは、SQL.DEF ファイルに保存されます。

新しいテーブル テンプレートを作成する

1. ［ツール］ペインで［**SQL アクセス マネージャ**］を展開して［**テーブル テンプレート**］を選択します。
2. ［新規作成］を選択します。

   ［テーブル テンプレートの定義］ダイアログ ボックスが表示されます。



3. ［テーブル テンプレート名］ボックスに、テーブル テンプレートの名前を入力します。

   テーブル テンプレート名は、インデックスなしで最大 32 文字まで使用できます。固有であるかどうかに関わらずインデックスを作成している場合、テーブル テンプレート名は 24 文字を超えてはなりません。
4. ［カラム名］ボックスに、テーブル テンプレートのカラム名を入力します。

   カラム名は最大 30 文字までです。
5. ［カラム タイプ］ボックスに、カラムのデータ タイプを入力します。データ タイプの選択は、使用されているデータベースによって異なります。
6. ［インデックス タイプ］領域で、以下のいずれかのオプションを選択します。
   - **固有**：各カラム値が固有である必要があります。
   - **非固有**：各カラム値は固有である必要はありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **なし**：インデックスはありません。

注記：SQLCreateTable() 関数を含むスクリプトを実行する場合、インデックス ファイルが自動的に作成されます。

7. このカラムに Null データを入力できるようにするには、［**Null データ入力可**］を選択します。

注記：InTouch では、Null データはサポートされていません。

データを挿入するときに、タグに値が入力されていない場合は、タグのタイプによって Null 値が割り当てられます。

データ型	値
論理型	0
整数型	0
メッセージ型	文字のない文字列

8. ［**アイテムの追加**］を選択して、新しいカラム名、カラム タイプ、長さ、およびインデックス タイプをテーブル テンプレートに追加します。
9. ［**OK**］を選択して、新しいテーブル テンプレート設定を保存して、ダイアログ ボックスを閉じます。

テーブル テンプレートを変更する

1. ［**ツール**］ペインで［**SQL アクセス マネージャ**］を展開して［**テーブル テンプレート**］を選択します。

   ［**テーブル テンプレートを選択してください**］ダイアログ ボックスが表示されます。
2. 変更するテーブル テンプレート名を選択して、［**修正**］を選択します。

   ［**テーブル テンプレートの定義**］ダイアログ ボックスが表示されます。
3. 必要なアイテムを変更します。
4. ［**OK**］を選択して変更を保存し、ダイアログ ボックスを閉じます。

テーブル テンプレートを削除する

1. ［**ツール**］ペインで［**SQL アクセス マネージャ**］を展開して［**テーブル テンプレート**］を選択します。

   ［**テーブル テンプレートを選択してください**］ダイアログ ボックスが表示されます。
2. 削除するテーブル テンプレート名を選択します。
3. ［**削除**］を選択します。テーブル テンプレートの削除を確認するよう促すメッセージが表示されます。
4. ［**はい**］を選択します。［テーブル テンプレートの定義］ダイアログ ボックスが再び表示されます。
5. ［OK］を選択して、ダイアログ ボックスを閉じます。

データベース アプリケーションの操作

SQL アクセス マネージャは、Oracle、Microsoft SQL Server、および Microsoft Access データベースをサポートします。各データベースの要件は固有です。このセクションには、各データベースと SQL アクセス マネージャ間の接続を設定する方法を説明する別のセクションが含まれています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SQL Server データベース アプリケーション

SQLConnect() 関数を InTouch QuickScript で使用して、Microsoft SQL Server データベースに接続します。SQLConnect() 関数は、ユーザーを SQL Server データベースにログオンし、接続を開きます。SQLConnect() 関数によって使用される接続文字列の形式は、以下のとおりです。

(SQLConnect(*ConnectionId*,"<*attribute*>=<*value*>;
<*attribute*>=<*value*>;..."));

ConnectionID 引数は、セッション番号を含む整数型タグ変数です。このセッション番号は、SQL Server データベースへの接続を参照するために、その他のほとんどすべての SQL Access 関数によって使用されます。セッション番号は、それぞれの SQLConnect() 関数呼び出しで 1 つずつ増分します。

次の表は、Microsoft SQL Server によって使用される **SQLConnect()** 関数について説明しています。

属性	値
プロバイダ	SQLOLEDB
データソース	データベースがインストールされるサーバー名
初期カタログ	データベース名
ユーザー<1> ID</1>	ログオン ID、大文字と小文字の区別あり
パスワード	パスワード、大文字と小文字の区別あり

```
"Provider=SQLOLEDB.1;User ID=UserIDStr; Password=PasswordStr;Initial
Catalog=DatabaseName;Data Source=ServerName;"
```

SQL アクセス マネージャは、4 つのタイプの InTouch タグ変数（論理型、整数型、実数型、メッセージ型）を他の SQL Server データベース データ タイプに関連付けます。

データ型	長さ	範囲	タグ タイプ
char 型	8,000 文字	1～131	メッセージ型
int 型		-2,147,483,648～2,147,483,647	整数型
浮動小数点型	15 桁	-1.79E+308 ～ 1.79E+308	実数型

char データ タイプには固定長文字列が含まれています。InTouch メッセージ型タグには char データ タイプが必要です。フィールド長を指定する必要があります。Microsoft SQL Server データベースは、最大 8,000 文字の char フィールドをサポートしています。ただし、InTouch メッセージ型タグは 131 文字に制限されています。メッセージ型タグ値にデータベース フィールドに指定された長さよりも多くの文字が含まれている場合、データベースに挿入されるときに char 文字列は切り捨てられます。

int データ タイプは InTouch 整数型タグ変数を表します。フィールドの長さが指定されていない場合、長さはデータベースのデフォルト値に設定されます。長さが指定されている場合は、Width の形式内にあります。Width は、カラムの最大桁数を決定します。

浮動小数点型のデータ タイプは InTouch 実数型タグ変数を表します。フィールドの長さ設定はデータベースによって固定されます。このデータ タイプのフィールドの長さは必要とされていません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Microsoft Access データベース アプリケーション

Microsoft Access と通信するには、SQLConnect() 関数を InTouch QuickScript で実行して接続する必要があります。

SQLConnect() 関数は、Microsoft Access データベースに接続するために使用されます。SQLConnect() 関数を含むスクリプトを実行すると、データベース サーバーにログオンし、その他の SQL 関数の実行を許可する接続を開きます。SQLConnect() によって使用される接続文字列の形式は、以下のとおりです。

SQLConnect(*ConnectionId*,"<*attribute*>=<*value*>;
<*attribute*>=<*value*>;...");

DSN は、Microsoft Access によって使用される固有の属性で、Microsoft ODBC Administrator で設定されるデータ ソースの名前を識別します。

```
SQLConnect(ConnectionId,"DSN=MSACC");
```

SQL アクセス マネージャがサポートする有効なデータ タイプは、使用されている ODBC ドライバのバージョンによって異なります。

データ型	長さ	デフォルト	範囲	タグ タイプ
テキスト型	255 文字	--	--	メッセージ型
数値型	--	--	--	整数型
数値型	--	--	--	実数型

テキスト データ タイプには固定長文字列が含まれ、InTouch メッセージ型タグ変数と共に使用されます。長さを指定する必要があります。Microsoft Access データベースは、最大 255 文字のテキスト フィールドをサポートします。InTouch メッセージ型タグ変数は 131 文字に制限されています。メッセージ変数がデータベース フィールドに指定された長さよりも多くの文字を含んでいる場合は、データベースに挿入されたときに文字列は切り捨てられます。Microsoft Access ODBC ドライバは、カラム名に対して最大 17 文字までサポートしています。SQLSetStatement(Select Col1, Col2, ...) を使用しているときにサポートされるカラムの最大数は 40 です。

Oracle データベース アプリケーション

SQL Access および Oracle データベース間に通信を確立するには、SQLConnect() 関数を含むスクリプトを実行することによって接続する必要があります。

Oracle 8.0 データベースと通信する

1. Oracle OLEDB プロバイダ（MSDAORA.DLL）ファイルが InTouch を実行しているコンピュータにインストールされていることを確認します。このファイルは、InTouch をインストールするときに一緒にインストールされる MDAC によってインストールされます。
2. **SQLConnect()** 関数を InTouch アクション スクリプトで実行することによって Oracle に接続します。

SQLConnect() 関数によって使用される接続文字列の形式は、以下のとおりです。

SQLConnect(*ConnectionId*,"<*attribute*>=<*value*>;
<*attribute*>=<*value*>;...");

次の表では、Oracle によって使用される関数属性について説明しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

属性	値
プロバイダ	MSDAORA
ユーザー<1> ID</1>	ユーザー名
パスワード	パスワード
データ ソース	Oracle サーバー マシン名

```
SQLConnect(ConnectionId, "Provider=MSDAORA; Data Source=OracleServer; User ID=SCOTT;
Password=TIGER;");
```

次の表は、SQL アクセス マネージャが Oracle データベースに対してサポートしている有効なデータ タイプを一覧表示しています。

データ型	長さ	デフォルト	範囲	タグ タイプ
char 型	2000 文字	1 文字		メッセージ型
数値型	38 桁	38 桁		整数型

Oracle 8.0 の日付フィールドに日時をログ記録するには、delim() 関数を使用してバインド リストを設定する必要があります。

Oracle 日付フィールドに日付と時間の両方をログ記録する

1. SQL アクセス マネージャのアプリケーション エクスプローラで、［バインド リスト］をダブル選択します。［バインド リストの定義］ダイアログ ボックスが表示されます。
2. ［タグ変数.フィールド名］ボックスに、使用するタグ変数を入力します。例：DATE_TIME_TAG。
3. ［カラム名］ボックスに、Oracle 日付フィールドの名前を入力します。Oracle 8.0 を使用している場合は、delim() 関数を使用して区切り文字を指定します。Oracle 9.2 以降を使用している場合は、delim() 関数は必要ありません。
4. InTouch アプリケーションで、QuickScript を作成して、現在の日時からの入力データを準備をします。以下に例を示します。

   ```
   DATE_TIME_TAG = "TO_DATE('" + $DateString + " " + StringMid($TimeString,1,8) +
   "','mm/dd/yy hh24:mi:ss')";
   ```

   QuickScript が WindowViewer で実行された後、以下の形式で日付が表示されます。

   ```
   TO_DATE('08/22/06 23:32:18' ,'mm/dd/yy hh24:mi:ss')
   ```

InTouch での共通 SQL 操作の実行

InTouch では SQL Access 関数を使用して、データベースに保存されている情報と対話します。これらの SQL Access 関数を使用すると、データベース レコードを選択、変更、挿入、または削除するスクリプトを記述できます。

SQL アクションは同期型です。データベース QuickScript を InTouch アプリケーションから実行すると、関数によって要求されたデータベース アクションが完了するまで、コントロールは InTouch に戻りません。

SQL Access 関数は、関数と関連付けられている引数タイプを説明する句読点基準に準拠しています。引数が引用符に囲まれたスクリプト文字列（"Arg1"）に入力されるとき、正確な文字列が使用されます。引用符が使用されない場合、引数値はタグ名とみなされ、タグの現在の値が引数と関連付けられます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ほとんどの SQL 関数は結果コードを返します。結果コードがゼロ以外の場合、関数は失敗し、他のアクションを行う必要があります。結果コードは SQLErrorMsg() 関数によって使用できます。

InTouch QuickScript エディタを使用して、SQL 関数を QuickScript に挿入します。SQL 関数をスクリプトに挿入する一般手順には以下の手順が含まれます。

SQL 関数をスクリプトに挿入する

1. InTouch WindowMaker を起動します。
2. QuickScript を QuickScript エディタで開きます。
3. スクリプト内で SQL 関数を挿入する場所にカーソルを置きます。
4. ［関数］領域で、［アドオン］を選択して、［関数の選択］ダイアログ ボックスを表示します。
5. QuickScript に挿入する SQL 関数を選択します。スクリプトが更新され、挿入した SQL 関数が表示されます。

SQL Access 関数に関連付けられている引数は、以下のように構成されています。

- *BindList*

  SQL.DEF ファイルに定義されているバインド リスト名に対応します。

- *ConnnectionID*

  ConnectionID 引数は、各データベース接続に **SQLConnect()** 関数によって割り当てられた番号（ID）を保持するメモリ型整数タグの名前を参照します。

- *ConnectString*

  ConnectString は、データベース システムと追加のログオン情報を識別します。入力形式は以下のとおりです。

  ```
  "DSN=data source name[;attribute=value
  [;attribute=value]...]"
  ```

  Microsoft SQL Server 接続文字列

  - SQL Server 用 Microsoft OLE DB プロバイダ（推奨）

    ```
    "Provider=SQLOLEDB.1;User ID=sa; Password=;Initial Catalog=MyDB;Data
    Source=MyServer;"
    ```

    SQL Server 用の OLE DB プロバイダは sqloledb です。

  - SQL Server 用 Microsoft OLE DB プロバイダ（推奨）

    ```
    "Provider=SQLOLEDB.1;uid=sa;pwd=;Database=MyDB"
    ```

  - ODBC 用 Microsoft OLE DB プロバイダ（SQL Server 用デフォルト プロバイダ MSDASQL 使用）：

    ```
    "DSN=Pubs;UID=sa;PWD=;"
    ```

  - ODBC 用 Microsoft OLE DB プロバイダ（SQL Server 用デフォルト プロバイダ MSDASQL 使用）：

    ```
    "Data Source=Pubs;User ID=sa;Password=;"
    ```

  Oracle 接続文字列

  - Oracle 用 Microsoft OLE DB プロバイダ（推奨）

    ```
    "Provider=MSDAORA;Data Source=ServerName;User ID=UserIDStr;
    Password=PasswordStr;"
    ```

  Microsoft Access 接続文字列

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Microsoft Jet 用 Microsoft OLE DB プロバイダ（推奨）。Microsoft.Jet.OLEDB.4.0 は、Microsoft Jet 用ネイティブ OLE DB プロバイダ（Microsoft Access Database エンジン）です。

```
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=d:\DBName.mdb;User
ID=UserIDStr;Password=PasswordStr;"
```

ODBC 用 Microsoft OLE DB プロバイダ（MS Access 用デフォルト プロバイダ MSDASQL の使用）:

```
"Provider=MSDASQL;DSN=DSNStr;UID=UserName; PWD=PasswordStr;"
```

- *ErrorMsg*

  エラーのテキスト説明を含むメッセージ型タグ

- *FileName*

  情報を含むファイルの名前

- *MaxLen*

  パラメータに関連付けられているカラムの最大サイズ。この引数によって、データが可変長文字かロング可変長文字のどちらの文字タイプであるかが決定されます。MaxLen がデータベースによって許可されている最大文字列に等しいかそれより少ない場合、このデータは可変長文字タイプとなります。それより大きい場合、データはロング可変長文字タイプとなります。

- *ORDERBY_EXPRESSIONS*

  カラムおよび昇順または降順のソート順序を定義します。ソートに使用できるのはカラム名のみです。式の形式は以下のとおりです。

  ColumnName [ASC|DESC]

  選択したテーブルをカラム名で昇順にソートするには:

```
"manager ASC"
```

  複数カラムでソートするには、式の形式は以下のようになります。

  ColumnName [ASC|DESC]、

  ColumnName [ASC|DESC]

  選択したテーブルを 1 つのカラム名（例: temperature）で昇順に並べ、別のカラム名（例: time）を降順にソートするには:

```
"temperature ASC, time DESC"
```

- *ParameterNumber*

  ステートメント内の実際のパラメータ番号。

- *ParameterType*

  指定したパラメータのデータ タイプ。有効な値は以下のとおりです。

タイプ	説明
Char	空白を補った固定長文字列
Var Char	可変長文字列
10 進数	BCD 番号
整数型	4 バイト符号付き整数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

タイプ	説明
Small Integer	2 バイト符号付き整数
浮動小数点	4 バイト浮動小数点
倍精度浮動小数点型	8 バイト浮動小数点
日付時刻	8 バイトの日時値
日付	4 バイトの日付時刻値
時間	4 バイトの日付時刻値
タイプなし	データ タイプなし

- *ParameterValue*

  設定する実際の値。

- *精度*

  10 進数値の精度、文字の最大サイズ、または日時値のバイト数。

- *RecordNumber*

  取得する実際のレコード番号

- *ResultCode*

  ほとんどの SQL 関数が返す整数型変数。ResultCode は、関数が正常に実行されるとゼロ（0）、正常に実行されなかった場合には負の値を返します。

- *スケール*

  10 進数値のスケール。この値は、Null に設定するパラメータに対して適用可能な場合にのみ必要です。

- *StatementID*

  拡張機能ステートメントを使用している場合、SQL は内部で使用する StatementID を返します。

- *SQLStatement*

  実際のステートメント。次に例を示します。

```
ResultCode=SQLSetStatement(ConnectionID, "Select LotNo, LotName from LotInfo");
```

- *TableName*

  TableName パラメータには、データベースでアクセスまたは作成するテーブルの名前が含まれます。

- *TemplateName*

  TemplateName パラメータは、テーブルを定義する SQL.DEF ファイルのテンプレートの名前です。

- *WhereExpr*

  テーブルの行に対する True または False の条件を定義します。関数はテーブルから Ture の条件を持つ行のみを抽出します。式の形式は以下のとおりです。

  ColumnName comparison_operator expression

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注記：カラムが文字データ タイプの場合、式を一重引用符で囲む必要があります。

次の例は、Name カラムに EmployeeID という値を含む行をすべて選択します。

```
Name='EmployeeID'
```

次の例は、100 ～ 199 までの部品番号を含むすべての行を選択します。

```
partno>=100 and partno<200
```

次の例は、temperature カラムで 350 を超える値があるレコードをすべて選択します。

```
temperature>350
```

データベースの接続と接続解除

SQLConnect() 関数および SQLDisconnect() 関数をスクリプトで使用して、SQL データベースとの接続または接続解除を行います。

SQLConnect() 関数

SQLConnect() 関数を InTouch QuickScript で使用して、ConnectString 引数で指定されたデータベースに接続できます。

SQLConnect() は、すべての後続 SQL 関数でパラメータとして使用される ConnectionID 引数に値を返します。**SQLConnect** 関数をスクリプトで使用する前に、アプリケーションでバインド リストを定義する必要があります。

カテゴリ

SQL

構文

```
[ResultCode=]SQLConnect(ConnectionID, "ConnectString");
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

ConnectString
データベースおよび SQLConnect() 関数で使用される追加ログオン情報を識別する文字列

備考

アプリケーション フォルダにバインド リスト（SQL.DEF ファイル）が必要です。この関数は、このファイルがないと機能しません。

SQLTrace=1 が win.ini ファイルの [InTouch] セクションに定義されている場合は、**SQLConnect** が正常に実行されるたびに、ADO、プロバイダ、およびデータベース システムのバージョン情報が Log Viewer にログ記録されます。

例

次のステートメントは、IBM OS/2 Database Manager と SAMPLE という名前のデータベースに接続します。

```
[ResultCode=]SQLConnect(ConnectionID,"DSN=OS2DM;
DB=SAMPLE");
```

この関数は、すべての後続 SQL 関数でパラメータとして使用される ConnectionID 変数に値が返されます。

```
"DSN=data source name[;attribute=value
[;attribute=value]...]"
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SQLConnectEx() 関数

SQL 認証で「ユーザー名とパスワード」タイプの資格情報を使用する場合、**SQLConnectEx()** 関数を InTouch QuickScript で使用して、ConnectString 引数で指定されたデータベースに接続できます。

SQLConnectEx() は、すべての後続 SQL 関数でパラメータとして使用される ConnectionID 引数に値を返します。**SQLConnectEx()** 関数をスクリプトで使用する前に、アプリケーション フォルダでバインド リストを定義する必要があります。

カテゴリ

SQL

構文

```
[ResultCode=]SQLConnectEx(ConnectionID, "ConnectString", "Credential Name");
```

引数

ConnectionID
各データベース接続に SQLConnectEx() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

ConnectString
データベースおよび SQLConnectEx() 関数で使用される追加ログオン情報を識別する文字列

資格情報名
資格情報マネージャに保存された資格情報の名前スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。

備考

アプリケーション フォルダにバインド リスト（SQL.DEF ファイル）が必要です。この関数は、このファイルがないと機能しません。

SQLTrace=1 が win.ini ファイルの [InTouch] セクションに定義されている場合は、**SQLConnectEx()** が正常に実行されるたびに、ADO、プロバイダ、およびデータベース システムのバージョン情報が Log Viewer に記録されます。

例

次のステートメントは、IBM OS/2 Database Manager と SAMPLE という名前のデータベースに接続します。

```
[ResultCode=]SQLConnectEx(ConnectionID,"DSN=OS2DM;
DB=SAMPLE");
```

この関数は、すべての後続 SQL 関数でパラメータとして使用される ConnectionID 変数に値が返されます。

```
"DSN=data source name[;attribute=value
[;attribute=value]...]"
```

SQLDisconnect() 関数

SQLDisconnect() 関数はデータベースとの接続を解除し、**SQLPrepareStatement()** 関数および **SQLInsertPrepare()** 関数用に取得されたすべての未リリース リソースをクリーン アップします。

カテゴリ

SQL

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
[ResultCode=]SQLDisconnect(ConnectionID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

参照項目

SQLConnect()

新規テーブルの作成

SQLCreateTable() 関数を InTouch QuickScript で使用し、指定したテーブル テンプレートからのパラメータを使用してデータベースにテーブルを作成します。

SQLCreateTable() 関数

SQLCreateTable() 関数を InTouch QuickScript で使用し、指定したテーブル テンプレートからのパラメータを使用してデータベースにテーブルを作成します。テーブル テンプレートは、データベース テーブルの構造を含む SQL.DEF ファイルで定義されます。

カテゴリ

SQL

構文

```
[ResultCode=]SQLCreateTable(ConnectionID, TableName,TemplateName);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
作成するデータベース テーブルの名前

TemplateName
使用するテンプレート定義の名前

例

次の **SQLCreateTable()** 関数の例では、OutputVal テンプレートで定義されたカラム名とデータを使用して、BATCH1 という名前のテーブルを作成します。

```
ResultCode=SQLCreateTable(ConnectionID,"BATCH1",
"OutputVal");
```

参照項目

SQLConnect()

テーブルの削除

SQLDropTable() 関数を InTouch QuickScript で使用して、データベースからテーブルを削除します。

SQLDropTable() 関数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SQLDropTable() 関数を InTouch QuickScript で使用して、データベースからテーブルを削除します。**SQLDropTable()** 関数を含む QuickScript が完了した後は、テーブルは認識されず、どの SQL ステートメントにも応答しません。

カテゴリ

SQL

構文

```
[ResultCode=]SQLDropTable(ConnectionID, TableName);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
データベースから削除するテーブルの名前

例

次の **SQLDropTable()** 関数の例では、データベースから BATCH1 テーブルを削除します。

```
ResultCode=SQLDropTable(ConnectionID,"BATCH1");
```

参照項目

SQLConnect()

テーブルからのデータの取得

一組の SQL 関数をスクリプトで使用して、データベースからデータを取得し、値を InTouch タグ変数に書き込みます。

- **SQLSelect()** 関数は、テーブルから情報を取得し、この情報をレコードの形式でメモリに作成された一時結果テーブルに配置します。
- **SQLGetRecord()** 関数は、RecordNumber によって指定されたレコードを現在の選択バッファから取得します。
- **SQLNumRows()** 関数は、前の **SQLSelect()** 関数で指定した条件に一致するテーブル行の数を返します。
- SQLFirst() 関数は、最後の **SQLSelect()** 関数によって作成された結果テーブルの最初のレコードを取得します。
- SQLNext() 関数は、最後の **SQLSelect()** 関数によって作成された結果テーブルの次のレコードを取得します。
- SQLPrev() 関数は、論理テーブルの前の行からデータを取得し、その行からの値を InTouch タグ変数にフェッチします。
- SQLLast() 関数は、論理テーブルの最後の行を取得し、その行からの値を InTouch タグ変数にフェッチします。
- SQLEnd() 関数は、ConnectionId と関連付けられている結果テーブルの内容を保存するメモリを解放します。

SQLFirst()、**SQLPrev()**、**SQLNext()**、**SQLLast()**、および **SQLGetRecord()** の関数は、論理テーブルの指定した行からデータを取得し、InTouch タグ変数値として保存します。フィールドが Null の場合は、関連付け

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

られている InTouch タグ変数の値が、タグ変数がアナログ型かメッセージ型かによってゼロまたはゼロ長文字列に設定されます。

データベース内の文字列が 131 文字よりも長い場合は、最初の 131 文字のみがデータベースから関連付けられている InTouch メッセージ型タグ変数にコピーされます。

SQLSelect() 関数

SQLSelect() 関数は、テーブルからレコードを取得します。SQLSelect() 関数を含むスクリプトが処理されると、取得されたレコードがメモリの一時結果テーブルに配置されます。これらのレコードは、SQLFirst()、SQLLast()、SQLNext() および SQLPrev() の関数を使用してブラウズできます。

重要：SQLSelect() 関数を含むスクリプトが完了したときには必ず SQLEnd() 関数を呼び出して、結果テーブルで使用されたメモリを解放します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLSelect(ConnectionID,TableName, BindList,WhereExpr,OrderByExpression);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
アクセスするデータベース テーブルの名前

BindList
使用する InTouch タグ変数とデータベースを定義します。

WhereExpr
テーブルの行に対する True または False の条件を定義します。SQLSelect() 関数は、*WhereExpr* 条件が Ture の行のみからデータを抽出します。式の形式は以下のとおりです。

ColumnName comparison_operator expression.

注記：比較が文字列式で行われる場合は、式を一重引用符で囲む必要があります。

次の例は、name カラムに EmployeeID という値を含む行をすべて選択します。

name='EmployeeID'
次の例は、100 ～ 199 までの部品番号を含むすべての行を選択します。

```
partno>=100 and partno<200
```

次の例は、temperature カラムに 350 を超える値が含まれるすべての行を選択します。

```
temperature>350
```

WhereExpr - Memory message Tag

```
OrderByExpr - Memory message Tag
Speed_Input - Memory Real - User Input Analog
Serial_Input - Memory Message - User Input String
```

アナログ型の例

```
WhereExpr = "Speed = " + text
(Speed_Input,"#.##");
```

Speed_Input は数字であるため、WhereExpr 文字列に連結できるようにテキストに変換する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

文字列の例

```
WhereExpr = "Ser_No = ‘" +
Serial_input + "’";
```

Serial_Input は文字列であるため、値を WhereExpr = "Ser_No='125gh'" のように一重引用符で囲む必要があります。

Like ステートメントを使用した文字列の例

```
WhereExpr = "Ser_No like ‘-’"
```

Like 比較演算子を使用するときには、% をワイルド カードとして使用できます。

ブール AND 演算子を使用した文字列およびアナログ型の例

```
WhereExpr = "Ser_No = ‘" + Serial_input + "’" + " and " + "Speed = " +
text(Speed_Input,"#.##");OrderByExpr = "";
```

順位が重要でなければ、上記のように null 文字列を使用します。

WhereExpr タグ変数を使用した SQLSelect

```
ResultCode = SQLSelect(Connect_Id,TableName,BindList
,
WhereExpr,OrderByExpr);
Error_msg = SQLErrorMsg( ResultCode );
```

SQLSelect WhereExpr 組み込み関数

```
ResultCode = SQLSelect(Connect_Id,TableName,BindList
,
"Ser_No = '" + Serial_input + "'", OrderByExpr);
Error_msg = SQLErrorMsg( ResultCode );
```

OrderByExpr

テーブル カラム内でデータをソートするための方向を定義します。ソートにはカラム名のみを使用することができ、式は次の形式である必要があります。

*ColumnNam*e [ASC|DESC]

次の例は、テーブルを manager カラムからのデータによって昇順でソートします。

```
"manager ASC"
```

また、以下のような形式を使用して、複数のカラムでソートすることもできます。

ColumnName [ASC|DESC]、
ColumnName [ASC|DESC]

次の例は、選択したテーブルを temperature カラムの昇順および time カラムの降順でソートします。

```
"temperature ASC,time DESC"
```

例

次のステートメントは、カラム名に cookie という値が含まれる List1 という名前の BindList を使用して、BATCH テーブルからレコードを選択します。その後、amount カラムで昇順に、sugar カラムで降順に情報をソートして表示します。

```
ResultCode=SQLSelect(ConnectionID,"BATCH","List1","type='cookie'","amount ASC,sugar DESC");
```

次のステートメントでは、データベースのすべてのデータが選択され、WhereExpr および OrderByExpr の値は指定されません。

```
ResultCode=SQLSelect(ConnectionID,"BATCH", "List1", "","");
```

参照項目

SQLFirst()、SQLConnect()、SQLLast()、SQLNext()、SQLPrev()、SQLEnd()

SQLGetRecord() 関数

SQLGetRecord() 関数は、RecordNumber によって指定されたレコードを現在の選択バッファから取得します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

SQL

構文

```
[ResultCode=]SQLGetRecord(ConnectionID, RecordNumber);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

RecordNumber
取得する実際のレコード番号

例

```
ResultCode=SQLGetRecord(ConnectionID,3);
```

参照項目

SQLConnect()

SQLNumRows() 関数

SQLNumRows() 関数は、最後の SQLSelect() 関数で指定した条件に一致する行数を示します。たとえば、WhereExpr 引数を使用して AGE という名前のカラム（AGE は 45 に等しい）を持つ行をすべて選択すると、返される行数は 40 または 4000 となります。これによって、その関数が次回処理されるかが決定されます。

カテゴリ

SQL

構文

```
SQLNumRows(ConnectionID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

例

次のステートメントは、選択された行数を NumRows 整数型タグ変数に返します。

```
NumRows=SQLNumRows(ConnectionID);
```

参照項目

SQLConnect()

SQLFirst() 関数

SQLFirst() 関数は、直前の SQLSelect() 関数によって作成された結果テーブルの最初のレコードを選択します。

カテゴリ

SQL

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
[ResultCode=]SQLFirst(ConnectionID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

参照項目

SQLConnect()、SQLSelect()

SQLNext() 関数

SQLNext() 関数は、最後の SQLSelect() 関数によって作成された結果テーブルのシーケンスの次のレコードを選択します。SQLSelect() 関数は、SQLNext() 関数をスクリプトで実行する前に処理する必要があります。

カテゴリ

SQL

構文

```
[ResultCode=]SQLNext(ConnectionID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

例

```
[ResultCode=]SQLNext(ConnectionID);
```

参照項目

SQLConnect()、SQLSelect()

SQLPrev() 関数

SQLPrev() 関数は、最後の SQLSelect() 関数によって作成された結果テーブルの前のレコードを選択します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLPrev(ConnectionID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

備考

このコマンドを使用する前に、SQLSelect() 関数を実行する必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

```
[ResultCode=]SQLPrev(ConnectionID);
```

参照項目

SQLConnect()、SQLSelect()

SQLLast() 関数

SQLLast() 関数は、前の SQLSelect() 関数によって作成された結果テーブルの最後のレコードを選択します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLLast(ConnectionID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

例

```
[ResultCode=]SQLLast(ConnectionID);
```

参照項目

SQLConnect()、SQLSelect()

SQLEnd() 関数

SQLEnd() 関数は、SQLSelect() 関数が結果テーブルの内容を保存するために使用しされたメモリを解放した後に実行されます。

カテゴリ

SQL

構文

```
[ResultCode=]SQLEnd(ConnectionID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

参照項目

SQLConnect()、SQLSelect()

テーブルへの新しいレコードの書き込み

SQLInsert() 関数を使用して、新しいレコードをデータベースに挿入できます。**SQLInsert()** 関数は、InTouch タグ変数の現在の値を使用して、1 つのレコードをテーブルに挿入します。**SQLInsert()** 関数は、ステートメントを準備、挿入、および終了するワン ステップの操作です。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch メッセージ型タグに関連付けられている文字列が、テーブルの対応するテキスト フィールドの定義されたサイズよりも長い場合は、メッセージ型タグから使用されている文字数がフィールドの定義サイズとなります。

注記：InTouch タグ変数に Null は使用できません。バインド リストにフィールドが含まれている場合は、これらの関数を使用したデータベースを更新したり、データベースに Null 値を挿入したりすることはできません。Null 値を許可するように定義されているはずであった、フィールドを含まない INSERT ステートメントに、**SQLExecute** を使用してフィールドに Null 値を挿入できます。

SQL Access は、別々に準備、挿入、およびレコード挿入後にクリーン アップを行う他の 3 つの関数を提供しています。これらの関数を一緒に使用すると、単一の prepare および end ステートメントを含むスクリプトを記述でき、必要に応じていくつでもレコード挿入ステートメントを追加できます。
SQLInsert() 関数ではなく個別の関数を使用してデータを挿入すると、コンピュータのリソースの使用率を削減できます。

SQLInsert() 関数

SQLInsert() 関数は、指定した BindList のタグ変数の値を使用して、参照されているテーブルに新しいレコードを挿入します。BindList パラメータは、使用される InTouch タグ変数とそれが関連付けられているデータベース カラムを定義します。

ステートメントを準備、挿入、および終了するには、SQLInsert() 関数を使用します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLInsert(ConnectionID, TableName, BindList);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
アクセスするデータベース テーブル名

BindList
使用される InTouch タグ変数と関連付けられるデータベース カラムを定義します。

例

次のステートメントは、List1 で指定されたタグ変数値を使用して、テーブル ORG に新しいレコードを挿入します。

```
ResultCode=SQLInsert(ConnectionID,"ORG","List1");
```

SQLInsertPrepare() 関数

SQLInsertPrepare() 関数は、関数を実行するたびに Insert ステートメントを作成および準備します。Insert ステートメントは処理されません。StatementID 引数は、ステートメントが処理された後の値を含む整数型タグ変数です。

カテゴリ

SQL

構文

```
[ResultCode=]SQLInsertPrepare (ConnectionID, TableName, BindList, StatementID);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
アクセスするデータベース テーブルの名前

BindList
使用される InTouch タグ変数と関連付けられるデータベース カラムを定義します。

StatementID
SQLPrepareStatement() 関数が使用されたときに SQL によって返される整数値

参照項目

SQLConnect()、SQLPrepareStatement()

SQLInsertExecute() 関数

SQLInsertExecute() 関数は、SQLInsertPrepare() 関数で指定された、以前に準備された Insert ステートメントを実行します。

SQLInsertExecute() 関数は InTouch タグ変数の現在の値を使用して、1 つの行を前の SQLInsertPrepare() 関数で識別されたテーブルに挿入します。BindList 引数に MS SQL Server テーブル用の Identity キー フィールドが含まれる場合は、SQLInsertExecute() を実行する前に IDENTITY_INSERT オプションを設定する必要があります。

StatementID 引数には、前の SQLInsertPrepare() 関数がスクリプト内で実行されたときに SQL によって返される整数値が含まれます。

カテゴリ

SQL

構文

```
[ResultCode=]SQLInsertExecute(ConnectionID, BindList, StatementID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

BindList
使用される InTouch タグ変数と関連付けられるデータベース カラムを定義します。

StatementID
SQLPrepareStatement() 関数が使用されたときに SQL によって返される整数値

参照項目

SQLConnect()、SQLPrepareStatement()

SQLInsertEnd() 関数

SQLInsertEnd 関数は、**SQLInsertPrepare** によって作成された StatementID 関数に関連付けられているリソースをクリーン アップします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

SQL

構文

```
[ResultCode=]SQLInsertEnd(ConnectionID,StatementID );
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

StatementID
SQLPrepareStatement() 関数が使用されたときに SQL によって返される整数値

例

次の例は、複数の Insert 関数をスクリプトに指定する方法を示しています。

```
ResultCode = SQLSetStatement(ConnectionId, "SET IDENTITY_INSERT Products ON");
ResultCode = SQLExecute(ConnectionId, "", 0);
ResultCode = SQLInsertPrepare(ConnectionId, TableName, Bindlist, StatementID);
ResultCode = SQLInsertExecute(ConnectionId, Bindlist, StatementID);
ResultCode = SQLInsertEnd(ConnectionId,StatementID);
```

参照項目

SQLConnect()、SQLPrepareStatement()

テーブルでの既存のレコードの更新

SQL Access は、InTouch タグ変数からの値を使用してテーブル レコードを更新するための 2 つの関数を提供します。

- SQLUpdate()
- SQLUpdateCurrent()

SQLUpdate() 関数

SQLUpdate() 関数は、InTouch タグ変数の現在の値を使用して、WhereExpr 引数によって設定された条件に一致するテーブルのすべての行を更新します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLUpdate(ConnectionID, TableName, BindList, WhereExpr);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
アクセスするデータベース テーブルの名前

BindList
使用される InTouch タグ変数と関連付けられるデータベース カラムを定義します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

WhereExpr
テーブルの行に対する True または False の条件を定義します。関数はテーブルから Ture の条件を持つ行のみを更新します。式の形式は以下のとおりです。
ColumnName comparison_operator expression.

注記 : カラムが文字データ タイプの場合は、条件式を一重引用符で囲む必要があります。

例

次の例は、name カラムに EmployeeID という値を含む行をすべて選択します。

```
name='EmployeeID'
```

次の例は、100 ～ 199 までの部品番号を含むすべての行を選択します。

```
partno>=100 and partno<200
```

次の例は、temperature カラムで 350 を超える値があるすべての行を選択します。

```
temperature>350
```

次のステートメントは、ロット番号が 65 である、BATCH テーブルのすべてのレコードを、BindList "List1" で指定したタグ変数の現在の値に更新します。

```
ResultCode=SQLUpdate(ConnectionID,"BATCH", "List1","lotno=65");
```

注記 : すべてのレコードが固有であることを確認してください。テーブルに該当するレコードが存在する場合、類似のレコードがすべて更新されます。

参照項目

SQLConnect()

SQLUpdateCurrent() 関数

SQLUpdateCurrent() 関数は、SQLSelect() 関数または SQLExecute() 関数のステートメントで指定されているバインド リストによってテーブル フィールドにマッピングされる InTouch タグ変数を使用して、論理テーブルの現在の行を更新します。現在の行と同一の行が存在する場合、すべての行が更新されます。

最大で 54 の同一レコードを一度に更新できます。SQL Access で更新される同一の行が多数存在する場合、SQLUpdateCurrent() 関数によってエラーが返されます。エラー メッセージは、「Microsoft Cursor Engine: Key column information is insufficient or incorrect. Too many rows were affected by update.（Microsoft カーソル エンジン: キー カラムの情報が不適切か不正です。更新される列が多すぎます。)」という内容となります。

このエラーを回避するには、各行を固有にする固有キー フィールドをテーブルに作成します。SQL Access で使用されるすべてのテーブルが固有のキーを持つことを強くお勧めします。キーがないテーブルの場合は、AutoNumber (Access) 型のフィールドまたは行 Identity（SQL Server）として使用される整数型フィールドをプライマリ キーとして使用して、SQLUpdateCurrent() 関数が一度に 1 つの行のみを更新するようにすることを推奨します。このプライマリ キー フィールドはバインド リストに含まれている必要はありません。

カテゴリ

SQL

構文

```
[ResultCode=]SQLLast(ConnectionID);
```

引数

ConnectionID

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

例

```
[ResultCode=]SQLLast(ConnectionID);
```

参照項目

SQLConnect()

テーブルにあるレコードを削除します

データベース テーブルからレコードを削除するには、2 つの SQL 関数を使用できます。

SQL Access は、テーブル レコードを削除するための 2 つの関数を提供します。

- SQLClearTable() はテーブルからレコードを削除します。
- SQLDelete() は、指定した条件に一致するテーブルからレコードを削除します。

SQLClearTable() 関数

SQLClearTable() 関数は、テーブルからすべてのレコードを削除します。テーブルはデータベースから削除されません。

カテゴリ

SQL

構文

```
[ResultCode=]SQLClearTable(ConnectionID,"TableName");
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
すべてのレコードがクリアされるテーブルの名前

例

次の例では、SQLClearTable() 関数が、BATCH1 テーブルからすべてのレコードをクリアします。

```
ResultCode=SQLClearTable(ConnectionID,"BATCH1");
```

参照項目

SQLConnect()、SQLClearStatement()

SQLDelete() 関数

SQLDelete() 関数は、*WhereExpr* 引数で指定された条件に一致するすべてのレコードをテーブルから削除します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLDelete(ConnectionID, TableName, WhereExpr);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

TableName
WhereExpr 引数で指定された条件に一致するレコードがクリアされるテーブルの名前

WhereExpr
テーブルの行に対する True または False の条件を定義します。**SQLDelete()** 関数は、*WhereExpr* 条件が True の行レコードのみを削除します。式の形式は以下のとおりです。
ColumnName comparison_operator expression

注記：SQLDelete() 関数に Null の *WhereExpr* 引数を含めることはできません。

例

次のステートメントでは、ロット番号が 65 に等しい BATCH1 テーブルのすべてのレコードが削除されます。

```
ResultCode=SQLUpdate(ConnectionID,"BATCH", "List1","lotno=65");
```

注記：カラムが文字データ タイプの場合は、条件式を "MachineID='AG_LX7_2'" のように一重引用符で囲む必要があります。

参照項目

SQLConnect()

パラメータ化ステートメントの実行

動的クエリを作成するには、SQLSetStatement() 関数および SQLAppendStatement() 関数を使用します。SQLSetStatement() 関数は、新しい SQL ステートメントを開始します。これはストアド プロシージャの名前を含む任意の有効な SQL ステートメントとなります。SQLAppendStatement() 関数は文字列の内容を使用して SQL ステートメントを続行します。

SQLSetStatement() 関数

SQLSetStatement() 関数は、SQLStatement の内容を使用して、確立された接続 ConnectionID で SQL ステートメント バッファを開始します。ConnectionID につき SQL ステートメント バッファは 1 つです。エラーは関数の戻りとして返されます。

カテゴリ

SQL

構文

```
[ResultCode=]SQLSetStatement(ConnectionID, SQLStatement);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

SQLStatement
実際の SQL ステートメントについては、以下の例を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

```
ResultCode=SQLSetStatement(ConnectionID,"Select LotNo, LotName from LotInfo");
```

次の例では、StatementID がゼロに設定され、ステートメントを実行する前に、ステートメントが SQLPrepare(Connect_Id, StatementID) を呼び出す必要はありません。StatementID が、この選択を適切に終了する SQLPrepare によって作成されていないため、SQLClearStatement() 関数の代わりに SQLEnd() 関数を使用します。

```
SQLSetStatement( Connect_Id, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLExecute(Connect_Id,0);
```

次の例では、StatementID が SQLPrepareStatement() 関数によって作成され、SQLExecute() 関数で使用されます。この SELECT ステートメントを終了するには、SQLClearStatement() 関数を使用して、リソースとハンドルを解放します。

```
SQLSetStatement( Connect_Id, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLPrepareStatement(Connect_Id,StatementID);
SQLExecute(Connect_Id,StatementID);
SQLSetStatement( Connect_Id、"Select Speed, Ser_No from tablename where Ser_No ='" +
Serial_input + "'");
SQLPrepareStatement(Connect_Id,StatementID);
SQLExecute(Connect_Id,StatementID);
```

参照項目

SQLConnect()

SQLAppendStatement() 関数 SQLAppendStatement()かんすう

SQLAppendStatement() 関数は、文字列の内容を使用して SQL ステートメントを続行します。関数呼び出し中にエラーが発生したかどうかが戻り値によって示されます。

InTouch タグ変数は最大 131 文字の文字列をサポートできます。一般的に、SQLAppendStatement() 関数を使用して、追加文字列をステートメントに連結します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLSetStatement(ConnectionID, SQLStatement);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

SQLStatement
付加する実際のステートメント

例

```
ResultCode=SQLAppendStatement(ConnectionID, "where tablename.columnname=TR-773-01")
```

参照項目

SQLConnect()、SQLClearStatement()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ステートメントの作成またはファイルからの既存のステートメントのロード

クエリは他のサードパーティ データベース ツールから作成でき、SQL Access を使用してクエリを実行できます。最初に、サードパーティ データベース ツールから作成された .SQL クエリ ファイルから SQL ステートメントをロードする必要があります。

```
ResultCode = SQLLoadStatement (ConnectionID, "c:\myappdir\lotquery.sql");
```

SQLLoadStatement() 関数を使用して SQL クエリをロードします。これでステートメントを実行する準備ができました。

SQLLoadStatement() 関数

SQLLoadStatement() 関数は、ファイルから SQL ステートメントを読み取ります。

ステートメントは 1 つのファイルにつき 1 つです。ただし、SQLAppendStatement() 関数は、SQLPrepareStatement() 関数または SQLExecute() 関数が呼び出されていない場合に、ステートメントに何かを付加するために使用できます。

カテゴリ

SQL

構文

```
[ResultCode=]SQLDevStatement(ConnectionID, FileName);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

FileName
SQL ステートメントを含むファイルの名前

備考

ステートメントをロードしてステートメント ハンドルを取得した後に、SQLPrepareStatement() 関数を使用してステートメントの実行を準備してください。

例

SQL.txt ファイルには次の SQL ステートメントが含まれます。

```
TableName から ColumnName を選択（ColumnName>100）;
```

SQLLoadStatement() 関数は、ファイルからステートメントをロードします。

```
ResultCode=SQLDevStatement(ConnectionID,
"C:\SQL.txt")
```

参照項目

SQLConnect()、SQLAppendStatement()、SQLExecute()、SQLPrepareStatement

ステートメントの準備

以下の関数を使用すると、必要なパラメータ化ステートメントを作成し、パラメータに 1 つずつ動的に入力できます。たとえば、以下の関数を使用すると、ファイルで汎用ステートメントを使用して、SQLLoadStatement() 関数を使ってそれをロードし、SQLPrepareStatement() 関数を使用してステートメント ID を取得するよう準備してから、ステートメントのパラメータに入力することができます。

- **SQLPrepareStatement()**

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- **SQLSetParamChar()**
- **SQLSetParamDate()**
- **SQLSetParamDateTime()**
- **SQLSetParamDecimal()**
- **SQLSetParamFloat()**
- **SQLSetParamInt()**
- **SQLSetParamLong()**
- **SQLSetParamNull()**
- **SQLSetParamTime()**
- **SQLClearParam()**
- **SQLClearStatement()**

SQL ステートメントでパラメータ代入を実行するには、後続のパラメータを指定する SQL ステートメントに "?" を配置します。ステートメントが準備され、パラメータがステートメントに設定されて、ステートメントが実行されます。

SQLPrepareStatement() 関数

SQLPrepareStatement() 関数は、SQL ステートメントの実行準備を行います。ステートメントが実行されるのではなく、ステートメントをアクティブにして、パラメータ値を設定できるようにします。

カテゴリ

SQL

構文

```
SQLPrepareStatement(ConnectionId, StatementID)
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

備考

デフォルト ステートメントを準備し、StatementID（1、2、3、･･･）を返します。この準備は、SQLSetParam{Type} 関数を使用して、設定する必要のあるパラメータを持つステートメントに対して役立ちます。

ステートメント パラメータの設定

SQL アクセス マネージャは、SQL ステートメントに含まれているパラメータに割り当てられている値を変更するための一組の関数を提供しています。

SQLSetParamChar() 関数

SQLSetParmChar() 関数は、スクリプトで使用して、指定した文字列に指定したパラメータの値を設定します。この関数は、実行前に複数回呼び出すことができ、その結果パラメータ値は送信されたすべての値の連続として設定できます。長さが 0（ゼロ）の場合は無視されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

SQL

構文

```
SQLSetParam Replicas(StatementID, ParameterNumber, ParameterValue, MaxLength);
```

引数

StatementID
SQLPrepareStatement() 関数が使用されたときに SQL によって返される整数値

ParameterNumber
ステートメントのパラメータ番号

ParameterValue
パラメータ値として設定される値

最大長
このパラメータが関連付けられているカラムの最大幅。この設定によって、パラメータが可変長文字であるかロング可変長文字のどちらのタイプであるかが決定されます。*MaxLength* がデータベースによって許可されている最大文字列に等しいまたはそれより少ない場合、このパラメータは可変長文字タイプとなります。大きい場合は、ロング可変長文字タイプとなります。

参照項目

SQLPrepareStatement()

SQLSetParamDate() 関数

SQLSetParamDate() 関数は、パラメータの値を指定した日付に設定します。

カテゴリ

SQL

構文

```
SQLSetParamDate(StatementID, ParameterNumber, "Value");
```

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントでパラメータを識別する整数値

値
二重引用符で囲まれるリテラルとしてパラメータに割り当てられる日付、または値が日付であるタグ変数の名前。日付に割り当てられる時間は 12:00:00 am です。

例

この例では、NewDate タグ変数に関連付けられている日付に 3 番目のステートメントの 2 番目のパラメータを設定します。

```
SQLSetParamDate(3, 2, NewDate);
```

参照項目

SQLPrepareStatement()

SQLSetParamDateTime() 関数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

SQLSetParamDateTime() 関数は、パラメータの値を指定した日時に設定します。

カテゴリ

SQL

構文

```
SQLSetParamDateTime(StatementID, ParameterNumber, Value, Precision);
```

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントでパラメータを識別する整数値

値
ParameterNumber 引数によって識別されるパラメータに割り当てられる日付と時間

精度
パラメータの値として割り当てられる日時値の文字数を指定する整数

参照項目

SQLPrepareStatement()

SQLSetParamDecimal() 関数

SQLSetParamDecimal() 関数は、パラメータの値を 10 進数に設定します。

カテゴリ

SQL

構文

```
SQLSetParamDecimal(StatementID, ParameterNumber, Value, Precision, Scale);
```

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントでパラメータを識別する整数値

値
Value は、文字列、または 10 進数（123.456）を表す InTouch メッセージ型タグ、または InTouch メモリ実数型タグとなります。
パラメータの精度を確実にするために、実数型タグの代わりにメッセージ型タグを使用することを推奨します。ただし、Value は浮動小数点値（たとえば、DAServer から受け取る実数値）の場合、関数は動作を続行します。ただし、浮動小数点表現の制限により、高精度は保証されない可能性があります。

精度
数値の桁の合計数を指定する整数

スケール
小数点以下の桁数を指定する整数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例では、3 番目の SQL ステートメントの 2 番目のパラメータを 123.456 に設定します。精度は 6 桁で、スケールは小数点以下の 3 桁です。

```
SQLSetParamFloat(3, 2, 123.456, 6, 3);
```

参照項目

SQLPrepareStatement()

SQLSetParamFloat() 関数

SQLSetParamFloat() 関数は、パラメータの値を 64 ビット符号付き浮動小数点値に設定します。

カテゴリ

SQL

構文

```
SQLSetParamFloat(StatementID, ParameterNumber, Value);
```

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントでパラメータを識別する整数値

値
指定したパラメータの値として割り当てる 64 ビット符号付き浮動小数点数

例

この例では、3 番目の SQL ステートメントの 2 番目のパラメータを -5 に設定します。

```
SQLSetParamFloat(3, 2, -5);
```

参照項目

SQLPrepareStatement()

SQLSetParamInt() 関数

SQLSetParamInt() 関数は、パラメータの値を 16 ビット符号付き整数に設定します。

カテゴリ

SQL

構文

```
SQLSetParamInt(StatementID, ParameterNumber, Value);
```

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントでパラメータを識別する整数値

値
指定したパラメータの値として割り当てる 16 ビット符号付き整数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

この例では、3 番目の SQL ステートメントの 2 番目のパラメータを -5 に設定します。

```
SQLSetParamFloat(3, 2, -5);
```

参照項目

SQLPrepareStatement()

SQLSetParamLong() 関数

SQLSetParamLong() 関数は、パラメータの値を 32 ビット符号付きアナログ数に設定します。

カテゴリ

SQL

構文

```
SQLSetParamLong(StatementID, ParameterNumber, Value);
```

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントでパラメータを識別する整数値

値
指定したパラメータの値として割り当てる 32 ビット符号付きアナログ数

例

この例では、最初のステートメントの 3 番目のパラメータを 2.1e9 に設定します。

```
SQLSetParamLong(1, 3, 2.1e9);
```

参照項目

SQLPrepareStatement()

SQLSetParamNull() 関数

SQLSetParamNull() 関数は、SQL ステートメント内の指定したパラメータを Null に設定します。

カテゴリ

SQL

構文

```
SQLSetParamNull(StatementID, ParameterNumber, ParameterType, Precision, Scale)
```

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントでパラメータを識別する整数値

ParameterType
ParameterNumber 引数で指定されたパラメータに関連付けられているデータ タイプを指定する整数値
ParameterType 引数は、次の値に関連付けることができます。
0: 文字列

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

1：日付時刻
2：整数型
3：浮動小数点数
4：進数

精度
パラメータ データ タイプに関連付けられているデータの精度

スケール
10 進数値のスケール。この値は、Null に設定するパラメータに対して適用可能な場合にのみ必要です。

備考

Null 値との比較は、SQL Server の ANSI_NULLS オプションによって管理されます。SQL Server 7.0 では、このオプションはクエリー実行時ではなく、オブジェクト作成時に解決されます。ストアド プロシージャが SQL Server 7.0 で作成される場合、このオプションはデフォルトで ON であり、"WHERE MyField = NULL" などの句は常に NULL (FALSE) を返し、この句を使用する SELECT ステートメントから行は返されません。

比較 = または <> が TRUE または FALSE を返すには、ストアド プロシージャを作成するときにオプションを OFF に設定する必要があります。ANSI_NULLS が OFF に設定されていない場合、SQLSetParamNull() は期待どおりに動作しません。この場合、Null 値に対する比較は "WHERE MyField IS NULL" または "WHERE MyField IS NOT NULL" の構文を使用する必要があります。

例

このトランザクション セットは、ProductName が Null ではない Products テーブル内のすべての行を返します。

```
ANSI_NULLS をオフに設定
GO
CREATE PROCEDURE sp_TestNotNull @ProductParam varchar(255)
AS SELECT * FROM Products WHERE ProductName <> @ProductParam
GO
ANSI_NULLS をオンに設定
GO
```

InTouch は、以下の SQL Access スクリプトを実行できます。

```
ResultCode = SQLSetStatement(ConnectionId, "sp_TestNotNull");
ResultCode = SQLPrepareStatement(ConnectionId, StatementID);
ResultCode = SQLSetParamNull(StatementID, 1, 0, 0, 0);
ResultCode = SQLExecute(ConnectionId, BindList, StatementID);
ResultCode =SQLTransact( ConnectionID);
ResultCode = SQLClearStatement(ConnectionId, StatementID);
```

参照項目

SQLPrepareStatement()

SQLSetParamTime() 関数

SQLSetParamTime() 関数は、指定した時間パラメータを指定した文字列に設定します。

カテゴリ

SQL

構文

```
SQLSetParamTime(StatementID, ParameterNumber, Value)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

StatementID
クエリー内で SQL ステートメントを識別する整数値

ParameterNumber
StatementID 引数によって識別される SQL ステートメントの実際のパラメータ数

値
設定する実際の値。ParameterNumber 引数で指定されたパラメータを時間値に設定します。関数を実行しているコンピュータからの現在の日付は指定した時間に含まれます。

例

この例では、4 番目の SQL ステートメントの 2 番目のパラメータを 10:00AM に設定します。

```
ResultCode=SQLSetParamTime(1, 3, "10:00:00 AM");
```

参照項目

SQLPrepareStatement()

ステートメント パラメータのクリア

SQLClearParam() 関数は、指定したパラメータの値をクリアします。

SQLClearParam() 関数

SQLClearParam() 関数は、指定したパラメータの値をクリアします。**SQLExecute()** 関数を呼び出してクエリを実行する前に、**SQLSetParam*xxx*()** 関数の 1 つを再び呼び出してパラメータを再ロードする必要があります。

カテゴリ

SQL

構文

```
ResultCode=]SQLClearParam(StatementID,ParameterNumber);
```

引数

StatementID
SQLPrepareStatement() 関数が実行するときに返される整数値

ParameterNumber
ParameterNumber 引数は、変更する SQL ステートメント内の実際の引数を識別します。*StatementID* に関連付けられている ParameterNumber の値を、引数が数値であるか文字列であるかによって、ゼロまたはゼロ長の文字列に設定します。

参照項目

SQLPrepareStatement()、SQLExecute()

ステートメントの実行

SQLExecute() 関数は、InTouch スクリプト内で使用されると、ランタイム時に SQL クエリーを実行できます。

SQLExecute() 関数

SQLExecute 関数は、スクリプト内で SQL クエリーを実行します。ステートメントに SELECT が含まれている場合、BindList 引数は、データベース カラムを InTouch タグにバインドするために使用するバインド リストの名前を指定します。バインド リストが Null の場合、タグの関連付けは行われません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

SQL

構文

```
SQLExecute(ConnectionID,BindList,StatementID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

BindList
BindList 引数は、ゼロ長文字列にすることができます。*StatementID* が、行を返すクエリーと関連付けられている場合、論理テーブルは **SQLExecute()** の結果によって更新されます。実際のバインド リストが指定されている場合、結果は BindList 引数と関連付けられます。ゼロ長バインド リストは、StatementID が行を返すクエリーと関連付けられていないことが事前に分かっている場合に役立ちます。

StatementID
SQLPrepareStatement() 関数が使用されたときに SQL によって返される整数値

備考

エラーは関数の戻りとして返されます。ステートメントが準備されている場合、その状態から返されたステートメント ハンドルを渡す必要があります。ステートメントが準備されていない場合、ステートメント ハンドルはゼロである必要があります。

注記：準備されていないステートメントに対する SQLExecute() 関数は、一度のみ呼び出すことができます。ステートメントが準備されていれば、何度でも呼び出すことができます。

デフォルト ステートメントは接続 ID と関連付けられています。テキスト SQL ステートメント（SELECT、INSERT、DELETE、または UPDATE）、MS Access のクエリーの名前（パラメータあり/なし）、または MS SQL Server のストアド プロシージャの名前（パラメータあり/なし）のいずれかになります。

デフォルト ステートメントは、**SQLLoadStatement()**、**SQLSetStatement()**、および **SQLAppendStatement()** の関数によって変更されます。デフォルト ステートメントは、StatementID = 0 が指定されるたびに **SQLExecute()** によって使用されます。

例

この例では、lotquery.sql ファイルから SQL ステートメントをロードし、SELECT ステートメントの結果をバインド リストで指定された InTouch タグに配置します。

```
ResultCode = SQLLoadStatement (ConnectionID, "c:\myappdir\lotquery.sql");
ResultCode = SQLExecute(ConnectionID, "BindList", 0);
ResultCode =SQLTransact( ConnectionID);
```

この **SQLSetStatement()** 関数は、複雑なクエリーおよび 131 文字を超える文字列式に使用する必要があります。文字列式が 131 文字を超える場合は、**SQLAppend()** 関数を使用します。

```
SQLSetStatement(ConnectionID, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLExecute(ConnectionID, "BindList", 0);
```

前の例では、StatementID 引数がゼロに設定され、ステートメントを実行する前に、SQLPrepareStatement(Connection_Id, StatementID) を呼び出す必要はありません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

StatementID はこの SELECT を適切に終了するために SQLPrepare によって作成されていないため、**SQLClearStatement() 関数**の代わりに SQLEnd() 関数を使用します。

```
SQLSetStatement(Connection_Id, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLPrepareStatement(Connection_Id, StatementID);
SQLExecute(Connection_Id, StatementID);
```

上の例では、StatementID が SQLPrepareStatement 関数呼び出しによって作成され、SQLExceute 関数によって使用されます。この SELECT ステートメントを終了するには、スクリプト内で SQLClearStatement() 関数呼び出しを使用して、リソースと StatementID を解放します。

SQLExecute() 関数は、いくつかのストアド プロシージャをサポートしています。たとえば、"Select LotNo, LotName from LotInfo" という選択ステートメントを含む "LotInfoProc" という名前のデータベース サーバーでストアド プロシージャを作成すると仮定します。

使用しているデータベースのタイプに基づいてストアド プロシージャを実行する InTouch QuickScript を記述します。次の例は、SQL Server データベースに対してストアド プロシージャを実行するためのスクリプト ステートメントを示しています。

```
ResultCode = SQLSetStatement (ConnectionID,"LotInfoProc");
ResultCode = SQLExecute(ConnectionID, "BindList", 0);
ResultCode = SQLNext (ConnectionID);
{Get results of Select}
```

次の例は、Oracle データベースに対してストアド プロシージャを実行するためのスクリプト ステートメントを示しています。

```
ResultCode = SQLSetStatement (ConnectionID, "{CALL LotInfoProc}");
ResultCode = SQLExecute(ConnectionID, "BindList", 0);
ResultCode =SQLTransact( ConnectionID);
{Get results of Select}
```

参照項目

SQLConnect()、SQLPrepareStatement()

占有されているリソースの解放

SQLClearStatement 関数は、*StatementID* によって指定されているステートメントに関連付けられているデータベース リソースを解放します。

SQLClearStatement() 関数

SQLClearStatement() 関数は、*StatementID* 引数で指定されたステートメントに関連付けられているデータベース リソースを解放します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLClearStatement(ConnectionID, StatementID);
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

StatementID
SQLPrepareStatement() 関数が使用されたときに SQL によって返される整数値

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

参照項目

SQLConnect()、SQLPrepareStatement()

トランザクション セットの操作

SQL Access には、データベースのレコードを変更、挿入、更新、または削除する一組のトランザクション関数が含まれています。一般に、これらのトランザクションはスクリプト内にトランザクション セットの形式でグループ化されています。トランザクション セットは一度に実行されます。

SQLTransact() 関数

SQLTransact() 関数は、トランザクション セットと呼ばれる SQL ステートメント グループの開始を定義します。トランザクション セットは、単一のトランザクションとして処理されます。**SQLTransact()** 関数が実行されると、後続のすべての操作は **SQLCommit()** 関数が正常に実行されるまではデータベースに対して実行されません。

カテゴリ

SQL

構文

```
[ResultCode=]SQLTransact(ConnectionID)
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

例

この例では、トランザクション セットに 3 つの Insert ステートメントが含まれています。

```
ResultCode = SQLTransact(ConnectionID);
ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertEnd(ConnectionID, StatementID);
ResultCode =SQLTransact( ConnectionID);
```

参照項目

SQLCommit()、SQLRollback()

SQLCommit() 関数

SQLCommit() 関数は、トランザクション セットの終了を定義します。**SQLTransact()** 関数が実行されると、トランザクション セット内のすべての SQL ステートメントは **SQLCommit()** 関数が正常に実行されるまではデータベースに対して実行されません。

注記：SQLCommit() 関数を含む QuickScript を記述するときは、慎重に行ってください。トランザクション セット内の SQL ステートメントの数によって処理時間は増加します。

カテゴリ

SQL

構文

```
[ResultCode=]SQLCommit(ConnectionID)
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

例

このサンプル スクリプトには、3 つのデータベース挿入を行うトランザクション セットが含まれています。

```
ResultCode = SQLTransact(ConnectionID);
ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertEnd(ConnectionID, StatementID);
ResultCode = SQLCommit(ConnectionID);
```

参照項目

SQLRollback()、SQLTransact()、SQLCommit()

SQLRollback() 関数

SQLRollback() 関数は、最後のトランザクション セットを反転またはロールバックします。トランザクション セットは、**SQLTransact()** 関数と **SQLCommit()** 関数の間で発行されるコマンド グループです。

トランザクション セットは、単一のトランザクションとして処理されます。SQLTransact() 関数が実行されると、後続のすべての操作はデータベースに対して実行されません。SQLCommit() 関数が実行された後で、データベースに対するクエリー変更が行われます。SQLRollback() 関数は、SQLCommit() 関数の前に実行される場合にトランザクション セットをロールバックします。

カテゴリ

SQL

構文

```
[ResultCode=]SQLRollback(ConnectionID)
```

引数

ConnectionID
各データベース接続に SQLConnect() 関数によって割り当てられた番号（ID）を保持するメモリ整数型タグの名前

例

この例は、スクリプト内の **SQLTransact** 関数の前にデータベース値をロールバックします。

```
ResultCode =SQLTransact(ConnectionID);
ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertEnd(ConnectionID, StatementID);
ResultCode =SQLRollback(ConnectionID);
```

参照項目

SQLCommit()、SQLTransact()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイムで［ODBC データソース アドミニストレータ］ダイアログ ボックスを開く

SQLManageDSN() 関数を使用して、InTouch アプリケーションの実行中に Microsoft ODBC Manager を実行します。

SQLManageDSN() 関数

SQLManageDSN 関数は、InTouch アプリケーションの実行中に Microsoft ODBC Manager セットアップ プログラムを実行します。**SQLManageDSN()** は、スクリプト内で使用して、SQL Server または Access データベースのデータ ソース名を追加、削除、および変更できます。

カテゴリ

SQL

構文

```
SQLManageDSN(ConnectionId)
```

引数

ConnectionID
ConnectionId は使用されません。これは、古いバージョンの SQL Access と下位互換性を保持しています。したがって、任意の数字を関数に渡すことができます。Microsoft ODBC Manager を開くために関数を実行する前に、データベース接続を確立する必要はありません。

例

```
SQLManageDSN(0);
```

SQL エラー メッセージについて

このセクションでは、SQL Access 関数を使用する InTouch アプリケーションのトラブルシューティング方法について説明しています。最初のセクションでは、SQLErrorMsg() 関数について説明し、SQL 結果コードのテーブルおよびそれに対応するエラー メッセージが記載されています。2 番目のセクションは、特定のデータベース エラー メッセージを含むテーブルが含まれています。

SQLErrorMsg() 関数

すべての SQL 関数は、トラブルシューティングに使用できる結果コードを返します。SQLErrorMsg() 関数は、結果コードに関連付けられているエラー メッセージを返し、それを InTouch メッセージ型タグの値として割り当てます。

カテゴリ

SQL

構文

```
ErrorMsg = SQLErrorMsg(ResultCode);
```

引数

ResultCode
前の SQL 関数によって返される整数値。SQLErrorMsg() 関数は、InTouch メッセージ型タグの値を結果コードに関連付けられているメッセージに設定します。結果コードに関連付けられているエラー メッセージについては、「SQL エラー メッセージの理解」を参照してください。

備考

ドキュメントに記載されていない結果コードについては、データベースのマニュアルを参照してください。また、その他のエラー メッセージについては、Log Viewer を参照してください。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

win.ini ファイルの [InTouch] セクションの下に定義されている SQLTrace=1 フラグは、SQL Access スクリプトのデバッグに役立ちます。

例

この例では、SQL アクセス マネージャの結果コードに関連付けられているエラー メッセージを ErrorMsg メッセージ型タグに割り当てます。

```
ErrorMsg = SQLErrorMsg(ResultCode);
```

参照項目

SQLConnect()

SQL アクセス マネージャの結果コードとメッセージ

以下の表には、いつくかの一般的な SQL Access 結果コードと、それに対応するエラー メッセージおよび説明が一覧表示されています。

ResultCode	エラー メッセージ	説明
0	エラーは発生していません	SQL 関数がエラーを生じることなく正常に実行されました。
-1	<データベース プロバイダからのメッセージ>	ベンダー データベースからの特定のエラー メッセージ
-2	空白のステートメントは実行できません	SQLExecute(ConnectionId, BindList, 0) が、事前に空白でないステートメントを持つ SQLSetStatement または SQLLoadStatement を呼び出さことなく実行されます。
-4	返される値は Null です	データベースから読み取られる整数値または実数値が Null です。これは単に Log Viewer に送信される警告メッセージです。
-5	フェッチできる行がありません	テーブルの最後のレコードに到達しました。
-7	無効なパラメータ ID です	SQLSetParamChar()、SQLSetParamDate()、SQLSetParamDateTime()、SQLSetParamDecimal()、SQLSetParamFloat()、SQLSetParamInt()、SQLSetParamLong()、SQLSetParamNull()、または SQLSetParamTime() の関数が無効なパラメータ ID で呼び出されています。
-8	無効なパラメータ リストです	無効なパラメーター リストの例：1、2、3、5 (欠落している 4)。
-9	NULL パラメータに対する無効なタイプです	SQLSetParamNull 関数が無効な *Type* 引数値で呼び出されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ResultCode	エラー メッセージ	説明
-10	パラメータのデータ タイプは変更できません	SQLSetParamChar()、SQLSetParamDate()、SQLSetParamDateTime()、SQLSetParamDecimal()、SQLSetParamFloat()、SQLSetParamInt()、SQLSetParamLong()、SQLSetParamNull()、または SQLSetParamTime() の関数が既存のパラメータに対して異なるタイプで呼び出されています。
-11	ステートメントが正しく実行された後にパラメータを追加することはできません。	SQLSetParamChar()、SQLSetParamDate()、SQLSetParamDateTime()、SQLSetParamDecimal()、SQLSetParamFloat()、SQLSetParamInt()、SQLSetParamLong()、SQLSetParamNull()、または SQLSetParamTime() の関数が、ステートメントが正常に実行された後に新しいパラメータ ID に対して呼び出されています。
-12	無効な日付/時刻形式です	たとえば、SQLSetParamTime()、SQLInsertExecute()、または SQLUpdateCurrent() の実行時に無効な日付/時刻形式が検出されました。
-13	無効な 10 進数形式です	たとえば、SQLSetParamDecimal()、SQLInsertExecute()、または SQLUpdateCurrent() の実行時に無効な 10 進数形式が検出されました。
-14	無効な通貨形式です	たとえば、SQLInsertExecute() または SQLUpdateCurrent() の実行時に無効な通貨形式が検出されました。
-15	この操作に対する無効なステートメント タイプです	SQLInsertEnd が SQLPrepareStatement() によって作成されたステートメント ID に対して呼び出されているか、SQLClearStatement() が SQLInsertPrepare() によって作成されたステートメント ID に対して呼び出されています。
-1001	メモリ不足です	この関数を実行するために十分なメモリがありません。
-1002	無効な接続	*ConnectionId* 引数に渡された値が有効ではありません。
-1003	バインド リストが見つかりません	指定したバインド リスト名が存在しません。
-1004	テンプレートが見つかりません	指定したテーブル テンプレート名が存在しません。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ResultCode	エラー メッセージ	説明
-1005	内部エラー	内部エラーが発生しました。テクニカル サポートに連絡してください。
-1006	文字列が Null です	警告 - データベースから読み取られた文字列が Null です。これは単に Log Viewer に送信される警告メッセージです。
-1007	文字列が切れています	警告 - データベースから読み取られた文字列が 131 文字よりも長く、選択したときに切れています。この警告は、Log Viewer に送信されます。
-1008	場所を示す文字列がありません	Delete に Where 句がありません。
-1009	接続に失敗	データベースへの接続の失敗に関する詳細情報については、Log Viewer を確認してください。
-1010	接続文字列の DB= portion で指定されたデータベースは存在しません	指定したデータベースが存在しません。
-1011	行が選択されていません	SQLSelect() 関数または SQLExecute() 関数を実行せずに SQLNumRows()、SQLFirst()、SQLNext()、SQLLast()、または SQLPrev() 関数が呼び出されています。
-1013	読み込むファイルがありません	SQLLoadStatement() 関数が、見つからないファイル名で呼び出されています。

ベンダー データベースからのエラー メッセージが ResultCode -1 を返します。SQL Access 関数 ResultCode は常に -1 ですが、メッセージがデータベース プロバイダから正確にコピーされます。

Oracle データベースを使用しているときに発生するエラー メッセージの場合、特定のエラー メッセージおよび解決方法については Oracle Server のマニュアルを参照してください。

以下の表には、Microsoft SQL Server または Access データベースを使用しているときに発生する可能性がある一般的なエラー メッセージが一覧表示されています。

エラー メッセージ	解決方法
一度に複数のステートメントをアクティブにすることはできません	SQLSelect() 関数を呼び出した後で、SQL コマンドを実行しようとしています。SQLEnd() を実行して SQLSelect() からリソースを解放するか、2 番目のステートメントに対して別の *ConnectionId* を使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

エラー メッセージ	解決方法
コマンドを処理するために使用できる十分なメモリがありません	クライアント ワークステーションの再起動を試行します。
無効なオブジェクト名テーブル名です	テーブル名が使用しているデータベースに存在しません。DB=database という名で試行します。

特定のエラー メッセージおよび解決方法については、Microsoft SQL Server のマニュアルを参照してください。

予約語リスト

このセクションでは、SQL Access バインド リスト、テーブル テンプレート、および ODBC インターフェイスでの使用から除外されているキーワードを一覧表示しています。

予約キーワードがバインド リストまたはテーブル テンプレート内のカラム名として使用されている場合、Log Viewer にエラー メッセージが表示されます。エラーのタイプは、使用されている ODBC ドライバとキーワードが検出される場所によって異なります。たとえば、最も一般的なエラーの一つは、DATE および TIME をバインド リストまたはテーブル テンプレートのカラム名に使用することです。このエラーを回避するには、たとえば "aDATE" および "aTIME" など少し異なる名前を使用します。

予約キーワードは、InTouch SQL Access によって使用される SQL（Structured Query Language）を定義します。キーワードは、使用されている特定の ODBC ドライバによっても認識されます。SQL コマンドを正しく解釈できない場合、SQL アクセス マネージャは Log Viewer で表示できるエラー メッセージを生成します。

次のアルファベット順のリストは、SQL Access および ODBC の予約キーワードを示しています。

ABSOLUTE	ADA	追加
ALL	ALLOCATE	ALTER
AND	ANY	ARE
AS	ASC	ASSERTION
AT	AUTHORIZATION	AVG
BEGIN	BETWEEN	BIT
BIT_LENGTH	BY	CASCADE
CASCADED	CASE	CAST
CATALOG	CHAR	CHAR_LENGTH
CHARACTER	CHARACTER_LENGTH	CHECK
CLOSE COALESCE	COBOL	COLLATE
COLLATION	COLUMN	COMMIT
CONNECT	CONNECTION	CONSTRAINT

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

CONSTRAINTS	CONTINUE	CONVERT
CORRESPONDING	COUNT	CREATE
CURRENT	CURRENT_DATE	CURRENT_TIME
CURRENT_TIMESTAMP	CURSOR	日付
DAY	DEALLOCATE	DEC
DECIMAL	DECLARE	DEFERRABLE
DEFERRED	削除	DESC
DESCRIBE	DESCRIPTOR	DIAGNOSTICS
DICTIONARY	DISCONNECT	DISPLACEMENT
DISTINCT	DOMAIN	DOUBLE
DROP	ELSE	END
ESCAPE	EXCEPT	EXCEPTION
EXEC	EXECUTE	EXISTS
EXTERNAL	EXTRACT	FALSE
FETCH	FIRST	FLOAT
FOR FOREIGN	FORTRAN	FOUND
FROM FULL	GET	GLOBAL
GO	GOTO	GRANT
GROUP	HAVING	HOUR
IDENTITY	IGNORE	IMMEDIATE
IN	INCLUDE	INDEX
INDICATOR	INITIALLY	INNER
INPUT	INSENSITIVE	挿入
INTEGER	INTERSECT	INTERVALL
INTO	IS	ISOLATION
JOIN	KEY	LANGUAGE
LAST	LEFT	LEVEL
LIKE	LOCAL	LOWER
MATCH	MAX	MIN
MINUTE	MODULE	MONTH
MUMPS	NAMES	NATIONAL

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

NCHAR	NEXT	NONE
NOT	NULL	NULLIF
NUMERIC	OCTET_LENGTH	OF
ｵﾌ	ｵﾝ	ONLY
OPEN	OPRN	OPTION
または	ORDER	OUTER
出力	OVERLAPS	PARTIAL
PASCAL	PLI	POSITION
PRECISION	PREPARE	PRESERVE
PRIMARY	PRIOR	PRIVILEGES
PROCEDURE	PUBLIC	RESTRICT
REVOKE	RIGHT	ROLLBACK
ROWS	SCHEMA	SCROLL
SECOND	SECTION	選択
SEQUENCE	設定	SIZE
SMALLINT	SOME	SQL
SQLCA	SQLCODE	SQLERROR
SQLSTATE	SQLWARNING	SUBSTRING
SUM	SYSTEM	TABLE
TEMPORARY	THEN	TIME
TIMESTAMP	TIMEZONE_HOUR	TIMEZONE_MINU
TO	TRANSACTION	TRANSLATE
TRANSLATION	TRUE	UNION
UNIQUE	UNKNOWN	UPDATE
UPPER	USAGE	USING
値	VALUES	VARCHAR
VARING	VIEW	WHEN
WHENEVER	WHERE	WITH
WORK	YEAR	

© 2015-2026 AVEVA Group Limited またはその子会社。 All rights reserved.

16 ペン トレンド ウィザードの使用

InTouch ウィザードを使用して、最大 16 個のタグ変数からのデータを表示できるリアルタイム トレンドおよび履歴トレンドを作成できます。16 ペン トレンドは、InTouch のインストール時にインストールできる補足コンポーネントです。

16 ペン トレンド ウィザードは、他の InTouch チャート ウィザードと同じ様に設定できます。16 ペン トレンド ウィザードを使用すると、以下のトレンド プロパティを設定できます。

- 各トレンド ペンに割り当てられているタグ変数
- トレンドの線の太さと色
- 履歴トレンドの開始日時と終了日時
- リアルタイム トレンドの更新率と時間範囲
- トレンド タグ変数に割り当てられている最小工学単位および最大工学単位
- トレンドの時間軸の主目盛りと小目盛り
- トレンドの縦軸の主目盛りと小目盛り

16 ペン トレンドの作成

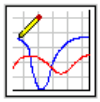
トレンドはオプションの 16 ペン トレンド ウィザードを WindowMaker から選択することによって作成できます。

16 ペン リアルタイム トレンドまたは履歴トレンドを作成する

1. 16 ペン トレンドを配置するウィンドウを WindowMaker で開きます。
2. ［**描画**］メニューの［**挿入**］グループで［**ウィザード**］を選択します。

   ［**ウィザードの選択**］ダイアログ ボックスが表示されます。
3. ウィザードのリストから［**トレンド**］を選択します。

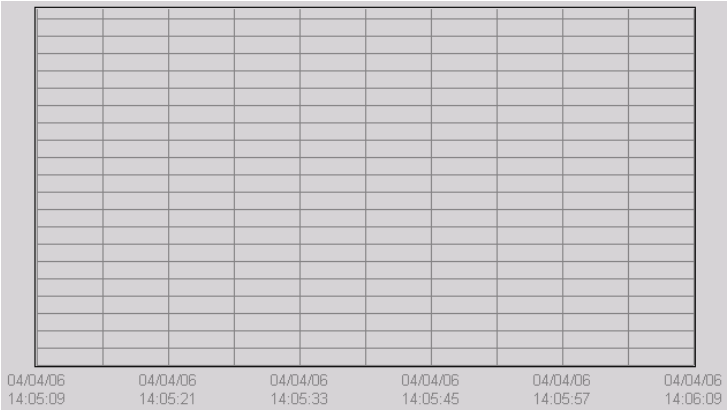
   ［**ウィザードの選択**］ダイアログ ボックスの右ペインに、トレンド ウィザードのアイコンのセットが表示されます。
4. ［**16 ペン トレンド**］ウィザードを選択し、［**OK**］を選択します。

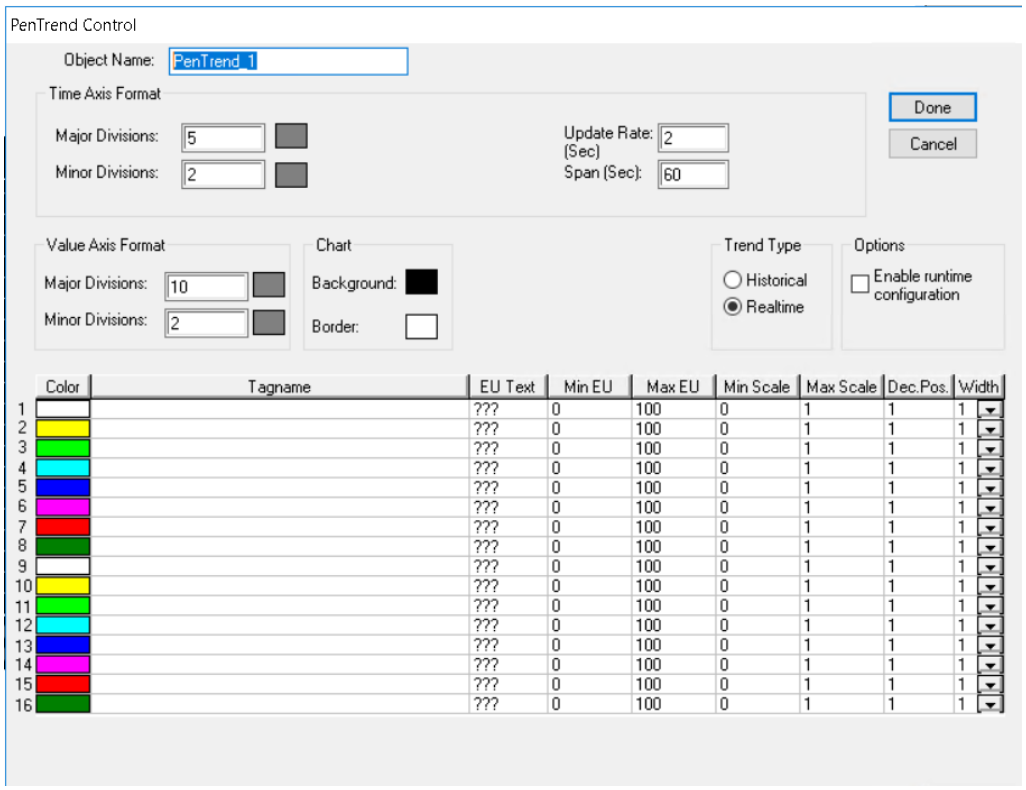
   ［**ウィザードの選択**］ダイアログ ボックスが閉じて目的のウィンドウが再度表示されます。
5. ウィンドウ内を選択して、16 ペン トレンドを選択します。

   ウィザードによって、16 ペン トレンド テンプレートがウィンドウに配置されます。

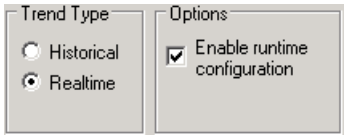
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. 16 ペン トレンド テンプレートをダブルクリックして、[**PenTrend コントロール**] ダイアログ ボックスを開きます。



7. [**トレンド タイプ**] 領域で、作成するトレンドのタイプとして [**履歴**] または [**リアルタイム**] を選択します。



[**PenTrend コントロール**] ダイアログ ボックスには、選択したトレンドのタイプに基づいて適切な時間および更新オプションが自動的に表示されます。

8. [**オプション**] 領域で、[**ランタイム構成を有効にする**] オプションをオンまたはオフにします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

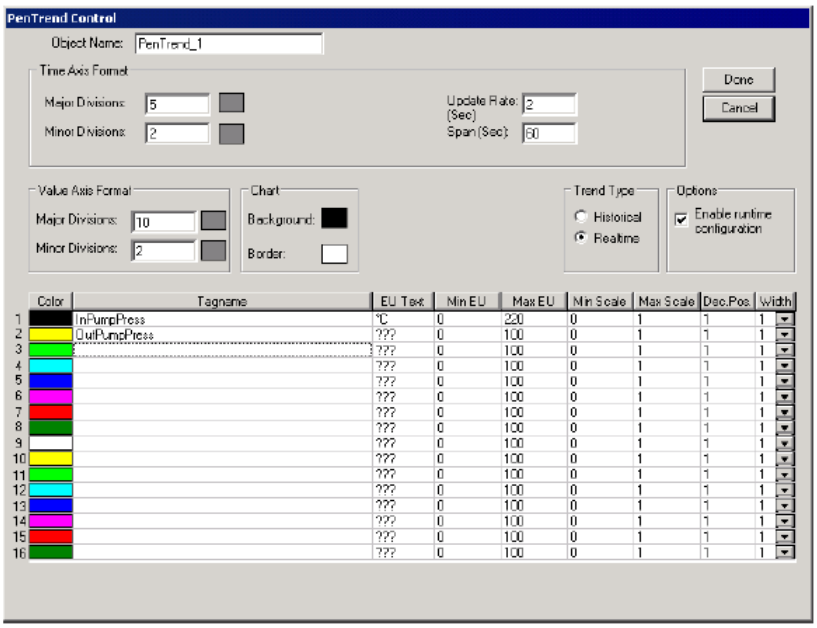
このオプションをオンにすると、16 ペン トレンドの実行中にオペレータはプロパティを変更できます。

トレンド グラフに表示するタグの設定

16 ペン トレンド ウィザードを使用すると、タグ変数をトレンド ペンに割り当てることができます。16 ペン トレンド ウィザードには、トレンドに表示されるタグ変数プロパティを指定する一組のカラムが含まれます。これらの列では、タグ名ディクショナリからタグに割り当てられるデフォルトのプロパティ値が使用されます。これらの割り当てられたタグ変数値は、トレンドを設定するときに他の値を指定することによって上書きできます。

16 ペン トレンド タグ変数を設定する

1. 16 ペン トレンド テンプレートを含むウィンドウを開きます。
2. 16 ペン トレンドをダブルクリックします。［**PenTrend コントロール**］ダイアログ ボックスが表示され、トレンド ペンに関連付けられているタグ変数を指定するためのグリッド領域が下部付近に表示されます。



3. ［**オブジェクト名**］ボックスで、16 ペン トレンドに名前を割り当てます。

   デフォルトの名前は PenTrend_1 で、新しいトレンドを作成するたびにこの名前の数が増分します。

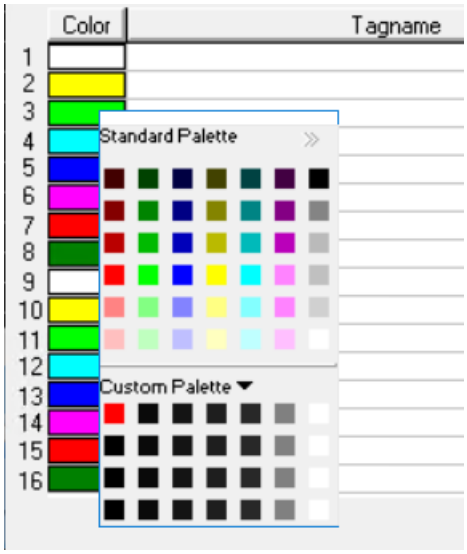
4. ［**タグ名**］ボックスに、グリッドの左に一覧表示されているペン番号に関連付けられているタグ変数の名前を入力します。

   ［**タグ名**］の下にあるセル内をダブルクリックすると、［**タグを選択してください**］ダイアログ ボックスが表示されます。タグを［**タグを選択してください**］ダイアログ ボックスから選択することによって、ペンにタグを割り当てることができます。

   注記：タグ変数を削除するには、タグ変数名を含む［Tagname］を選択して、キーボードのスペースバーを押します。

5. ［**色**］列でカラー ボックスをクリックして、カラー パレットを開きます。ペンの色を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. ［**EU テキスト**］カラムに、ランタイムでそれぞれのペンのペン軸のヘッダ テキストとして最初に使用するテキストを入力します。

   このテキストは、ペンがアクティブに設定されているときの軸テキストです。［**EU テキスト**］カラムには、初めはタグ名ディクショナリからのタグの工学単位が割り当てられています。これらの 16 ペン トレンド用のデフォルトの工学単位を上書きできます。

7. ［**工学値最小値**］カラムに、ペンに割り当てる最小工学単位値を入力します。

   ［**工学値最小値**］カラムには、初めはタグ名ディクショナリからのタグの最小工学単位値が表示されます。16 ペン トレンドにのみ適用する別の最小工学単位値を割り当てることができます。

8. ［**工学値最大値**］カラムに、ペンに割り当てる最大工学単位値を入力します。

   ［**工学値最大値**］カラムには、初めはタグ名ディクショナリからのタグの最大工学単位値が表示されます。16 ペン トレンドにのみ適用する別の最大工学単位値を割り当てることができます。

注記 : 最小/最大工学単位は、履歴トレンド データを表示するために非常に重要です。履歴トレンドは、工学単位スケールの 0 ～100 % を表示します。

---

1. ［**最小スケール**］カラムに、それぞれの工学単位スケールの最小ペン軸グリッドを計算するためにランタイムで最初に使用するパーセンテージを入力します。
2. ［**最大スケール**］カラムに、それぞれの工学単位スケールの最大ペン軸グリッドを計算するためにランタイムで最初に使用するパーセンテージを入力します。
3. ［**小数点以下の桁数**］カラムに、ペン軸グリッドをラベル付けする場合に、ランタイム時に最初に使用する小数点以下の桁数を入力します。
4. ［**幅**］カラムで、トレンドに表示されるデータ値を描画するペン線の幅をピクセル単位で選択します。
5. 次の手順に進み、16 ペン トレンドのトレンド時間と更新率を更新します。

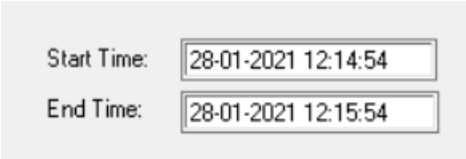
トレンドの時間範囲と更新レートの設定

［**PenTrend コントロール**］ダイアログ ボックスには、リアルタイムまたは履歴の 16 ペン トレンドのどちらを作成しているかに基づいて異なるオプションが表示されます。履歴トレンドに対しては時間範囲、リアルタイム トレンドに対しては更新レートを設定できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

16 ペン履歴トレンドの時間範囲を設定する

1. ウィンドウで 16 ペン履歴トレンドをダブルクリックします。［**Pen Trend Control**］ ダイアログ ボックスが表示され、トレンドの開始日時および終了日時を設定するためのオプションが表示されます。



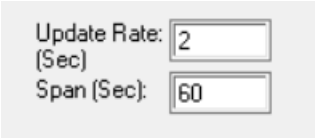
2. 履歴トレンドの開始日時および終了日時を設定します。

   開始日時および終了日時には、次の形式を使用します。

   DD-MM-YYYY HH:MM:SS AM/PM

16 ペン リアルタイム トレンドの更新レートを設定する

1. ウィンドウ内の 16 ペン リアルタイム トレンド オブジェクトをダブルクリックします。［**PenTrend コントロール**］ダイアログ ボックスが表示され、リアルタイム トレンドの更新レートと範囲を設定するためのオプションが表示されます。



2. ［**Update Rate**］ボックスに、履歴トレンドのリフレッシュ間隔を秒数で入力します。
3. ［**Span**］ボックスに、トレンドに表示されるリアルタイム間隔の秒数を入力します。

トレンド表示オプションの設定

16 ペン トレンド ウィザードを使用して、トレンドの視覚的外観を設定できます。

16 ペン トレンドの表示オプションを設定する

1. WindowMaker で **16 ペン トレンド**をダブルクリックします。

   ［**PenTrend コントロール**］ダイアログ ボックスが表示されます。

2. ［**時間軸の形式**］領域で、主目盛りの数を［**主目盛り**］に入力します。このオプションは、トレンドの横軸に主目盛りを設定します。
3. 主目盛りの線に別の色を割り当てる場合は、［Major Divisions］の右にあるカラー ボックスを選択してカラー パレットを開き、色を選択します。 それ以外の場合は、この手順を省略して主目盛りの線にデフォルトで割り当てられている色を受け入れます。
4. ［**小目盛り**］ボックスに、トレンドの横軸に表示される小目盛りの数を入力します。
5. 小目盛りの線の色を選択します。
6. ［**値軸の形式**］領域で、主目盛りの数を［**主目盛り**］に入力します。このオプションでは、縦の値軸に表示される主目盛り数が設定されます。
7. 主目盛りの色を設定します。
8. ［**小目盛り**］ボックスに、トレンドの縦軸に表示される小目盛りの数を入力します。
9. 小目盛りの色を設定します。

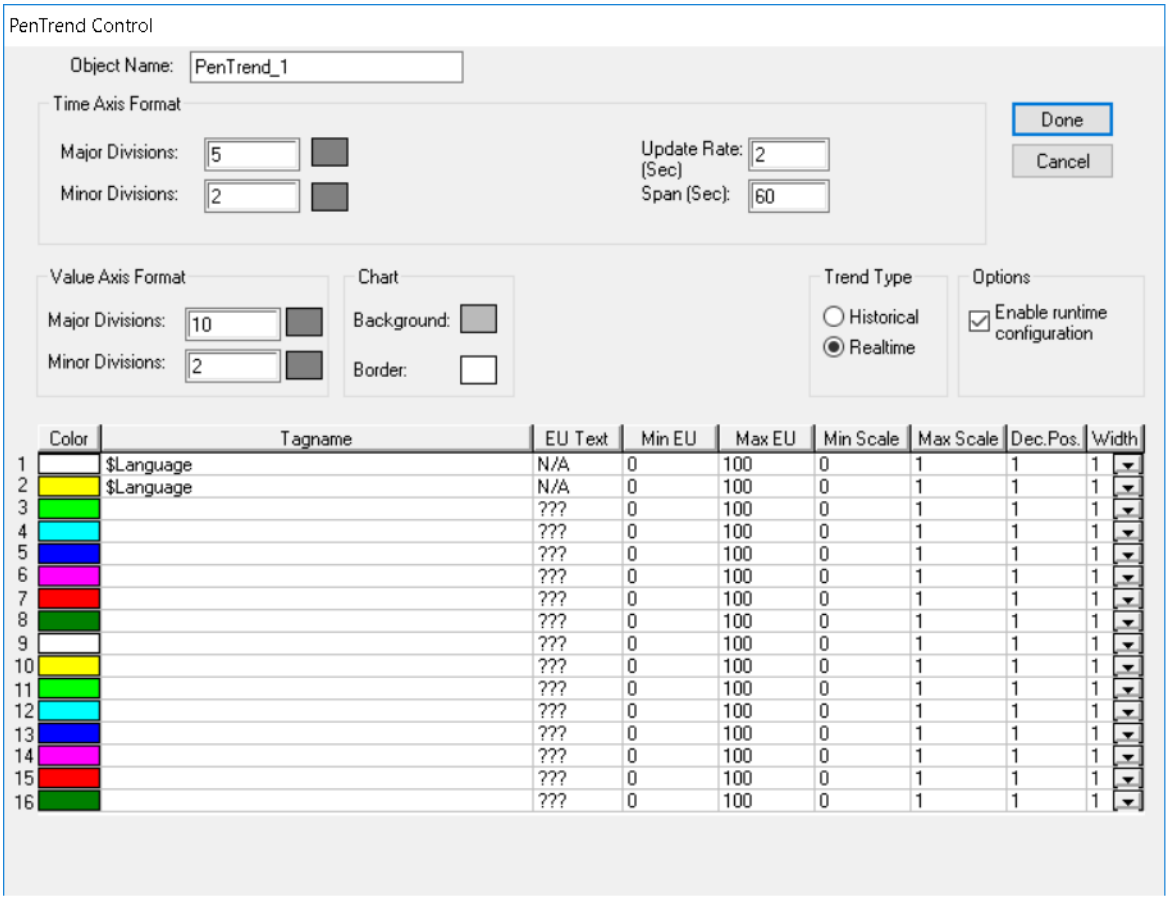
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

10. ［チャート］領域で、トレンドの背景色および境界色を選択します。
11. ［完了］を選択して、16 ペン トレンドに行われた設定変更を保存します。

ランタイム時のトレンド設定の変更

［**PenTrend コントロール**］ダイアログ ボックスで［**ランタイム構成を有効にする**］オプションが選択されている場合、アプリケーションが実行している間、オペレータは 16 ペン トレンドの一部の特性を変更できます。

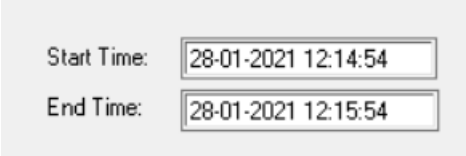
ランタイムでの 16 ペン トレンドに対する変更は永続的なものではありません。オペレータが WindowViewer を閉じ、再びアプリケーション ウィンドウを開始した場合、16 ペン トレンドでは WindowMaker で元々定義されている設定が保持されます。次の図は、16 ペン トレンドを実行中に選択したときに表示される［**PenTrend コントロール**］ダイアログ ボックスを示しています。



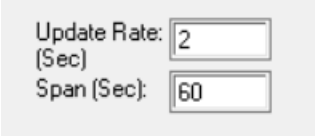
ランタイムでは、以下の内容を変更できます。

- トレンド ペンに割り当てられているタグ変数または式
- トレンド タグ変数または式の特性
- トレンドのタイプ（履歴またはリアルタイム）
- 履歴トレンドの日時範囲

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



- リアルタイム トレンドの更新レートおよび間隔



［完了］をクリックした後で、現在の WindowViewer アプリケーション セッションの間、トレンドでは設定変更が保持されます。

スクリプトの使用による 16 ペン トレンド ウィザードのコントロール

QuickScript で一組の関数を使用すると、ランタイムで 16 ペン トレンド オブジェクトを管理できます。たとえば、チャートにペンを接続したり、チャートに新しいイベントを追加したり、グリッドを削除または再描画したり、スクータを削除または再描画したりすることができます。

ptGetTrendType() 関数

ptGetTrendType() 関数はスクリプトで使用して、16 ペン トレンドの現在のモードが履歴データまたはリアルタイム データのどちらであるかを示す値を返すことができます。

カテゴリ

ペン トレンド

構文

```
ptGetTrendType(TrendName);
```

引数

TrendName
トレンドの名前。*TrendName* は、文字列定数またはメッセージ型タグ変数である必要があります。

戻り値

トレンド タイプを返します。

0 = 履歴トレンド

1 = リアルタイム スクロールなし

2 = リアルタイム トレンド

例

以下の例では、PumpPress トレンドが履歴データまたはリアルタイム データのどちらを表示しているかを示す値を返しています。

```
ptGetTrendType("PumpPress");
```

ptLoadTrendCfg() 関数

ptLoadTrendCfg() 関数はスクリプトで使用して、ファイルからトレンド設定値をロードできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

ペン トレンド

構文

```
ptGetTrendCfg(TrendName,FileName);
```

引数

TrendName
トレンド オブジェクトの名前。*TrendName* に割り当てられる値は、文字列定数またはメッセージ型タグである必要があります。

FileName
設定ファイルの名前。設定ファイルへのフォルダ パスは *FileName* 引数に含める必要があります。

例

TankFarm トレンドは、C:\TrendCfg.txt ファイルからの値で設定されます。

```
pt ReplicasTrendCfg("TankFarm","C:\TrendCfg.txt");
```

ptPanCurrentPen() 関数

ptPanCurrentPen() 関数はスクリプトで使用して、16 ペン トレンドのペンを縦の値軸で上方または下方にスクロールできます。縦スクロールは引数値として指定されるトレンドの主ユニットまたは小ユニットの単位数によって決定されます。

カテゴリ

ペン トレンド

構文

```
ptPanCurrentPen(TrendName,MajorUnits, MinorUnits);
```

引数

TrendName
トレンド オブジェクトの名前。*TrendName* は、文字列定数またはメッセージ型タグ変数である必要があります。

MajorUnits
主目盛りで定義されたユニット数によってスクロールする乗数。負の数は縦軸の下方へのスクロールを示しています。

MinorUnits
小目盛りで定義されたユニット数によって追加スクロールするための乗数。負の数は縦軸の下方へのスクロールを示しています。

例

この例では、ペンを主目盛りで上に 1 つスクロールします。

```
ptPanCurrentPen("TrendName", 1, 0);
```

この例では、ペンを小目盛りで半分上にスクロールします。

```
ptPanCurrentPen("TrendName", 1, 0);
```

この例では、ペンを主目盛りで 2 つ、小目盛りで半分下にスクロールします。

```
ptPanCurrentPen("TrendName", 1, 0);
```

この例では、主目盛りで 1 つ上に、小目盛りで 2 つ下にスクロールします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

```
ptPanCurrentPen("TrendName", 1, 0);
```

ptPanTime() 関数

ptPanTime() 関数はスクリプトで使用して、16 ペン トレンドのペンを指定したトレンドの主ユニットまたは小ユニット数に基づいて、横軸で左または右にスクロールできます。

カテゴリ

ペン トレンド

構文

```
ptPanTime(TrendName、MajorUnits、MinorUnits);
```

引数

TrendName
トレンド オブジェクトの名前。*TrendName* は、文字列定数またはメッセージ型タグ変数である必要があります。

MajorUnits
横軸の主目盛りの数によってスクロールする乗数。負の数はトレンドで左にスクロールすることを示しています。

MinorUnits
小目盛りで定義されたユニット数によって追加スクロールするための乗数。負の数はトレンドで左にスクロールすることを示しています。

備考

開発時に［**PenTrend Control**］ダイアログ ボックスで指定した［Major Division］および［Minor Division］の設定は、スクロール量が計算される基本となります。トレンドの時間範囲が 120 秒で、主目盛りの値が 10、および小目盛りの値が 2 の場合、主目盛りを 12 秒ごと、小目盛りを 6 秒ごとにスクロールするトレンドとなります。ptPanTime("TrendName",1,0.5) 関数では、時間軸が 1*12 + 0.5*6 = 15 秒でスクロールされます。

例

この例では、ペンを主目盛りで 1 つ横軸で右にスクロールします。

```
ptPanCurrentPen("TrendName", 1, 0);
```

この例では、ペンを小目盛りで 0.5 ほど横軸で右にスクロールします。

```
ptPanCurrentPen("TrendName", 1, 0);
```

この例では、ペンを主目盛りで 2.5 ほど横軸で左にスクロールします。

```
ptPanCurrentPen("TrendName", 1, 0);
```

この例では、ペンを主目盛りで 1 つ右に、小目盛りで 2 つ左にスクロールします。

```
ptPanCurrentPen("TrendName", 1, 0);
```

ptPauseTrend() 関数

ptPauseTrend() 関数はスクリプトで使用して、16 ペン トレンドがグラフを更新することを一時的に停止できます。トレンドは、ptPauseTrend を値 0 で再度呼び出すまで停止したままです。

カテゴリ

ペン トレンド

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
ptPauseTrend(TrendName, Value);
```

引数

TrendName
トレンド オブジェクトの名前。*TrendName* は、文字列定数またはメッセージ型タグ変数である必要があります。

値
1 はトレンド更新を一時停止します。0 はトレンド更新を再開します。

例

この例では、*Value* 引数が 1 の間、16 ペン トレンドへのさらなる更新が一時停止します。

```
ptPauseTrend ("TrendName",1);
```

ptSaveTrendCfg() 関数

ptSaveTrendCfg() 関数はスクリプトで使用して、トレンドの現在の設定値をファイルに保存できます。

カテゴリ

ペン トレンド

構文

```
ptSaveTrendCfg(TrendName,FileName);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグのいずれかである必要があります。

FileName
トレンドの設定値を保存するファイルの名前。設定ファイルへのフォルダ パスは FileName 引数で指定できます。

例

ptSaveTrendCfg() 関数は、PumpTrend 16 ペン トレンドからの値を C:\Config.txt ファイルに保存します。

```
ptSaveTrendCfg ("PumpTrend", "C:\Config.txt")
```

ptSetCurrentPen() 関数

ptSetCurrentPen() 関数はスクリプトで使用して、割り当てられた数によってペンを選択して、ペン軸を管理できます。

カテゴリ

ペン トレンド

構文

```
ptSetCurrentPen(TrendName, PenNum);
```

引数

TrendName
トレンドの名前。文字列定数またはメッセージ型タグのいずれかである必要があります。

PenNum
現在のトレンド ペンとして割り当てられているペンの数（1 ～ 16）。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

ptSetCurrentPen() 関数はペン 2 を PumpPress トレンドの現在のペンとして割り当てます。

```
ptSetCurrentPen("PumpPress",2);
```

ptSetPen() 関数

ptSetPen() 関数はスクリプトで使用して、トレンド ペンにタグ変数を割り当てることができます。

カテゴリ

ペン トレンド

構文

```
ptSetPen(TrendName,PenNum,TagName);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグ変数である必要があります。

PenNum
現在のトレンド ペンとして割り当てられているペンの数。

TagName
トレンド ペンとして割り当てられているタグ変数の名前。

例

ptSetPen() 関数は **PumpInP** タグ変数を PumpPress トレンドのペン 2 に割り当てます。

```
ptSetPen("PumpPress",2,"PumpInP");
```

ptSetPenEx() 関数

ptSetPenEx() 関数はスクリプトで使用して、特定のトレンド ペンにタグ変数を割り当て、タグ変数ディクショナリで指定されたタグ変数の設定値を上書きできます。

カテゴリ

ペン トレンド

構文

```
ptSetPenEx(TrendName, PenNum, TagName, minEu, maxEU, minPercent, maxPercent, Decimal, EU);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグのいずれかである必要があります。

PenNum
現在のトレンド ペンとして割り当てられているペンの数。

TagName
トレンド ペンとして割り当てられているタグ変数の名前。

minEU
指定したタグ変数の最小工学単位値。

maxEU
指定したタグ変数の最大工学単位値。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

minPercent
それぞれの工学単位スケールの最小ペン軸グリッドを計算するためにランタイムで最初に使用するパーセンテージ。

maxPercent
それぞれの工学単位スケールの最大ペン軸グリッドを計算するためにランタイムで最初に使用するパーセンテージ。

10 進数
トレンドのタグ変数値の小数点精度。

EU
タグ変数の工学単位のラベル。

例

ptSetPenEx() 関数は **PumpInP** タグ変数を PumpPress トレンドのペン 2 に割り当てます。タグ変数の工学単位範囲は 0 ～ 1500 の間で設定され、ユニットは PSI です。グリッドのパーセンテージ範囲は 0 ～ 1 で、小数点精度は 2 に設定されています。

```
ptSetPenEx ("PumpPress", 2, "PumpInP", 0, 1500, 0, 1, 2, "PSI");
```

ptSetTimeAxis() 関数

ptSetTimeAxis() 関数はスクリプトで使用して、トレンドの開始日時および終了日時を設定できます。

カテゴリ

ペン トレンド

構文

```
ptSetTimeAxis(TrendName,StartDateTime,EndDateTime);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグのいずれかである必要があります。

StartDateTime
トレンドを開始する日時。開始日時の形式は dd/mm/yyyy hh:mm:ss AM/PM です。

EndDateTime
トレンドを終了する日時。終了日時の形式は dd/mm/yyyy hh:mm:ss AM/PM です。

例

ptSetTimeAxis() 関数は、トレンドの開始日時および終了日時を 2007 年 5 月 22 日月曜日の 8:30 から開始する 25 時間単位で設定します。

```
ptSetTimeAxis ("PumpPress", "05/22/2007 08:30:00 AM", "05/23/2007 09:30:00 AM");
```

ptSetTimeAxisToCurrent() 関数

ptSetTimeAxisToCurrent() 関数はスクリプトで使用して、現在のチャートの範囲およびチャートの終了時間を計算できます。

カテゴリ

ペン トレンド

構文

```
ptSetTimeAxisToCurrent(TrendName);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

TrendName
トレンド オブジェクトの名前。*TrendName* は、文字列定数またはメッセージ型タグ変数である必要があります。

例

ptSetTimeAxisToCurrent() 関数は、PumpPress トレンドの終了日時を現在の日時に設定します。

```
ptSetTimeAxisToCurrent("PumpPress");
```

ptSetTrend() 関数

ptSetTrend() 関数はスクリプトで使用して、16 ペン トレンドへの更新を一時停止または再開します。

カテゴリ

ペン トレンド

構文

```
ptSetTrend(TrendName, EnableUpdates);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグのいずれかである必要があります。

EnableUpdates
1 はトレンドの更新を開始します。0 はトレンド更新を停止します。

例

ptSetTrend() 関数は PumpPress トレンドを更新します。

```
ptSetTrend("PumpPress",1);
```

ptSetTrendType() 関数

ptSetTrendType() 関数はスクリプトで使用して、トレンドが履歴データまたはリアルタイム データのどちらを表示しているのかを指定できます。

カテゴリ

ペン トレンド

構文

```
ptSetTrendType(TrendName, TrendType);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグ変数のいずれかである必要があります。

TrendType
1 は履歴トレンドを示します。2 はリアルタイム トレンドを示します。

例

ptSetTrendtype() 関数は PumpPress トレンドがリアルタイム データを表示するように指定します。

```
ptSetTrendType("PumpPress",2);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ptZoomCurrentPen() 関数

ptZoomCurrentPen() 関数はスクリプトで使用して、トレンドの Y 軸に表示される値範囲を変更できます。トレンドの縦軸の範囲は、指定したズーム率によって増減できます。

カテゴリ

ペン トレンド

構文

```
ptZoomCurrentPen(TrendName,ZoomFactor);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグのいずれかである必要があります。

ZoomFactor
1.0 よりも大きい数値を割り当てると、現在の範囲限界をズーム因数で乗算することによって、トレンドの値範囲が増加します。1.0 よりも小さいズーム因数を割り当てると、トレンドの縦軸に表示される値範囲が減少します。

備考

<1>ズーム率は、現在のペンの</1> Y <2>軸の既存の範囲に適用されます。</2> たとえば、トレンドの Y 軸が -50 ～ 50 の範囲で始まり、2.0 の比率でズームすると、新しい範囲は -100 ～ 100 となります。2.0 の比率でもう一度ズームすると、新しい範囲は -200 ～ 200 となります。ズーム率は元の Y 軸範囲ではなく、現在有効である範囲に適用されます。

ズーム率は各トレンドのペンを実行している間維持されます。**ptSetCurrentPen()** 関数を使用して 1 つのペンから別のペンに切り替える場合、Y 軸の値範囲は選択したペンに対して現在のスケールを反映します。

例

ptZoomCurrentPen 関数は、トレンドの縦 Y 軸に表示される「PumpPress」というトレンドにある現在のタグ変数の Y 軸範囲を 2 倍にします。

```
ptZoomCurrentPen("PumpPress",2);
```

ptZoomTime() 関数

ptZoomTime() 関数はスクリプトで使用して、トレンドの横軸に表示される時間範囲を変更できます。

カテゴリ

ペン トレンド

構文

```
ptZoomTime(TrendName,Zoom);
```

引数

TrendName
トレンド オブジェクトの名前。文字列定数またはメッセージ型タグのいずれかである必要があります。

ズーム
1.0 よりも大きい数値を割り当てると、トレンドの横軸に表示される期間が増加します。1.0 よりも小さい数値を割り当てると、横軸に表示される期間が減少します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

例

ptZoomTime() 関数は、トレンドの横軸に表示される期間を 17 パーセント増加させます。

```
ptZoomTime("PenTrend_1",1.17);
```

ptZoomTime() 関数は、トレンドの横軸に表示される期間を 50 パーセント削減させます。たとえば、**ptZoomTime()** 関数は、元の時間範囲が 1 時間に設定されている場合は、トレンドの期間を 30 分に短縮します。

```
ptZoomTime("PenTrend_1", 0.5);
```

Symbol FactorySymbol Factory

Symbol Factory には、InTouch アプリケーション ウィンドウで視覚要素として使用できる 4,000 を超える産業用シンボル コレクションが含まれています。Symbol Factory は、InTouch HMI のインストール時にインストールできる補足コンポーネントです。

注記：産業用グラフィック エディタを使用して、Application Server と相互作用する InTouch アプリケーション用の視覚要素を作成します。グラフィック エディタを使用して、InTouch HMI とは別のアプリケーション用にインテリジェントな視覚要素を作成することもできます。産業用グラフィック エディタについては、Application Server のマニュアルを参照してください。

AVEVA では、この製品に関する一切の保証は提供していません。問題が発生した場合は、グローバル テクニカル サポートに連絡してください。新しいユーティリティまたはアプリケーションをインストールまたは使用する前に、常にアプリケーションおよびデータをバックアップすることを強くお勧めします。

シンボル タイプ

Symbol Factory には以下の 4 つのタイプのウィザードが含まれます。

- ピクチャ ウィザード
- ビットマップ ウィザード
- テクスチャ ウィザード
- InTouch オブジェクト

ピクチャ ウィザード

Symbol Factory ピクチャ ウィザードは、装置図またはフロー図のベクトル ベースの画像です。アプリケーションを作成するときに、以下の手順でピクチャ ウィザード画像を変更できます。

- アニメーションを画像に割り当てる
- 画像を横または縦に回転させる
- 画像の縦横の視点を変更する
- 画像を軸上で回転させる
- 画像の塗りつぶし色およびパターンを変更する
- 画像の線のサイズ、パータン、および色を変更する

ビットマップ ウィザード

ビットマップ ウィザードは、ウィンドウ アイコンまたはテキスト ブロックなどのビットマップ画像です。アプリケーションを作成するときに、以下の手順でピクチャ ビットマップ ウィザード画像を変更できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- アニメーションをビットマップに割り当てる
- 画像を横または縦に回転させる
- ビットマップの縦横の長さを変更する
- ビットマップ境界の周囲に境界線または影を配置する
- ビットマップ イマージを軸上で 90 度ずつ回転させる
- 透明色を定義する
- ビットマップ内の最大 3 色を他の色に置換する

テクスチャ ウィザード

テクスチャ ウィザードはビットマップ ウィザードと似ていますが、連続パターンを形成するためにサイズ変更できる点が異なります。テクスチャ ウィザードは一般的に、ウィンドウの背景の作成またはグラフィック オブジェクトの塗りつぶしとして使用されます。テクスチャ ウィザードは、Symbol Factory テクスチャ カテゴリから選択できます。

InTouch オブジェクト

InTouch オブジェクトは、Symbol Factory で現状のまま保存される InTouch セルまたはウィザードです。

InTouch オブジェクトを Symbol Factory から WindowMaker に貼り付けた後は、Symbol Factory では編集できません。

WindowMaker でオブジェクトをダブル選択すると、オブジェクトがセルの場合は［**タグ変数の変更**］ダイアログ ボックスが表示され、個別のグラフィック オブジェクトに対してはアニメーション リンク選択ダイアログ ボックスが表示されます。

Symbol Factory の使用

Symbol Factory ウィザードの使用方法は、その他のウィザードと似ています。ウィザードを選択してウィンドウに配置し、設定します。

簡単な使用方法

Symbol Factory の使用に慣れている場合は、以下のヒントを参照して簡単に使用を開始できます。

- ウィザードのオプションを設定するには、ウィンドウでウィザードをダブルクリックし、［**Symbol Factory by Reichard Software**］ダイアログ ボックスの［**Options**］を選択します。
- ウィザードを別のカテゴリにコピーするには、サムネイル画像を［**カテゴリ**］ウィンドウにドラッグして別のカテゴリにドロップします。移動するには、Shift キーを押します。
- ウィザードの説明を編集するには、ウィザードのサムネイルを右クリックします。カテゴリの説明を編集するには、カテゴリを右クリックします。
- ウィザードを削除するには、ウィザードのサムネイルを右クリックし、［**Delete Symbol**］を選択します。
- Symbol Factory は、開発者グループがネットワーク間で同じウィザードのライブラリを使用および提供できるように設定できます。
- 特定のカテゴリのすべてのウィザードは、.cat という拡張子の付いたファイルで保存されます。Symbol Factory カテゴリ ファイルは、通常は c:\program files\wonderware\intouch\symfac フォルダに配置されます。このファイルは別のコンピュータの c:\program files\wonderware\intouch\symfac フォルダにコピーできます。

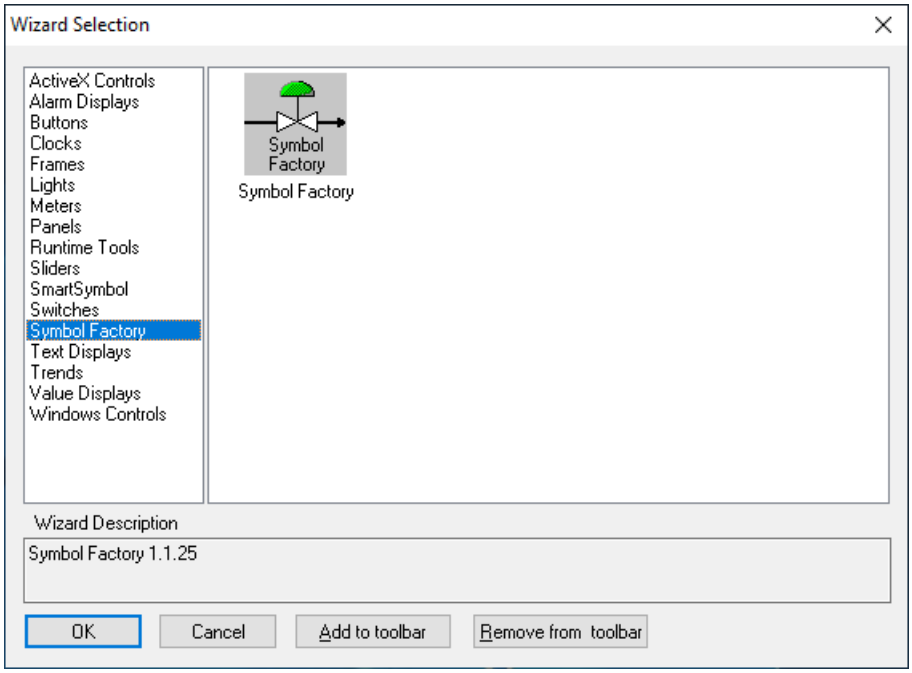
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウィンドウへの Symbol Factory ウィザードの配置

Symbol Factory ウィザードは、他のウィザードを配置するのと同様にウィンドウに配置します。

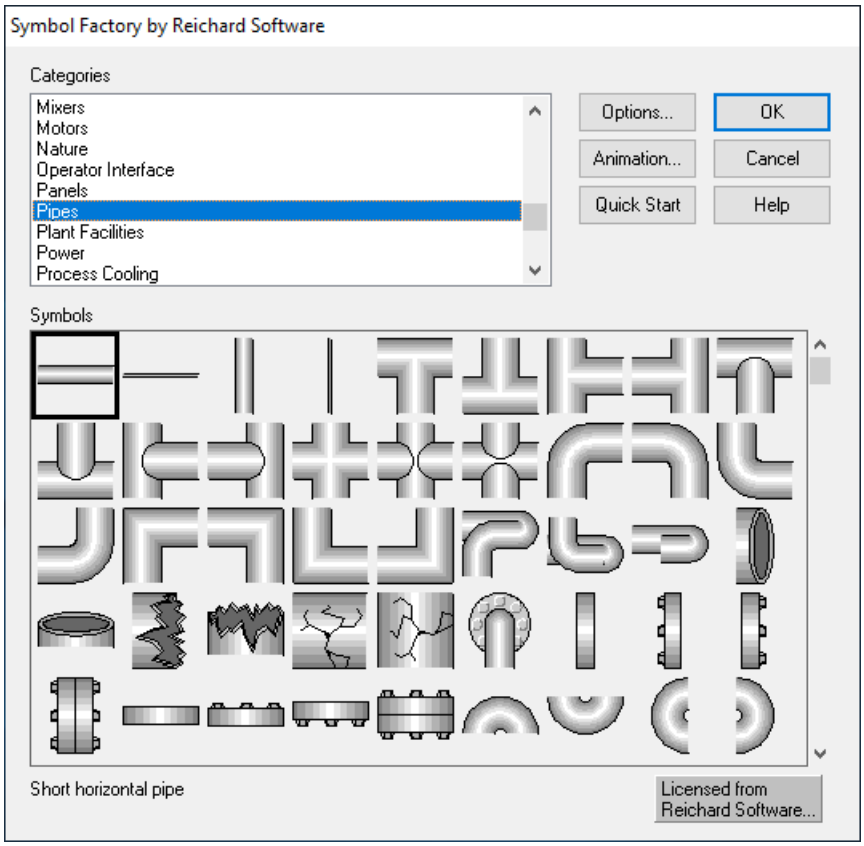
ウィンドウに Symbol Factory ウィザードを配置する

1. WindowMaker でアプリケーションを開きます。
2. ［描画］メニューの［挿入］グループで［ウィザード］を選択します。
   ［ウィザードの選択］ダイアログ ボックスが表示されます。



3. 左ペインに表示されているウィザードのリストで、［**Symbol Factory**］を選択します。
4. 表示領域で Symbol Factory ウィザードを選択して、［**OK**］を選択します。
5. ウィンドウ内を選択して、ウィザードを配置します。
   ［**Reichard Software による Symbol Factory**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. ［カテゴリ］リストから、カテゴリを選択します。［**Symbol**］ウィンドウに、選択したカテゴリのウィザードが表示されます。
7. 配置するウィザードを選択して、［**OK**］を選択します。

シンボル オプションの設定

ウィザード オプションはウィザードによって異なります。色を変更すると、各ピクセルがスキャンされ、変更される場合があるので、ビットマップおよびテクスチャの描画速度に影響が与えられます。

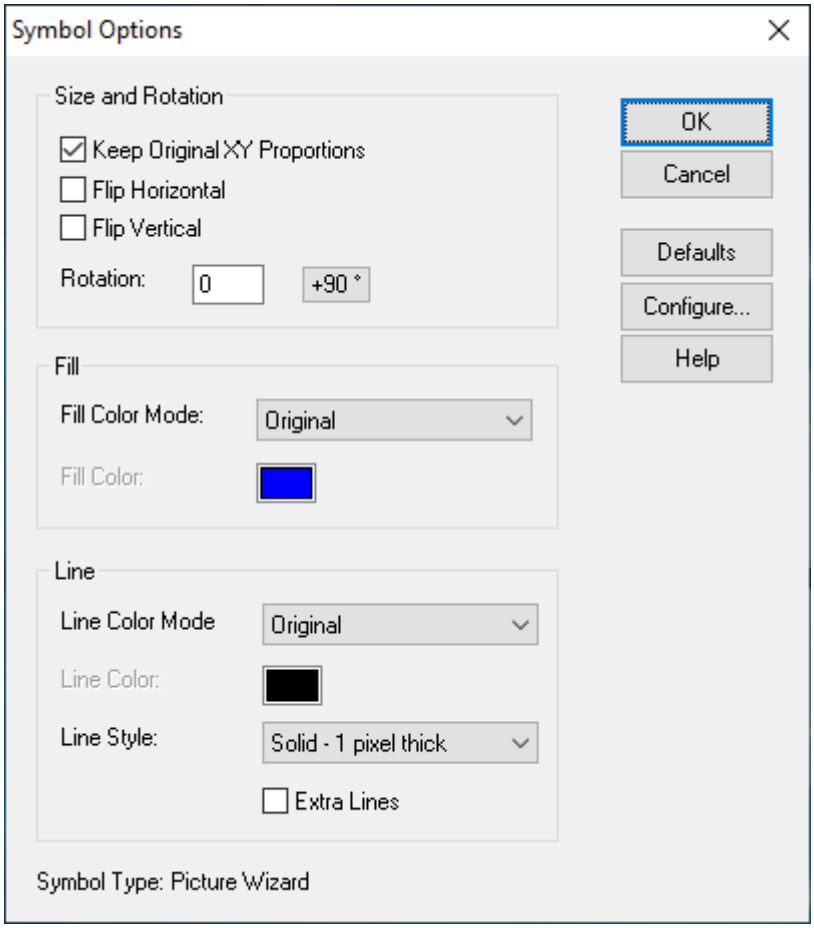
ウィザード オプションを設定する

1. Symbol Factory ウィザードを含むウィンドウを開きます。
2. ウィザードをダブルクリックします。

   ［**Reichard Software による Symbol Factory**］ダイアログ ボックスが表示されます。
3. ウィザードを選択したままで、［**Options**］を選択します。

   ［シンボル オプション］ダイアログ ボックスが表示されます。［シンボル オプション］ダイアログ ボックスに表示される画像プロパティは、編集するために選択したウィザードのタイプによって異なります。以下の例は、ピクチャ ウィザード シンボルを選択した場合に使用できるオプションを示しています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



ヒント：［**Symbol Factory の設定**］ダイアログ ボックスで［マウスの右ボタンの代替ボタンを有効にする］オプションが選択されている場合は、［**シンボルの編集**］ボタンが表示され、ボタンを選択すると選択したウィザードを設定できます。

4. ［**サイズと回転**］領域で、以下のいずれかを実行します。
   - ウィザードのオリジナルの縦横比を保持するには、［**元の XY 比率を維持**］チェック ボックスをオンにします。
   - ウィザードを左右反転するには、［**左右反転**］チェック ボックスをオンにします。
   - ウィザードを上下反転するには、［**上下反転**］チェック ボックスをオンにします。
   - ［**回転タイプ**］ボックスで、ウィザードを回転する角度の数を入力します。ピクチャ ウィザードは任意の角度に回転できます。ビットマップ ウィザードおよびテクスチャ ウィザードは、90 度の倍数（0、90、180、または 270）でのみ回転させることができます。ボタンを選択すると、自動的に回転角度が 90 度ずつ増えます。
5. ピクチャ ウィザードを設定している場合は、［**線**］領域および［**塗りつぶし**］領域で、以下のいずれかを実行します。
   - ［**Fill Color Mode**］リストで、塗りつぶしタイプを選択します。［**塗りつぶしの色**］ボックスをダブルクリックして、カラー パレットを開きます。
   - ［**Line Color Mode**］リストで、線の色を選択します。［**線の色**］ボックスをダブルクリックして、カラー パレットにアクセスします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ［**Line Style**］リストで、線のスタイルを選択します。
- ［**余分な線**］チェック ボックスをオンにして、ウィザード内の斜めの境界線に線を追加します。

6. ビットマップ ウィザードまたはテクスチャ ウィザードを設定している場合は、［**効果**］領域および［**色の変更**］領域で、以下のいずれかを実行します。
   - ウィザードの周囲に黒い境界線を作成するには、［**境界線を含める**］チェック ボックスをオンにします。
   - ウィザードの後ろに暗い灰色の影を作成するには、［**影を含める**］チェック ボックスをオンにします。
   - それぞれのカラー ボックスを選択すると、カラー パレットが開き、ウィザード内の色を変更できます。
7. ［**OK**］ を選択します。

ウィザードをアニメーション化する

すべての Symbol Factory ウィザードはアニメーション化できます。Symbol Factory は、最も一般的なアニメーション リンクへのアクセスを提供します。

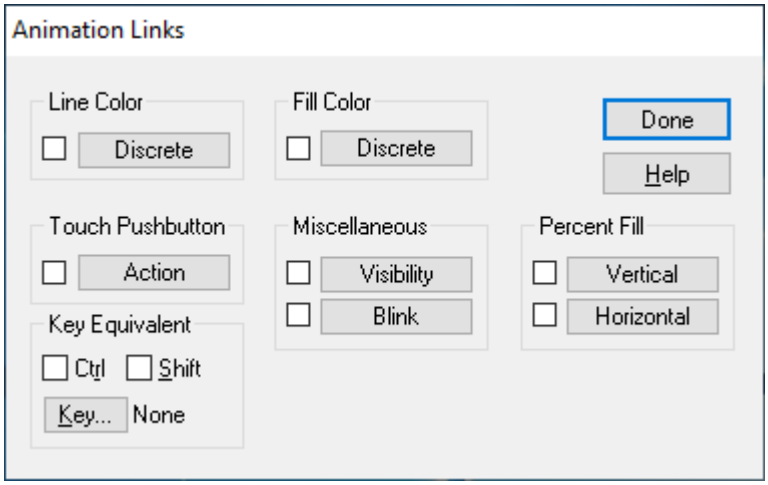
別のタイプのアニメーションが必要な場合は、ウィザードを分解して、標準の InTouch アニメーション リンクを使用してアニメーション化します。

ウィザードをアニメーション化する

1. Symbol Factory でウィザードを選択するか、すでにウィンドウに貼り付けられている場合は、ウィザードをダブルクリックします。

   ［**Reichard Software による Symbol Factory**］ダイアログ ボックスが表示されます。
2. ［**アニメーション**］ を選択します。

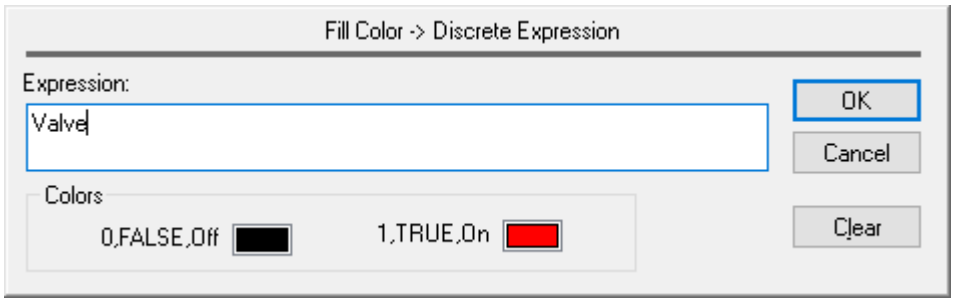
   ［**アニメーション リンク**］ダイアログ ボックスが表示されます。



3. 各アニメーション リンクのタイプのボタンを選択して、選択したウィザードに適用します。

   式ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［式］ウィンドウに式を入力します。

   ウィンドウでダブルクリックして、［**タグの選択**］ダイアログ ボックスを開きます。既存のタグを使用する場合は、式でタグをダブルクリックして**タグ名ディクショナリ**を開いて、タグ定義を参照できます。

   未定義のタグを使用する場合は、式ダイアログ ボックスを閉じるときに、定義するよう促すメッセージが表示されます。

5. アニメーション リンクのタイプの詳細を設定します。
6. ［**OK**］ を選択します。

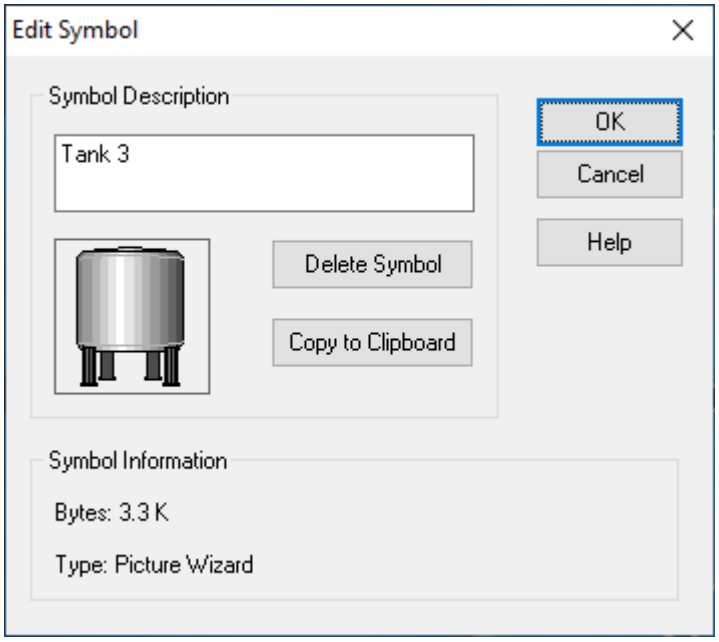
シンボルの編集

ウィザード ツールヒントの説明を変更すること、ウィザードを削除すること、またはウィザードを Windows クリップボードへコピーすることができます。

ウィザードを編集する

1. ［**Reichard Software による Symbol Factory**］ダイアログ ボックスで、ウィザードを含むカテゴリを選択します。
2. ウィザードを右クリックします。

   ［**シンボルの編集**］ダイアログ ボックスが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ウィザードを編集します。以下のいずれかを実行します。
   - ［シンボル説明］ボックスに、ツールヒント テキストを入力します。説明の最大文字数は 80 文字です。
   - ウィザードを削除するには、［**Delete Symbol**］を選択します。
   - ウィザードを Windows クリップボードにコピーするには、［**Copy to Clipboard**］を選択します。ウィザードがピクチャ ウィザードの場合は、Windows メタファイルとしてコピーされます。ウィザードがビットマップ ウィザードまたはテクスチャ ウィザードの場合は、Windows ビットマップとしてコピーされます。
4. ［**OK**］ を選択します。

編集のためのウィザードの分解

Symbol Factory ウィザードは、分解することによって個々に編集できます。ただし、ウィザードを分解した後は、ウィザードのプロパティは失われます。誤ってウィザードを分解した場合、元に戻すツールを使用して再アセンブルできます。

ウィザードを分解する

1. WindowMaker を開きます。
2. ［アニメーション］メニューの［セル］グループで［分解］を選択します。

ネットワークでのシンボルのカテゴリの共有

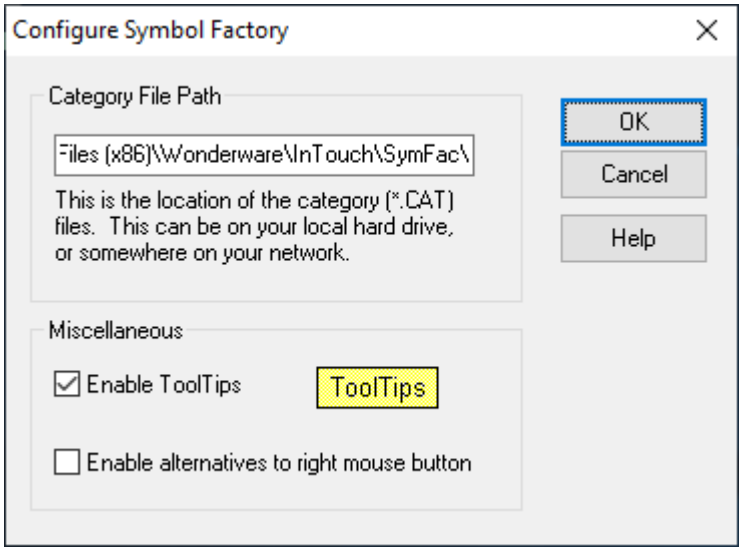
複数の開発者がネットワーク上でウィザードのカテゴリ ファイルを使用および提供できるように Symbol Factory を設定することができます。

カテゴリ ファイルをネットワーク フォルダに移動する

1. ［Symbol Factory］ダイアログ ボックスで、［オプション］を選択します。

   ［シンボル オプション］ダイアログ ボックスが表示されます。
2. 設定を選択します。

   ［**Symbol Factory** の設定］ダイアログ ボックスが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. ［カテゴリ ファイル パス］ボックスに、カテゴリ ファイルが保存されているネットワーク フォルダへのフル パスを入力します。
4. ［**OK**］ を選択します。

カテゴリを読み取り専用にする

ウィザード ファイルをネットワーク フォルダに保存する場合は、他のユーザーがウィザードを移動したり、名前を変更したりすることを防ぐためにカテゴリを読み取り専用にすることをお勧めします。

カテゴリを読み取り専用にする

- ファイルを Windows エクスプローラで読み取り専用に設定します。

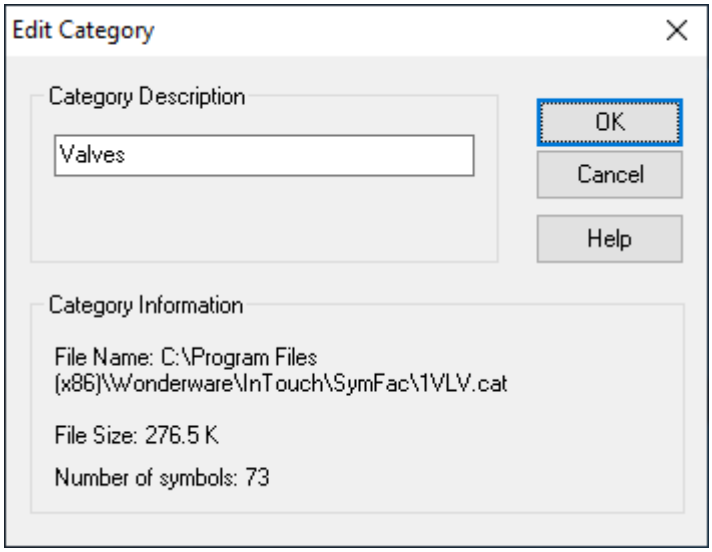
カテゴリ プロパティの表示

カテゴリ パス、ファイル サイズ、およびウィザードの数を参照できます。

カテゴリのプロパティを表示する

1. ［**Symbol Factory**］ダイアログ ボックスで、［カテゴリ］リスト内のカテゴリを右クリックします。

   ［カテゴリの編集］ダイアログ ボックスが表示されます。



2. ［**Category Description**］ボックスに、カテゴリの新しい説明を入力して［**OK**］を選択します。説明の最大文字数は 40 文字です。
3. ［カテゴリ情報］領域で、プロパティを表示します。

カテゴリ情報	説明
ファイル名	カテゴリ（.cat）ファイル パス。デフォルトでは、このパスは C:\Program Files(x86)\Wonderware\InTouch\SymFac です。
ファイル サイズ	カテゴリ ファイルのサイズ（キロバイト単位）。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ情報	説明
シンボルの数	カテゴリに含まれるウィザードの合計数。最大値は 32,767 です。

4. ［**OK**］ を選択します。

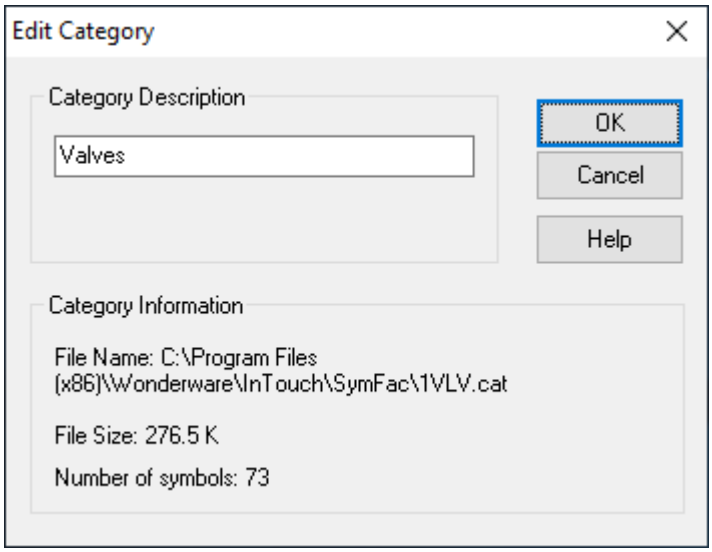
既存のカテゴリを編集する

カテゴリ名のみ編集できます。

既存のカテゴリを編集する

1. ［Symbol Factory］ダイアログ ボックスで、［**カテゴリ**］リスト内のカテゴリを右クリックします。

   ［**カテゴリの編集**］ダイアログ ボックスが表示されます。



2. ［**Category Description**］ボックスに、カテゴリの新しい説明を入力して［**OK**］を選択します。説明の最大文字数は 40 文字です。
3. ［**OK**］ を選択します。

カテゴリの削除

Windows エクスプローラを使用して、カテゴリのファイル名を指定することによって、カテゴリ（.cat）ファイルを削除します。

ヒント：カテゴリ ファイル名は［Edit Category］ダイアログ ボックスで確認できます。

Symbol Factory を設定する

Symbol Factory を設定する場合、以下の項目を指定できます。

- ウィザードを選択するときにツールヒントを表示するかどうか
- ［**Reichard Software による Symbol Factory**］ダイアログ ボックスに追加オプションを表示するかどうか。デフォルトでは、カテゴリおよびウィザードの説明を編集するには、アイテムを右クリックする必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- カテゴリ（.cat）ファイルの場所。各カテゴリのすべてのウィザードのすべてのデータは、1 つのカテゴリ ファイルに保存されます。パフォーマンス上の理由から、このパスには .cat ファイルのみを含める必要があります。ウィザードを他の開発者と共有するには、このパスをネットワーク フォルダに設定します。「ネットワークでのシンボルのカテゴリの共有」を参照してください。

警告：ローカルの InTouch アプリケーション フォルダにはカテゴリ ファイルを配置しないでください。代わりに、カテゴリファイルを C:\Program Files\Wonderware\intouch\symfac に保存します。

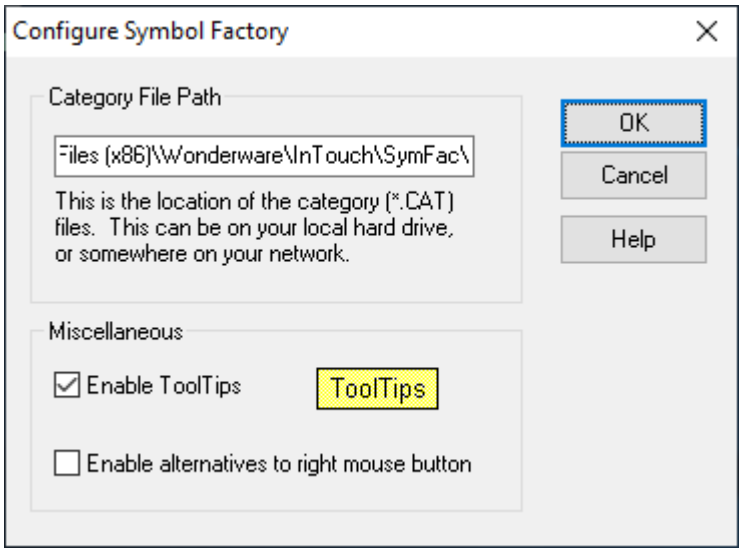
Symbol Factory を設定する

1. ［Symbol Factory］ダイアログ ボックスで、［**オプション**］を選択します。

   ［**シンボル オプション**］ダイアログ ボックスが表示されます。

2. **設定**を選択します。

   ［**Symbol Factory の設定**］ダイアログ ボックスが表示されます。



3. Symbol Factory を設定します。以下の手順を実行します。
   - ［**カテゴリ ファイル パス**］ボックスに、Symbol Factory カテゴリ（.cat）ファイルを保存する場所を入力します。
   - ツールヒントを［**Reichard Software による Symbol Factory**］ダイアログ ボックスのウィザードに対して表示する場合は、［**ツールヒントを有効にする**］チェック ボックスをオンにします。
   - ボタンを［**Reichard Software による Symbol Factory**］ダイアログ ボックスに追加する場合は、［**マウスの右ボタンの代替ボタンを有効にする**］チェック ボックスをオンにします。これらのボタンは、カテゴリおよびウィザードを編集するために右マウス ボタンの代わりに使用できます。

   カテゴリの編集

   選択したカテゴリの［カテゴリの編集］ダイアログ ボックスを表示します。

   シンボルの編集

   選択したウィザードの［シンボルの編集］ダイアログ ボックスを表示します。

4. ［**OK**］ を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

トラブルシューティング

誤って Symbol Factory ウィザードをアンインストールしてしまった場合は、もう一度インストールする必要があります。ウィザードのインストールの詳細については、『*AVEVA™ InTouch HMI アプリケーション開発ガイド*』の「ウィザード」を参照してください。

誤ってウィザードを削除した後で、そのウィザードを元に戻す場合は、そのウィザードを取得する必要があります。

削除したウィザードをカテゴリから取得する

1. ファイルの名前を ~cat.bak から temp.cat に変更します。
2. Symbol Factory を実行して、削除したウィザードが元に戻っているかどうかを確認します。ウィザードを元のカテゴリに移動して、temp.cat ファイルを削除します。
3. 上記の手順で取得できない場合は、Ctrl キーを押しながら削除したウィザードのカテゴリを右クリックします。この操作により、カテゴリ ファイルが圧縮され、新しいバックアップ ~cat.bak が作成されます。

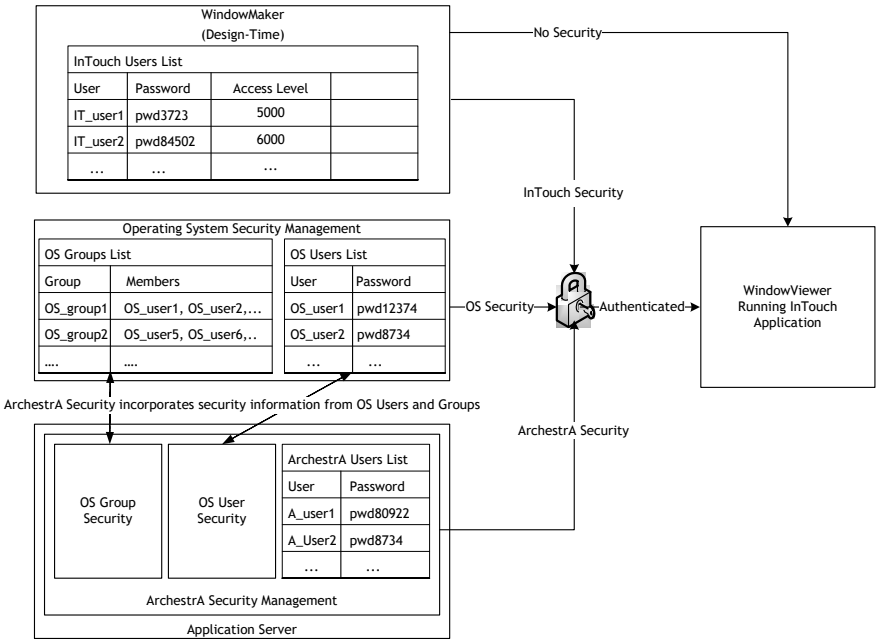
削除したウィザードが見つかるまで前の手順を繰り返します。

InTouch のセキュリティ保護

InTouch アプリケーションは、以下を使用して保護できます。

- 従来の InTouch ベースのセキュリティ
- オペレーティング システム ベースのセキュリティ
- ArchestrA ベースのセキュリティ

以下の図は、3 つのタイプのセキュリティの関係を示しています。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch セキュリティ機能

InTouch アプリケーションの実行中に保護するには、以下の操作を実行できます。

- 無操作のタイムアウト時間を設定する
- キーをロックする
- メニューを非表示にする

無操作タイムアウトを設定する

無操作オペレータを WindowViewer アプリケーションから自動的にログオフするように InTouch を設定できます。オペレータは無操作のためにログオフされた場合、後で再びログオンする必要があります。自動無操作ログオフ期間を設定することによって、オペレータがワークステーションを離れる場合、InTouch アプリケーションへの不正アクセスを防ぐことができます。

タイマーは、オペレータが実行中の InTouch アプリケーションと対話していない時間を測定します。タイマーは、オペレータがマウスまたはその他の入力デバイスを使用してデータを入力するたびにリセットされます。タイマーで設定した時間が経過した場合、ユーザーは自動的にログオフされます。

注記：無操作タイマーは、Active-X コントロールおよび OLE オートメーション コントロールに対してはリセットされません。

オペレータの自動ログオフは 2 段階のステップで行われます。

1. WindowViewer は、オペレータの無操作時間が指定された警告期間を超えると、$InactivityWarning システム タグを 1 に設定します。条件 QuickScript で $InactivityWarning タグを使用して、オペレータに保留中の無操作ログオフについて警告するウィンドウを表示できます。オペレータは、指定したタイムアウト時間が発生する前に応答することによって、ログオンした状態を維持できます。オペレータが何らかの操作を行うと、$InactivityWarning タグと無操作タイマーがゼロにリセットされます。
2. オペレータが無操作警告の後で応答しなかった場合、タイムアウト時間に達すると、$InactivityTimeout システム タグが 1 に設定されます。$InactivityTimeout が 1 になると、WindowViewer はログオンしているオペレータ名を予約名 None に設定して、$AccessLevel セキュリティ タグを 0 に設定します。ユーザーは自動的にログオフされます。

タイムアウト機能は警告機能とは無関係に使用できます。

無操作タイムアウトを設定するには

1. WindowMaker を開きます。
2. ［**ファイル**］メニューで、［**設定**］をポイントし、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［**無操作時間の設定（秒）**］領域で、警告とタイムアウトの値を設定します。以下の手順を実行します。
   - ［**警告**］ボックスに、$InactivityWarning タグが 1 に設定されるまでの秒数を入力します。
   - ［タイムアウト］ボックスに、$InactivityTimeout タグが 1 に設定されてユーザーが自動的にログオフされるまでの時間を秒で入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. ［**OK**］ を選択します。
5. 無操作警告時間が経過した後に「警告 - ログオフが保留中です」という名前のウィンドウを表示するには、条件として "$InactivityWarning" を使用し、スクリプト本文が以下のような条件スクリプトを作成します。
```
   Show "ログオフが保留中です";
```
6. 無操作タイムアウト時間が経過した後に「ログオフしました」という名前のウィンドウを表示するには、"$InactivityTimeout" を条件とし、スクリプト本文が以下のような条件スクリプトを作成します。
```
   Show "ログオフしました";
```

$InactivityTimeout システム タグ変数

無操作に設定された時間が経過したことを示します。

カテゴリ

セキュリティ

使用法

$InactivityTimeout

備考

無操作時間タイマーが経過すると 1 に設定されます。ログオフ時間の設定の詳細については、「無操作タイムアウトの設定」を参照してください。

注記：無操作時間タイマーは、Active-X コントロール、OLE コントロール、およびオートメーション コントロールに対してはリセットされません。

データ型

論理型（読み取り専用）

参照項目

$InactivityWarning

例

以下の例は「True の時」条件スクリプトです。
```
If $InactivityTimeout == 1 THEN
   Show "ログオフしました";
ENDIF
```

参照項目

$InactivityWarning

$InactivityWarning システム タグ変数

ログオフが発生しようとしていることをユーザーに警告するために設定された時間が経過したことを示します。

カテゴリ

セキュリティ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

使用法

$InactivityWarning

備考

無操作警告時間が経過すると 1 に設定されます。無操作時間タイマーは、マウスまたはキーボード操作によってのみリセットされます。ログオフ警告の設定の詳細については、「無操作タイムアウトの設定」を参照してください。

注記：無操作時間タイマーは、ActiveX コントロール、OLE オートメーション コントロール、および SPC ウィザードに対してはリセットされません。

データ型

論理型（読み取り専用）

例

以下の例は「True の時」条件スクリプトです。

```
If $InactivityWarning == 1 THEN
   Show "ログオフが保留中です";
ENDIF;
```

参照項目

$InactivityTimeOut

システム キーのロック

InTouch アプリケーションを実行しているコンピュータでシステム キーを無効にすることによって、標準の Windows 機能へのオペレータ アクセスを制限できます。たとえば、オペレータが Windows の CTRL+ALT+DEL キーの組み合わせを使用して、［**タスク マネージャ**］ダイアログ ボックスを表示することを防ぐことができます。システム キーを無効にすることによって、オペレータが InTouch HMI から別の Windows アプリケーションに切り替えることを防ぎます。

WindowViewer には、InTouch アプリケーションが起動するときにシステム キーのデフォルト状態を設定するキー フィルタ オプションがあります。キー フィルタは、アクティブである場合にシステム キーを無効にします。

さまざまな InTouch ユーザーによって完了される予測されるタスクに基づき、システム キーを無効にします。オペレータに対して、ほとんどのファンクション キーを無効にする必要があります。管理者は InTouch タスクに対してファンクション キーが引き続き必要となります。

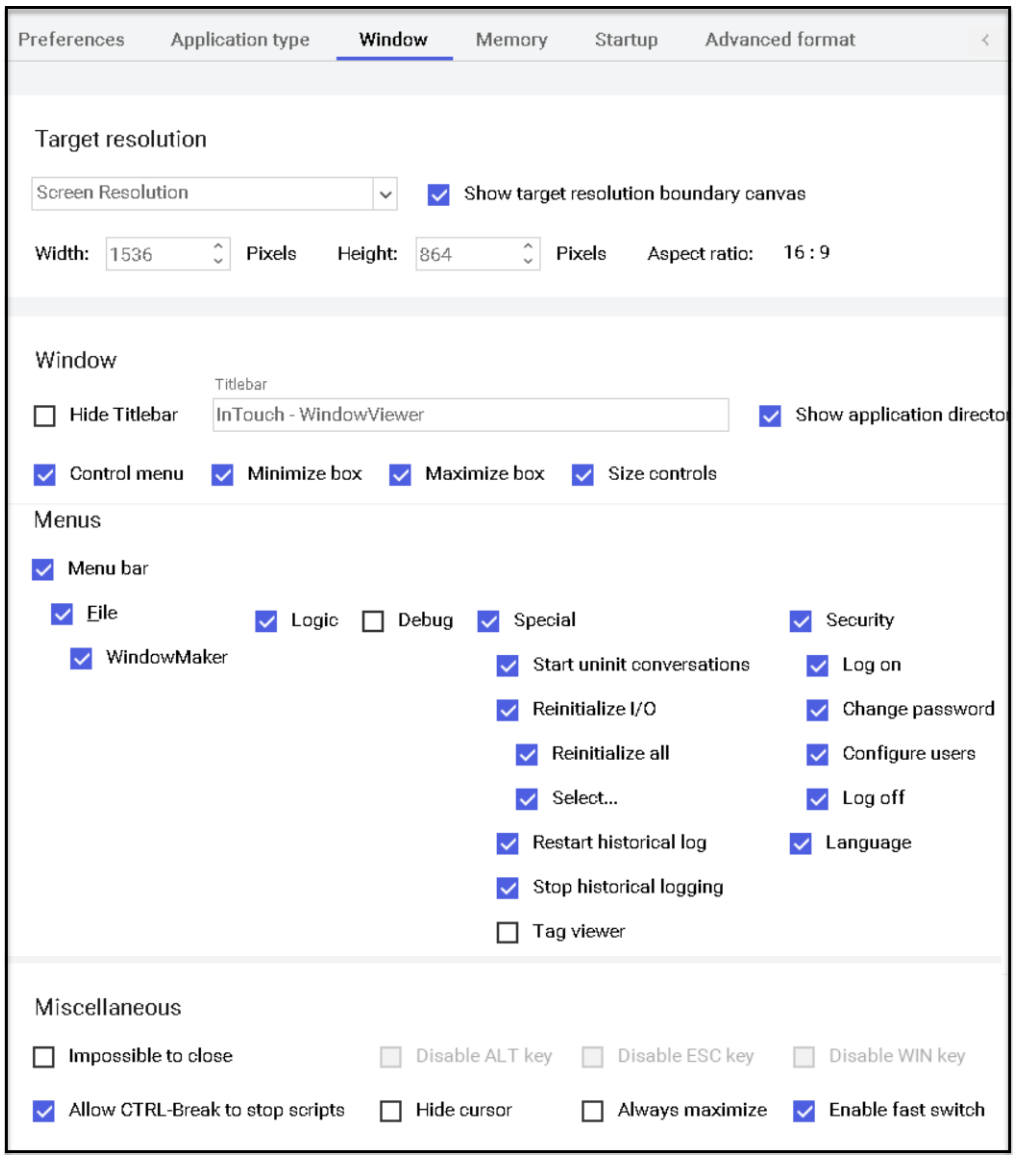
WindowViewer にログオンするユーザーのアクセス レベルに基づき、システム キーを有効または無効にするスクリプトを記述できます。スクリプトに EnableDisableKeys() 関数を使用して、Windows のファンクション キーを選択的に有効または無効にします。

キー フィルタを有効にする

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］を選択し、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［**ウィンドウ**］タブを選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



1. ［**その他の設定**］領域で、WindowViewer システム キーを無効にします。以下の手順を実行します。
   - ［**高速切り替えを有効にする**］チェック ボックスをオフにして、WindowViewer からユーザーを WindowMaker に切り替えるための［**開発**］ボタンを削除します。
   - ［**ALT キーの使用を禁止**］チェック ボックスをオンにして、InTouch アプリケーションを実行しているコンピュータで ALT キーを無効にします。
   - ［**WIN キーの使用を禁止**］チェック ボックスをオンにして、InTouch アプリケーションを実行しているコンピュータで WIN キーを無効にします。
   - ［**ESC キーの使用を禁止**］チェック ボックスをオンにして、InTouch アプリケーションを実行しているコンピュータで ESC キーを無効にします。
2. ［**OK**］ を選択します。
3. WindowViewer が InTouch アプリケーションの実行を開始するときに実行するスクリプトを記述します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スクリプトには、WindowViewer にログオンするユーザーのアクセス レベルに基づき、キーを動的にロックまたはアンロックするステートメントを含める必要があります。

スクリプトに EnableDisableKeys() 関数を含め、ALT キー、ESC キー、および WIN キーを有効または無効にします。EnableDisableKeys() 関数は、その引数の論理値に基づきシステム キーを有効または無効にします。

```
EnableDisableKeys(AltKey,EscKey,WinKey);
```

引数値 1 の引数値を指定すると、キー フィルタが有効になり、キーが無効になります。

EnableDisableKeys() 関数

Alt キー、Esc キー、および Windows キーのキー フィルタを有効または無効にします。

カテゴリ

表示

構文

```
EnableDisableKeys(AltKey, EscKey, WinKey);
```

パラメータ

AltKey
Alt キーのキー フィルタを有効または無効にする整数型：

1 = フィルタの有効化（Alt キーを無効化)

0 = フィルタの無効化（Alt キーを有効化）

EscKey
Esc キーのキー フィルタを有効または無効にする整数型：

1 = フィルタの有効化（Esc キーの無効化）

0 = フィルタの無効化（Esc キーの有効化）

WinKey
Windows キーのキー フィルタを有効または無効にする整数型：

1 = フィルタの有効化（Win キーの無効化）

0 = フィルタの無効化（Win キーの有効化）

備考

Alt キーを無効にすると、Win+L キーの組み合わせも無効になります（Windows デスクトップをロックする場合）。Win+L は、Alt キーを含むキーの別の組み合わせのショートカットです。したがって、Windows デスクトップのロックで Alt キーを無効にすると、ショートカットも無効になります。

Esc キーを無効にすると、そのすべてのアクションが無効になります。

例

```
EnableDisableKeys(0,0,0); // 3 つのキーをすべて有効にします
EnableDisableKeys(1,1,1); // 3 つのキーをすべて無効にします
EnableDisableKeys(0,0,1); //  Alt キー、Esc キーを有効にし、Windows キーを無効にします
```

ランタイム時にメニュー項目を非表示にする

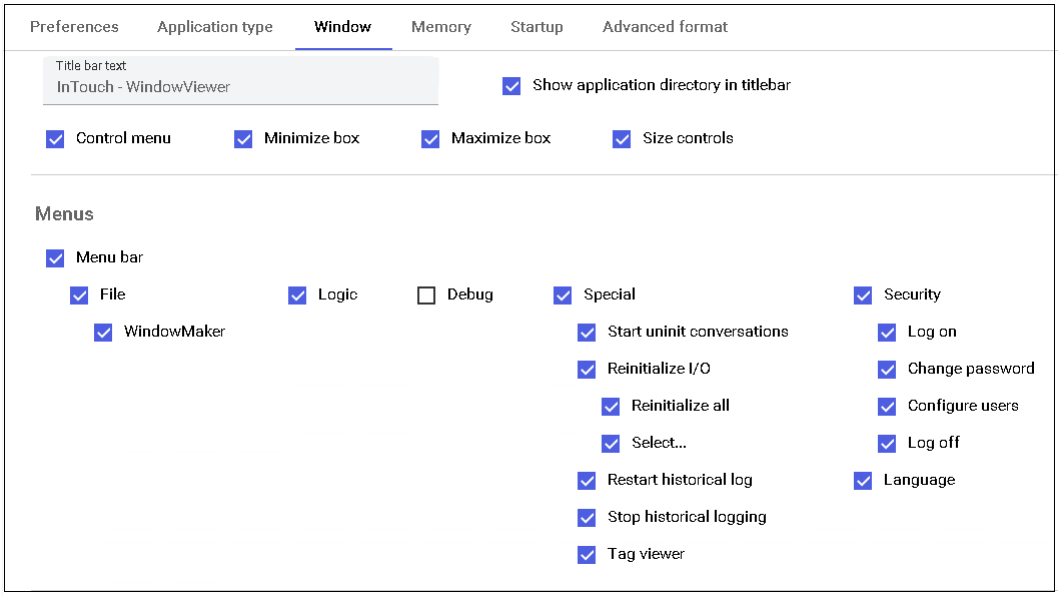
InTouch アプリケーションが実行している間に WindowViewer メニューとコマンドを非表示にすることによって、それらのメニューやコマンドへのオペレータ アクセスを制限します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイム時にメニュー項目を非表示にする

1. WindowMaker を開きます。
2. ［ファイル］メニューの［設定］を選択し、［**WindowViewer**］を選択します。

   WindowViewer の設定画面が表示されます。
3. ［ウィンドウ］タブを選択します。



4. ［メニューの設定］領域で、オペレータに対して表示する WindowViewer メニューとコマンドを選択します。以下の手順を実行します。
   - ［WindowMaker］チェック ボックスをオフにして、［WindowMaker］コマンドを WindowViewer の［ファイル］メニューから使用できないようにします。このオプションをオフにしても、WindowMaker への高速切り替えには影響しません。
   - ［ロジック］チェック ボックスをオフにして、QuickScript を起動および停止するコマンドを含む WindowViewer の［ロジック］メニューを非表示にします。

   注記：$LogicRunning システム タグを使用すると、オペレータがすべての QuickScript を開始および停止できます。
   ［**Ctrl-Break でスクリプトの停止**］オプションを選択すると、［ロジック］メニューの表示の有無に関わらず、オペレータはすべての QuickScript の実行を停止できます。
   現在実行中の非同期クイック関数は停止できません。ただし、オペレータが新しい非同期クイック関数を開始することを防ぐことはできます。

   - アプリケーションをテストしている場合、［デバッグ］チェック ボックスをオンにします。それ以外の場合、［デバッグ］チェック ボックスをオフにして、ランタイム中は［デバッグ］メニューを非表示にします。
   - ［システム］メニュー項目をオフにすると、オペレータがロギングや I/O 接続のような継続的な InTouch 関数を停止することを防ぐことができます。
   - ［セキュリティ］チェック ボックスをオフにすると、オペレータがセキュリティ関連のオプションを変更することを防ぐことができます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

5. ［**ウィンドウ コントロール**］領域で、WindowViewer からオペレータに対して利用可能とするウィンドウ コントロールを選択します。これらのオプションは、InTouch アプリケーションを実行しているウィンドウに影響を与えます。以下の手順を実行します。
   - ［**コントロール メニュー**］チェック ボックスをオフにすると、ウィンドウを閉じるコントロール、および最小化、最大化、またサイズ変更するコントロールを非表示にできます。
   - ［**アイコン化ボタン**］チェック ボックスをオフにすると、オペレータがウィンドウを最小化することを防ぐことができます。
   - ［**最大化ボタン**］チェック ボックスをオフにすると、オペレータがウィンドウを最大化することを防ぐことができます。
   - ［**サイズ コントロール**］チェック ボックスをオフにすると、オペレータがウィンドウをサイズ変更することを防ぐことができます。
6. ［**タイトル**］バー領域で、InTouch アプリケーションを実行しているウィンドウのタイトル バーを設定します。以下の手順を実行します。
   - ［**タイトル バーのテキスト**］ボックスに、WindowViewer タイトル バーに表示するタイトルを入力します。
   - ［**アプリケーションのディレクトリの表示**］チェック ボックスをオンにして、InTouch アプリケーションのフォルダへのパスをタイトル バーに表示します。
   - ［**タイトル バーを隠す**］チェック ボックスをオンにして、ウィンドウのタイトル バーを非表示にします。
7. ［**その他の設定**］領域で、以下の手順を実行します。
   - ［**WindowViewer の強制終了不可**］チェック ボックスをオンにして、オペレータが InTouch アプリケーションを実行している WindowViewer ウィンドウを閉じることを防ぎます。このオプションを選択すると、ウィンドウの［**閉じる**］ボタンが無効になります。

     ［**閉じる**］ボタンを非表示にする場合、［**ウィンドウ コントロール**］領域で［**コントロール メニュー**］チェック ボックスをオフにします。
   - ［**CTRL-Break でスクリプトの停止**］チェック ボックスをオフにすると、オペレータが QuickScript の停止を有効にする CTRL + BREAK キーの組み合わせが無効となります。

注記：現在実行中の非同期クイック関数は停止できません。ただし、新しい非同期クイック関数が実行されることを防ぐことができます。

   - ［カーソルの非表示］チェック ボックスをオンにすると、ランタイム中にマウス ポインタが非表示になります。この機能は、タッチスクリーン用のアプリケーションを設計している場合に役立ちます。
   - ［WindowViewer を常時最大表示］チェック ボックスをオンにすると、InTouch アプリケーションを実行しているウィンドウがオペレータの画面で最大表示されます。
8. ［**OK**］ を選択します。
9. WindowViewer を再起動して、変更を適用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

認証ベースおよび承認ベースのセキュリティ

InTouch セキュリティは、アプリケーションを使用しようとしているユーザーが有効なユーザーとして認識されるかどうかを決定する 2 段階プロセスの最初の段階です。2 番目の段階では、認証されたユーザーに与える InTouch 権限を決定します。

認証と承認の比較

認証は、ユーザーの識別を確認するプロセスです。一般に、オペレータは InTouch アプリケーションを使用する前にユーザー名とパスワードを入力して自分自身を認証します。セキュリティの 3 つのタイプはすべて、認証プロセスの一部としてログイン プロセス中にユーザーのアカウント情報を確認します。

承認は、認証されたユーザーが要求されたリソースにアクセスできるかどうかを判断するプロセスです。一般に、InTouch 機能へのアクセスはグループ内または割り当てられたアクセス レベルのユーザーのメンバーシップに基づいて与えられます。

さまざまな認証セキュリティ モード

InTouch セキュリティのすべてのタイプは、ログオン プロセス中、ユーザー名とパスワードの組み合わせを使用してユーザーを認証します。それぞれのタイプのセキュリティでは、認証プロセス中に異なるメカニズムを使用して、ユーザー名とパスワードを確認します。

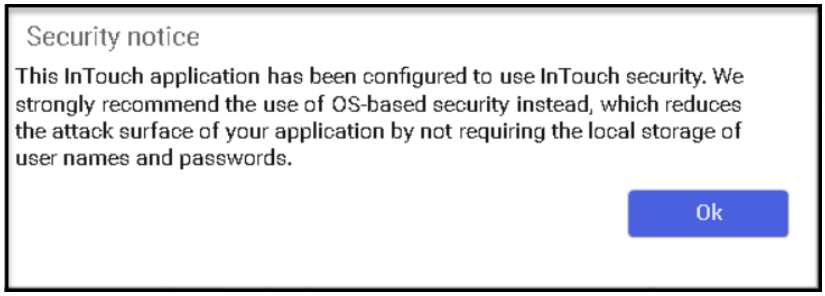
InTouch ベースのセキュリティの使用

アプリケーションへのセキュリティの適用はオプションです。デフォルトでは、InTouch アプリケーションにセキュリティは適用されていません。しかし、対象となる機能を内部タグにリンクすることによって、操作者が実行できる機能を制限できます。さらに、アプリケーションにセキュリティを確立するとき、アラームとイベントを InTouch HMI にログオンしているオペレーターに関連付ける監査記録を作成できます。

InTouch にセキュリティ タイプを設定する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］を選択し、［**セキュリティ**］を選択します。

   セキュリティ設定画面が表示されます。
3. セキュリティ タイプを［**InTouch**］として選択します。



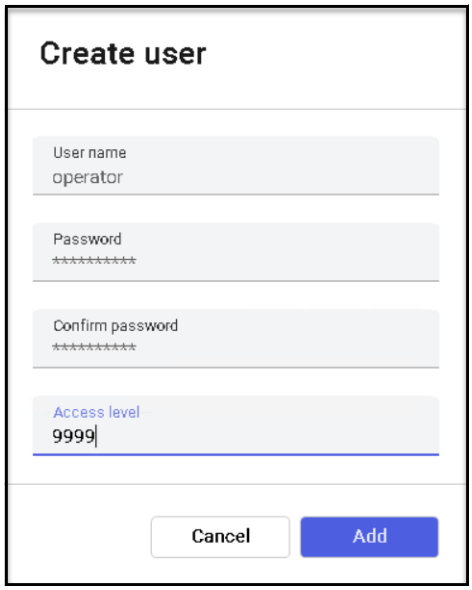
セキュリティ タイプを InTouch として追加すると、［**ユーザーの設定**］セクションが有効になります。セキュリティは、InTouch アプリケーションにログオンするためにユーザー名とパスワードを入力することによって、オペレータが自分自身を認証する過程に基づきます。各オペレータに対してユーザー名、パスワード、およびアクセス レベルを割り当てる必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ユーザーを設定するには

1. ［追加］アイコンを選択します（Alt+A）。

   ［ユーザーを作成］ダイアログが表示されます。



2. ［ユーザー名］ボックスに、オペレータに割り当てる名前を入力します。
3. ［パスワード］ボックスに、最大 29 文字までのオペレータ パスワードを入力します。
4. ［パスワードの確認］ボックスに、同じパスワードを再度入力します。
5. ［アクセス レベル］ボックスに、オペレータのアクセス レベル（最低 0 ～最高 9999）を入力します。

新しいユーザー名をセキュリティ リストに追加して WindowMaker または WindowViewer を再起動すると、デフォルトのユーザー名はアクセス レベル 0 の None に自動的にリセットされるので、WindowMaker と WindowViewer の両方で［ユーザーの設定］コマンドへのアクセスが防止されます。ただし、Administrator アカウントとパスワードは維持され、引き続き使用できます。

オペレータがアプリケーションにログオンすると、保護された機能へのアクセスは、機能にリンクされた内部セキュリティ タグで指定された値に対してそのオペレータのパスワードとアクセス レベルが照合された後で許可されます。

スタンドアロン アプリケーションの場合、管理者特権を持つユーザーだけが InTouch WindowMaker でアプリケーションを開くこと、および編集することができます。管理特権を持たないユーザーが InTouch WindowMaker を起動しようとすると、その処理を続行するには管理特権が必要であることを通知するエラー ダイアログが表示されます。管理特権を持たないユーザーは、マネージド アプリケーション用の IDE から WindowMaker を起動できます。

オペレーティング システム ベースのセキュリティの使用

オペレーティング システム ベースの認証方法では、Windows オペレーティング システムのアカウント ポリシーの一部が継承され、その他のポリシーは InTouch HMI から施行されます。最大有効期間、最低有効期間、最低文字数などのパスワード ポリシーは、オペレーティング システムによって施行されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

インストール中に使用されたユーザー名は、オペレーティング システムの一部として動作します。これらの基準を施行するには、Windows ドメインに希望するアカウント ポリシーが設定されている必要があります。無操作タイムアウト時間は InTouch HMI で施行されます。

オペレーティング システム ベースの認証方法では、Windows ネットワーク ドメインまたはワークグループに関連付けられているユーザー リストからユーザー名を選択できます。各ユーザー名には、特定のアクティビティに対するユーザーの認証を決定するアクセス レベルが割り当てられます。オペレーティング システムはパスワードを内部で管理するため、InTouch HMI ではアプリケーションをホストするノードに対するパスワードは保存されません。

オペレーティング システム ベースのセキュリティでは、InTouch AddPermission() スクリプト関数を使用して、ユーザー リストおよびそれらのユーザーに対応するアクセス レベルを定義して、管理します。AddPermission() 呼び出しの実行後に作成されるこのリストは、ディスクに書き込まれます。ユーザーの認証の詳細が含まれるファイルは、NAD クライアント ノードにコピーされません。

オペレータは、WindowViewer の［**システム**］メニューで（［**システム**］メニューが表示されている場合）、［**セキュリティ**］から［**ログオン**］メニュー コマンドを実行することによってアプリケーションにログオンできます。または、内部セキュリティ タグにリンクされているタッチ入力オブジェクトを使用してカスタムのログオン ウィンドウを作成することもできます。

アプリケーションにセキュリティを確立するために使用するコマンドは、WindowMaker および WindowViewer の両方の［**セキュリティ**］に配置されています。セキュリティ コマンドは、アプリケーションに対するログオンおよびログオフ、パスワードの変更、有効なユーザー名の設定、パスワード、およびアクセス レベルのリストの設定に使用されます。

たとえば、ログオンしているオペレータのアクセス レベルを 2000 より大きく指定して、ウィンドウへのアクセス、オブジェクトの表示などを制御できます。

ArchestrA ベースのセキュリティの使用

ArchestrA セキュリティを使用するようにノードを設定する場合、InTouch HMI はログオンやログオフ操作に対して、Application Server からのメソッドとダイアログ ボックスを使用します。ユーザーは Application Server の Galaxy レポジトリ ノードで設定されます。詳細については、Application Server のマニュアルを参照してください。

ArchestrA セキュリティを使用すると、ユーザーを簡単に定義して、実行を許可された操作を割り当てることができます。ユーザーがオートメーション オブジェクトを使用して実行できる操作の観点から、セキュリティ上の許可を定義します。基本的な手順は、以下のとおりです。

1. セキュリティ モデルを定義します。
2. 保護のセキュリティ モデルに従って、オートメーション オブジェクトを構成します。
3. セキュリティ モデルに従って、ユーザーを定義します。

システム管理者は、対応するユーザー プロファイルを作成して、システム ユーザーを定義します。システム管理者は、セキュリティ モデルで事前定義されたユーザー役割のリストから選択することによって、1 つまたは複数の役割を各ユーザーに割り当てます。

InTouch を ArchestrA ベースのセキュリティと同時に使用した場合、パスワードの最大文字数は 31 です。

InTouchView ユーザーは通常、パスワードベースのログオンによって認証されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

認証にスマート カードの使用

スマート カードは、集積回路が内蔵されているポケットサイズのカードです。カードは、プライベート キーおよびパブリック キーの証明書を含むデータを安全に保管します。カードの所有者は個人識別番号（PIN）によって認証され、カード上の特定のデータにのみアクセスできるよう承認されます。

InTouch アプリケーションは、ユーザー認証でスマート カードをサポートするように設定することができます。アプリケーションにユーザー名、パスワード、およびドメインを要求させる代わりに、スマート カード証明書および関連する PIN 番号を認証に使用できます。また、スマート カードの代わりに、名前、パスワード、およびドメイン名でログオンすることもできます。

ログオンやセキュリティ書き込み/確認書き込みのようなユーザー認証を必要とする操作でも、スマート カード認証を利用することができます。詳細については、「セキュリティ書き込みおよび確認書き込みの使用」を参照してください。

スマート カード認証の設定

スマート カード認証を設定するには、以下の手順を実行する必要があります。

- InTouch アプリケーションを InTouch OS または ArchestrA OS セキュリティのいずれかを使用するように設定します。ArchestrA セキュリティは、ユーザーベースまたはグループベースのいずれかにすることができます。ArchestrA セキュリティは System Platform IDE を使用して設定します。詳細については、IDE のマニュアルを参照してください。
- WindowViewer コンピュータをネットワークの適切なドメインに参加させます。
- WindowMaker で、InTouch アプリケーション用のスマート カード認証を有効にします。詳細については、「WindowMaker でスマート カード認証を有効にする」を参照してください。
- スマート カードを、使用するドメインに合わせて設定します。
- WindowViewer コンピューターにカードのドライバをインストールします。スマート カードと専用のドライバはハードウェアに特化しています。スマート カード リーダーのインストールおよび設定の詳細については、リーダーに付属しているマニュアルを参照してください。
- スマート カード リーダーを WindowViewer コンピュータの適切なポートに接続します。手順については、スマート カードに付属するマニュアルを参照してください。
  - 複数のユーザーが関与する確認書き込みの機能を実施するには、複数のスマート カード リーダーが必要になります。
  - ターミナル サーバーと RDP クライアントでスマート カードを使用するには、スマート カード認証を有効にするため、スマート カード リーダーをクライアント システムに接続しておく必要があります。RDP を使用するターミナル サーバーにスマート カードを接続するには、RDP クライアントの接続設定で、［**ローカル デバイスとリソース**］の下の［**スマート カード**］オプションを有効にしてあることを確認してください。

WindowMaker でスマート カード認証を有効にする

スマート カード リーダーを認証に使用するには、WindowMaker でスマート カード認証を有効にしておく必要があります。

スマート カード オプションを設定する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］を選択し、［**セキュリティ**］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

3. 使用可能なオプション（なし、**InTouch**、**OS** および **ArchestraA**）からセキュリティ タイプを選択してください。
   - ［**ArchestrA**］を選択する場合には、事前に IDE を使用して ArchestrA OS セキュリティ（OS ユーザーベースまたは OS グループベース）を設定しておいてください。
   - ［**OS**］または［**ArchestrA**］のセキュリティ タイプを選択した場合、［**スマート カードの認証**］チェック ボックスが有効になります。スマート カード認証を有効にするには、このチェック ボックスをオンにします。

スマート カードによるログオン

スマート カードを使用して InTouch WindowViewer にログオンすることができます。スマート カードで InTouch アプリケーションにログオンするには、スマート カードによる認証が有効になっているアプリケーションが必要です。

スマート カードは、ドメインで設定された証明書を少なくとも 1 つ含んでいる必要があります。WindowViewer を実行しているコンピュータにスマート カード リーダーを接続する必要があります。使用するスマート カードの PIN を入力することが求められます。

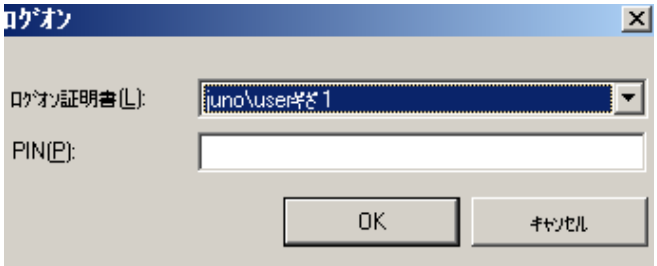
ログオンしようとしたときにリーダーでスマート カードが検出されなかった場合には、代わりにユーザー名とパスワードで認証するように求められます。

セキュリティ書き込みおよび確認データ書き込みのための認証では、スマート カードを使用することができます。詳細については、「セキュリティ書き込みおよび確認書き込みの使用」を参照してください。

スマート カードによるログオン

1. WindowViewer を実行します。
2. スマート カードを挿入します（まだ挿入されていない場合）。
3. ［**システム**］ メニューで、［**セキュリティ**］をポイントして、［**ログオン**］を選択します。

   ［ログオン］ダイアログ ボックスが表示されます。

   スマート カードを挿入していた場合には、ログオン証明書（ドメインとユーザー名）がダイアログ ボックスに表示されます。

   スマート カード ログオン ダイアログ ボックスは、WindowViewer 内のスクリプトから LogonCurrentUser() 関数または PostLogonDialog() 関数を実行した場合にも表示されます。これらの関数は InTouch スクリプトのみで使用可能です。Archestra クライアント スクリプトでは使用できません。

4. ［**PIN**］ボックスに、スマート カードの PIN を入力します。

   スマート カードが利用できない場合には、システムはユーザー ID とパスワードでログオンするように要求します。

5. ［**OK**］ を選択します。これで WindowViewer にログオンできます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注記：スマート カード ユーザーとしてログオンしたら、カードをスマート カード リーダーに挿入した状態にしておく必要があります。スマート カードをリーダーから取り外すと、ログオフされます。

セキュリティ書き込みおよび確認書き込みの使用

InTouch アプリケーションを設定して、オペレータが、特定のセキュリティ分類に設定されている Galaxy 属性にデータを書き込めるようにすることができます。

- 「セキュリティ書き込み」分類では、書き込み処理を完了させるために、ランタイムのオペレータに資格情報を要求します。
- 「確認書き込み」分類では、2 つの署名が要求されます。オペレータは、適切な資格情報を提供すればデータを書き込むことができますが、書き込み処理を完了するには、追加の確認者による承認が要求されます。

セキュリティ書き込みおよび確認書き込みでは、次のことが必要になります。

- Galaxy に対してセキュリティが有効にされている必要があります。
- InTouch アプリケーションに対して ArchestrA セキュリティが有効にされている必要があります。
- ランタイム オペレータは Galaxy 内で設定された適切な権限を持っている必要があります。
  - オペレータは、セキュリティ書き込みまたは確認書き込みのいずれかを行うには、「操作属性変更可能」の権限を持っている必要があります。
  - 確認書き込みの確認を行うには、確認者は「確認書き込み可能」の権限を持つユーザーである必要があります。

現在誰が InTouch アプリケーションのランタイム ユーザーとしてログインしているかには関わりなく、セキュリティ書き込みまたは確認書き込みでは、常にユーザー認証が必要とされます。オペレータとしてログオンしていない場合でも、セキュリティ書き込みまたは確認書き込みセキュリティ分類の設定された属性を変更することはできます。この操作はログオンしているユーザーのセッションには影響を与えません。

重要：セキュリティ設定が有効になっていて、Application Server バージョン 3.5 に移行された Galaxy の場合、「操作属性変更可能」の権限の設定は、「確認書き込み可能」の属性にコピーされます。Application Server 3.5 移行では、Galaxy の「確認書き込み可能」の権限の設定は、デフォルトで無効にされています。

InTouch Tag Viewer では、ランタイム ユーザーが行えるのは、Application Server 属性への参照を持つ間接タグへの書き込みだけです。

セキュリティ書き込みおよび確認データ書き込みのための認証では、スマート カードを使用することができます。詳細については、「セキュリティ書き込みおよび確認書き込みの使用」を参照してください。

セキュリティ書き込みを実行する

セキュリティ書き込みセキュリティ分類が設定されている Application Server Galaxy 属性の値を変更しようとする場合には、有効なセキュリティ アカウント（ドメイン名、ユーザー名、およびパスワード）またはスマート カードのいずれかを使用して、自分自身を認証する必要があります。スマート カードのオプションは、WindowViewer コンピュータにスマート カード リーダーが接続されている場合にのみ使用できます。

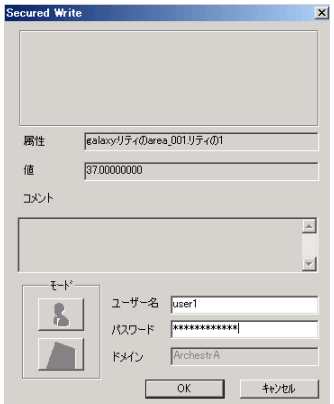
セキュリティ書き込みを行う場合、オペレータは Galaxy 内で「操作属性変更可能」の権限を持っている必要があります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

オペレータのセキュリティ書き込みの認証は、現在ログオンしているユーザーのセッションには影響しません。オペレータがスマート カードでログオンしていた場合には、自分自身を再認証する必要があります。

セキュリティ書き込みを実行する

1. **セキュリティ書き込み**セキュリティ分類が設定された属性の値を変更することを試みます。［**セキュリティ書き込み**］ダイアログボックスが表示されます。スマート カードが利用できない場合には、［**モード**］ ボタンは無効になります。



2. 事前定義された［**コメント**］リストから選択するか、［**コメント**］テキスト ボックスにコメントを入力して、書き込み操作に対するコメントを追加します。コメントは 200 文字までです。

   SignedWrite() スクリプト関数を使えば、コメントのリストを事前定義できます。または、［**コメント**］テキスト ボックスに新しいコメントを入力することができます。事前定義されたコメント リストは、SignedWrite() スクリプト関数を使用する場合にのみアクセスできます。

3. ネットワーク ユーザー アカウントを使用して認証する場合には、ユーザー アカウントのオプションが表示されます。

   以下の手順を実行します。

   a. ［**ユーザー名**］ボックスに、ユーザー名を入力します。デフォルトでは現在ログオンしているユーザーの名前が表示されます。現在ログオンしているユーザーがいない場合、空白のボックスが表示されます。

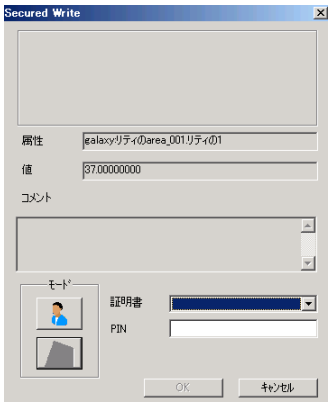
   b. ［**パスワード**］ボックスに、ユーザー名に関連付けられているパスワードを入力します。

   c. ［**ドメイン**］ボックスに、ドメイン名を入力します。

   d. ［**OK**］ を選択します。

4. スマート カードを使用して認証する場合には、スマート カードのオプションが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



スマート カードを使用して認証するには、以下の手順を実行します。

a. ［**証明書**］リストで、自分のスマート カード証明書を選択します。証明書のリストは、ドメイン名／ユーザー名の形式で表示されます。証明書のリストが表示されるには、その時点でコンピュータに接続されているリーダーにスマート カードが挿入されている必要があります。デフォルトでは現在ログオンしているユーザーの証明書が表示されます。［**セキュリティ書き込み**］ダイアログ ボックスが開いているときにカードの抜き差しを行った場合、証明書のリストは自動的に更新されます。

b. ［**PIN**］ボックスに、スマート カードの PIN を入力します。

c. 選択 ［**OK**］をクリックします。

スマート カードがある場合でも、名前、パスワード、およびドメイン名で認証を行う場合には、［モード］ボタンを選択します。手順 3 に移動します。

確認書き込みを実行する

確認書き込みセキュリティ分類が設定されている Application Server Galaxy 属性の値を変更しようとする場合には、有効なセキュリティ アカウント（ドメイン名、ユーザー名、およびパスワード）またはスマート カードのいずれかを使用して、自分自身を認証する必要があります。この書き込みは、別の人によって確認される必要があります。

- 確認書き込みを実行するには、オペレータは「操作属性変更可能」の操作権限を持っている必要があります。
- 確認書き込みの確認を行うには、確認者も「確認書き込み可能」の権限を持つユーザーである必要があります。

オペレータの確認書き込みの認証は、現在ログオンしているユーザーのセッションには影響しません。

スマート カードのオプションは、WindowViewer コンピュータにスマート カード リーダーが接続されている場合にのみ使用できます。オペレータとしてログインする場合、確認者としてログインする場合、またはその両方としてログインする場合でもスマート カードを使用できますが、オペレータと確認者は別の人である必要があります。オペレータがスマート カードでログオンしていた場合には、自分自身を再認証する必要があります。

以下のオプションがあります。

- 2 台のスマート カード リーダーと 2 枚のスマート カードを使用します。
- スマート カード リーダーが 1 台しか利用できない場合には、オペレータまたは確認者のために、1 台のスマート カード リーダーと 1 枚のスマート カードを使用することができます。オペレータが

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

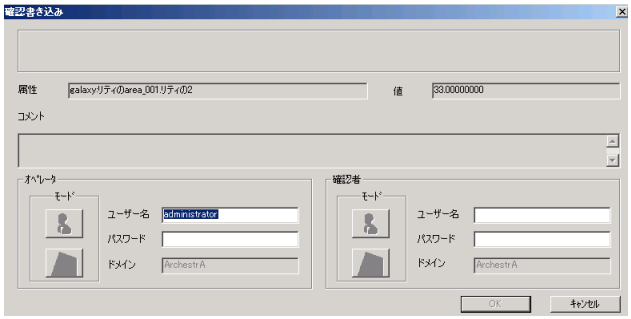
証明書番号と PIN を使用してログオンしている場合には、確認者はユーザー名とパスワードを使用してログオンする必要があります。その逆も可能です。

- オペレータと確認者の両方がユーザー名とパスワードを使用して認証することもできます。

確認書き込みを実行する

1. ［**確認書き込み**］セキュリティ分類が設定された属性の値を変更することを試みます。

   ［**確認書き込み**］ダイアログボックスが表示されます。スマート カードが利用できない場合には、［**モード**］ ボタンは無効になります。



2. 事前定義された［**コメント**］リストから選択するか、［**コメント**］テキスト ボックスに自分自身のコメントを入力して、書き込み操作に対するコメントを追加します。コメントは 200 文字までです。

   事前定義されたコメント リストは、SignedWrite() スクリプト関数を使用する場合にのみアクセスできます。

3. ネットワーク ユーザー アカウントを使用して認証する場合には、ユーザー アカウントのオプションが表示されます。

   以下の手順を実行します。

   a. ［**ユーザー名**］ボックスに、ユーザー名を入力します。デフォルトでは現在ログオンしているユーザーの名前が表示されます。現在ログオンしているユーザーがいない場合、空白のボックスが表示されます。

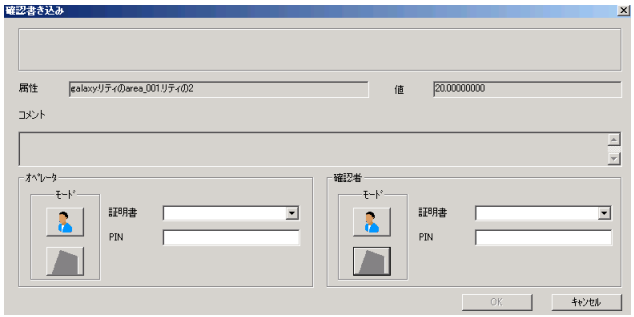
   b. ［**パスワード**］ボックスに、ユーザー名に関連付けられているパスワードを入力します。

   c. ［**ドメイン**］ボックスに、ドメイン名を入力します。

   d. ［**OK**］ を選択します。

   e. 代わりにスマート カードを使用して認証するには、［**証明書**］ボタンを選択します。（ステップ 4 に進む）

4. スマート カードを使用して認証する場合には、スマート カードのオプションが表示されます。



© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

スマート カードを使用して認証するには、以下の手順を実行します。

a. ［**証明書**］リストで、自分のスマート カード証明書を選択します。証明書のリストは、ドメイン名／ユーザー名の形式で表示されます。証明書のリストが表示されるには、その時点でコンピュータに接続されているリーダーにスマート カードが挿入されている必要があります。現在ログオンしているユーザーがスマート カードを使用していた場合、そのユーザーの証明書がデフォルトで表示されます。［**セキュリティ書き込み**］ダイアログ ボックスが開いているときにカードの抜き差しを行った場合、証明書のリストは自動的に更新されます。

b. ［**PIN**］ボックスに、スマート カードの PIN を入力します。

c. ［**OK**］ を選択します。

d. スマート カードがある場合でも、名前、パスワード、およびドメイン名で認証を行う場合には、［モード］ボタンを選択します。手順 3 に移動します。

［セキュリティ書き込み］/［確認書き込み］ダイアログ ボックスのカスタマイズ

SignedWrite() スクリプト関数を使用すれば、［**セキュリティ書き込み**］または［**確認書き込み**］ダイアログ ボックスの以下の点を設定できます。

- 理由メッセージの表示
- ［**コメント**］リストの事前設定

- ［**コメント**］テキスト ボックスの編集を許可する

SignedWrite() 関数の詳細、および構文、パラメータ、詳細例などを始めとする使用方法については、『産業用グラフィック エディタ ユーザー ガイド』を参照してください。

SignedWrite() 関数をランタイム時に操作する

SignedWrite() 関数を使えば、セキュリティ書き込みまたは確認書き込みの署名を必要とする属性に、直接値を割り当てることができます。

セキュリティ書き込みまたは確認書き込みセキュリティ分類が設定されている値を設定して、それを変更すると、［**セキュリティ書き込み**］または［**確認書き込み**］ダイアログ ボックスが表示されます。［**セキュリティ書き込み**］または［**確認書き込み**］ダイアログ ボックスに表示される内容は、値の変更方法に応じて変わります。

- 値を SignedWrite() 関数を使用して変更する場合、関数からのパラメータ設定に基づいて、［**セキュリティ書き込み**］または［**確認書き込み**］ダイアログ ボックスにオプションが表示されます。
- 値をユーザー操作によって変更する場合、理由メッセージのエリアにフィールド属性の説明が表示されます（存在する場合）。属性がフィールド属性でない場合、または説明がない場合には、理由メッセージのエリアにその属性が属している ApplicationObject の説明が表示されます。事前定義された［**コメント**］リストは利用できません。

InTouch WindowViewer で属性の値を変更する場合、［**セキュリティ書き込み**］または［**確認書き込み**］ダイアログ ボックスに表示される理由を説明するメッセージを参照することができます。ダイアログ ボックスには、属性の名前と、その属性に書き込まれる新しい値が表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

注記：理由についての説明および事前定義された［**コメント**］リストとテキスト ボックスは、InTouch WindowViewer を使用している場合にのみ、［**セキュリティ書き込み**］または［**確認書き込み**］ダイアログ ボックスに表示されます。Tag Viewer では表示されません。

ユーザーの管理と承認レベルの設定

InTouch HMI を使用する必要のあるユーザーのグループに対してセキュリティを実装するには、以下を実行する必要があります。

- 各ユーザーにユーザー名とパスワード認証アカウント情報を割り当てます。
- 各ユーザーに InTouch 承認レベル（アクセス レベル）を割り当てます。

InTouch セキュリティ認証と承認の設定

各オペレータに対して、ユーザー名、パスワード、およびアクセス レベルを割り当てる必要があります。

None と **Administrator** 名前は予約されており、Administrator のパスワードのみを変更できます（デフォルトは wonderware です）。アプリケーションのユーザー名を設定した後で、Administrator パスワードを変更します。Administrator のデフォルトのアクセス レベル（9999）は最高の値であり、［**ユーザーの設定**］コマンドを含むすべての InTouch 機能にアクセスできます。

［ユーザー入力 - 論理型］ボタンを $ConfigureUsers タグにリンクさせて、アクセス レベルが 9000 以上の認証されたオペレータが［**ユーザーの設定**］ダイアログ ボックスにアクセスして、セキュリティ ユーザー名リストを編集することを許可できます。オペレータがこのボタンを選択すると、$ConfigureUsers タグ変数の値が 1 に設定され、［ユーザーの設定］ダイアログ ボックスが表示されます。 オペレータがダイアログ ボックスを閉じると、システムは値を 0 にリセットします。これは、書き込み操作のみを対象としたシステム論理型タグ変数です。

注記：$ConfigureUsers タグは、セキュリティ タイプが InTouch に設定されている場合のみ機能します。ArchestrA ベースおよびオペレーティング システム ベースのセキュリティでは機能しません。

アプリケーションに対してオペレータのセキュリティを設定する

1. WindowMaker を開きます。
2. ［**ファイル**］メニューの［**設定**］を選択し、［**セキュリティ**］を選択します。

   セキュリティ設定画面が表示されます。
3. セキュリティ タイプを［**InTouch**］として選択します。

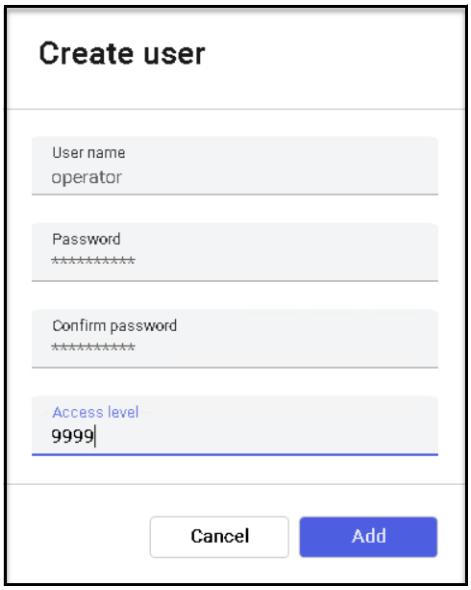
   ［ログオン］ダイアログが表示されます。
4. ユーザー名とパスワードを入力します。

   注記：InTouch のオペレータを設定できるのは、管理者アカウントを持つユーザーか 9000 以上のアクセス レベルを持つユーザーだけです。

   ［**ユーザーの設定**］セクションが有効になります。
5. ［追加］アイコンを選択します。

   ［**ユーザーを作成**］ダイアログが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



6. セキュリティ アカウントを追加するには、以下の手順を実行します。
   a. ［**ユーザー名**］ボックスに、オペレータに割り当てる名前を入力します。
   b. ［**パスワード**］ボックスに、最大 29 文字までのオペレータ パスワードを入力します。
   c. ［**アクセス レベル**］ボックスに、オペレータのアクセス レベル（0～ 9999）を入力します。
   d. ［**追加**］を選択して、ユーザー名を InTouch セキュリティ リストに追加します。
7. ユーザー名を変更するには、［ユーザーの設定］グリッドからエントリを選択します。
8. 必要な変更を行い、［**更新**］を選択します。
9. ユーザー名を変更するには、［ユーザーの設定］グリッドからエントリを選択し、［**削除**］を選択します。
10. ［**OK**］ を選択します。

InTouch オペレータ パスワードの変更

オペレータは、WindowMaker または WindowViewer から自分のパスワードを変更できます。

WindowMaker でオペレータ パスワードを変更する

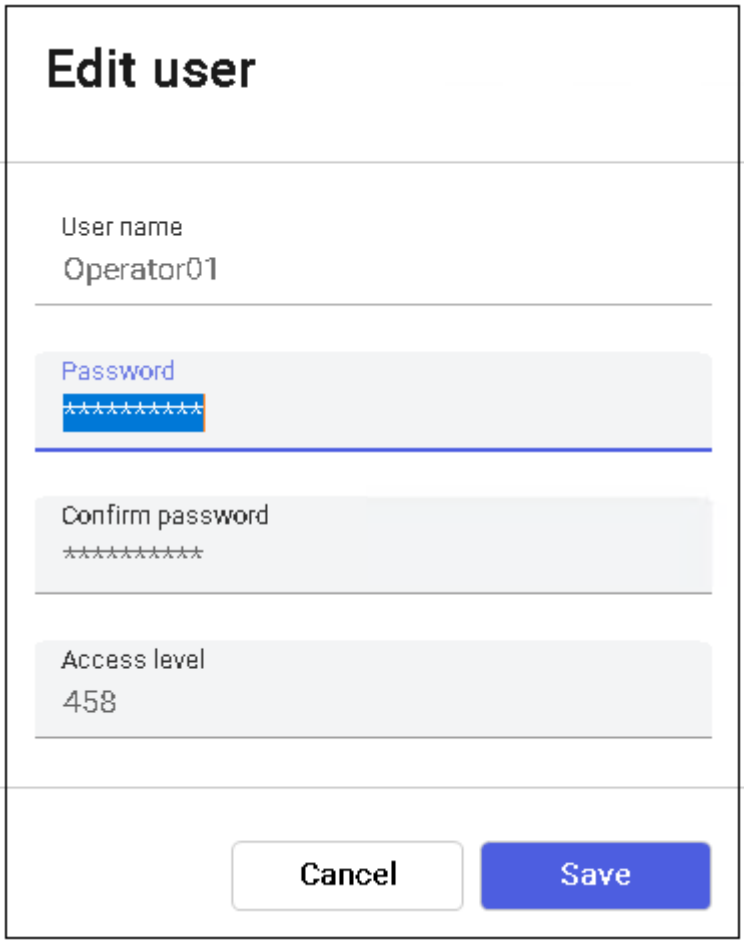
1. ［**ファイル**］メニューの［**設定**］を選択し、［**セキュリティ**］を選択します。

   セキュリティ設定画面が表示されます。
2. ユーザーのセキュリティ タイプを選択します。

   サインインを要求される場合があります。パスワードを変更できるのは管理者または 9000 以上のアクセス レベルを持つユーザーのみです。
3. ユーザー名とパスワードを入力します。
4. ［**ユーザーの設定**］リストで、パスワードを変更するユーザー アカウントを選択し、［**編集**］を選択します。

   ［**ユーザーを編集**］画面が表示されます。

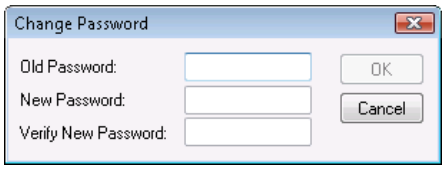
© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



5. ［パスワード］フィールドにパスワードを入力します。
6. ［パスワードの確認］フィールドにパスワードを再度入力して確認します。
7. 保存を選択します。

WindowViewer でオペレータ パスワードを変更する

1. WindowViewer を起動します。
2. ［システム］メニューの［セキュリティ］を選択して、［パスワードの変更］を選択します。
   ［パスワードの変更］ダイアログ ボックスが表示されます。



3. パスワードを設定します。以下の手順を実行します。
   - ［旧パスワード］ボックスに、現在のパスワードを入力します。
   - ［新パスワード］ボックスに、新しいパスワードを入力します。
   - ［新パスワードの確認］ボックスに、新しいパスワードを再入力します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

4. ［**OK**］ を選択します。

論理値ボタンを使用してオペレータ パスワードを変更する

WindowViewer で［**システム**］メニューを表示しない場合、論理値ボタンを作成して、それを $ChangePassword 内部タグにリンクできます。$ChangePassword タグの値を 1 に設定すると、［**パスワードの変更**］ダイアログ ボックスが表示されます。オペレータはそのパスワードを変更できます。オペレータがダイアログ ボックスを閉じると、$ChangePassword の値は 0 にリセットされます。これは書き込み操作のみを対象としたシステム論理型タグ変数です。

オペレーティング システム ベースの認証と承認の設定

オペレーティング システム ベースのセキュリティでは、承認済み Windows ユーザー グループのリストから InTouch を認証します。ローカル コンピュータまたは Active Directory サーバーのいずれかで、Windows ユーザー グループを作成します。Windows ユーザーを特定のグループに追加することによって、グループに関連付ける必要があります。ユーザー グループの作成の詳細については、Windows オペレーティング システムのマニュアルを参照してください。

スクリプトの AddPermission() 関数を使用して、InTouch アクセス レベルを Windows グループに割り当てます。AddPermission() 関数は、ユーザーがログオンする準備が整っている場合、WindowViewer で承認済みユーザー グループが認識されるように、アプリケーションの起動時に通常は呼び出されます。

一般に、InTouch アプリケーションを作成した後ですぐに、オペレーティング システム ベースのセキュリティを指定します。

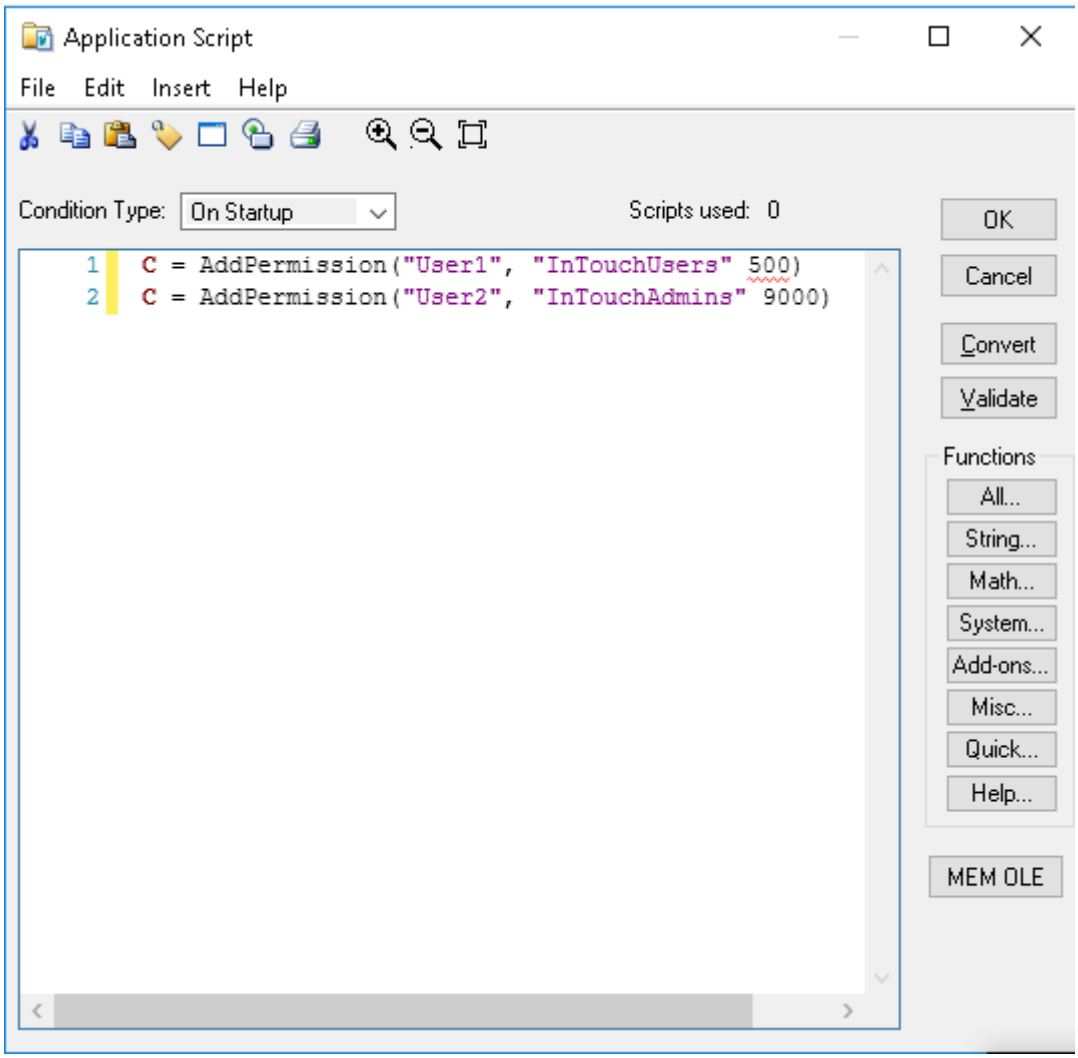
オペレーティング システムの認証と内部 InTouch 承認を使用するように InTouch アプリケーションを設定した後、［システム］メニューの［セキュリティ］で、［パスワードの変更］、［ログオン］、［ユーザーの設定］、および［ログオフ］コマンドは使用できない状態となります。

オペレーティング システム ベースのセキュリティを設定してアクセス レベルを設定する

1. ［**スクリプト**］ペインで［**アプリケーション**］を選択します。

   ［**アプリケーション スクリプト**］ダイアログ ボックスが表示されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



2. ［実行条件］リストで、［起動時］を選択します。
3. AddPermission() 関数を使用して、グループ名と対応するアクセス レベルを指定します。AddPermission() の引数はオペレーティング システム（またはドメイン）、グループ名、およびアクセス レベルです。
4. ［OK］ を選択します。

ArchestrA ベースのセキュリティの設定

ArchestrA セキュリティ システムはグローバル機能であり、Galaxy データベースの全オブジェクトに適用されます。ユーザーおよび Galaxy のオブジェクトと機能の関係に基づいたシステムです。このシステムは、セキュリティの役割（設定、システム管理、ランタイム権限）およびセキュリティ グループに基づいています。これにより、特定のセキュリティ役割のランタイム権限がオブジェクト レベルで決定されます。セキュリティ システムの設定は、IDE（Integrated Development Environment）で実行され、そのエディタを通して全オブジェクトに適用されます。

ArchestrA 認証を使用するように InTouch アプリケーションを設定した後で、WindowMaker の［パスワードの変更］、［ログオン］、［ユーザーの設定］、および［ログオフ］コマンドは使用できなくなります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ArchestrA ベースのセキュリティを設定する

1. WindowMaker でウィンドウを開きます。
2. ［ファイル］メニューの［設定］を選択し、［セキュリティ］を選択します。

   ［セキュリティ］設定画面が表示されます。
3. セキュリティ タイプのオプションから［**ArchestrA**］を選択します。

AddPermission() 関数

特定の InTouch アクセス レベルをローカル システムまたはドメインの指定したユーザー グループに割り当てます。AddPermission() 関数が呼び出された後でそのグループに属すユーザーが InTouch HMI にログオンするとき、そのユーザーは指定されたアクセス レベルを受け取ります。

カテゴリ

セキュリティ

構文

```
DiscreteTag=AddPermission( "Domain", "Group", AccessLevel);
```

引数

ドメイン
グループが配置されているドメインまたはローカル コンピュータの名前

Group
Windows ユーザー グループ

AccessLevel
指定したグループに関連付ける InTouch アクセス レベル

備考

オペレーティング システムのセキュリティの場合のみ有効です。この関数が呼び出されると、指定したドメインまたはワークグループの指定したグループの存在を確認します。成功した場合、TRUE が返され、指定したアクセス レベルが後続のユーザー ログオンのグループに関連付けられます。その他すべての場合（つまり、引数に対して無効な値が指定されている場合）、FALSE が返されます。

この関数は通常、アプリケーションの起動時に実行されるように設定されています。現在ログオンしているユーザーには影響は与えられません。AddPermission() が正常に呼び出された後でログオンするユーザーのみがグループに関連付けられているアクセス レベルを受け取ります。

例

```
DiscreteTag=AddPermission( "corporate_hq", "InTouchAdmins", 9000);
DiscreteTag=AddPermission( "johns01", "InTouchUsers", 5000);
```

Operations Control 接続エクスペリエンス

AddPermission() メソッドが Operations Control 接続エクスペリエンスで受け入れるのは 2 つのパラメータだけです。

- CONNECT グループ
- アクセス レベル

Operations Control 接続エクスペリエンスの AddPermission() のスクリプト関数を以下に示します。

```
DiscreteTag=AddPermission("", "CONNECT group", AccessLevel);
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ランタイムユーザーが CONNECT の複数のグループのメンバーである場合、アクセスレベルは最上位のアクセスレベルを持つグループによって決定されます。

参照項目

PostLogonDialog()、InvisibleVerifyCredentials()、IsAssignedRole()、AttemptInvisibleLogon()、QueryGroupMembership()

ChangePassword() 関数

ログオンしているオペレータがパスワードを変更することを許可する［**パスワードの変更**］ダイアログボックスを表示します。

カテゴリ

セキュリティ

構文

```
[Result=]ChangePassword();
```

戻り値

以下の整数値のいずれかを返します。

0 = ［キャンセル］がクリックされました。

1 = ［OK］がクリックされました。

備考

オペレータがタッチ スクリーンを使用している場合、オペレータは英数字キーボードを使用して、新しいパスワードを入力できます。

例

以下のスクリプトはボタンに配置したり、条件スクリプトまたはデータ変化スクリプトから呼び出したりすることができます。

```
Errmsg=ChangePassword();
```

$AccessLevel システム タグ変数

現在ログインしているユーザーのアクセス レベルを定義します。

カテゴリ

セキュリティ

使用法

$AccessLevel

備考

このタグの値は、InTouch HMI で現在ログインしているユーザーのセキュリティ プロファイルに割り当てられているアクセス レベルによって決まります。このプロファイルには、WindowViewer の［**ユーザーの設定**］メニューを使用してアクセスできます。

$AccessLevel の実際の数字は WindowViewer に対して何の意味も持ちません。ただし、9000 またはそれより大きい値は管理特権を示し、WindowViewer の［**セキュリティ**］メニューを有効にします。システム内のセキュリティをより詳細にカスタマイズするには、$AccessLevel システム タグを使用します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

データ型

整数型（読み取り専用）

有効値

0 ～ 9999

例

以下のステートメントは、ログオンしているユーザーのアクセス レベルに基づいて表示されるボタンなどのオブジェクトを作成するための表示リンクに対して使用されます。

```
$AccessLevel >= 2000;
{オブジェクトは、$AccessLevel に基づく式を持つ「無効化」リンクを関連付けることができます。}
$AccessLevel < 5411;
IF $AccessLevel <=500 THEN
Show "アクセス不可"; {アクセスを拒否するポップアップ ウィンドウ}
ELSE
Show "アクセス可"; {アクセスを許可するポップアップ ウィンドウ}
ENDIF;
```

参照項目

$Operator、$OperatorEntered、$PasswordEntered、$ConfigureUsers

$ChangePassword システム タグ変数

［パスワードの変更］ダイアログ ボックスを表示します。

カテゴリ

セキュリティ

使用法

```
$ChangePassword
```

備考

［パスワードの変更］ダイアログ ボックスを表示するには、この値を 1 に設定します。ダイアログ ボックスが閉じられると、$ChangePassword システム タグ変数の値は 0 にリセットされます。このシステム タグ変数を 1 以外の値に設定すると、結果は定義されません。

データ型

論理型（書き込み専用）

有効値

1

例

［パスワードの変更］ダイアログ ボックスを開く論理値押しボタンを作成できます。セット オプションが選択された状態の、単一の論理値押しボタン リンクを押しボタンに割り当てます。ボタンが押されると、$ChangePassword システム タグ変数が 1 に設定され、［パスワードの変更］ダイアログ ボックスが表示されます。

参照項目

$AccessLevel、$OperatorEntered、$PasswordEntered、$Operator、$ConfigureUsers

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

$ConfigureUsers システム タグ変数

［ユーザーの設定］ダイアログ ボックスを表示します。

カテゴリ

セキュリティ

使用法

```
$ConfigureUsers
```

備考

この関数は InTouch セキュリティでのみ動作します。

［ユーザーの設定］ダイアログ ボックスを開くには、値を 1 に設定します。

ダイアログ ボックスが閉じられると、このシステム タグ変数の値は 0 にリセットされます。このシステム タグ変数を 1 以外の値に設定すると、結果は定義されません。

このダイアログ ボックスを表示するには、ユーザーは 9000 以上の $AccessLevel を持つ必要があります。

データ型

論理型（書き込み専用）

有効値

1

例

［ユーザーの設定］ダイアログ ボックスを開く論理値押しボタンを作成できます。セット オプションが選択された状態の、単一の論理値押しボタン リンクを押しボタンに割り当てます。ボタンが押されると、$ConfigureUsers システム タグ変数が 1 に設定され、［ユーザーの設定］ダイアログ ボックスが表示されます。

参照項目

$Operator、$OperatorEntered、$ChangePassword、$PasswordEntered、$AccessLevel

ログオンとログオフ

InTouch アプリケーションに対するログオンやログオフは、アプリケーションを保護するために使用されるセキュリティのタイプによって異なります。

InTouch によりセキュリティが保護されたアプリケーションへのログオン

ログオン情報が間違って入力されたり、無効である場合、ログオン試行が失敗したことを示すメッセージが表示されます。

ログオンが正常に行われた場合、$AccessLevel システム タグは InTouch セキュリティ ユーザー リストでユーザーに関連付けられた事前定義の値に設定されます。

注記：PostLogonDialog() 関数を使用して、［ログオン］ダイアログ ボックスを表示することもできます。詳細については、「PostLogonDialog() 関数」を参照してください。

アプリケーションにログオンする

1. ［ファイル］メニューの［設定］を選択し、［セキュリティ］を選択します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

2. 関連するセキュリティ タイプを選択します。

   ［ログオン］ダイアログが表示されます。

3. ［名前］ボックスに、ユーザー名を入力します。
4. ［パスワード］ボックスに、パスワードを入力します。
5. ［サインイン］を選択します。

オペレーティング システムによりセキュリティが確保されたアプリケーションへのログオン

ユーザーが InTouch アプリケーションにログオンする際、以下を要求するダイアログ ボックスが表示されます。

- ユーザー名
- パスワード
- ドメインまたはローカル コンピュータ名

ドメインとユーザー名の組み合わせはオペレーティング システムに渡され、ユーザーのアカウント情報が認証されます。オペレーティング システムのキャッシュを有効または有効にしない状態で、ログオンが試行されます。ユーザーが（ネットワーク障害などの原因で）キャッシュなしでログオンできなくてもキャッシュが有効な状態で以前に認証されている場合、ローカルの InTouch キャッシュからユーザーの氏名とアクセス レベルが取得されます。

セキュリティ チェックがすべて正常に完了すると、ユーザーは InTouch HMI にログオンしていると見なされ、関連するデータ構造（たとえば、$Operator など）が更新されます。それ以外の場合は、エラー メッセージが表示されます。

オペレータが今まで正常にログオンしたことがなく、ドメインが使用できない場合、ログオン試行は失敗します。InTouch HMI はシステム イベントをエラー ログにログ記録します。

パスワードの期限が切れている場合、エラー メッセージが表示されます。オペレータが［**OK**］を選択すると、［パスワードの変更］ダイアログ ボックスが表示され、オペレータはパスワードを変更して、新しいパスワードを使用して再びログオンを試行できます。

ArchestrA によりセキュリティが保護されたアプリケーションへのログオン

ArchestrA によりセキュリティが保護された InTouch アプリケーションには通常、有効なユーザー名とパスワードを入力してログオンし、ログオフします。

InTouch アプリケーションが ArchestrA セキュリティ「なし」に設定されている場合、デフォルト ユーザーのログオン アカウント情報が使用され、オペレータはログオンするように要求されません。以下の手順では、システムが「Galaxy」、「OS User based」、または「OS Group based」など ArchestrA 認証モードに対して設定されていると仮定しています。

ArchestrA によりセキュリティが保護されたアプリケーションへのログオン

1. ArchestrA によりセキュリティが保護された InTouch アプリケーションを起動します。ログイン ダイアログ ボックスが表示されます。
2. 有効なユーザー名とパスワードを入力します。システムで認証されないと、再びログオンするよう促すメッセージが表示されます。

システムによってログオン アカウント情報が認証されると、セキュリティ モデルでユーザーに関連付けられている役割または許可に応じて、今後のすべての操作へのアクセスが許可されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

InTouch アプリケーションからのログオフ

オペレータは作業が完了した後で、InTouch アプリケーションからログオフします。オペレータによる操作なしで一定時間が経過した場合、オペレータを自動的にログオフするようにアプリケーションを設定することもできます。詳細については、「無操作タイムアウトの設定」を参照してください。

アプリケーションからのログオフ

1. ［ファイル］メニューで、［資格情報］を選択します。
2. InTouch HMI WindowMaker ユーザー セクションで ［サインアウト］を選択します。

注記：コンフィグレータの［ライセンス モード］タブで［**Operations Control 接続エクスペリエンス**］が選択されている場合、［資格情報］オプションは使用できません。

カスタム ログオン ウィンドウを作成する

［システム］メニューが WindowViewer で表示されていない場合、オペレータがアプリケーションにログオンするためのカスタム ログオン ウィンドウを作成できます。

カスタム ログオン ウィンドウを作成する

- $OperatorEntered、$PasswordEntered、および $OperatorDomainEntered システム タグ変数をデータ入力オブジェクトにリンクするか、スクリプトで使用して、ユーザー名、パスワード、およびドメイン名を設定します。これらのタグ変数は、書き込み操作のみを対象とした内部メッセージ型タグ変数です。

$OperatorDomainEntered タグ変数は、セキュリティ モードがオペレーティング システム ベースの場合のみ必要とされます。それ以外の場合、このタグ変数は無視されます。セキュリティ モードがオペレーティング システム ベースで、$OperatorDomainEntered が null の場合、ローカル コンピュータを指していると扱われます。

値が $PasswordEntered システム タグ変数に書き込まれると、$OperatorEntered、$PasswordEntered、および $OperatorDomainEntered システムタグ変数の値を使用してログオン試行が実行されます。値が $OperatorEntered または $OperatorDomainEntered システム タグ変数にのみ書き込まれた場合、ログオンは実行されません。

エントリが有効である場合、$AccessLevel と $Operator の内部タグ変数は（セキュリティ ユーザー リストで設定した）定義済みの値に設定されます。

［ユーザー入力 - 論理型］ボタンを $ConfigureUsers タグにリンクさせて、アクセス レベルが 9000 以上の認証されたオペレータが［ユーザーの設定］ダイアログ ボックスにアクセスして、セキュリティ ユーザー名リストを編集することを許可できます。オペレータがこのボタンを選択すると、$ConfigureUsers タグ変数の値が 1 に設定され、［ユーザーの設定］ダイアログ ボックスが表示されます。 オペレータがダイアログ ボックスを閉じると、システムによって値は 0 にリセットされます（これは書き込み専用のみを対象としたシステム論理型タグ変数です）。

注記：$ConfigureUsers タグは、セキュリティ タイプが InTouch に設定されている場合のみ機能します。ArchestrA ベースのセキュリティに対しては機能しません。

PostLogonDialog() 関数

InTouch の［ログオン］ダイアログ ボックスを表示し、TRUE を返します。

カテゴリ

セキュリティ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

構文

```
DiscreteTag=PostLogonDialog();
```

例

```
DiscreteTag=PostLogonDialog();
```

参照項目

InvisibleVerifyCredentials()、AttemptInvisibleLogon()、IsAssignedRole()、QueryGroupMembership()、AddPermission()

LogonCurrentUser() 関数

現在 Windows オペレーティング システムにログオンしているユーザー アカウントで InTouch にログオンします。

- OS セキュリティが設定された InTouch の場合：ユーザーは WindowViewer にログオンします。
- ArchestrA セキュリティが設定された InTouch の場合：ユーザーは Archestra OS ユーザーベースまたは OS グループベース セキュリティのメンバーである必要があります。
- ArchestrA OS ユーザーベースまたは OS グループベースのセキュリティが設定された InTouch およびスマート カードの資格情報で設定されたユーザー アカウントの場合：ユーザーはスマート カードのアカウント情報を使用してログオンします。スマート カードがリーダーから取り外されている場合、ユーザーはログオフされます。

カテゴリ

セキュリティ

構文

```
IntegerResult = LogonCurrentUser();
```

戻り値

ログオンが失敗すると -1 を返し、$Operator、$OperatorName、$OperatorDomain、および $AccessLevel に割り当てられる値に変更はありません。

備考

この関数は InTouch スクリプトのみで使用可能です。Archestra クライアント スクリプトでは使用できません。

例

```
IntegerResult = LogonCurrentUser();
```

次もご確認ください。

AddPermission() 関数

AttemptInvisibleLogon() 関数

InvisibleVerifyCredentials() 関数

IsAssignedRole() 関数

PostLogonDialog() 関数

QueryGroupMembership() 関数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Logoff() 関数

InTouch アプリケーションからユーザーをログオフします。

カテゴリ

セキュリティ（書き込み専用）

構文

```
DiscreteTag = LogOff();
```

備考

現在ログオンしているユーザーをログオフし、現在のユーザーのステータスをデフォルトの none に設定します。

例

```
DiscreteTag = LogOff();
```

参照項目

PostLogonDialog()、InvisibleVerifyCredentials()、IsAssignedRole()、AttemptInvisibleLogon()、QueryGroupMembership()、AddPermission()

AttemptInvisibleLogon() 関数

AttemptInvisibleLogon() 関数はスクリプトで使用して、指定したアカウント情報でユーザーを InTouch にログオンできます。ユーザーはパスワードまたはユーザー ID を入力する必要はありません。

カテゴリ

セキュリティ

構文

```
DiscreteTag=AttemptInvisibleLogon( "UserId", "Password", "Domain" );
```

引数

UserId
有効なユーザー アカウント名

パスワード
ユーザーのパスワード

ドメイン
ユーザーが所属するローカル コンピュータ、ワークグループ、またはドメインの名前。この引数は、現在のセキュリティ モードがオペレーティング システム ベースの場合のみ適用されます。

戻り値

認証が正常に行われると、TRUE を返します。それ以外の場合は、FALSE を返します。

備考

指定したアカウント情報を使用して、InTouch HMI へのログオンが試行されます。

- ログオン試行が正常に行われると、TRUE が返され、$OperatorDomain、$OperatorName、$AccessLevel、および $Operator のシステム タグ変数が適宜更新されます。
- ログオン試行に失敗した場合、FALSE が返され、現在ログオンしているユーザー（存在する場合）がそのまま現在のユーザーとなります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

Domain 引数は、オペレーティング システム ベースのセキュリティの場合のみ有効です。ArchestrA セキュリティ モードが使用中で、ArchestrA セキュリティが代わりにオペレーティング システム ベースのセキュリティを使用している場合、*UserId* 引数にはドメイン名またはコンピュータ名を使用した完全修飾ユーザー名が含まれます。

例

オペレーティング システム ベースのセキュリティの場合：

```
DiscreteTag=AttemptInvisibleLogon( "UserId", "Password", "Domain" );
```

InTouch ベースまたは ArchestrA ベースのセキュリティの場合：

```
DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "" );
```

次もご確認ください。

AddPermission() 関数

InvisibleVerifyCredentials() 関数

IsAssignedRole() 関数

PostLogonDialog() 関数

QueryGroupMembership() 関数

AttemptInvisibleLogonEx() 関数

AttemptInvisibleLogonEx() 関数はスクリプトで使用して、資格情報マネージャに保存された資格情報でユーザーを InTouch にログオンできます。ユーザーはパスワードまたはユーザー ID を入力する必要はありません。

カテゴリ

セキュリティ

構文

```
DiscreteTag=AttemptInvisibleLogonEx("Credential Name");
```

引数

資格情報名
資格情報マネージャに保存された資格情報の名前スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。

戻り値

認証が正常に行われると、TRUE を返します。それ以外の場合は、FALSE を返します。

備考

資格情報マネージャに保存された資格情報を使用して InTouch HMI へのログオンが試行されます。スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。

- ログオン試行が正常に行われると、TRUE が返され、$OperatorDomain、$OperatorName、$AccessLevel、および $Operator のシステム タグ変数が適宜更新されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- ログオン試行に失敗した場合、FALSE が返され、現在ログオンしているユーザー（存在する場合）がそのまま現在のユーザーとなります。

Domain 引数は、オペレーティング システム ベースのセキュリティの場合のみ有効です。ArchestrA セキュリティ モードが使用中で、ArchestrA セキュリティが代わりにオペレーティング システム ベースのセキュリティを使用している場合、*UserId* 引数にはドメイン名またはコンピュータ名を使用した完全修飾ユーザー名が含まれます。

例

オペレーティング システム ベースのセキュリティの場合：

```
DiscreteTag=AttemptInvisibleLogonEx("TestCredentialName01");
```

参照項目

PostLogonDialog()、InvisibleVerifyCredentials()、IsAssignedRole()、QueryGroupMembership()、AddPermission()

$OperatorEntered システム タグ変数

有効なユーザー名を入力するために使用します。

カテゴリ

セキュリティ

使用法

```
$OperatorEntered
```

備考

このタグ変数を使用して、カスタム ログオン ウィンドウを作成できます。タッチ入力オブジェクトや QuickScript をこのタグ変数にリンクして、ログオンのユーザー名を設定できます。

注記 : $OperatorEntered システム タグ変数が有効である場合、$AccessLevel や $Operator のシステム タグ変数は事前定義された値に設定されます。

データ型

メッセージ型（書き込み専用）

参照項目

$AccessLevel、$Operator、$PasswordEntered、$ChangePassword、$ConfigureUsers

$PasswordEntered システム タグ変数

有効なパスワードを入力するために使用します。

カテゴリ

セキュリティ

使用法

```
$PasswordEntered
```

備考

$PasswordEntered システム タグ変数は常に空の文字列として読み取られます。このシステム タグ変数に関連付けられた表示リンクは常に空白です。タグ変数は常に空の文字列を返すため、データ変化スクリプトはこのタグ変数をトリガオフしません。このタグ変数を使用して、カスタム ログオン ウィンド

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ウを作成できます。タッチ入力オブジェクトや QuickScript をこのタグ変数にリンクして、ユーザーのパスワードを設定できます。

注記 : $PasswordEntered が有効である場合、$AccessLevel や $Operator のシステム タグ変数は事前定義された値に設定されます。

データ型

メッセージ型（書き込み専用）

参照項目

$AccessLevel、$Operator、$OperatorEntered、$ChangePassword、$ConfigureUsers

$OperatorDomainEntered システム タグ変数

ドメイン名はオペレータによって入力されます。

カテゴリ

セキュリティ

備考

$PasswordEntered タグ変数が変更されると、ダイアログ ボックスを表示しない状態でログオンが試行されます。このログオン試行では、入力ユーザー名として $*Entered タグ変数が、ドメイン名として $OperatorDomainEntered の文字列値が使用されます (現在のモードがオペレーティング システム ベースのセキュリティの場合のみ使用)。モードがオペレーティング システム ベースでない場合、このタグ変数は無視されます。

データ型

文字列

例

```
$OperatorEntered == "john";
$OperatorDomainEntered == "Corporate_HQ";
$PasswordEntered == "password";
```

参照項目

$Operator

オペレータまたはアクセス レベルに基づく機能の有効化と無効化

アプリケーションのセキュリティを実装した後で、$AccessLevel および $Operator セキュリティ タグ変数を、ボタン、アニメーション リンクの式、または QuickScript で使用すると、ログオンしているオペレータが特定のアプリケーション機能を実行することが許可されるかどうかを制御できます。

たとえば、ログオンしているユーザーのアクセス レベルに基づいてオブジェクトを表示させるには、以下のステートメントを表示アニメーション リンクの式に使用します。

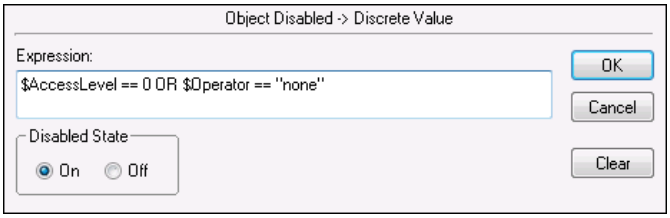
```
$AccessLevel >= 2000;
```

または、IF ステートメントによってスクリプトをバインドできます。

```
IF $Operator == "DayShift" THEN
   ［コントロールパネルウィンドウ］を表示します。
   {DayShift Operator に対してのみ実行するその他のライン}
ENDIF;
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

無効化アニメーション リンクを使用して、内部セキュリティ タグ変数の値に基づいてオブジェクトのタッチ リンク機能を制御することもできます。以下に例を示します。



この式を使用することによって、ログオンしているユーザーがいない場合、オブジェクトやボタンが変更されないように保護されます。

InvisibleVerifyCredentials() 関数

InvisibleVerifyCredentials() 関数は同期 QuickScript で使用して、ユーザーを InTouch HMI にログオンすることなく特定のユーザーのアカウント情報を確認できます。

カテゴリ

セキュリティ

構文

```
AnalogTag=InvisibleVerifyCredentials( "UserId", "Password", "Domain" );
```

引数

UserId
ローカル コンピュータ、ワークグループ、またはドメインの一部である Windows オペレーティング システムのユーザー アカウント名

パスワード
アカウントのパスワード

ドメイン
アカウントの Windows ドメイン

備考

指定したユーザー、パスワード、およびドメインが有効である場合、そのユーザーに関連付けられた該当するアクセス レベルが整数として返されます。それ以外の場合は、-1 が返されます。

注記： InvisibleVerifyCredentials() 関数は同期 QuickScript から実行する必要があります。非同期 QuickScript から実行されると、関数は常に -1 を返します。

この関数によって、現在ログオンしているユーザーは変更されません。Domain 引数は、オペレーティング システム ベースのセキュリティの場合のみ有効です。ArchestrA セキュリティ モードが使用中で、ArchestrA セキュリティが代わりにオペレーティング システム ベースのセキュリティを使用している場合、UserId 引数にはドメイン名またはコンピュータ名を使用した完全修飾ユーザー名が含まれる必要があります。

例

```
AnalogTag=InvisibleVerifyCredentials( "john", "Password", "corporate_hq" );
```

次もご確認ください。

AddPermission() 関数

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

AttemptInvisibleLogon() 関数

IsAssignedRole() 関数

PostLogonDialog() 関数

QueryGroupMembership() 関数

InvisibleVerifyCredentialsEx() 関数

InvisibleVerifyCredentialsEx() 関数は同期 QuickScript で使用して、ユーザーを InTouch HMI にログオンすることなく特定のユーザーの資格情報を確認できます。

カテゴリ

セキュリティ

構文

```
AnalogTag=InvisibleVerifyCredentialsEx("Credential Name");
```

引数

資格情報名
資格情報マネージャに保存された資格情報の名前スタンドアロン InTouch アプリケーションの場合、資格情報はアプリケーション マネージャから取得されます。マネージド InTouch アプリケーションの場合、資格情報は Application Server の資格情報マネージャから取得されます。

備考

指定したユーザー、パスワード、およびドメインが有効である場合、そのユーザーに関連付けられた該当するアクセス レベルが整数として返されます。それ以外の場合は、-1 が返されます。

注記：InvisibleVerifyCredentialsEx() 関数は同期 QuickScript から実行する必要があります。非同期 QuickScript から実行されると、関数は常に -1 を返します。

この関数によって、現在ログオンしているユーザーは変更されません。Domain 引数は、オペレーティング システム ベースのセキュリティの場合のみ有効です。ArchestrA セキュリティ モードが使用中で、ArchestrA セキュリティが代わりにオペレーティング システム ベースのセキュリティを使用している場合、UserId 引数にはドメイン名またはコンピュータ名を使用した完全修飾ユーザー名が含まれる必要があります。

例

```
AnalogTag=InvisibleVerifyCredentialsEx("john", "Password", "corporate_hq");
```

参照項目

PostLogonDialog()、AttemptInvisibleLogon()、IsAssignedRole()、QueryGroupMembership()、AddPermission()

現在ログオンしているオペレータに関する情報の取得

監査は、すべてのセキュリティ システムでのプライマリ機能です。一組のシステム タグ変数を使用して、InTouch アプリケーションにログオンしているユーザー、ユーザーのログオン元であるドメイン、およびログオンがいつ試行されたかなどの情報を特定できます。

GetAccountStatus() 関数

ユーザーのパスワードが期限切れとなるまでの日数を返します。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

セキュリティ

構文

```
Result=GetAccountStatus(Domain, UserID);
```

引数

ドメイン
ユーザー アカウントが配置されているドメインまたはローカル コンピュータの名前

UserID
ローカル コンピュータ、ワークグループ、またはドメインの一部である Windows ユーザー アカウント名

戻り値

この関数は以下の値も返します。

結果	説明
-1	アカウント パスワードが期限切れです
-2	アカウント パスワードは期限切れとなりません
-3	アカウントがロック アウトされています
-4	アカウントが無効です
-5	アカウント情報が失敗しました

備考

オペレーティング システム ベースのセキュリティでのみこの関数を使用します。ArchestrA セキュリティ モードではこの関数を使用しないでください。

GetAccountStatus() 関数が ArchestrA セキュリティで使用されると、スクリプトはアカウント情報をドメイン コントローラから直接取得しようと試みます。ArchestrA Galaxy が同じドメインでオペレーティング システム セキュリティを使用している限り、この処理は機能します。

例

```
Status = GetAccountStatus("Corporate_HQ","Operator");
```

IsAssignedRole() 関数

現在ログオンしているユーザーが指定したユーザー役割のメンバであるかどうかを判断します。ArchestrA セキュリティに対してのみ適用されます。

カテゴリ

セキュリティ

構文

```
DiscreteTag=IsAssignedRole( "RoleName" );
```

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

引数

RoleName
Application Server ユーザーに関連付けられている役割

備考

ArchestrA セキュリティ モードのみに対して有効で、現在ログオンしているユーザーに適用されます。ユーザーが現在ログオンしており、Galaxy IDE で *RoleName* 役割が割り当てられている場合、TRUE が返されます。それ以外の場合は、FALSE が返されます。

例

```
DiscreteTag=IsAssignedRole( "Administrators" );
```

参照項目

AttemptInvisibleLogon()、PostLogonDialog()、InvisibleVerifyCredentials()、QueryGroupMembership()、AddPermission()

QueryGroupMembership() 関数

現在ログオンしているユーザーが指定したユーザー グループのメンバであるかどうかを判断します。オペレーティング システム セキュリティに対してのみ適用されます。

カテゴリ

セキュリティ

構文

```
DiscreteTag=QueryGroupMembership( "Domain", "Group" );
```

引数

ドメイン
グループが配置されているドメインまたはローカル コンピュータの名前

グループ
グループの名前

備考

オペレーティング システム セキュリティ モードに対してのみ有効で、現在ログオンしているユーザーに適用されます。ユーザーが現在ログオンしており、ドメインに配置されているグループの一部である場合、TRUE が返されます。それ以外の場合は、FALSE が返されます。

QueryGroupMembership() 関数はオペレーティング システム ベースのセキュリティで動作し、ArchestrA セキュリティがオペレーティング システム ベースのセキュリティに設定されている場合のみ ArchestrA セキュリティで動作します。

例

```
DiscreteTag=QueryGroupMembership( "corporate_hq", "InTouchAdmins" );
DiscreteTag=QueryGroupMembership( "Domain", "Group" );
```

参照項目

PostLogonDialog()、InvisibleVerifyCredentials()、IsAssignedRole()、AttemptInvisibleLogon()、AddPermission()

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

$OperatorName システム タグ変数

オペレーティング システム ベースまたは ArchestrA の認証が使用されており、誰かがログオンし、まだログオフしていない場合、オペレータのフル ネームが含まれます。それ以外の場合、ログオンしたユーザーの名前が含まれます（$Operator タグ変数の内容と同じ）。

カテゴリ

セキュリティ

データ型

文字列（読み取り専用）

例

```
IF $OperatorName <> "" THEN
   {デフォルト値を設定}
ENDIF;
```

参照項目

$Operator

$OperatorDomain システム タグ変数

使用されているセキュリティのタイプに基づいて、異なる値が含まれます。

- オペレーティング システム ベースのセキュリティが選択され、オペレータが正常にログオンしている場合、$OperatorDomain タグ変数にはログオン時に指定されたドメインまたはノード名が含まれます。
- ArchestrA セキュリティが選択され、ユーザーがログオンしている場合、$OperatorDoamin には "ArchestrA" が含まれます。
- セキュリティが選択されている場合、$OperatorDomain タグ変数には文字列 "InTouch" が含まれます。
- 「なし」が選択されている場合、空白の文字列（""）となります。

カテゴリ

セキュリティ

データ型

文字列

例

```
IF $OperatorDomain == "PRODUCTION" THEN
   {限界値の変更を許可}
ELSE
   {変更が拒否されました}
ENDIF;
```

参照項目

$Operator

$Operator システム タグ変数

ログオンしているユーザーのログオン名が含まれます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

カテゴリ

セキュリティ

データ型

文字列

$VerifiedUserName システム タグ変数

InvisibleVerifyCredentials() 関数の呼び出しが正常に行われ、セキュリティ モードがオペレーティング システム ベースまたは ArchestrA Application Server ベースに設定されている場合、確認されたユーザーのフル ネームが含まれます。呼び出しが失敗した場合、システム タグ変数は null に設定されます。

カテゴリ

セキュリティ

使用法

```
$VerifiedUserName
```

備考

$VerifiedUserName システム タグ変数が変更されると（InvisibleVerifyCredentials() 関数が呼び出されるとき）、イベントが生成されます。

データ型

メッセージ型（読み取り専用）

有効値

ユーザーのフル ネーム

例

```
Tag = InvisibleVerifyCredentials( "john","password", "Plant_Floor");{ If the call is
successful, the $VerifiedUserName is set to "John Smith" and an Operator Event is
generated. 上記の呼び出しが正常に行われなかった場合、$VerifiedUserName は "" に設定されます。}
```

参照項目

InvisibleVerifyCredentials()、$OperatorName, $Operator

セキュリティ システム タグと関数の概要

以下の表には、それぞれのセキュリティ モードで使用できるセキュリティ システム タグと関数が示されています。

	InTouch セキュリティ	オペレーティング システム セキュリティ	**ArchestrA** セキュリティ
$AccessLevel	可	Yes	可
$ChangePassword	可	Yes	可
$ConfigureUsers	可	いいえ	不可
$InactivityTimeout	可	Yes	可

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

	InTouch セキュリティ	オペレーティング システム セキュリティ	ArchestrA セキュリティ
$InactivityWarning	可	Yes	可
$Operator	可	Yes	可
$OperatorDomain	不可	可	使用可能*
$OperatorDomainEntered	不可	可	使用可能*
$OperatorEntered	可	Yes	可
$OperatorName	可	Yes	可
$PasswordEntered	可	Yes	可
$VerifiedUserName	不可	Yes	可
AddPermission()	不可	可	不可
AttemptInvisibleLogon()	可	Yes	可
AttemptInvisibleLogonEx()	可	Yes	可
ChangePassword()ChangePassword()	可	いいえ	不可
EnableDisableKeys()	可	Yes	可
GetAccountStatus()	不可	可	使用可能*
InvisibleVerifyCredentials()	不可	可	使用可能*
InvisibleVerifyCredentialsEx()	不可	可	使用可能*
IsAssignedRole()	不可	いいえ	可
Logoff()	可	Yes	可
LogonCurrentUser()	不可	可	使用可能*
PostLogonDialog()	可	Yes	可
QueryGroupMembership()	不可	可	使用可能*

* ArchestrA セキュリティが OS ユーザーまたはグループ ベースの場合

管理者以外のユーザーに許可されたアプリケーション マネージャの操作

管理者以外のユーザーが実行できる操作は、管理者に許可された操作のサブセットです。管理者以外のユーザーによる基本的な操作へのアクセスを制限することによってセキュリティの脅威を防止できます。以下の表には、InTouch HMI アプリケーション マネージャの各タブで実行できる操作が一覧表示されています。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

［**InTouch**］タブ

変換処理	管理者以外
新規アプリケーション	不可
WindowMaker の起動	不可
WindowViewer の起動	可
DBLoadDBLoad	不可
DBDumpDBDump	不可
アプリケーションの削除	不可
アプリケーション名の変更	不可
アプリケーションのプロパティ	不可
テンプレートとしてのエクスポートする	不可
アプリケーションのインポート	不可
OPC UA サーバー設定	可
アプリケーションの検索	可
IoT 用にエクスポート	可
CONNECT へのアップロード	可
エッジ デバイスのユーザーの設定	不可
AVEVA Insight データ ソースとしてのタグ データのパブリッシュ	可
ノードのプロパティ	
アプリ開発 > サービスとしての WindowViewer でのアプリケーションの起動	不可
アプリ開発 > ネットワーク アプリケーション開発の有効化	可
解決策	可
メモリ設定	不可
パフォーマンス	不可
[最新の情報に更新]	可
表示	可
CONNECT ログイン	可
サムネイルの変更	可

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

変換処理	管理者以外
アプリケーション フォルダを開く	可

［**Web Client**］タブ

変換処理	管理者以外
Web Client の有効化	不可
Web Client の起動	可
Web Client の設定	
グラフィックのリフレッシュ レート	不可
アラームのリフレッシュ レート	不可
Web Client サイト名	不可
ヘッダーを表示	不可
ナビゲーション バーを有効化	不可
匿名アクセスを許可	不可
AIM 登録設定	
AIM サーバーを認証サーバーとして使用する	不可
Identity Server	該当なし
ユーザー名	不可
Password	不可
セキュア ゲートウェイ	不可
Web サイトへの産業用グラフィックの埋め込みを許可する	不可

アプリケーション フォルダへの役割ベースのセキュリティの適用

役割ベースのセキュリティをアプリケーション フォルダに適用すると、承認されたユーザーは、アプリケーション フォルダ内のリソースにアクセスしてアプリケーションを編集できます。役割ベースのセキュリティが適用されたアプリケーションはセキュア アプリケーションと呼ばれます。役割ベースのセキュリティが適用されていないアプリケーションは非セキュア アプリケーションと呼ばれます。既存のアプリケーションをインポートまたは変更するときでもアプリケーション フォルダを保護できます。

既定では、InTouch アプリケーションは、作成された場所から親フォルダからアクセス許可を継承し、追加のアクセス許可は適用されません。スタンドアロンアプリケーションのアプリケーションフォルダへのアクセスを制限するには、［**ツール**］>［**ノードのプロパティ**］で［**セキュリティ**］の下の［アプリケーションマネージャ］で［**InTouchDevelopers** と **InTouchOperators グループ**］**オプションを使用するユーザーに対して、スタンドアロンの InTouch** Application へのアクセス >を制限します。ノード上でセキ

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

ュリティ機能を有効にすると、厳格な読み取りおよび書き込み権限が InTouch View アプリケーションに適用されます。

セキュリティ ユーザー役割の定義

InTouch HMI がインストールされている場合、以下の InTouch ユーザー グループがローカル ユーザー グループに自動的に追加されます。

- InTouch 開発者 - このグループのユーザーには InTouch アプリケーション ルート フォルダのフル コントロールが付与されます。アプリケーション全体への読み取り/書き込み権限でアプリケーションを編集および管理できます。
- InTouch オペレータ - このグループのユーザーには InTouch アプリケーション ルート フォルダの制限付きコントロールが付与されます。アプリケーションを実行できます。ほとんどのファイルへの読み取り専用アクセスがあり、いくつかのファイルには読み取り/書き込み権限が必要になります。

InTouch プリケーションへのアクセス制限は、'InTouchDevelopers' と 'InTouchOperators' グループのユーザーに対してのみ適用できます。

ドメイン グループの割り当て

ドメイン グループを 'InTouchDevelopers' または 'InTouchOperators' グループに割り当てることができます。ドメイン グループに属するユーザーには、開発者またはオペレータ レベルのアクセスが付与されます。

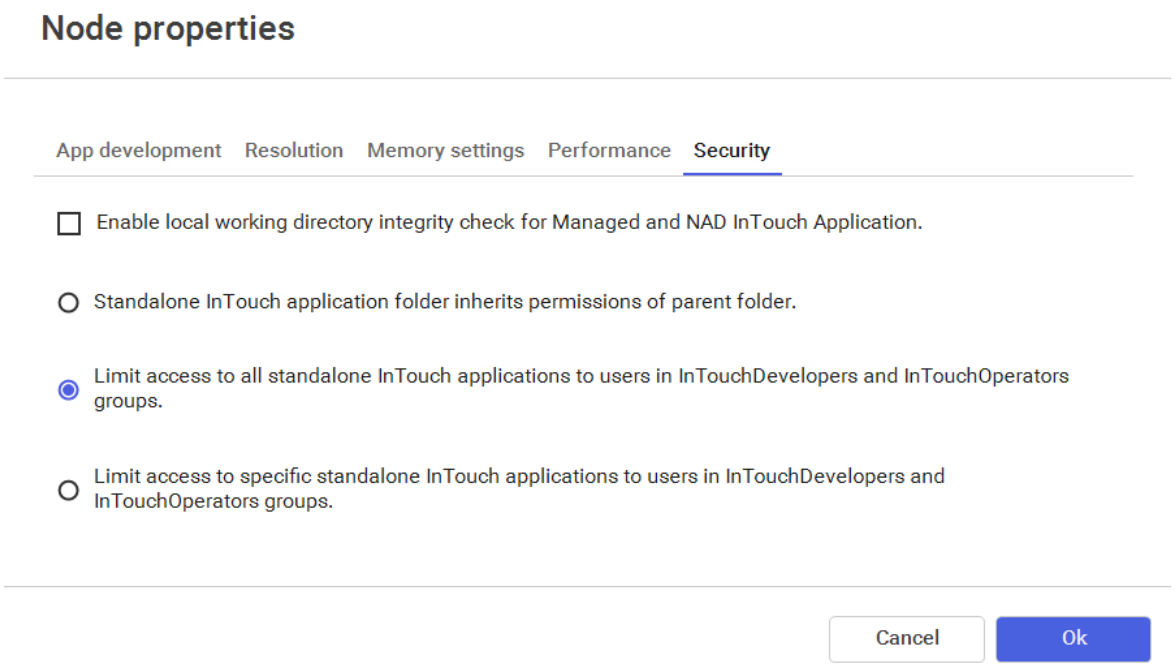
ローカル グループを 'InTouchDevelopers' または 'InTouchOperators' グループに割り当てることはできません。ユーザーを 'InTouchDevelopers' または 'InTouchOperators' グループに追加することが推奨されます。

InTouch アプリケーション フォルダのセキュリティを有効にするには

1. 管理者としてアプリケーション マネージャを起動します。
2. ［ツール］メニューの［ツール］タブで［セキュリティ］をクリックします。

   ［セキュリティ］画面が表示されます。
3. **InTouchDevelopers および InTouchOperators グループ内のユーザーに対して、スタンドアロンのインタッチアプリケーションへのアクセスを制限する**を選択し、すべてのアプリケーションフォルダに対するセキュリティを有効にするか、 **InTouchDevelopers および InTouchOperators グループ内のユーザーに特定のスタンドアロンの InTouch アプリケーションへのアクセスを制限する**を選択し、選択したアプリケーションフォルダのみにセキュリティを有効にします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



4. ［**OK**］ を選択します。

追加情報

- ［このノードのプロパティ］ダイアログで［OK］を選択すると、アプリケーションフォルダへのアクセス制御リスト(ACL)が適用されます(マシン単位またはアプリケーション単位でのセキュリティ保護を目的とします)。
  - マシンごとに、フォルダを持つすべてのアプリケーションの ACL が更新されます。
  - アプリケーションごとに、ユーザーが選択したアプリのみが更新されます。
- UNC(ユニバーサル命名規則)パスやマッピングされたネットワークドライブ上にあるアプリケーションはすべて除外されます。
- 「アプリケーションの検索」または「アプリケーションのインポート」を実行し、アプリケーションフォルダのセキュリティをマシンレベルで有効にしている場合、そのセキュリティは新しく追加されたアプリケーションフォルに適用されます。
- この構成の設定は、「SecureApplicationFolder」エントリを持つ次のパスのファイルにあります。

  C:\ProgramData\wonderware\InTouch\CONFIGURATION.ini

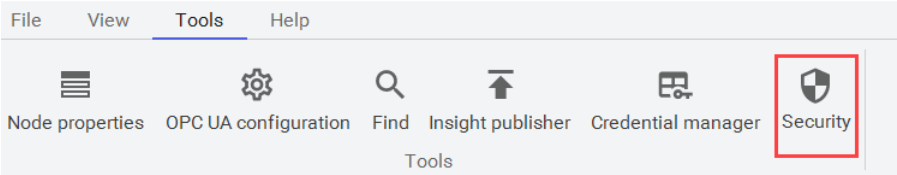
ユーザーとグループの設定

［セキュリティ］ダイアログでは、ユーザーとグループを InTouchDevelopers ロールまたは InTouchOperators ロールのメンバーに設定できます。

ユーザーとグループを設定するには

1. 管理者としてアプリケーション マネージャを起動します。
2. ［ツール］メニューの［ツール］タブで［セキュリティ］をクリックします。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



［セキュリティ］画面が表示されます。

3. ［+ ドメインユーザーとグループ］] または ［+ ローカルユーザー］をクリックします。

   ［ユーザーまたはグループを選択］または［ユーザーを選択］ダイアログがそれぞれ表示されます。

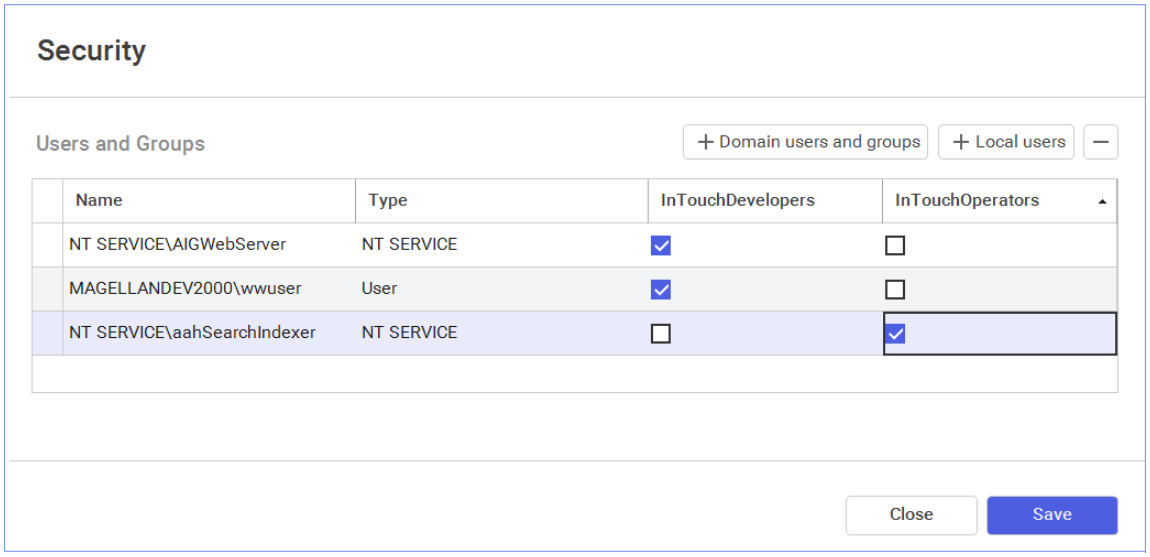
   注記：Windows Active Directory(AD)ユーザーとグループ、Windows ローカルユーザーがサポートされています。Windows ローカルグループはサポートされていないため、追加できません。

4. ユーザー名またはグループ名を入力して［**OK**］をクリックします。

   注：ユーザーとグループは、オブジェクト名、タイプ、または位置情報で検索できます。［名前の確認］をクリックして、選択したオブジェクトが AD 名またはグループに解決されることを確認します。

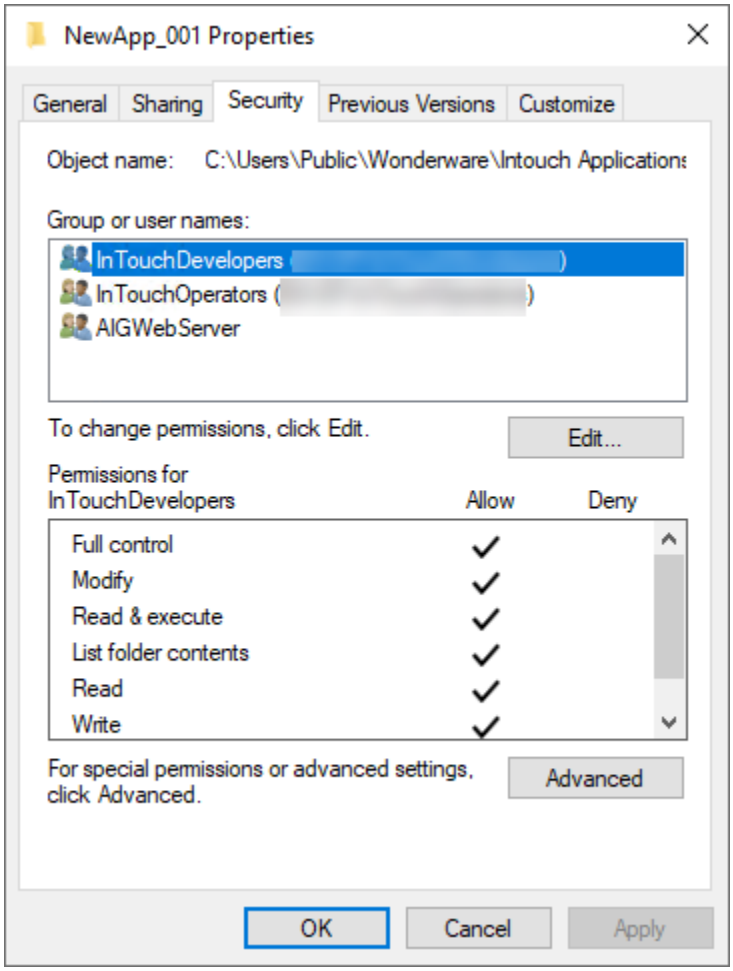
   入力したユーザーまたはグループが［**ユーザーとグループ**］グリッドに表示されます。

5. ユーザー/グループを InTouch 開発者として割り当てるには、［**InTouchDevelopers**］列のチェック ボックスをオンにします。
6. ユーザー/グループを InTouch オペレータとして割り当てるには、［**InTouchOperators**］列のチェックボックスをオンにします。
7. ［保存］をクリックします。



既定では、NT SERVICE\AIGWebServer と現在ログオンしているユーザーがメンバーとして追加され、NT SERVICE\aahSearchIndexer が InTouchOperators グループに追加されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.



ユーザーとグループを削除するには

1. ［ユーザーとグループ］グリッドで、削除するエントリを選択します。
2. アイコンをクリックします。

   選択したユーザー/グループが削除されます。

追加情報

- InTouchDevelopers と InTouchOperator は、マシン上のローカルグループとして維持されます。
- InTouchDevelopers および InTouchOperators のセキュリティ権限は、セキュリティアクセス権は PIVAS アプリケーションには適用されません。ネットワークパス上にあるアプリケーションやマッピングされたネットワークドライブは、すべて、そのアプリケーションは、例えば、ネットワークドライブ（例: ネットワークドライブ）と見なされます。
- 上から［セキュリティ］ダイアログに追加されたユーザーはすべて、対応するグループに追加されます。
- これらのグループには、スタンドアロンの InTouch アプリケーションの各アプリケーションフォルダへのアクセス権が付与されます。
- この機能を有効にして新しいアプリケーションを作成した場合、InTouchDevelopers と InTouchOperators に適切なアクセス権を持つフォルダーのアクセス制御リストが更新されます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

- 移行されたアプリケーションには、移行プロセス中に適用されるセキュリティ権限が付与されます。
- アプリケーションフォルダを開いてフォルダのプロパティを表示する場合、［セキュリティ］タブを開いて、適用されたアクセス許可を表示できます。

マネージド InTouch アプリケーションおよび NAD InTouch アプリケーションのローカル作業ディレクトリの完全性チェック

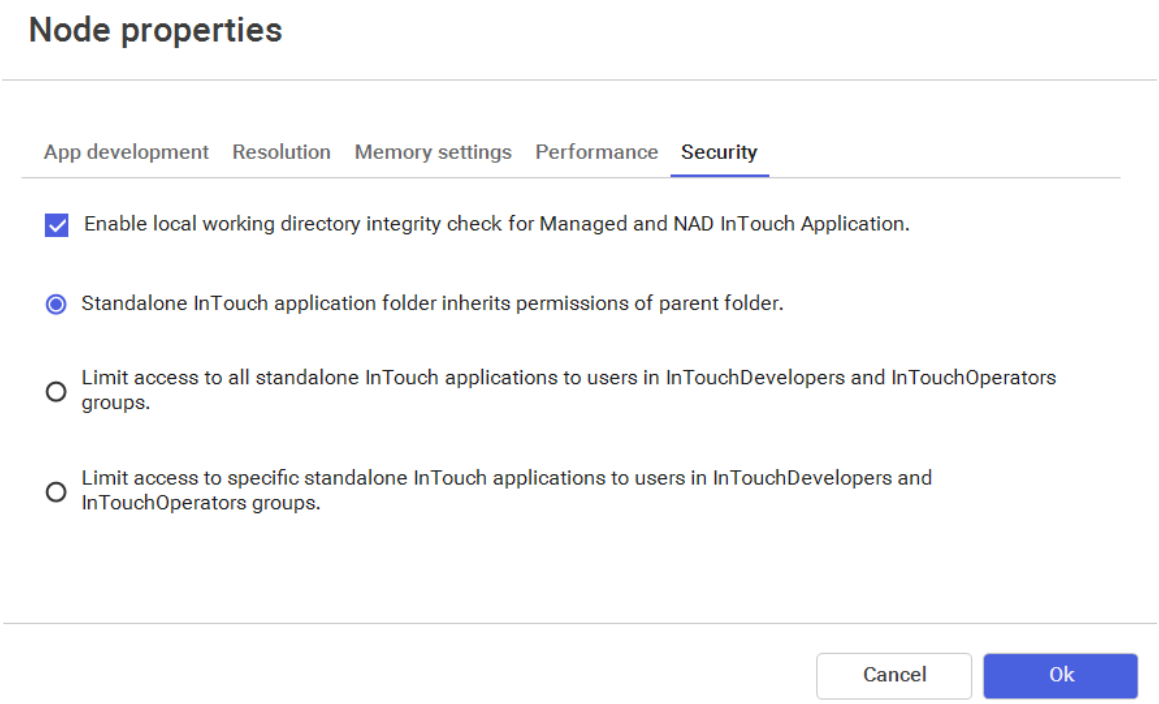
この機能を使用して、配置されたアプリケーションとローカル作業ディレクトリ内のアプリケーションを比較できます。マネージドおよび NAD アプリケーション用の WindowViewer を起動すると、配置済みアプリケーションとローカル作業ディレクトリ内のアプリケーションの間でファイルタイムスタンプとファイルコンテンツハッシュの比較が実行されます。比較は特定のファイルタイプ（.dlls、.exe、.vedef など）に制限されています。ファイルが一致しない場合、WindowViewer は配置済みアプリケーションをローカル作業ディレクトリにコピーしてアプリケーションを実行します。

完全性チェックを有効にするには

1. アプリケーションマネージャで、**[ツール]>[ノードのプロパティ]>[セキュリティ]**に移動します。
2. **[Enable local working directory integrity check for Managed and NAD InTouch Applications]**チェックボックスをオンにします。

   注：このオプションは**[Limit access to Standalone InTouch applications to users in InTouchDevelopers and InTouchOperators groups]**オプションと連動していません。これらは同じノード上で両方または個別に有効にできます。

3. ［**OK**］ を選択します。



注：コピー処理によって WindowViewer の起動に遅延が発生する場合があります。ノード上でネットワークアプリケーション開発（NAD）が有効になっている場合、セキュリティモードのオプションは無効になります。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

マネージド InTouch アプリケーションの動作

次のいずれかに当てはまる場合は比較後にファイルが一致しません。

- 配置済みアプリケーションに新しいバージョンがある。
- ローカル作業ディレクトリ内のアプリケーションファイルの内容が変更されている。

どちらのシナリオでも、WindowViewer は配置済みアプリケーションをローカル作業ディレクトリにコピーしてアプリケーションを実行します。これが原因で起動時間の遅延が発生します。

アプリケーションの InTouch.ini ファイルには次の 2 つの追加設定があります。

1. 名前：NewDefaultLocalWorkingDirectory
   - 2023R2 以降の新規アプリケーションの場合、値は 1 になります。
   - 2023R2 以降の移行済みアプリケーションの場合、値は 0 になります。
2. 名前：VIEWEDMANAGEDDIALOG

   [ファイル]>[設定]>[WindowViewer]>[マネージドアプリケーション]に移動すると、値は 1 に設定されます。これは、ローカル作業ディレクトリが更新されたという通知が 1 回のみ表示されるためです。

通知：

新しいマネージドアプリケーションの場合は、既定のローカル作業ディレクトリが%LOCALAPPDATA%\ArchestrA\ManagedApp に更新されたという内容の通知になります。

移行済みマネージドアプリケーションの場合は、推奨されるローカル作業ディレクトリが%LOCALAPPDATA%\ArchestrA\ManagedApp であるという内容の通知になります。

既定の作業ディレクトリを復元するには：

マネージドアプリケーションの WindowViewer で、［**ファイル**］>［**設定**］>［**WindowViewer**］>［**マネージドアプリケーション**］に移動し、ページ下部にある［**デフォルトを復元**］を選択します。

[Local working directory]フィールドが次の内容に更新されます。

%LOCALAPPDATA%\ArchestrA\ManagedApp

NAD InTouch アプリケーションの動作

次のいずれかに当てはまる場合は比較後にファイルが一致しません。

- NAD リポジトリが更新されている。
- ローカル作業ディレクトリ内のアプリケーションファイルの内容が変更されている。

どちらのシナリオでも、WindowViewer は NAD リポジトリ アプリケーションをローカル作業ディレクトリにコピーしてアプリケーションを実行します。

SP 2023 R2 より前のノードで NAD が以前から有効になっており、SP 2023 R2 以降にアップグレードされた場合、ローカル作業ディレクトリパスは変更されません。新しいファイルパスを推奨する通知が表示されます。

System Platform 2023 R2 以降の新規インストールにおける、NAD の既定のローカル作業ディレクトリパスは以下のとおりです。

C:\Users*USER*\AppData\Local\NAD

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

付録 AL InTouch HMI のセキュリティの管理

このセクションでは、InTouch HMI のさまざまなセキュリティ ガイドラインとセキュリティ設定について説明します。詳細については、『サイバーセキュリティ導入ガイド』を参照してください。

InTouch HMI のセキュリティ設定

InTouch HMI で設定する必要のあるセキュリティ領域、および該当する手順を示すセクションの詳細を以下の表に示します。

セキュリティ領域	このガイド内のトピック	サマリ
ホスト上のアプリケーションおよびコンテンツの保護	無操作タイムアウトの設定	操作を行っていないオペレータを WindowViewer アプリケーションから自動的にログオフするように InTouch を設定します。 接続エクスペリエンスが有効になっている場合、非アクティブタイムアウトはサポートされません。
認証および承認によるシステムのセキュリティ	認証ベースおよび承認ベースのセキュリティ	ユーザーは InTouch アプリケーションを使用する前に認証を受ける必要があります。InTouch では、認証されたユーザーが特定の機能を使用できるかどうかが検証されます。
	仮想アカウントの使用	仮想アカウントを使用すると、アラーム関数にアクセスする際のセキュリティを強化することができます。『InTouch HMI アラームとイベント ガイド』の「仮想アカウントの使用」を参照してください。
ホスト上のアプリケーションおよびコンテンツの保護	システム キーのロック	InTouch アプリケーションを実行しているコンピュータでシステム キーを無効にすることによって、標準の Windows 機能へのオペレータ アクセスを制限します。
認証および承認によるシステムのセキュリティ		

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.

セキュリティ領域	このガイド内のトピック	サマリ
ICS ソフトウェアでのユーザーとグループの管理	InTouch ベースのセキュリティの使用	機能を内部タグにリンクすることによって、オペレータが実行できる機能を制限します。
Windows でのユーザーとグループの管理	オペレーティング システム ベースのセキュリティの使用	Windows オペレーティング システムからユーザー/グループ アカウント ポリシーを継承します。
アプリケーション データ フォルダの保護	アプリケーション フォルダへの役割ベースのセキュリティの適用	ユーザーの役割とユーザー グループを定義して、アプリケーション データ フォルダへのアクセスを管理できます。

© 2015-2026 AVEVA Group Limited またはその子会社。All rights reserved.