AVEVA™ System Platform Deployment

2023 R2

Contents

Welcome

The AVEVA™ System Platform Deployment Guide provides recommendations and best practices information to help plan, design and implement integration projects.

The information contained in this guide is based on past experience from building multiple projects using the System Platform/Application Server infrastructure to build AVEVA™ OMI and InTouch HMI applications. Recommendations contained in this document should not prevent you from discovering and using other methods and procedures that work effectively.

What's new in the System Platform Deployment Guide

The release history for the System Platform Deployment Guide is listed by calendar year.

Releases in 2024

Releases in 2023

Releases in 2024

August 12, 2024	Remove broken hyperlink from "Script Editing Styles and Syntax" topic.
January 17, 2024	Updates to System Platform port information

Releases in 2023

October 16, 2023	This is the initial release of the AVEVA System Platform Deployment Guide. The purpose of this guide is to provide an overview of the requirements and processes needed to deploy AVEVA System Platform successfully.

Introduction

Application Server is the underlying infrastructure for System Platform. All System Platform-based applications, whether they use AVEVA OMI or InTouch HMI visualization components, are built on top of an Application Server framework.

Assumptions

This deployment guide is intended for:

- Engineers and other technical personnel who will be developing and implementing System Platform solutions.
- Sales engineers and other sales personnel who need to define system topologies in order to submit System Platform project proposals.

You should be familiar with Microsoft Windows and Windows Server operating systems, as well as with a scripting, programming, or macro language. Also, an understanding of concepts such as variables, statements, functions, and methods will help you to achieve best results.

System Platform functional components

This Deployment Guide has been updated to include current AVEVA™ software product versions (current Service Packs are assumed for all software). However, the information is designed to apply to previous versions except where noted and is applicable to the following System Platform component products:

AVEVA™ Application Server

AVEVA Application Server provides an object-based framework repository to construct an asset hierarchy of your physical processes. This repository, called a Galaxy, is built on a SQL Server database that manages the deployment and operation of the run-time elements that form a System Platform application. Application Server functions can be divided by function onto multiple nodes. The Bootstrap module is automatically installed on each Application Server node.

- GR node (configuration): Galaxy Repository node. Requires a supported version of SQL Server.
- IDE node (configuration): Engineering development node. Requires installation of the IDE and optionally, InTouch WindowMaker.
- AOS node (run time): Application Object Server node. Requires Bootstrap module only. Object instances run on AOS nodes when they are deployed.
- DAS node (run time): Data Acquisition Server or I/O node. Requires installation of AVEVA™ Communication Drivers. Serves as the communication bridge between AOS nodes and the PLCs, RTUs, etc. in your control network.
- OMI node (run time): Visualization client node. Requires Bootstrap module only. OMI ViewApps run on these

nodes after being deployed.

- InTouch HMI node (run time): Visualization client node. Requires Bootstrap module only. InTouch WindowViewer runs on these nodes after being deployed.

AVEVA™ Development Studio

AVEVA Development Studio is an engineering environment for developing, maintaining, and managing for all supervisory SCADA and HMI application development. It provides a shared development environment that helps you drive standards and best practices across your company. Development Studio includes the System Platform IDE, the Industrial Graphic Editor, and optionally, InTouch HMI WindowMaker.

AVEVA™ Historian

AVEVA Historian is a process database integrated with operations control, enabling access to your process, alarm, and event history data. It stores plant data from Communication Drivers and other data sources, allowing operators to make real-time decisions from a secure and trustworthy set of industrial data. AVEVA Historian also contains summary, configuration, and system monitoring information. AVEVA Historian enhances and is tightly coupled to Microsoft SQL Server.

Visualization and Analysis Clients

Visualization and analysis clients enable you to visualize real-time and historical data from System Platform.

- AVEVA™ InTouch HMI WindowViewer provides an optional visualization capability for control and optimization of any industrial and manufacturing process through unparalleled situational awareness.
- AVEVA™ Operations Management Interface (OMI) is an advanced visualization component bundled with Application Server. OMI delivers immersive control applications that weave context throughout the visual design, including situational awareness concepts, for improved operator performance. With AVEVA OMI you can create responsive visualization that organizes content based on user device and utilize AVEVA System Platform's asset model hierarchy to drive intuitive navigation, adapting content and context based on user selection.
- AVEVA™ Insight is an AI-infused SaaS application for asset reliability and operational performance visualization for hybrid markets. Users can access critical asset and process data with powerful visualization tools that drive prescriptive actions, ranging from self-service analytics with no programming required, to comprehensive analytics for in-depth analysis of critical assets and processes.

AVEVA™ Enterprise Licensing

All System Platform products must have their licenses activated to be fully functional. The License Server can be installed on a standalone node or can be installed on a node with other components. If the License Server is combined with other components on a node, it should be installed on the node with the Galaxy Repository.

See the *AVEVA Licensing Deployment Guide* for information about deploying licenses and licensing-related recommendations for different topologies.

System Management Server

The System Management Server (SMS) is used to implement secure communications between System Platform nodes. To enable security, every System Platform node must communicate with the System Management Server. There should only be one System Management Server in your System Platform topology, otherwise, communication disruptions may occur. The System Management Server stores shared security certificates and establishes a trust relationship between machines. You can configure one additional node as a redundant SSO server, which functions as a backup for single sign-on if the System Management Server cannot be reached.

The System Management Server node is also the recommended node to use as the time master in multi-node configurations.

If you are implementing redundancy for Application Server, a System Management Server is required. Every redundant Application Server run-time node must be configured to use the System Management Server if data is being historized. Redundant nodes have an instance of HCAP running, which is used to synchronize tags and store-and-forward data between redundant AppEngines. With the release of System Platform 2023 R2, secure communication is required for HCAP, and thus, redundant nodes will not function without the SMS.

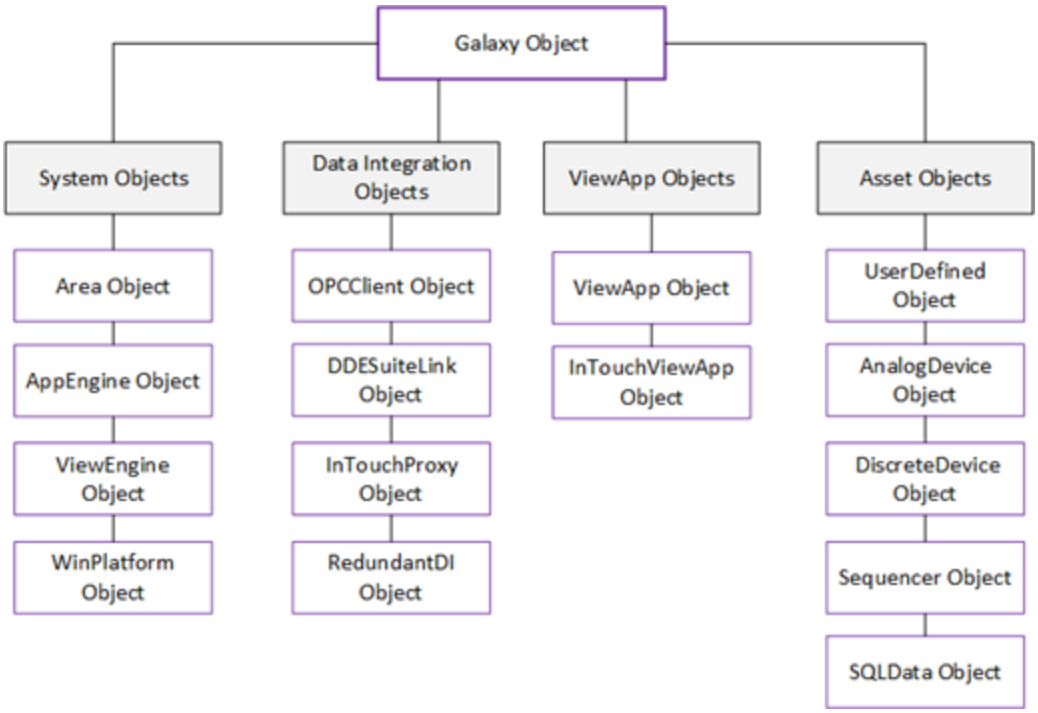
AVEVA™ System Monitor

System Monitor can monitor and manage the performance and availability of your System Platform installation, including the core software, engineered software application(s), and the related hardware and network infrastructure. It monitors key system attributes, and then generates alerts when those attributes exceed defined operational limits, and reports system performance issues, errors, and trends. Notifications can be sent to an internal support team and/or the AVEVA Knowledge and Support Center so the issue can be responded to proactively to prevent production interruption or downtime.

- If an activated license is detected (Licensed mode), System Monitor provides FULL monitoring of an unlimited number of machines.
- If no activated license is detected (Basic mode), System Monitor provides license server and license acquisition monitoring for all machines using software that requires activation-based licensing and FULL monitoring for one machine (user-selected).

Application Server terminology

The following figure shows basic object classifications and their relationships within Application Server. This document focuses on the Application/Device Integration/Engine/Platform/Area Objects level except where otherwise noted:



The following terms are used throughout this document:

- **Area Object:** A System object that represents an Area of a plant within a Galaxy. The area object acts as an alarm concentrator and is used to put other ApplicationObjects into the context of the actual physical automation layout.
- **ApplicationEngine (AppEngine) Object:** A type of System object, an AppEngine object is real-time engine that hosts and executes ApplicationObjects.
- **Asset Object:** An object that represents some element of a user application. This may include elements such as (but not limited to) a piece of equipment (for example, a thermocouple, pump, motor, valve, reactor, tank, etc.) or associated application component (for example, a function block, PID loop, sequential function chart, ladder logic program, batch phase, SPC data sheet, etc.).
- **Device Integration Object (DIObject):** An ApplicationObject that represents the communication with external devices, which all run on an AppEngine. The **PLCClient Object** is typically used to represent an external device such as a PLC or RTU, but other DIObjects, such as the DDESuiteLinkClient and InTouchProxy objects can also be used. The OIGW (OI Gateway) and Sim (Simulator) objects are derived from the PLCClient object.
- **Platform Object:** A representation of the physical hardware on which the System Platform software is running. Platform objects host engine objects (see WinPlatform, below).
- **System Object:** An object, such as a Device Integration object, platform object, or system object that is a representation of computer hardware or software with a user-defined, unique name within the galaxy. It provides a standard way to create, name, download, execute or monitor the represented component.
- **ViewApp Object:** An object that is used to host a run-time InTouch HMI or OMI application. ViewApp objects are hosted by ViewEngine objects.
- **WinPlatform:** A single computer in a galaxy consisting of Network Message Exchange, a set of basic services, the operating system and the physical hardware. This object hosts AppEngines and ViewEngines. The latter are used for visualization platforms and host ViewApp objects.

Where to find additional information

AVEVA offers a variety of support options to answer questions on AVEVA products and their implementation. For more information, refer to the AVEVA Global Customer Support (GCS) website.

https://softwaresupport.aveva.com/

For access to resources available to your organization, refer to the AVEVA Connect website.

https://connect.aveva.com/

Technical support

Before contacting Technical Support, please refer to the appropriate chapter(s) of this manual and to the User Guide and Online Help for the relevant System Platform component(s).

For local support in your language, please contact an AVEVA certified support provider in your area or country. For a list of certified support providers, see https://www.aveva.com/en/about/partners/sales-and-support/.

- Receive technical support by sending an email to your local distributor or to AVEVA Customer Support:
  - Priority email for Customer FIRST Members: custfirstsupport@aveva.com
  - Email for customers without a support agreement: wwsupport@aveva.com
- Online:
  - Contact information is available at https://www.aveva.com/en/support-and-success/support-contact/wonderware/
  - You can sign into the AVEVA Knowledge and Support Center at https://softwaresupport.aveva.com/.
- Telephone: You can reach the AVEVA Global Customer Support Hotline from 7:00 a.m. to 5:00 p.m. Pacific Time at:
  +1 949-639-8500
  US and Canada: 1-800-966-3371

If you need to contact technical support for assistance, please have the following information available:

- The type and version of the operating system you are using. For example, Windows 10 Pro 21H2.
- The exact wording of the error messages encountered.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of the attempts you made to solve the problem(s) and your results.
- Details of how to reproduce the problem.
- If known, the GCS case number assigned to your problem (if this is an ongoing problem).

When requesting technical support, please include:

- Your first and last name
- Your company name
- Your telephone number and/or email address and preferred contact method

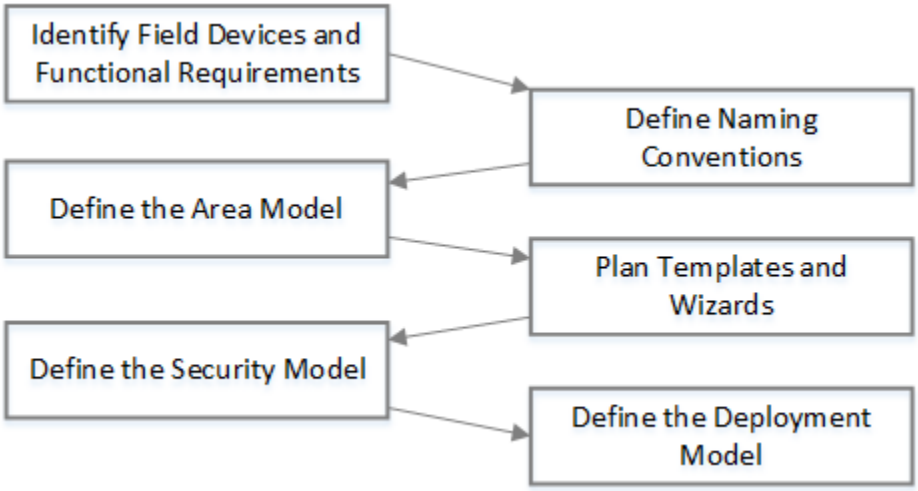
Planning

Your AVEVA™ System Platform project begins with a thorough planning phase.

This section explains the System Platform project workflow, with Application Server and its Integrated Development Environment (IDE) as the development environment. The workflow is designed to make engineering efforts more efficient by completing specific tasks in a logical and consistent (repeatable) sequence.

System Platform project workflow

The project information resulting from the planning phase becomes a roadmap (project template) when using the System Platform Integrated Development Environment (IDE) and AVEVA Integration Studio. The more detailed the project plan, the less time it takes to create and implement the application, with fewer mistakes and rework.



Each task is detailed on the following pages, and includes a checklist (summary) where applicable.

- Identify field devices and functional requirements
- Define object naming conventions
- Define the area model
- Plan templates
- Define the security model
- Define the deployment model

Identify field devices and functional requirements

The first project workflow task identifies field devices that are included in the system. Field devices include components such as valves, agitators, rakes, pumps, Proportional-Integral-Derivative (PID) controllers, totalizers, and so on. Some field devices consist of more than one base-level device. For example, a motor may be a

component of an agitator or a pump.

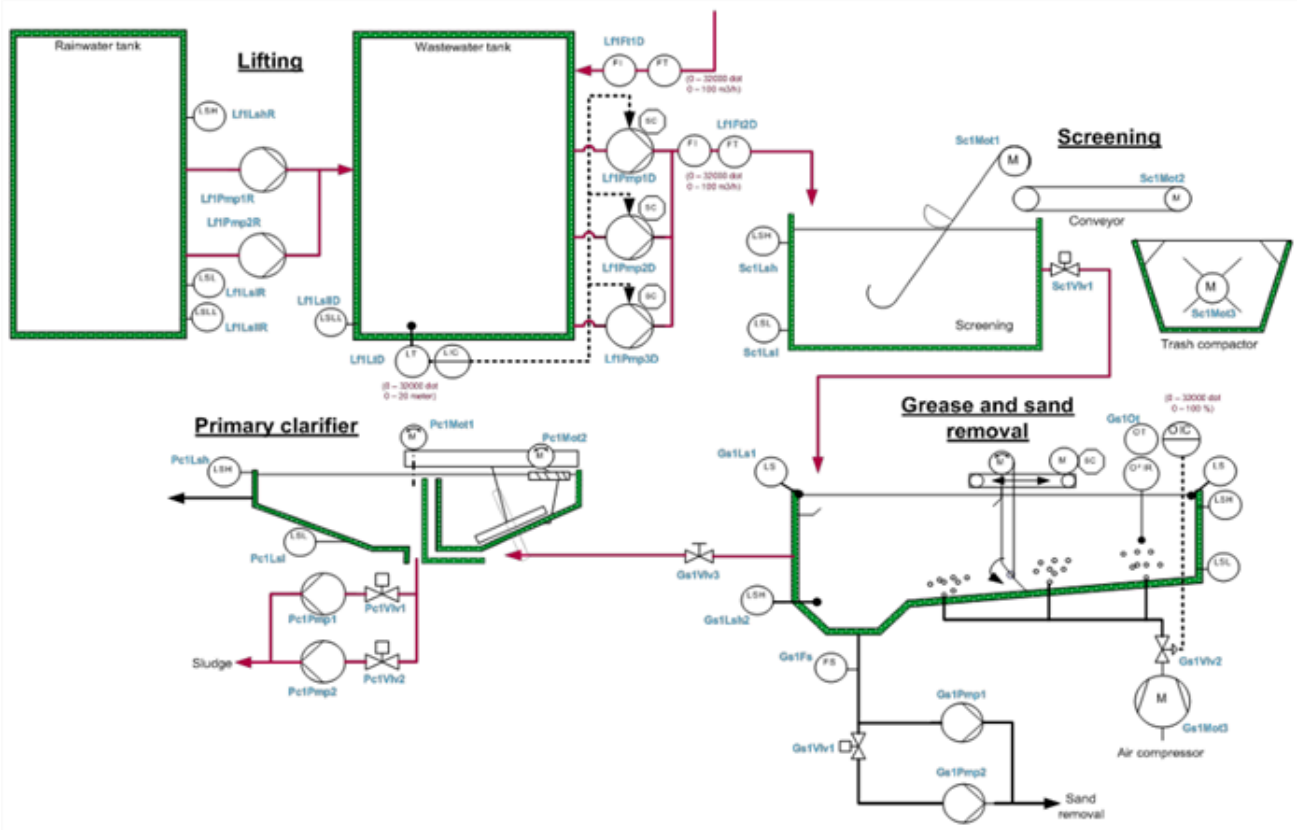
After identifying all field devices, determine the functionality for each.

Field Devices Checklist

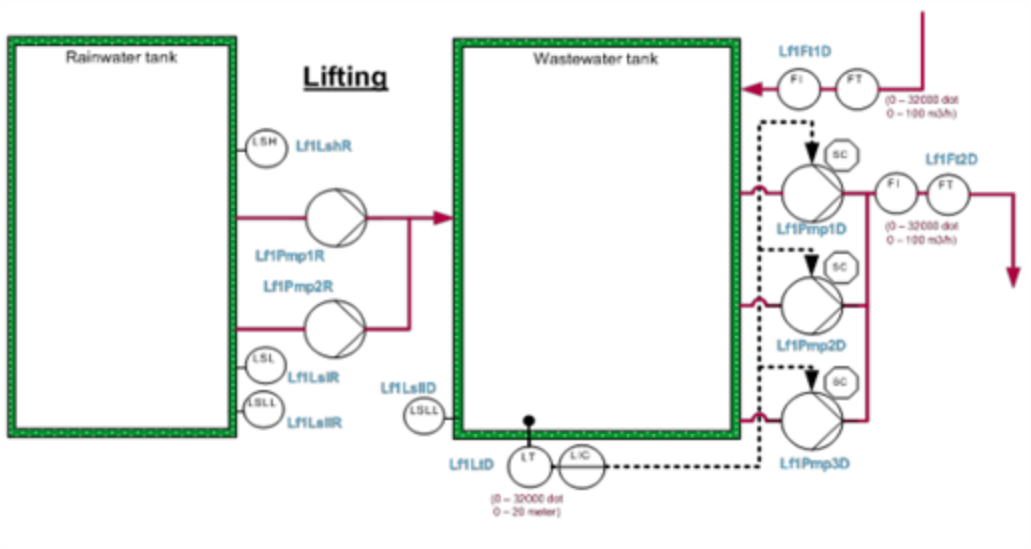
1. The Piping and Instrumentation Diagram (P&ID) below is shown as example for identifying field devices. This diagram shows all field devices and illustrates the flow between them.

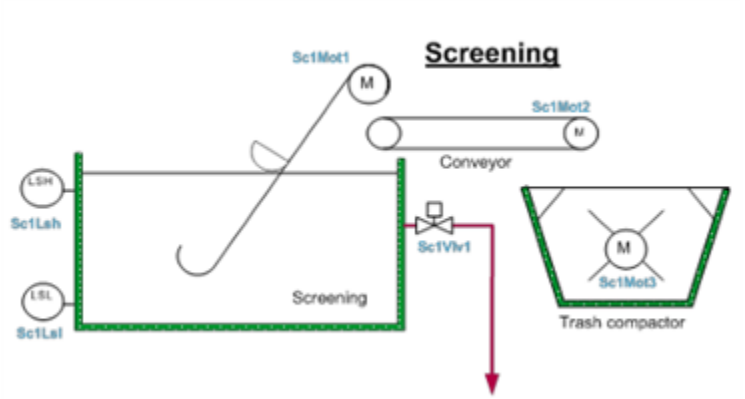
   A good P&ID can make the application planning process faster and more efficient. Verify that the P&ID is correct and up-to-date before beginning the planning process.

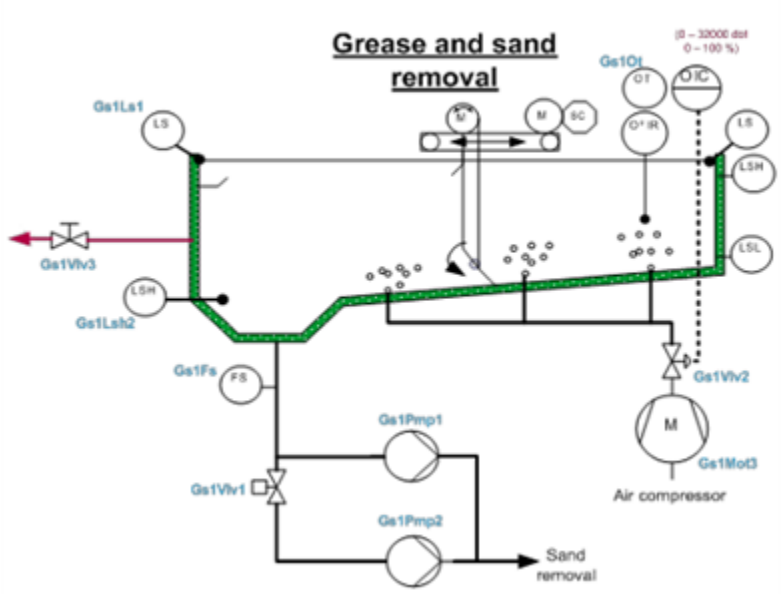
   The following P&ID example diagram shows the equipment used in the pre-treatment process for a wastewater plant that consists of four separate areas.

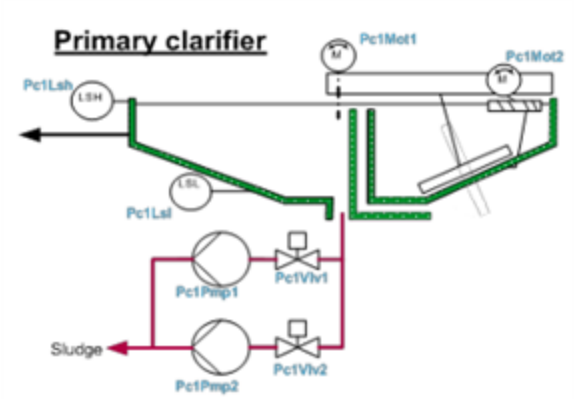


   Each of the four areas in the PI&D example are shown in greater detail below.









2. Examine each component in the P&ID and identify each basic device. For example, a valve can be a basic device. A motor, however, may consist of multiple basic devices.
3. Once a complete list is created, group the devices according to type, such as valves, pumps, and so on. Consolidate any duplicate devices into common types so that only a list of unique basic devices remains, and then document them in the project planning worksheet.

Each basic device is represented in the System Platform IDE as an AutomationObject. An instance of an object must be derived from a defined template. The number of device types in the final list will help determine how many object templates are necessary for your application. Object wizards can allow a single object templates to derive many different types of instances. For example, a wizard for valves can let you derive instances that represent 2-way, 3-way, 4-way valve, right-angle, and other valve types, from a single template. Similarly, a wizard for motors can be used to select between fixed and variable speed drives, as well to select other features.

Group multiple basic objects to create more complex objects (containment).

For more information on objects, templates, object wizards, and containment, see the *Application Server User Guide*.

Functional requirements checklist

Define the functional requirements for each unique device. The functional requirements list includes:

- Attributes: Determine the attributes needed to define the object. Attributes are parameters of the object that can access data from other objects as well as provide access to their own data to other objects (inputs and outputs).
- Scripting: What scripts will be associated with the device? Specify scripts both for self-configuring the object as well as for run-time operation.
- Historization: Are there process values associated with this device that you want to historize? How often do you want to store the values? Do you want to add change limits for historization?
- Alarms and Events: Which attributes require alarms? What values do you want to be logged as events?
- Security: Which users will access to the device? What type of access is appropriate? For example, you may grant a group of operators read-only access for a device, but allow read-write access for an administrator. You can set up different security for each attribute of a device.

All the above functional requirement areas are discussed in detail in this Deployment Guide.

Define object naming conventions

The second workflow task defines naming conventions for templates, objects, and object attributes. Naming conventions should adhere to:

- Conventions in use by the company.
- System Platform IDE naming restrictions. The following are reserved Galaxy and template names and cannot be used as object names:
  - Me
  - MyArea
  - MyContainer
  - MyEngine
  - MyHost
  - MyPlatform
  - System
- The extension "_New" cannot be used when naming an object, if the name already exists in the Galaxy without the "_New" extension. This is because automatic renaming of imported objects will append an existing object name with the "_New" extension.
- For more information on allowed names and characters, see the *Application Server User Guide*.

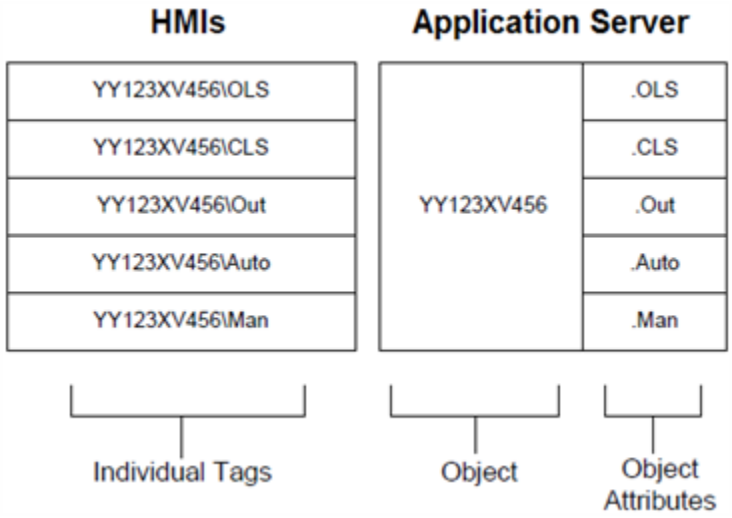
The following (instance) tagname is used as an example:

YY123XV456

It has the following attributes:

OLS, CLS, Out, Auto, Man

The following figure shows the differences between the HMI and Application Server naming conventions:



Create references using the following IDE naming convention:

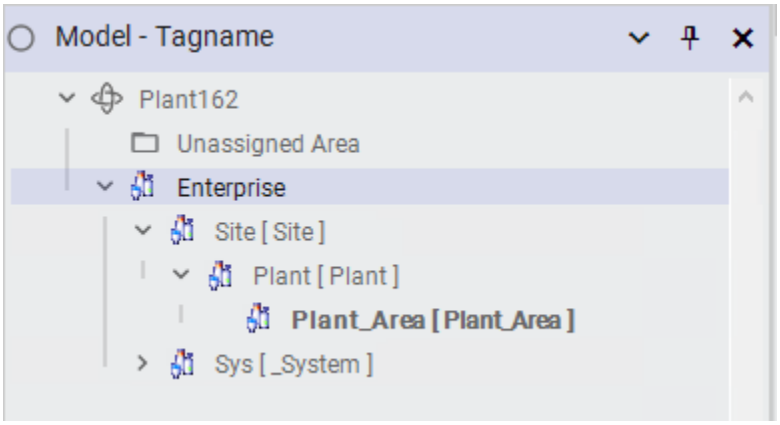
<objectname>.<attributename>

For example:

YY123XV456.OLS

Define the area model

The third workflow task defines the area model. The Model View in the IDE shows the Galaxy organized by area object.



An area represents a logical grouping of objects within the physical plant layout. For example, Receiving Area, Process Area, Packaging Area and Dispatch Areas are all logical representations of a physical plant area. In the previous example shown under Identify field devices and functional requirements, the areas are Lifting, Screening, Grease and Sand Removal, and Primary Clarifier. Alarms originating from an object in an area are aggregated to the area object.

Define and document all necessary plant areas to ensure each object is assigned to its relevant area.

Note: The default installation creates an Unassigned area. All object instances are assigned to this area unless a different area is designated as the default.

Area model checklist

1. Using the Model view of the IDE, create all areas first. An object instance can then be easily assigned to the correct area; otherwise, you will have to move them out of the unassigned area later.
2. Create a System Area. Assign instances of WinPlatform and AppEngine objects to the System Area. WinPlatform and AppEngine objects are used to support communications for the application, and are not necessarily relevant to a plant-related area.

   Area and application objects (such as the UserDefined object) are independent of the physical environment that they represent. Therefore, these objects can be moved easily between a development galaxy and a production galaxy.

   Platform and engine system objects, and device integration (DI) client objects, which link to PLCs or RTUs, link directly with the physical plant/environment. These environment-dependent system and DI client objects are not transportable between different physical environments.
3. Areas can be nested. Sub-areas can be assigned to a different AppEngine on a different platform. Note that while area objects host application objects, the objects assigned to one area object cannot scan across multiple AppEngines.
4. It may be practical to create an object instance for one area at a time. If using this development approach, mark the area as Default, so that each object instance is automatically assigned to the Default area. Before

creating instances in another area, change the default setting to the new area.

5. Equate various areas to Alarm Groups. Alarm displays can easily be filtered at the area level.

Note that areas and the application objects that they host are not environment-dependent, and thus can easily be moved between a development galaxy and a production galaxy. System objects such as platforms, engines, and Device Integration client objects are, in contrast, linked to the physical infrastructure, and cannot be easily moved because they have been configured to communicate with a specific machine name or IP address, or in the case of DI client objects, with a particular PLC. Therefore, we recommend that you create an infrastructure specific to the environment in which you are operating. A development environment should have its own infrastructure. Similarly, if you have a separate test environment, it should also have its own infrastructure. Your production environment should have its own infrastructure as well. You can export the area objects and the application objects that they host and move the objects between environments as needed.

For more information on areas, see the *Application Server User Guide*.

Plan templates

The fourth workflow task determines the necessary object "shape" templates.

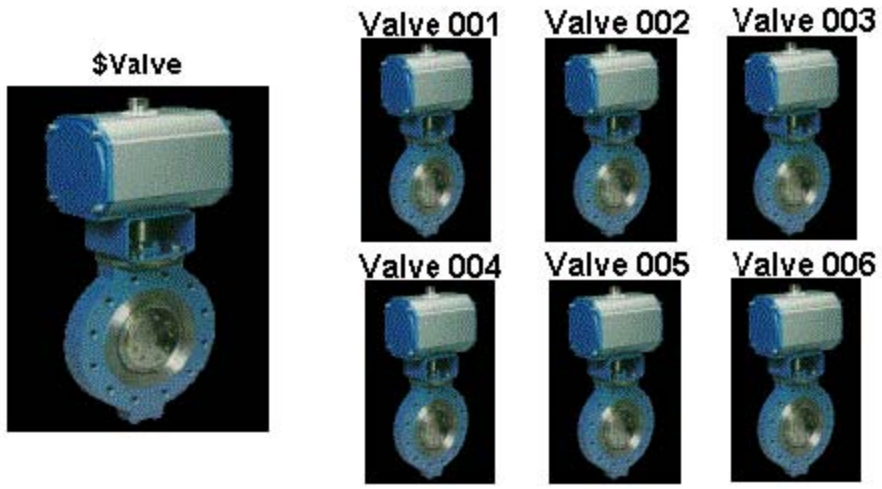
A shape template is an object that contains common configuration parameters for derived objects (objects used multiple times within a project). The shape template can now be reused multiple times, either as another template or an object instance. The shape template is derived from a $Master_template object and is designed to represent baseline or "generic" physical objects, and to encapsulate specific, baseline functionality within the production environment. The derived objects are "digital twins" of the physical objects they represent, for example, a valve. In most cases, you can use the $Master_UserDefined object to create the digital twin.

For example, multiple instances of a certain valve type may exist within the production environment. Create a shape valve template that includes the required valve properties.

- Adding an object wizard to the shape template lets you limit the number of required templates through the ability to address a wide range of requirements within a single template. Object wizards allow project developers to select from a list of choices and options contained in a single template to create a variety of derived objects.

If changes are necessary, they are propagated to the derived object instances. Use the drag-and-drop operation within the IDE to create object instances.

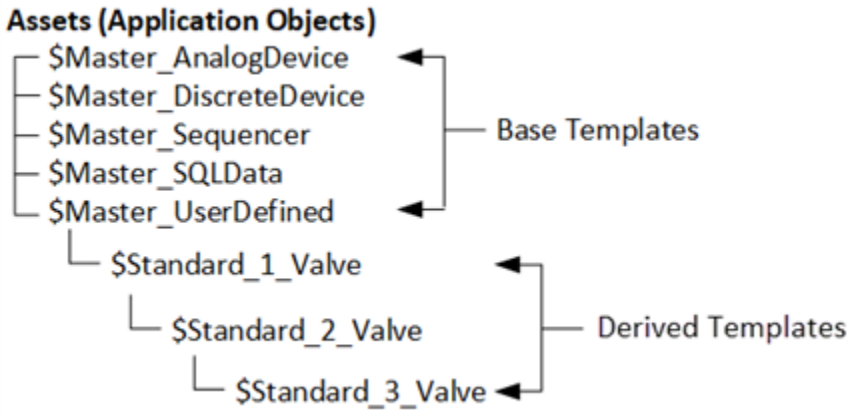
The following figure shows multiple instances (Valve001, -002, etc.) derived from a single object template (**$Valve**):



Application Server is shipped with a number of pre-defined master templates to help you create your application quickly and easily.

The master (base) templates provided with Application Server are summarized in the Base Template Functional Summary. Determine if any of their functionally match the requirements of the devices on your list.

If the base templates do not satisfy the design requirements, create (derive) new shape templates or object instances from the $UserDefined object base template.



A child template derived from a base (parent) template can be highly customized. Add attributes, scripts, and features such as alarm, history, and I/O extensions to the derived templates as needed.

Template derivation

Since templates can be derived from other templates, and child templates can inherit properties of the parents, establish a template hierarchy that defines what is needed before creating other object templates or instances. Always begin with the most basic template for a type of object, then derive more complicated objects. Incorporating object wizards into your template design can help to minimize how many layers of derivation are needed. A basic template hierarchy might consist of the following:

- First layer: corporate template definitions
- Second layer: templates by geographic or areas of operational similarities
- Third layer: plant or local templates

If applicable, lock object attributes at the template level, so that changes cannot be made to those same attributes for any derived objects.

A production facility typically uses many different device models from different manufacturers. For example, a process production environment has a number of flow meters in a facility. A base flow meter template would contain those fields, settings, and so on, that are common to all models used within the facility.

Derive a new template from the base flow meter template for each manufacturer, or alternatively, include options within a single template for each manufacturer (preferred method). The derived template for the specific manufacturer includes additional attributes specific to the manufacturer. A new set of templates would then be derived from the manufacturer-specific template to define specific models of flow meters. Finally, instances would be created from the model-specific template.

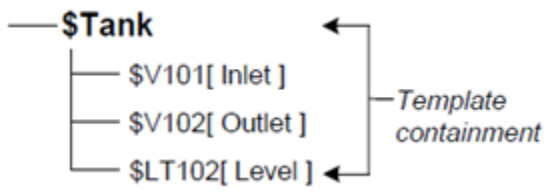
Note: For detailed examples of template derivation, see Templates. For more information on templates, template derivation, and object wizards, refer to the *Application Server User Guide*.

Template containment

In addition to object wizards, template containment allows more complex structures to be modeled as a single object. For example, a new template called "$Tank" is derived from the $UserDefined base or shape template. Use the template to derive other asset objects that represent components of the tank, such as pumps, valves, and levels.

For example, derive two instances from the $UserDefined template called "Inlet" and "Outlet," and configure them as valves. Derive another instance from the $UserDefined template called "Level," and contain all three within the $Tank template.

The containment hierarchy is as follows:



Note: Deeply nested template/container structures can slow the check-in of changes in IDE development and propagation. You can leverage object wizards to limit the template derivation depth.

The preferred method for defining object properties is to create a high-level template and object wizard to define enterprise-wide standards.

- Use template containment to create a higher-level object with lower-level objects. This practice works best when the lower-level object also has many components to it and may contain even lower-level objects.

  However, when adding the lowest-level object to a template, it is possible to use either template containment or attributes. Both allow for an external I/O point link and historization.

  If required attributes (such as complex alarms, setpoints, or other features) are readily available in a template, use template containment. If the lower-level object is very basic, use an attribute. It is always valid to use a contained object, even if it is a simple property.
- Always use a contained object for I/O points and use a user-defined attribute for memory or calculated values. How this is accomplished is up to the application designer, and should be decided in advance for project consistency.

Object wizard best practices

An object wizard can be added to any derived template. However, to maximize the benefits of using object wizards, add your object wizard to the template as close to the top level of the derivation hierarchy as possible. To build an object wizard you must:

- Try to minimize the depth of template derivation by using object wizards. You can provide options that allow subsequent derived objects, either templates or instances, to be easily configured from the choices provided by the wizard. For example, the top level template might encompass corporate standards, while the second level, derived from the corporate standards, can be modified to include specific, local plant requirements.
- Add and configure attributes, symbols, links to external content, and scripts before you build the object wizard.
- An object wizard requires at least one choice group or option.
- Configure choices and options by associating attributes, graphics/content, and scripts. You should limit the number of items associated with each choice or option to avoid configuration mismatches.
- When configuring choices and options, keep the workflow as simple and direct as possible. Remember that users will be working sequentially through the object wizard. If a feature is enabled at one level of the wizard hierarchy, it cannot be disabled at a subsequent level.

  For example, if a symbol has Custom Property "X" enabled and is associated with a choice, do not associate the same symbol with "X" disabled to an option further down in the hierarchy. Instead, leave "X" disabled and override the setting when deriving an instance.

See the *Application Server User Guide* for additional information about configuring templates and object wizards.

Object template checklist

1. Document which existing templates and object wizards can be used for which objects, and which templates and object wizards need to be created from scratch. For information on a particular object template, see the help file for that object.
2. Design your containment model at the template level before generating large object instance quantities.
3. Create instances from the top "container" template of a hierarchical set of contained templates.

   Such template hierarchies should be tested with one or two instances before proceeding to the generation of numerous instances. Any change by insertion or removal of a contained template in the hierarchy does not result in propagation of new insertions or removals in the instance hierarchies.

   For instances of the containment hierarchy, insertions and removals must be managed individually. However, changes within already included contained templates can be automatically propagated by locking.

   Note: For detailed information about using templates, see Templates.

Define the security model

The fifth workflow task defines the security model.

The following basic concepts are reviewed in order to reinforce understanding of the System Platform security model:

- **Users:** A user is each individual person that will be using the system (for example, John Smith and Polly Perez).
- **Roles:** Roles define groups of users within the security system. Roles usually reflect the type of work performed by different groups within the production environment. For example, Operators and Technicians.
- **Permissions:** Permissions determine what users are allowed to do within the system. For example: Operate, Tune, and Configure.
- **Security Groups:** A security group is a group of objects with the same security characteristics. The purpose of a security group is to simplify object security management by avoiding the need to assign security permission for each role to each individual object.

  Security groups let you define groups of objects, and determine who within your organization is allowed to make changes to the different object groups. Security groups typically map to areas within your System Platform installation, but more than one security group for each area may be needed within a single area. For example, you may need to assign specific operators to the "Line_1" security group, but assign a different set of operators to the "Line_2" security group.

Security model checklist

1. Configure a System Management Server (SMS) node. Set up only one SMS node in your topology. The SMS node secures the communication between all System Platform nodes.
2. If you are using an external authentication provider for Single Sign-On (SSO), such as AVEVA Connect or Azure Active Directory, configure the SMS to utilize the authentication provider as described in the *System Platform Installation Guide*. Note that the Certificate Manager authenticates users, not the SMS.

   You can also configure other deployed nodes as Redundant SSO server nodes. More than one RSSO node can be configured, but additional nodes can incur more network overhead..
3. Define Users, Roles, Permissions, and Security Groups necessary to implement security for the production environment. Select Users and Roles previously defined within the Operating System Security model, or define them within the IDE.

   Using Operating System Users and Roles facilitates object deployment and makes future maintenance easier.
4. Determine security settings for writable attributes of objects. Security permissions reflect the rights of different groups of users to change the attribute value. The available security options for writeable attributes are:
   - Read Only
   - Operate
   - Tune
   - Secured Write
   - Verified Write
   - Configure

   Review the functional worksheet that lists the objects (and their attributes).

   An Operate permission requirement does not mean that the user must be an Operator. A QA inspector might have Operate permissions to change a value on an object that that collects QA data, while an operator on the same production line would not have this permission.

To set up security using the IDE

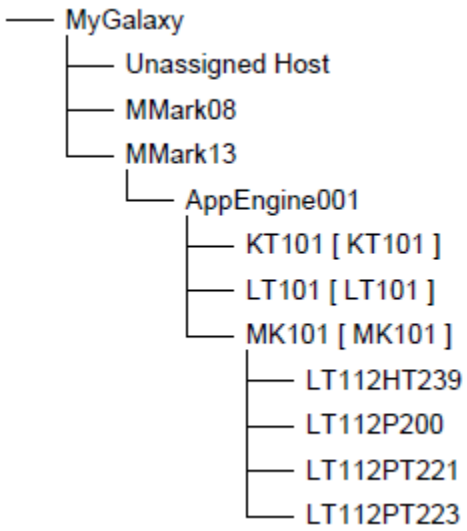
1. Set the Galaxy security mode:
   - Galaxy
   - OS based (user or group)
   - Authentication provider
2. Create security groups.
3. Create roles and assign them to security groups.
4. Select permissions and grant them to roles.
5. Define users and assign them to roles.
6. Configure attribute security at the Template level.

See Security for more information. For details about security configuration and options, see the *Application Server User Guide* and the *System Platform Installation Guide*. In addition to procedures for implementing securing, the user guide also contains a reference section that lists general security guidelines. Additional information about security across AVEVA products is available in the *AVEVA Cybersecurity Deployment Guide*.

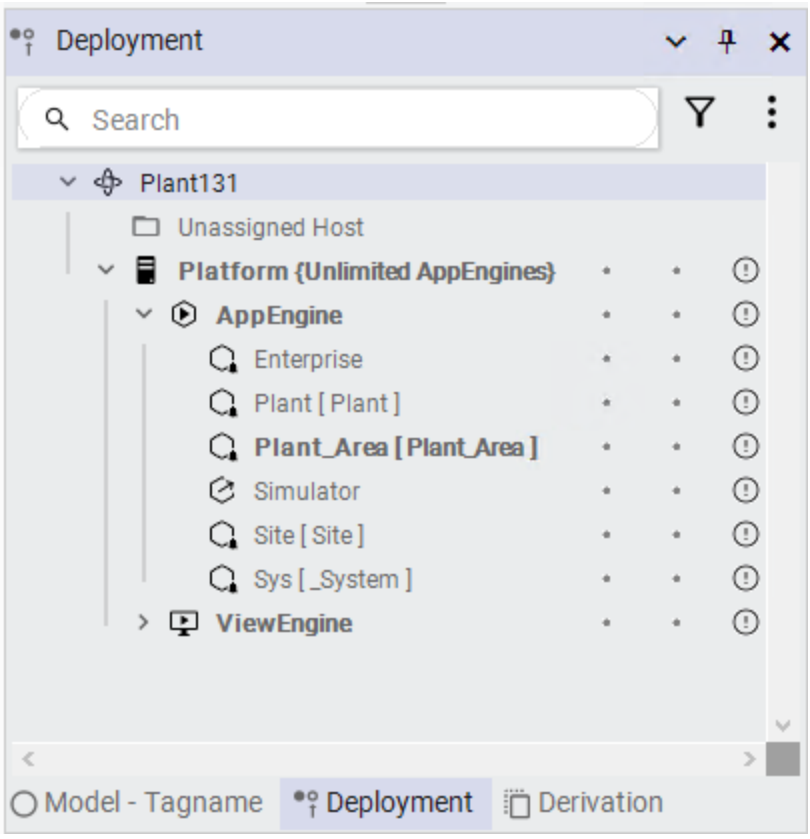
Define the deployment model

The last workflow task defines the deployment model that specifies where objects are deployed. In other words, the deployment model defines which nodes will host the various objects that make up the Galaxy.

Each computer in the System Platform network must have a WinPlatform object, AppEngine object, and Area object deployed to it. For example, KT101, LT 101 and MK101 are all areas in the following figure:



The Deployment View in the IDE shows the Galaxy organized by Platform (WinPlatform object). Each engine object assigned to the Platform is shown next. Area objects assigned to each engine are next in the hierarchy.



The objects deployed on particular platforms and engines define the objects' "load" on the platform. The load is based on the number of I/O points, the number of attributes, derivation depth, etc. More complex objects consume more resources. No more than two AppEngines should be assigned to a single logical processor. In a redundant system, these are typically the primary and backup AppEngines. See System sizing guidelines for more information.

Note: For object types and target deployment node recommendations (such as DIObjects), see Assessing System Size and Performance.

After deployment, you can use the Object Viewer to check communications between nodes and determine if the system is running optimally. For example, a node may be executing more objects than it can easily handle, and it will be necessary to deploy one or more objects to another computer.

Note: For more information on deployment, see the *Application Server User Guide.*

Document the planning results

Determine how to document the project planning results before beginning the planning phase.

Use a spreadsheet application, such as Microsoft Excel, to document the list of devices, the functionality of each device, process areas to which the devices belong, etc. For example:

Field Device	**Valves**										
Attribute Name	Description	Data Type	In	Out	Default Value	Min	Max	Default Security	Alarm	Hist	Event

Inputs (2)											
OpenLimit Switch	Indicates Opened State	Boolean	Y		False			View Only	N	N	Y
ClosedLimit Switch	Indicates Closed State	Boolean	Y		True			View Only	N	N	N
Outputs (1)											
OpenCommand	Commands Valve to Open/Close	Boolean		Y	False			Operate	N	N	N
Scripting											
Control Mode	Switches Valve Manual/Auto Mode	Boolean		Y	Auto			Tune	Y	N	N

Selecting System Platform Components

The following lists show the System Platform components available for installation.

AVEVA Application Server and OMI Components

- System Platform IDE
- Application Server Galaxy Repository
- Application Server Platform (runtime)
- Operations Management Interface (OMI) ViewApp (runtime)
- OMI Web Server

AVEVA OMI Apps

- OMI ContentPresenter App
- OMI Map App
- OMI Standard Apps

AVEVA OMI Widgets

- Carousel Widget
- DataGrid Widget

- QRCode Scanner Widget
- Teamwork Widget
- Web Browser Widget

AVEVA InTouch HMI and InTouch Access Anywhere Components

- InTouch WindowMaker
- InTouch WindowViewer
- InTouch Web Server (Web Client)
- InTouch Workspaces
- InTouch Access Anywhere Server
- InTouch Access Anywhere Secure Gateway

AVEVA Historian

- Historian Server (Desktop/Server)
- Historian Client Web (Insight Local)
- Historian Client Desktop
- Insight Publisher

AVEVA Common Services

- Communication Driver Pack
- Common Services Framework
- AVEVA Enterprise Licensing
- System Monitor

Supported operating systems

Important! System Platform is supported on 64-bit operating systems only.

System Platform 2023 R2 is supported on the following Windows client and server operating systems (64-bit only). This list was compiled at the release of System Platform 2023 R2. Check the AVEVA GCS web site for the latest information. Apply operating system patches and updates prior to installing or upgrading System Platform.

Note: The same operating system support applies to InTouch Access Anywhere.

Note that when Windows updates run in the background, there is the possibility that different software processes can be adversely affected. Therefore, it is important to schedule the updates to run only during planned shutdown periods.

Configuring Automatic Windows Updates

If Windows is configured to update automatically, these automatic updates, when running in the background, can disrupt System Platform components, including Application Server and OMI during installation/upgrade,

configuration and run-time operations. These updates may cause the IDE, GR, OMI Web Client and related components, and other services to shutdown unexpectedly. Therefore, we recommend that you disable automatic Windows updates, or otherwise ensure the updates will be installed only when System Platform applications are not being actively used.

Client Operating Systems

Semi-Annual Channel Releases:

- Windows 10 21H2 Pro, Enterprise, and IoT Enterprise [Microsoft support ends 11 Jun 2024]
- Windows 10 22H2 Pro, Enterprise, and IoT Enterprise [Microsoft support ends 14 Oct 2025]
- Windows 11 21H2 Pro, Enterprise, and IoT Enterprise [Microsoft support ends 8 Oct 2024]
- Windows 11 22H2 Pro, Enterprise, and IoT Enterprise [Microsoft support ends 14 Oct 2025]
- Windows 12 24?? Pro, Enterprise, and IoT Enterprise [End of support date not yet announced]

Long Term Service Channel Releases:

- Windows 10 Enterprise, IoT Enterprise 2015 LTSB (1507) [Microsoft support ends 14 Oct 2025]
- Windows 10 Enterprise, IoT Enterprise 2016 LTSB (1607) [Microsoft support ends 13 Oct 2026]
- Windows 10 Enterprise, IoT Enterprise 2019 LTSC (1809) [Microsoft support ends 09 Jan 2029]
- Windows 10 Enterprise 2021 LTSC (21H2) [Microsoft support ends 12 Jan 2027]
- Windows 10 IoT Enterprise (Only) 2021 LTSC (21H2) [Microsoft support ends 13 Jan 2032]
- Windows 11 Enterprise, IoT Enterprise 2024 LTSC [End of support not yet announced]

Server Operating Systems

Long Term Service Channel Releases:

- Windows Server 2016 LTSC Standard and Datacenter [Microsoft support ends 12 Jan 2027]
- Windows Server 2019 LTSC (Datacenter, Essentials, Standard) [Microsoft support ends 9 Jan 2029]
- Windows Server IoT 2016 LTSC [Microsoft support ends 12 Jan 2027]
- Windows Server IoT 2019 LTSC [Microsoft support ends 9 Jan 2029]
- Windows Server 2022 LTSC Standard and Datacenter [Microsoft support ends 14 Oct 2031]
- Windows Server 2025 LTSC Standard and Datacenter [End of service to be announced]

Note: System Platform is not supported on any version of Windows prior to Windows 10, or on Windows Server versions prior to 2016.

Supported InTouch Access Anywhere Clients

InTouch Access Anywhere has been tested in the following HTML5-capable browsers:

- Google Chrome version 98.0.4758.80 and newer
- Firefox version 96.03 ESR and newer

- Microsoft Edge Non-Chromium
- Microsoft Edge Chromium 97.0.1072.76 and newer
- Safari version 15.2 and newer (Mac and iOS only) (Not Windows)
- Opera version 83.0.4254.16 and newer

System sizing guidelines

The following table provides guidelines for System Platform hardware configurations, based on application size. These guidelines are subject to the limitations of your Windows operating system, and if applicable, to the SQL Server edition (Express, Standard, or Enterprise). See the Technology Matrix on the AVEVA Knowledge and Support Center website (https://softwaresupport.aveva.com/) for supported versions of Windows operating systems and SQL Server.

- An HD display is recommended for engineering tools such as the System Platform IDE.
- A 64-bit operating system is required, regardless of system size.
- A Windows Server operating system is required for large installations.
- SQL Server Express is supported only for small systems, that is, installations with less than 25,000 I/O per node.
- Pagefile.sys, the Windows paging file (also called the swap file or virtual memory file), must be enabled. The Windows default setting is enabled.

To access the relevant information from the Technology Matrix, go to the Knowledge and Support Center website, select the Technology Matrix icon, and then enter the name of the System Platform product (for example, Application Server or Historian), or enter the Windows or SQL Server version you wish to use (for example, SQL Server 2022 Standard x64).

Definitions

In the table below, hardware guidelines for different types of System Platform are listed. Definitions for the terminology used in the table are:

Level (Minimum and Recommended)

Minimum level describes the baseline hardware configuration that will provide at least minimally acceptable performance for the role. Recommended level describes an expanded hardware set that provides improved performance.

IDE Node

IDE nodes are engineering workstations. These are used for creating, editing, and deploying objects.

Application Object Server Node

Application Object Server nodes, also called AOS nodes, are remote run-time nodes. AppEngines, and the objects assigned to them, are deployed from the Galaxy Repository to AOS nodes, where the AppEngines run on the AOS WinPlatform object. Each active AppEngine requires one logical processor and runs as a 32-bit process. We recommend that each AppEngine in a redundant pair is also assigned one logical processor (one for active and one for standby). If redundant AppEngines consume less than 40% of CPU and memory resources, you can allocate one active and one standby AppEngine to a single logical processor. However, if the AppEngines exceed 40% of the computing resources, you run the risk of overleveraging the node (i.e., CPU and/or memory usage

hits 100%) when a failover occurs.

AOS resource allocation

Areas are assigned to AppEngines, and objects are assigned to areas. The total number of objects that can be assigned to a single AppEngine is very variable, and depends on the complexity of the objects, including the number of attributes, attribute datatypes, if the object is running scripts, script complexity, and if the object contains graphics (owned graphics will take more memory than linked graphics). In most system configurations, an AppEngine can host anywhere from 5,000 to 50,000 objects, but even from this broad range does not cover non-typical configurations, depending on the factors just mentioned (attributes, datatypes, owned graphics, etc.). For example, a single object attribute of datatype BigString can, conceivably, consume 2 GB of memory. All of the areas and objects under them that are assigned to an AppEngine cannot require more than 2 to 3 GB of total memory. Do not forget to take into account CPU, memory, and disk requirements for running Windows when provisioning the AOS nodes. Device Integration objects also run on the AppEngine and consume resources.

AOS deployment performance

When you deploy a galaxy, the GR node is deployed first. After the GR, remote AOS platforms are deployed. Deployment of AppEngines to the AOS platforms is done in parallel. The AppEngines, along with the areas and the objects they contain are deployed serially. Thus, deployment is much quicker if you use multiple AOS nodes, each hosting fewer AppEngines, rather than using a single AOS node to host, for example, 30 active AppEngines. The improvement in deployment performance that you gain by using multiple nodes is nearly linear. Using two AOS nodes instead of one can reduce deployment time by half, using four AOS nodes reduces the time to a quarter, eight nodes reduces the time to an eighth. Once the AppEngines are deployed, deployment of areas and their contained objects to each AppEngine occurs serially. Thus, deployment is much more efficient if you use multiple, AOS nodes that are provisioned with fewer hardware resources, rather than using a few, highly-resourced nodes.

Galaxy Repository Node

Galaxy Repository nodes, also called GR nodes, host the galaxy database. The GR is tightly coupled to a Microsoft SQL Server database.

Historian Server Node

Historian Server nodes host the AVEVA Historian. The Historian is tightly coupled to a Microsoft SQL Server database.

Thin Client

Thin clients include include smart phones and tablets. In the context of System Platform, thin clients are platforms for web browsers and remote desktop sessions (for example, InTouch Access Anywhere clients).

Client

In the context of System Platform, clients are computers that can be used to develop and/or view and interact with applications. Remote IDE workstations, as well as for run-time applications like WindowViewer, AVEVA OMI ViewApps, and Historian Insight can be System Platform clients.

The following guidelines are provided for reference only. The system configuration required for your application will depend on multiple factors, including but not limited to the size and complexity of the application, and the features and components used.

Application	Level	Logical Processors [1]	RAM [3]	Free Disk Space [2, 3]	Network Speed
Application Object Server (AOS) Nodes [5, 6]					
Small AOS Node 1 - 6 AppEngines	Minimum	4	4 GB	100 GB	100 Mbps
	Recommended	8	8 GB	200 GB	1 Gbps

Medium AOS Node 6 - 15 AppEngines	Minimum	8	8 GB	200 GB	1 Gbps
	Recommended	16	16 GB	500 GB	1 Gbps
Large AOS Node 15 - 30 AppEngines	Minimum	16	16 GB	500 GB	1 Gbps
	Recommended	32	24 GB	1 TB	Dual 1 Gbps
Galaxy Repository Nodes					
Small Galaxy 1 - 50,000 I/O per Node	Minimum	4	2 GB	100 GB	100 Mbps
	Recommended	8	4 GB	200 GB	1 Gbps
Medium Galaxy 50,000 - 200,000 I/O per Node	Minimum	8	8 GB	200 GB	1 Gbps
	Recommended	16	12 GB	500 GB	1 Gbps
Large Galaxy > 200,000 I/O per Node	Minimum	16	16 GB	500 GB	1 Gbps
	Recommended	32	24 GB	1 TB	Dual 1 Gbps
Historian Server Nodes					
Small Historian 1 - 50,000 Historized Tags per Node	Minimum	4	2 GB	100 GB	100 Mbps
	Recommended	8	4 GB	200 GB	1 Gbps
Medium Historian 50,000 - 200,000 Historized Tags per Node	Minimum	8	8 GB	200 GB	1 Gbps
	Recommended	16	12 GB	500 GB	1 Gbps
Large Historian > 200,000 Historized Tags per Node	Minimum	16	16 GB	500 GB	1 Gbps
	Recommended	32	24 GB	1 TB	Dual 1 Gbps
Thin Client Node					
RDP clients, InTouch Access Anywhere browsers, mobile devices	Minimum	2	512 MB	N/A	100 Mbps
	Recommended	4	2 GB	N/A	100 Mbps
Client Node					
WindowViewer, ViewApp, Historian Client, Remote IDE	Minimum	4	1 GB	100 GB	100 Mbps
	Recommended	8	4 GB	200 GB	1 Gbps
Remote Desktop Server Nodes					
Basic RDS, InTouch Access Anywhere Server Supports up to 15 concurrent remote sessions	Minimum	8	8 GB	200 GB	1 Gbps
	Recommended	16	12 GB	500 GB	1 Gbps

Large RDS, InTouch Access Anywhere Server Supports up to 30 concurrent remote sessions	Minimum	16	16 GB	500 GB	1 Gbps
	Recommended	32	24 GB	1 TB	Dual 1 Gbps
All-In-One Node [4] (all products on a single node)					
Small Application 1,000 I/O max	Minimum	8	8 GB	200 GB	100 Mbps
	Recommended	12	12 GB	500 GB	1 Gbps
Medium Application 20,000 I/O max	Minimum	12	16 GB	500 GB	1 Gpbs
	Recommended	16	32 GB	1 TB	1 Gbps
Large Application [7] 100,000 I/O max	Minimum	20	32 GB	2 TB	1Gbps
	Recommended	24	64 GB	4 TB	1 Gbps

1) To calculate the number of logical processors: multiply the number of physical cores by the number of threads each core can run. A four core CPU that runs two threads per core provides eight logical processors. The terms "Hyper-Threading and "simultaneous multithreading" (SMT) are also used to describe logical processors.
2) SSD drives are highly recommended.
3) In redundant environments, increase CPU and RAM to maintain a maximum of 40% typical resource utilization.
4) For optimal performance of all-in-one nodes, a high clock speed (>2.8 GHz) is recommended.
5) You can deploy two AppEngines (one active and one standby) per logical processor provided that the CPU and memory load is less than 40% for each AppEngine.
6) Using multiple Application Object Server platform nodes reduces deployment time.
7) For large applications on all-in-one nodes, dual XEON processors are recommended.

Supported and recommended node hardware types

Product	Server Node	Thin Client- Server Node	Client Node	Thin Client	All-In-One
Application Server					
Galaxy Repository	Preferred	Supported	Supported	No	Supported
ApplicationObject Server (AOS)	Preferred	Supported	Supported	No	Supported
System Platform IDE	Preferred	Supported	Supported	RDP	Supported
AVEVA OMI ViewApp Runtime	Supported	Supported	Preferred	ITAA/RDP	Supported
InTouch HMI Standalone					
InTouch WindowMaker	Supported	Supported	Preferred	RDP	Supported
InTouch WindowViewer /	Supported	Supported	Preferred	ITAA/RDP	Supported

InTouch ViewApp (runtime only)					
InTouch for System Platform					
InTouch WindowMaker (with Managed Apps)	Preferred	Supported	Supported	RDP	Supported
InTouch WindowViewer / InTouch ViewApp (runtime only)	Supported	Supported	Preferred	ITAA/RDP	Supported
InTouch Access Anywhere					
InTouch Access Anywhere Server	Supported	Preferred	No	No	Supported
InTouch Access Anywhere Client (HTML5 Browser)	Browser	Browser	Browser	Browser	Browser
InTouch Access Anywhere Secure Gateway	Supported	No	No	No	No
Historian					
Historian Server	Preferred	Supported	Supported	No	Supported
AVEVA™ Insight	Browser	Browser	Browser	Browser	Browser
Historian Client	Supported	Supported	Preferred	RDP	Supported
Support Components					
OI Gateway	Preferred	Supported	Supported	No	Supported
AVEVA Enterprise License Server	Preferred	Supported	Supported	No	Supported
AVEVA Enterprise License Manager	Preferred	Supported	Supported	No	Supported
AVEVA Enterprise License Manager Client	Browser	Browser	Browser	Browser	Browser

Windows network configuration

If you are installing System Platform products on more than one node, we recommend that you use domain based networking. Domain based (client-server) networks provide better user account security and management than workgroup based (peer to peer) networks.

System Platform does not support mixed Windows workgroup/domain environments. While workgroups are supported, you cannot use workgroup nodes within a domain environment.

Note: Do not install the Galaxy Repository on a computer that is used as a domain controller or as an Active Directory server.

Operations that rely on inter-node communications may not function correctly in a workgroup based Application Server installation. Examples of this type operation include connecting to a remote IDE, or viewing the status of a remote platform.

If you must use workgroup based networking, you can avoid communications issues by enabling "everyone permissions" for anonymous users. To enable these permissions, open the Local Security Policy app and set network access permissions for anonymous users as follows:

```
Local Security Policy > Local Policies > Security Options > Network Access: Let everyone permissions apply to anonymous.
```

Or, you can enter the following command from an administrator command prompt:

```
reg add HKLM\System\CurrentControlSet\Control\Lsa /v EveryoneIncludesAnonymous /t REG_DWORD /d 1
```

Ports Used by System Platform Products

The following tables list the ports used by System Platform products.

Note: Firewall settings for all destination ports must allow INBOUND connections.

Application Server

Port	Can be configured	Protocol	Subsystem	Purpose
135	No	TCP	Bootstrap	DCOM and RPC
139 445	No	TCP	Bootstrap	DCOM and NetBios
443	No	TCP (HTTPS)	AVEVA.AppServer.BootstrapProxy.exe	AVEVA.AppServer.BootstrapProxy.exe
808	Yes	TCP	Multi-Galaxy	Galaxy Pairing ASBAuthentication Service ASBGRBrowsing Service IOM BLS Service ASMBMxDataProvider Service
5026	Yes	TCP	NMXSVC	NMXSVC

Port	Can be configured	Protocol	Subsystem	Purpose
8090	Yes	TCP	aaGR	aaGR
30000 30001	Yes	TCP/UDP TCP	Bootstrap, Redundancy PMC	Local redundancy messaging (WinPlatform)
32568	Yes	TCP	aaEngine.exe	aaEngine.exe
48031	Yes	TCP	Platform Common Services	OPC UA Server
49152 – 65535	No	TCP	aaGlobalDataCacheMonitorSvr aaGR aaIDE aaObjectViewer aaPIM aaPlatformInfoSvr aaUserValidator Bootstrap	DCOM

AVEVA Historian

Port	Can be configured	Protocol	Subsystem	Purpose
32565	Yes	TCP	aaClientAccessPointNG.exe	Historian Client Access Point NG
32568	Yes	TCP	AVEVA Historian	AVEVA Historian as a real-time service
32569	Yes	TCP (HTPPS)	Insight	Insight on-premise gateway
32573	Yes	TCP (HTTPS)	Historian Secured	REST

Port	Can be configured	Protocol	Subsystem	Purpose
			Gateway	communications

Device Integration (Communication Drivers Pack)

Port	Can be configured	Protocol	Subsystem	Purpose
102	No	TCP	SiDirect OI Server	Siemens PLC communication to OI Server
135	No	TCP	DASEngine, OPC	DCOM and RPC
443	Yes	TCP (HTTPS)	GDIWebServer	MQTT and Auto-Build configuration
502	No	TCP	MBTCP OI Server	Modbus communication to OI Server
1883 8883	Yes	TCP	MQTT	MQTT broker communication to OI Server
2221 2222 2223	No	TCP	ABTCP OI Server	Allen-Bradley PLC communication to OI Server
5413	No	TCP	SuiteLink	SuiteLink communication
18245	No	TCP	GESRTP OI Server	GE PLC communication to OI Server
44818	No	TCP	ABCIP OI Server	Allen-Bradley CIP PLC communication to OI Server
See note, below	Yes	TCP	OPC UA Services	Remote access to the OPC UA servers

Note: The Communication Drivers Pack uses the default OPC ports, which are are configurable. For details, refer

to the OPC Foundation documentation:
https://opcfoundation.github.io/UA-.NETStandard/help/firewall_settings.htm

InTouch

Port	Can be configured	Protocol	Subsystem	Purpose
51218	No	TCP	Alarmmgr.exe	Alarm Manager
48032 – 65000	Yes	TCP	InTouch.OPCUA.ServiceHost.exe	InTouch OPC UA

InTouch Access Anywhere (ITAA)

Port	Can be configured	Protocol	Subsystem	Purpose
443	Yes	TCP	EricomSecureGateway.exe	Secure Gateway
7433	Yes	TCP	EricomAuthenticationServer.exe	Access Anywhere Authentication Server
8080	Yes	TCP	EricomSecureGateway.exe AccessServer64.exe	Communication between ITAA Server and ITAA Secure Gateway
57111	No	UDP	EricomSecureGateway.exe	Secure Gateway
57733 57734 57735	No No No	TCP	AccessServer64.exe	Server

Licensing

Port	Can be configured	Protocol	Subsystem	Purpose
80	Yes	TCP (HTTP)	License Manager	Web Service
443	Yes	TCP	License Manager	License Manager outbound to activation server

Port	Can be configured	Protocol	Subsystem	Purpose
50051	Yes	TCP (HTTPS)	Licensing Platform	Serve licensing requests from products
55555	Yes	TCP (HTTP)	License Server	License Server Translator service. Also required to support prior client versions from Server 4.0
55559	Yes	TCP (HTTP/ HTTPS)	License Server	License Server core service
59200	Yes	TCP	License Server	License Server Agent Service
59201	Yes	TCP (HTTPS)	License Server	License Server Agent Service

OMI Web Client

Port	Can be configured	Protocol	Subsystem	Purpose
80 808	No	TCP (HTTP)	VCP	vcp.services.onprem.DataAccess.exe vcp.services.onprem.WebServer.exe
443 80	No	TCP (HTTPS)	VCP	vcp.services.onprem.frontdoor.exe

Operations Control Logger

Port	Can be configured	Protocol	Subsystem	Purpose
135	No	TCP	RPC	Used for dynamic

Port	Can be configured	Protocol	Subsystem	Purpose
				port mapping

Platform Common Services (PCS)

Port	Can be configured	Protocol	Subsystem	Purpose
80	No	TCP (HTTP)	PCS	PCS.ServiceManager.exe
443	Yes	TCP (HTTPS)	PCS	PCS.Agent.exe(Discovery) PCS.IdentityManager.Host.exe
808	Yes	TCP	PCS	WCF shared port
1900	No	UDP (SSDP)	PCS	PCS.IdentityManager.Host.exe SSDP
7084 7085	No No	TCP	PCS	System authentication during node registration

SQL Server

Port	Can be configured	Protocol	Subsystem	Purpose
1433	Yes	TCP	SQL Server	SQL Server
1434	No	UDP	SQL Server	SQL Server browser

System Monitor

Port	Can be configured	Protocol	Subsystem	Purpose
25	Yes	TCP (SMTP)	System Monitor	SMTP Server
80	Yes	TCP (HTTPS)	System Monitor	Sentinel Console Service

Port	Can be configured	Protocol	Subsystem	Purpose
443	Yes	TCP (HTTPS)	System Monitor	Secure Sentinel Console Service
587	Yes	TCP (SMTP)	System Monitor	Secure SMTP Server

FDA compliance

Application Server provides compliance with the guidance issued by the Federal Food and Drug Administration (FDA) for electronic records and electronic signatures (21 CFR Part 11). Since both Managed InTouch applications and AVEVA OMI ViewApps are underpinned by Application Server, you can use either product to provide visualization for your application and maintain compliance with 21 CFR Part 11.

Topology

System and information requirements are unique to each manufacturing domain. To control equipment, computers must provide real-time response to interrupts. To plan production, scheduling systems must consider sales commitments, routing costs, equipment downtime, and numerous other variables.

Enterprise system and information requirements are satisfied by designing effective network topologies and implementing software to leverage the topology.

The topology configurations include descriptions and "best practice" recommendations for specific components and functionality.

Note: For information on system requirements, see the user guides or readme files in the installation directory of the appropriate installation media. The most up-to-date information is available online from the AVEVA Global Customer Support (Knowledge and Support Center) website: https://gcsresource.aveva.com/TechnologyMatrix. Pay particular attention to the requirements regarding the version and Service Pack level of the operating system and other application components.

System Platform component descriptions

The System Platform framework consists of server-side configuration and deployment related components. In System Platform, these components include:

- A centralized object repository called the Galaxy Repository
- An integrated development environment (IDE) for engineering, configuration, and maintenance
- Run-time nodes
- A database to store historical data (the Historian)

Engineering Station

Runs the tools necessary to develop and configure the application, such as the System Platform IDE, Industrial Graphic Editor, and InTouch WindowMaker. There can be multiple engineering stations for multi-user development teams.

Galaxy Repository

Runs the Galaxy Repository service and hosts the configuration project database. There is only one Galaxy Repository per Galaxy.

Historian Server

Runs the Historian Server software and hosts the history and alarm databases. Typically, there is only one historian server per Galaxy, but there can be more than one if needed, such as in largely distributed Galaxies hosting local historian servers per location.

Application Object Server

Computer where application objects operate in run time after being deployed. There can be multiple Application Object Servers (AOS) for load distribution, redundancy, or both.

Device Integration Server

Computer connected to the control network and running the corresponding Communication Drivers, or legacy OI Servers. A single Device Integration Server can run multiple drivers, or you can have multiple Device Integration Servers, depending on the control network topology.

RDP Server

The RDP Server allows the supervisory clients to run the operator's interface through a RDS session.

InTouch Access Anywhere (ITAA)

ITAA allows the supervisory clients to run the operator's interface via an HTML5 compliant browser.

ITAA Secure Gateway

Provides secure web remote connections from InTouch Access Anywhere clients running on unsecured networks to internal network resources. The Secure Gateway provides authentication and authorization services as well as data encryption.

ITAA Authentication Server

Authenticates InTouch Access Anywhere users before they are allowed to connect to internal resources.

Supervisory Client
Thick client

Runs the operator's interface or HMI through OMI runtime tools installed in a workstation. There can be multiple visualization stations.

RDS or Web Client

Runs the operator's interface or HMI through OMI runtime tools through a RDS or HTML5 web browser. There can be multiple visualization stations.

License Server

Provides the functionality for users acquire, store, maintain, and serve licenses to all installed AVEVA Enterprise software.

System Management Server

Encrypts communications between all System Platform nodes.

System Platform and Application Server

Application Server is a core component of System Platform. The Galaxy, stored in the Galaxy Repository (GR), is the top level object in Application Server, and encompasses the whole supervisory control system. A Galaxy is represented by a single logical namespace and a collection of Platforms, AppEngines, and application objects. The Galaxy defines the namespace in which all components and objects reside.

Because of its distributed nature and common services, Application Server does not require expensive server-class or fault-tolerant computers to enable a robust industrial application.

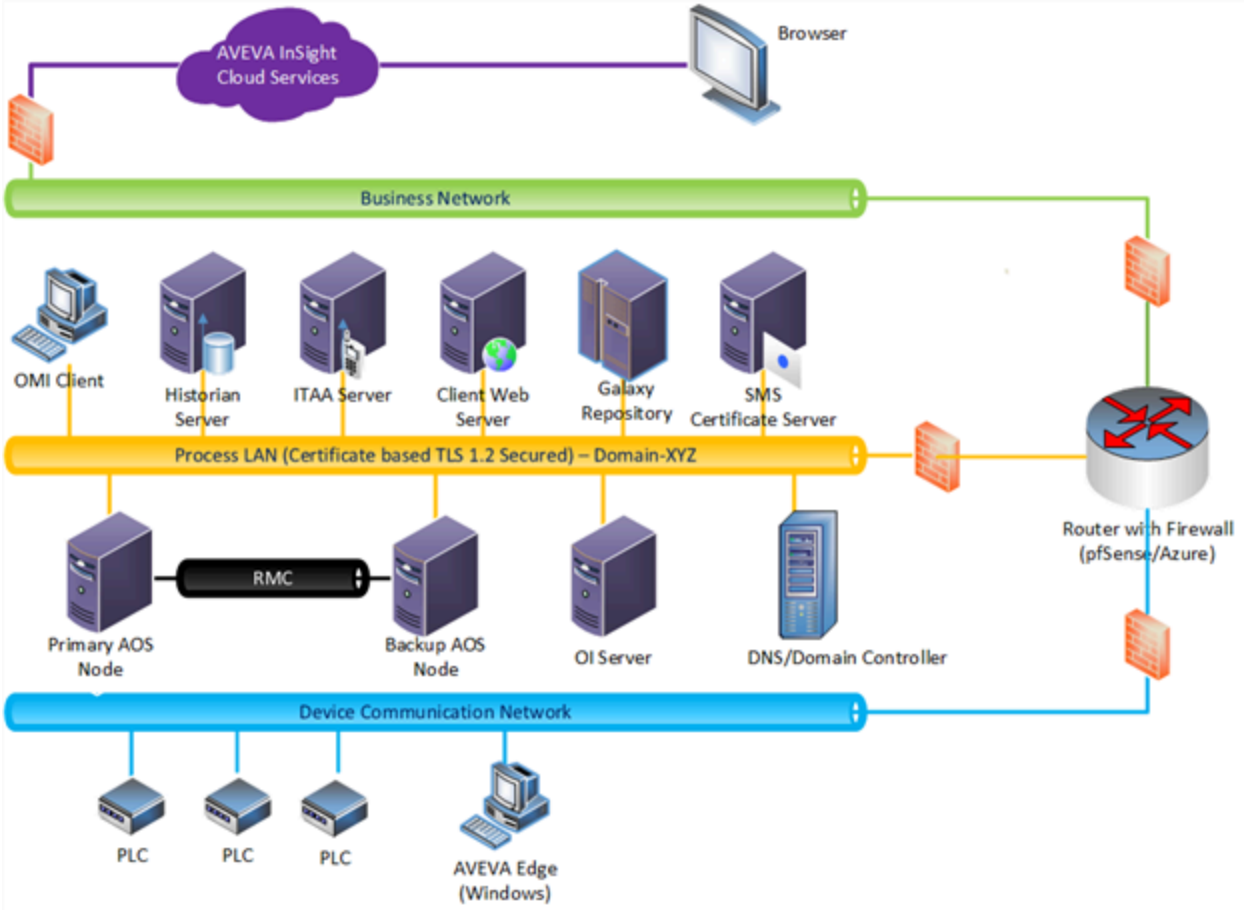
Application Server distributes objects throughout a distributed (networked) System Platform environment, allowing a single application to be split into a number of different component objects, each of which can run on a different computer.

Before exploring the following System Platform topologies, review the main components and how they will be distributed based on requirements and functionality. The main components are:

- IDE: Application Server development node (Engineering workstation). The System Platform IDE runs on this node. If you are using InTouch HMI for System Platform for visualization, WindowMaker can reside on this node, in addition to the IDE.
- GR: Galaxy Repository. This is the configuration database. The GR uses a Microsoft SQL Server database.
- AOS: ApplicationObject Server (run time) nodes. Multiple sets of redundant AOS nodes can be implemented.
- OI: Operations Integration nodes contain the communication drivers that function as the interface to the PLCs in the control network.
- Historian Server. This runs the Historian database and saves historical data. The Historian also uses a Microsoft SQL Server database.
- Visualization nodes running AVEVA OMI ViewApps or InTouch WindowViewer. There are differences in how InTouch HMI and AVEVA OMI handle different types of controls. Consider the following:
  - ActiveX controls: Allowed by InTouch, but cannot be used with OMI.
  - .NET controls: For InTouch, .NET controls can be embedded directly in an Industrial Graphic (the Industrial Graphic functions as a container for the control). For OMI, .NET controls must be placed in panes (one control per pane). If an Industrial Graphic used in OMI contains an embedded .NET control, OMI will strip out the control before it displays the graphic. The only way that Industrial Graphics can interact with .NET controls in OMI is through layout scripting. You must first build a layout that hosts both the Industrial Graphic and the .NET control in separate panes, and embed that layout into a pane. In this case, the layout with the .NET control and Industrial Graphic is the content that is embedded into another layout.

Each System Platform and Application Server component can be installed on its own node, or a single node can contain multiple components. In a small system, all Application Server components can exist on a single node. The same principles and resource requirements apply to both physical environments and virtual environments that leverage Hyper-V or VMware.

The following figure shows a multi-node System Platform topology, running the IDE, GR, Historian, System Management Server, License Server, OI Server, and AOS Servers on separate nodes. The IDE, GR, OI Server, and AOS Servers are components of Application Server.



Common node configurations

How you configure a particular node in your System Platform topology depends on number of I/O, requirements for high availability and/or disaster recovery, and network bandwidth. Typical System Platform node configurations include the components and capabilities described below.

Application Server nodes

Application Server nodes can be broadly categorized as development nodes, Galaxy Repository nodes, run-time nodes, or all-in-one nodes.

All Application Server nodes require the following two software elements:

- Bootstrap
- Platform Common Services (PCS) Runtime

Development node (engineering workstation)

A **development node** (IDE node) is used for configuring Application Server, and creating, managing, maintaining and deploying a Galaxy and OMI applications (ViewApps). The IDE is not part of the run-time environment.

The core components of a development node are as follows:

- Bootstrap
- PCS Runtime
- System Platform IDE
- System Monitor

In addition to the components listed above, the following AVEVA/System Platform components and applications are also installed:

- Historian Search
- Historian Client
- Operations Control Management Console
- Operations Control Logger
- AVEVA Communications Drivers Pack
- AVEVA Application Manager

A development node can be combined with other types of Application Server nodes, such as:

- Galaxy Repository
- Application Object Server
- OI Server
- OMI Client (visualization node)

Other System Platform products that are often combined onto an Application Server development node include:

- InTouch development (WindowMaker)
- InTouch run time (WindowViewer)

  Adding InTouch components adds other associated software components. See InTouch HMI node for additional information.

Best practice

When remote off-site access to the Galaxy Repository is required by means of the IDE, use a Remote Desktop connection to the Galaxy repository where the IDE has also been installed. The remote user must have appropriate permissions.

If InTouch HMI is used, the Engineering Station node also hosts WindowMaker to create and modify InTouch applications.

Galaxy Repository

The **Galaxy Repository** is a database that contains the configuration data for your Galaxy. Galaxy data is stored in a Microsoft SQL Server database. The following components are installed on a Galaxy Repository (GR node):

- Bootstrap

- PCS Runtime
- PCS Service Repository
- Galaxy Repository (requires Microsoft SQL Server)
- System Monitor
- AVEVA License Server

  While AVEVA Licensing is automatically installed when the GR component is selected for installation, you do not need to load licenses onto the node if you are using a different node as the License Server.
- AVEVA License Manager

In addition to the components listed above, the following System Platform components and applications are also installed:

- Historian Search
- Historian Client
- Operations Control Management Console
- Operations Control Logger
- AVEVA Communications Drivers Pack
- AVEVA Application Manager

A Galaxy Repository node can be combined with other types of Application Server nodes, such as:

- System Platform IDE (Development Node)
- Object Server
- I/O Server
- OMI Client (visualization node)

The GR can be installed on a dedicated node, in combination with the System Platform IDE, or with other System Platform components, depending on your system size.

Note: For information on installing the GR on the same node with any other components, see Single Node System Implementation.

The Galaxy Repository manages the configuration data associated with one or more Galaxies. This data is stored in individual databases, one for each Galaxy in the system. Microsoft SQL Server is the relational database used to store the data.

During run-time, the GR communicates with all nodes in the Galaxy to keep them updated on global changes such as security model modifications, etc. Even though it is possible to disconnect the GR from the Galaxy and still keep remaining nodes in production, it is recommended to maintain the GR connection to the Galaxy in order to transfer all global changes when they occur.

The Galaxy Repository is accessed when the objects in the database are viewed, created, modified, deleted, deployed, or uploaded. The Galaxy Repository is also accessed when a running object attempts to access another object that has not been previously referenced.

Application Object Server

An **Application Object Server (AOS)** node is a run-time platform used for hosting run-time objects (instances) in

a deployed Galaxy.

The Application Object Server (AOS) node provides the run-time resources for AppEngines, Areas, AutomationObjects, and DeviceIntegration (DI) Objects. The AOS node requires a Platform to be deployed. The OPCClient objects are generally used for device integration.

AOS functionality can be combined with a visualization node, depending on the process requirements and system capabilities. A distributed local network topology takes advantage of this type of configuration to provide flexibility to the system.

There is in interplay between the number of AppEngines, scan rate, and I/O count that affect how an AOS node should be sized. No more than two AppEngines (primary and backup) should be deployed to each logical core on an AOS node. Faster scan rates will result in a lower I/O threshold. In other words, the higher scan rate, the lower the I/O count that can be supported on the AOS node.

The following are the core components of an AOS node:

- Bootstrap
- PCS Runtime
- System Monitor

In addition to the components listed above, the following AVEVA/System Platform components and applications are also installed:

- Historian Client
- Operations Control Management Console
- Operations Control Logger
- AVEVA Communications Drivers Pack
- AVEVA Application Manager

OI server functions are frequently combined on an AOS node, if the I/O load permits.

OI Server

An **OI server** is a run-time platform used for hosting Communication Drivers in a deployed Galaxy. Communication Drivers function as interfaces between the PLCs in your control network and the System Platform supervisory network. The following are the core components of an OI Server:

- Bootstrap
- PCS Runtime
- System Monitor

In addition to the components listed above, the following AVEVA/System Platform components and applications are also installed:

- Historian Client
- Operations Control Management Console
- Operations Control Logger
- AVEVA Communications Drivers Pack

- AVEVA Application Manager

These are the same components as on an AOS node. The difference is that you do not deploy objects to an OI server.

OMI Client

An **Operations Management Interface (OMI)** visualization node is a run-time platform that hosts an OMI ViewApp to provide graphical representation of data in a deployed Galaxy. At run time, the ViewApp is deployed to the OMI client, which contains the following the core components:

- Bootstrap
- PCS Runtime
- System Monitor

In addition to the components listed above, the following AVEVA/System Platform components and applications are also installed:

- Historian Client
- Operations Control Management Console
- Operations Control Logger
- AVEVA Communications Drivers Pack
- AVEVA Application Manager

OI server functions are frequently combined on an AOS node, if the I/O load permits.

All-in-one node

An **all-in-one node** combines all of Application Server components and functionality onto a single node. It functions as a development node, as a Galaxy Repository, as a run-time node for hosting objects and Communication Drivers, as as visualization node that hosts OMI ViewApps (and optionally, InTouch HMI ViewApps). The following are the core components of an All-in-One node:

- Bootstrap
- PCS Runtime
- System Platform IDE
- System Monitor

System Management Server node

The System Management Server is used to implement important security measures for System Platform.

Important! Using a **System Management Server (SMS)** is highly recommended to ensure the security of System Platform. It is **required** when redundancy is enabled for Application Server nodes.

Security measures implemented by the SMS include:

- Enabling secure communication between System Platform nodes.
- Synchronizing data between redundant Application Server AppEngines.
- Setting the System Platform installation type and license mode.
- Setting port numbers for inter-node communications.
- Setting the SuiteLink security mode and user access to the AVEVA Network Message Exchange.
  - Communication over a SuiteLink connection can be configured to use only encrypted (secure) communications, or to allow unencrypted communications, if a secure (TLS) connection cannot be established. SuiteLink is used for a number of different applications in System Platform.
  - The AVEVA Network Message Exchange (NMX) is an application communication protocol that leverages a DCOM-based transport mechanism for communication between nodes.
- Certificate management
- User authentication via the OpenID connect standard, which allows single sign on (SSO) via an external identity provider.

To enable security, every System Platform node must communicate with the System Management Server. There should only be one System Management Server in your System Platform topology, otherwise, communication disruptions may occur. The System Management Server stores shared security certificates and establishes a trust relationship between machines. You can configure one additional node as a redundant SSO server, which functions as a backup for single sign-on if the System Management Server cannot be reached.

Beginning with System Platform 2023 R2, every redundant Application Server run-time node must use the System Management Server if data is being historized. Redundant nodes have an instance of HCAP running, which is used to synchronize tags and store-and-forward data between redundant AppEngines. Secure communication is required for HCAP, and thus, redundant nodes will not function if the SMS is not configured.

If some nodes have not been upgraded to System Platform 2017 Update 3 or later, communication with those older nodes may need to utilize unsecure communication. However, communication between nodes running System Platform 2017 Update 3 or later will be encrypted, as long as the nodes are configured to communication with the System Management Server.

For more information about configuring the System Management Server with an authentication provider, see Design a robust SSO system with an external authentication provider.

Design a robust SSO system with an external authentication provider

Adding Azure AD as an authentication provider to the System Management Server allows several different ways to configure your System Platform installation. The following configurations provide varying degrees of system robustness, redundancy, and complexity. Use the architecture that most closely aligns with your requirements.

Recommended SMS architecture utilizing an authentication provider

This system design contains a minimum of three nodes for user authentication, and provides the highest level of robustness and redundancy. It is also the most architecturally complex. Use the System Platform Configurator to configure the System Management Server.

Node 1 - standalone SMS

Configure the System Management Server on the license server or System Monitor server.

- On the **System Management Server** tab of the Configurator, select the option "**This machine is the System Management Server**."
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."
  - Configure the token host.

  Note: This node is not deployable since it does not contain a WinPlatform object. As a result, it may not be reachable by other nodes under certain circumstances. Therefore, Redundant SSO nodes are required.

Node 2 - redundant SSO node on the GR

Configure the GR node or other deployable node, such as an IDE node, as a Redundant SSO node.

- On the **System Management Server** tab of the Configurator, select the option "**Connect to an existing System Management Server**."
  - Select node 1 as the existing SMS node.
  - Select the checkbox "**Configure this machine as a Redundant SSO Server**."
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."
  - Configure the token host.

Node 2 is now now configured to provide user authentication via the SSO provider in the event node 1 is unreachable.

Node 3 - second redundant SSO Node on a deployed platform

Configure an IDE node or other deployable node, such as an Application Object Server node, as a second Redundant SSO node.

- On the **System Management Server** tab of the Configurator, select the option "**Connect to an existing System Management Server**."
  - Select node 1 as the existing SMS node.
  - Select the checkbox "**Configure this machine as a Redundant SSO Server**."
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."
  - Configure the token host.

Node 3 is now now configured as a second redundant authentication provider.

Node 4 though *n*

- On the **System Management Server** tab of the Configurator, select the option "**Connect to an existing System Management Server**."
  - Select node 1 as the existing SMS node.
  - For the option to configure the node as a Redundant SSO Server, leave the checkbox unchecked.
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."

    Note: Since this node is not a redundant authentication provider, the fields to configure a token host are not shown.

Simplified SMS architecture utilizing an authentication provider

This system design contains a minimum of two nodes for user authentication, and provides robustness and redundancy.

Node 1 - SMS on the GR or other deployed platform

In this simplified architecture, the System Management Server is installed on the GR node.

- On the **System Management Server** tab of the Configurator, select the option "**This machine is the System Management Server**."
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."
  - Configure the token host.

Node 2 - redundant SSO node on the IDE

The System Management Server is installed on an IDE node or an Application Object Server (run-time) node.

- On the **System Management Server** tab of the Configurator, select the option "**Connect to an existing System Management Server**."
  - Select node 1 as the existing SMS node.
  - Select the checkbox "**Configure this machine as a Redundant SSO Server**."
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."
  - Configure the token host.

Node 2 is now configured as a redundant authentication provider.

Node 3 though *n*

- On the **System Management Server** tab of the Configurator, select the option "**Connect to an existing System Management Server**."
  - Select node 1 as the existing SMS node.
  - For the option to configure the node as a Redundant SSO Server, leave the checkbox unchecked.
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."

    Note: Since this node is not a redundant authentication provider, the fields to configure a token host are not shown.

Minimum SMS architecture utilizing an authentication provider

This system design uses a single node for user authentication. This design does not provide for redundancy.

Node 1 - SMS on the GR or other deployed platform

In this minimum architecture, the System Management Server is installed on the GR node.

- On the **System Management Server** tab of the Configurator, select the option "**This machine is the System Management Server**."
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."
  - Configure the token host.

Node 2 though *n*

- On the **System Management Server** tab of the Configurator, select the option "**Connect to an existing System Management Server**."
  - Select node 1 as the existing SMS node.
  - For the option to configure the node as a redundant SSO Server, leave the checkbox unchecked .
- On the **Authentication Provider** tab:
  - Select the checkbox to "**Configure this machine to provide SSO via an external Authentication Provider**."

    Note: Since this node is not a redundant authentication provider, the fields to configure a token host are not shown.

InTouch HMI node

An **InTouch HMI node** is typically used as a run-time platform to host an InTouch ViewApp (view application). You can configure an InTouch node for both configuration and run-time, or you can combine an InTouch

configuration/run-time node on an engineering workstation with the System Platform IDE.

The following are the core components of an InTouch HMI:

- Bootstrap
- PCS Runtime
- PCS Service Repository
- System Monitor
- AVEVA License Server
- AVEVA License Manager

In addition to the components listed above, the following System Platform components and applications are also installed:

- Historian Search
- Historian Client
- Operations Control Management Console
- Operations Control Logger
- AVEVA Communications Drivers Pack
- AVEVA Application Manager

An All-in-One node can be combined with other System Platform products, such as:

- Historian Server (additional hardware/virtual resources may be required)
- InTouch Access Anywhere

Other System Platform products that are often combined onto an Application Server All-in-One node include:

- InTouch development (WindowMaker)
- InTouch run time (WindowViewer)

  Adding InTouch components adds other associated software components. See xxxx for additional information.

Historian Server

The Historian node is leverages SQL Server to run the AVEVA Historian software. The Historian stores all historical process data and provides real-time data to System Platform client applications such as AVEVA OMI and InTouch HMI.

The Historian node does not require a Platform. The Application Object Server pushes data (configured for historization) to the Historian node using the Historian software.

In most cases, Historian Server should be the only product installed on the Historian node because of the I/O and memory requirements associated with running the Historian.

Prior to System Platform 2023 R2, the default HCAL port for Historian was 32568. In System Platform 2023 R2, the default is 32565. For REST APIs, the default port is 32569.

Best practice

Most system topologies combine the Historical and Alarm databases on the Historian Node. Configure the alarm system using the Alarm Logger™ utility, which creates the appropriate database and tables in Microsoft SQL Server. For requirements and recommendations for alarm configuration, see Implementing Alarms and Events. For information about historization, see Historizing Data.

Topology categories

The following information describes high-level topology categories using System Platform components.

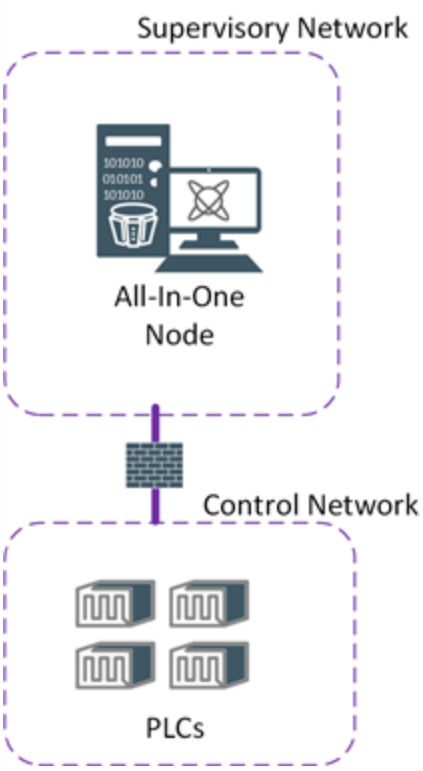
All-in-one configuration

Using the All-In-One System Platform installation option places all System Platform components on a single node. While this is the simplest topology from a logical viewpoint, it also places the greatest resource challenge for that single computer. For optimal performance, a CPU clock speed greater than 2.8 GHz is recommended. For large applications on an All-In-One node, dual XEON processors are recommended.

Application Size	Level	Logical Processors	RAM	Free Disk Space	Network Speed
Small: 1,000 I/O max	Minimum	8	8 GB	200 GB	100 Mbps
	Recommended	12	12 GB	500 GB	1 Gbps
Medium: 20,000 I/O max	Minimum	12	16 GB	500 GB	1 Gpbs
	Recommended	16	32 GB	1 TB	1 Gbps
Large: 100,000 I/O max	Minimum	20	32 GB	2 TB	1Gbps
	Recommended	24	64 GB	4 TB	1 Gbps

The All-In-One option, by default, places the Application Server GR, IDE, and run-time components on the node, as well as InTouch HMI, Historian, Licensing, the System Management Server (and other System Platform components). See All-in-one node for additional information.

The following figure shows an all-in-one node:



Medium-sized network

This topology is generally applicable to medium-sized systems where the processing requirements of each software component can be easily handled by the nodes providing the projected performance support. If other dimensions of the system, such as I/O, engine loading, and complexity of the galaxy.

The primary characteristic of this topology type is that Visualization and ApplicationObject Server functionalities coexist in the same node, and the IDE and GR coexist on another node within the distributed local network.

AOS/Visualization node

The Visualization and ApplicationObject Server components are combined on the same node. Both components share the Platform, which handles communication with other nodes in the Galaxy. The Platform also allows for deployment/undeployment of ApplicationObjects.

If you plan to combine the Visualization and Application Object Server components on the same node, evaluate the resource requirements for the following:

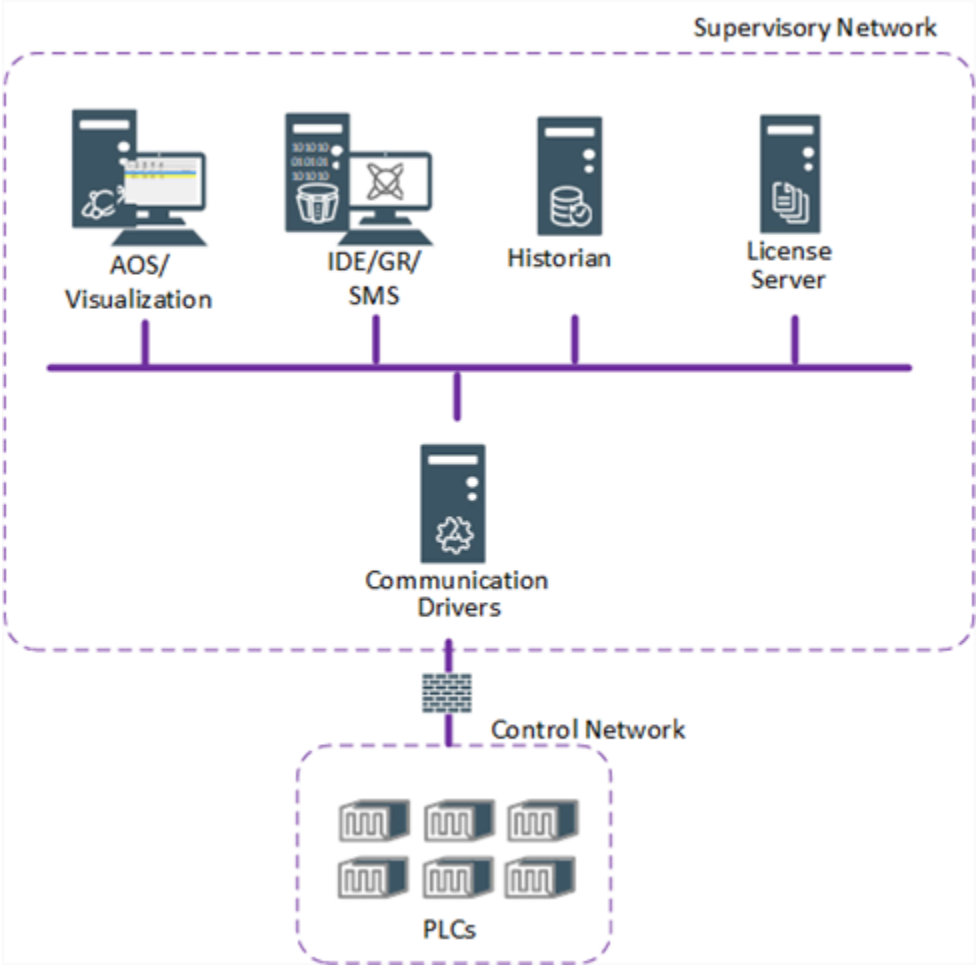
- Active tags-per-window if using InTouch HMI
- Number of I/O
- Alarm displays
- Historized tags

These values will impact ApplicationObject service performance. Refer to System sizing guidelines for computer resource recommendations.

Note: For details on Alarm System configuration, see Implementing Alarms and Events.

IDE/GR/SMS node

As in the case of the AOS/Visualization node, the System Platform IDE and Galaxy Repository, along with the System Management Server are combined onto a second node. Additional consolidation is possible, for example, by installing the Communication Drivers on this node as well. All components share the Platform for communication with other nodes and components.



Client operating systems such as Windows 10 can manage up to 10 simultaneous active connections with other nodes. If the system contains more than 10 simultaneously-active nodes, Windows Server must be used for all nodes.

Communication driver nodes

Different I/O data sources have different requirements. Two main groups are identified:

- Legacy I/O Server applications (SuiteLink, DDE, and OPC Servers) do not require a platform on the node on which they run. They can reside on either a standalone or workstation node.

  However, the DI client objects used to communicate with those data sources such as the DDESuiteLinkClient object, OPCClient object, and InTouchProxy objects must be deployed to an AppEngine on a Platform.

Although it is not required that these DI client objects be installed on the same node as the data server(s) they communicate with, it is highly recommended in order to optimize communication throughput.

- Communication Drivers and their corresponding DIObjects must reside on the same computer hosting an AppEngine.

Best practice

Historian node: The Historian software should run on a designated node. The number of historized tags will determine the sizing of the node. Refer to System sizing guidelines for for computer resource recommendations.

Engineering Station and GR (Configuration Database): The Engineering Station node hosts the System Platform IDE and optionally, InTouch WindowMaker to facilitate Application Server and InTouch Software application development. As the GR node, it hosts the SQL Server database that stores the Galaxy's Configuration Data.

Large network

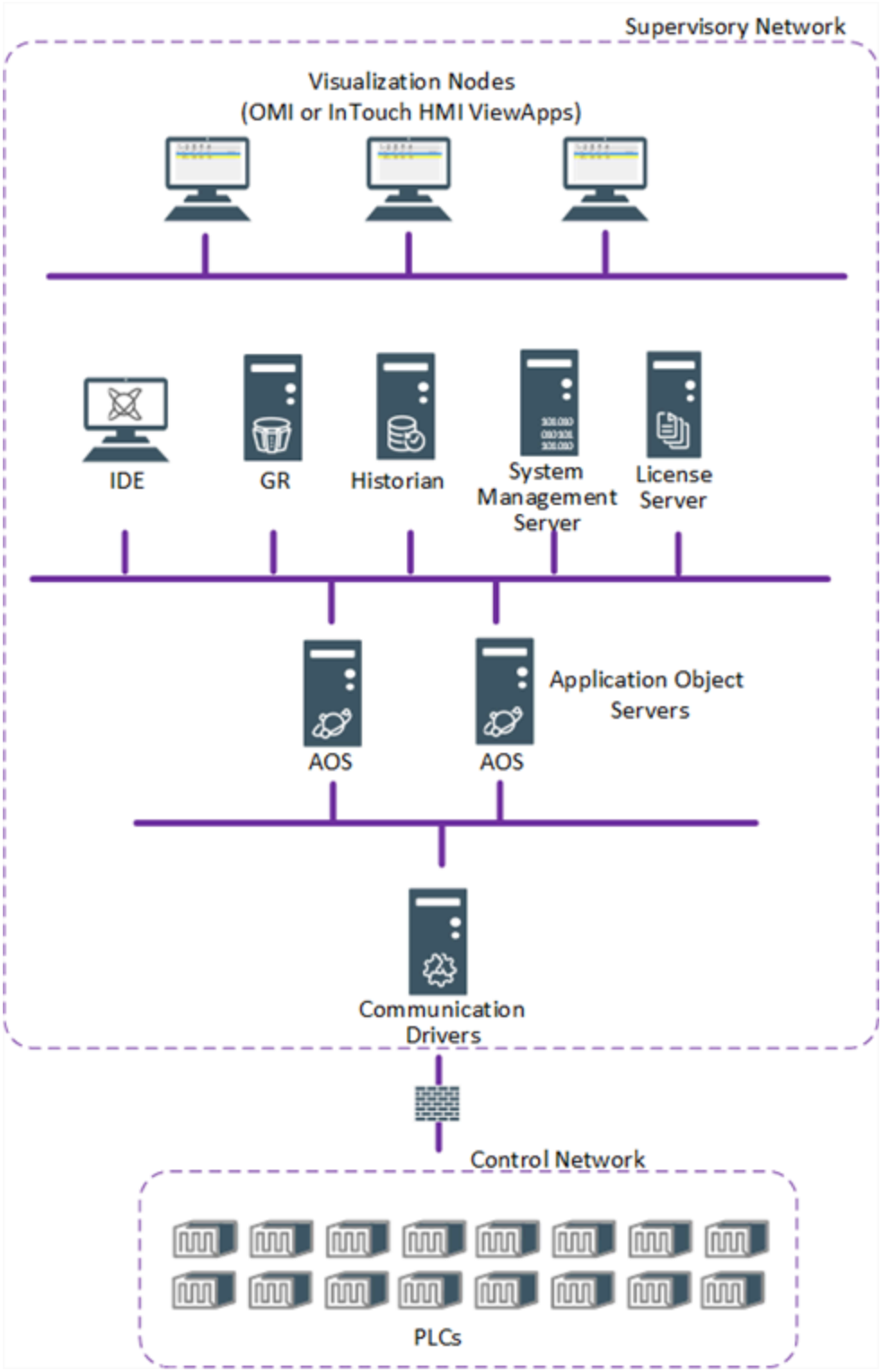
This topology configuration includes dedicated nodes running Application Object Servers, while visualization tasks are performed on separate nodes.

The benefits of this topology include usability, flexibility, scalability, system reliability, and ease of system maintenance, since all configuration data resides on dedicated servers.

The client components (represented by the visualization nodes) provide the means to operate the process using applications that provide data updates to process graphics. The clients have a very light data processing load.

The ApplicationObject Server nodes share the load of data processing, alarm management, communication to external devices, security management, etc.

The following figure illustrates a client/server topology:



This topology is scalable to include a greater number of servers. Including more servers distributes data processing loads and enables a higher load of I/O reads/writes. Client nodes can be added when additional operator stations are needed.

Working in wide-area networks and SCADA systems

This section discusses application of System Platform products in wide-area network (WAN) and Supervisory Control and Data Acquisition (SCADA) systems environments.

The WAN network environment exhibits the following characteristics:

- Low bandwidth.
- High latency.
- Intermittent communication.

Wide-Area Networks overview

Wide-Area Networks (WANs) consist of computers located across large distances. Communication between the computers is typically handled by modems, T1 lines, or satellite links. Data transmitted in this environment must travel through a large number of network components (routers, satellites, modems). By doing so, latency (delay from when the data was sent to when it was received) is increased.

Further, the underlying technologies used for communication is often limited to low bandwidth. As a result, these distributed networks may experience delays or breaks in communication due to relatively high amounts of network traffic, or interference by external conditions such as severe weather.

WANs are used in industries such as water/waste-waster, telecommunications, natural gas production, and oil production/distribution, where they are implemented as part of a SCADA system. The SCADA system is usually a central computer that communicates over the WAN to remote PLCs or RTUs.

Note: In this context, a Remote Terminal Unit (RTU) is defined as an industrial data collection device typically located at a remote location, which communicates data to a host system by using telemetry (such as radio, dial-up telephone, or leased lines).

A SCADA system gathers real-time data, transfers the data to a central site, performs the necessary analysis and control, and displays the information visually in an appropriately organized fashion. SCADA topologies can easily be expanded to handle additional remote sites and I/O points.

The SCADA system collects and records data events and alarms. A SCADA host performs centralized alarm management, data trending, and operator display and control.

Current status and commands are handled by remote controllers consisting of RTUs and PLCs. SCADA systems employ RTU or PLC protocols including Modbus, AB-DF1, and DNP3.0. SCADA communications can use a range of wired (lease line, dialup line, fiber, ADSL, cable) and wireless media (licensed radio, spread spectrum, cellular, CDPD, satellite). Communication drivers collect data from remote units, then send this data to the Application Server using OPC or SuiteLink protocols.

Network terminology

When metric prefixes (k for kilo, M for Mega) are used in a network context, they retain their original definitions. That is, k = 1,000 and M = 1,000,000. This usage differs from disk-storage terminology, where KB = 1024 Bytes and MB = 1,048,576.

The following table summarizes the conventions used in this section:

k	1,000
M	1,000,000
b	bit
B	Byte
bps	bits per second
Bps	Bytes per second

Network and operating system configuration

The following information refers to both network component configuration and the operating system configuration necessary to function successfully. Some information is included for contextual purposes only.

Minimum bandwidth requirements

The slowest recommended network connection speed for System Platform for platform, history, and alarm communications is 128 kbps of dedicated bandwidth.

This is totally independent of communications between OPC/IO/Communication Driver and field devices (RTUs, etc.), which typically occurs over a different device network using RTU and/or other industrial protocols, often at slower data rates (i.e.: 56 kbps and lower).

Check ping times for remote nodes and consider improving the available bandwidth or reducing I/O polling frequencies for those nodes that exhibit very slow ping times. For information about using Ping for checking and diagnostics, see "Diagnostics" on page 266.

Subnets

Set up subnets and sub-areas in the IP network. Most SCADA systems use routers and switches to isolate traffic within a particular site so as not to burden the network. Be sure routers are configured to isolate and route information correctly.

DCOM

When assessing and setting up the network, be careful in setting blocked ports. Some DCOM ports need to be open to communicate with OPC. Leave open the ports that interact between System Platform components.

See List of System Platform Ports for more detailed information.

Domain controller

The Domain Controller is a Windows server computer that stores user account information, authenticates users, and enforces security policy for a Windows domain. Domain controllers detect changes to user accounts and synchronize changes made in Directory Server user entries.

Distributed SCADA networks typically employ multiple Domain Controllers at strategic network locations.

The Domain Controller node includes the DNS service and Time Synchronization to manage network communication requests.

Domain name server (DNS)

An Internet service that translates domain names and host names into IP addresses.

Best practice

Implement Universal Time Synchronization (UTS) at each site. Doing so provides absolute certainty that data is properly time-stamped and forwarded to the historian under all circumstances. Current technology uses GPS or radio broadcast software along with dedicated hardware devices that may linked to one or more computers at the site using serial ports, Ethernet or USB. See Synchronizing time across a galaxy for detailed information.

The following considerations apply when implementing UTS:

- If time master provider is not in same geographical area, there is a risk of losing a connection to the time server through time "drift." In other words, nodes in the same geographical area should have access to a master in the same area, or use a GPS.
- Synchronize all Domain Controllers with a common time provider.

  The Windows service uses time stamp as a part of the Kerberos security implementation. Kerberos is a system that provides a central authentication mechanism for a variety of client/server applications, using passwords, secret keys, and time-sensitivity.

  If the time between clients/Domain Controllers, or between Domain Controllers (in other geographical areas) drifts, the operating systems may fail.

Synchronizing time across a galaxy

Certain Application Server functions like scripting, alarming, and historizing require that all member computers of a galaxy are synchronized to the same time. A time master is a network-time-protocol server that provides a time to which other nodes on your network can synchronize.

The System Management Server node should be designated as the time master. The Application Server nodes in the galaxy periodically synchronize their clocks to the time master.

Using time synchronization in Windows domains

The system administrator of the Windows domain may have time synchronization configured already. If this is the case, configuring a galaxy time master is unnecessary and can conflict with the existing time synchronization.

Application Server does not implement its own time synchronization algorithm, but supports time synchronization for your galaxy through windows time service. If needed, you can configure a galaxy time master to utilize the Windows time service on a node that you designate as the time master, preferably the System Management Server node. If a node in your galaxy has a system time that is outside of a predetermined amount (for example, five minutes), certain operations such as deployment may fail.

For information about configuring the Windows time server, refer to the following Microsoft documentation:

https://learn.microsoft.com/en-us/windows-server/networking/windows-time-service/configuring-systems-for-high-accuracy

Synchronization schedule

The designated node clock serves as the master clock for all timestamping functions. Time synchronization is based on the Microsoft Windows time service.

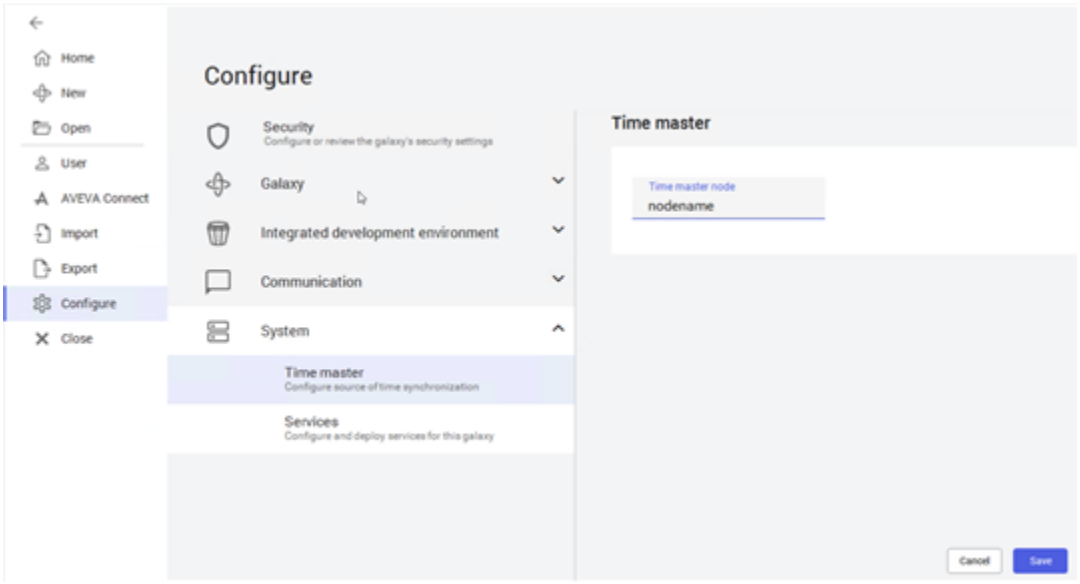
All WinPlatforms begin synchronizing the time on their node when they are deployed.

You can specify a time master node in another time zone. The time on each Application Server node is set to the time specified on the node in the other time zone.

Configure the time master

To configure a time master node

1. Select the **Galaxy** backstage, then navigate to **Configure** and then expand the **System** option.
2. Select **Time master**. The **Time master** dialog box appears.



3. Enter the node name in the **Time master** dialog. We recommend that you use the System Management Server node as the time master.
4. Click **Save**.

Remote Desktop Services

Remote Desktop Services (RDS) is optimized to use minimal network bandwidth. When managing a remote Application Server node using the IDE, connect with a dedicated RDS node to manage the connection with the remote node.

Ensure that the remote can be pinged successfully. To do so, the DNS server must be able to resolve all remote Node Names.

Application Server

IDE does not support slow network connections between the GR and the IDE. An RDS session (from the remote

client) is the preferred method of configuring the Galaxy.

Historian Server

No special configuration is required in this context.

Security

This section contains security information specific to a SCADA environment.

Domain-level security

The System Platform network account must be configured on the domain controller and used by all local and remote component installations.

In order to maintain a central point of security administration, the following configurations are recommended:

- For Application Server, configure OS Group-based security.
- For InTouch HMI, configure ArchestrA security from WindowMaker.

These settings facilitate a centrally-administered security model within the distributed Microsoft domain.

Distributed SCADA systems will likely use more than one domain controller. Such distributed topologies join computer nodes at different sites to different domain controllers; the operators, engineers and technicians log into those domains.

For systems with remote sites that are prone to long disconnected periods (from the central network), it is important to distribute additional DNS servers and backup domain controllers at strategic points in the network.

A single System Platform network account is accessible from any domain. A galaxy can span multiple domains, but a single network account must be used for all nodes. This account can be a local account on each node, each account having the identical name and password.

Domain control and authenticated token cache expiration time are features of Microsoft security. Before changing domain parameters, refer to Microsoft documentation.

It is critical to properly configure the DNS settings for the NIC adapter to ensure that multiple domains are visible to the computer. This configuration is performed when installing the bootstrap on each node. Microsoft security with Active Directory and DNS supports invoking such "cross-domain" accounts at installation time.

Tune expiration times relating to domain control and security. For example, in a scenario where Application Server security is enabled as OS User or OS Group for the galaxy and the node is temporarily disconnected from a domain controller, logins to Operations Control Management Console, Object Viewer, and OMI/InTouch ViewApps still succeed, but for a limited time.

If a domain is out of communication for a period of time, tokens are locally cached until the configured timeout. If the operating system's default expiration time is too short for your operation, modify/extend the expiration timeout setting for cache security.

Application Server

Galaxy security (run-time) is configured via the Security dialogue of the IDE. Users and gGrous are assigned, states are created and mapped to galaxy privileges, and individual Application Objects are allocated to security

groups. As long as their membership is then authenticated against galaxy authorized groups, they will have access to the capabilities of the system.

All Application Server installations must use a common domain, user name and password for authentication, even for the case in which there are multiple domain controllers in the system. The same-named domain account must exist as a member of the local administrator's group on each node; i.e. it must be one domain, one user name and its associated password. This ensures a contiguous galaxy.

Ensure the Login Time is at its default value of 1000 ms, and not 0 (disables the login).

This setting limits the role-validation part of the login to 1 second and improves login time on an application in a SCADA system using "OS Group based security." Role-validation on a large system might otherwise take many seconds.

To change the default login time

1. Launch the IDE on the GR node.
2. Select Galaxy > Configure > Security
3. Select OS Group based and set the login time.

InTouch HMI

Use the ArchestrA security model selection within InTouch WindowMaker.

Historian Server

When historizing data, the System Platform network account used in Application Server must also exist on the Historian Server node.

Workgroup-level security

If you are installing System Platform products on more than one node, domain based networking is recommended. Domain based (client-server) networks provide better user account security and management than workgroup based (peer to peer) networks.

System Platform does not support mixed Windows workgroup/domain environments. While workgroups are supported, you cannot use workgroup nodes within a domain environment.

Note: Do not install the Galaxy Repository on a computer that is used as a domain controller or as an Active Directory server.

Operations that rely on inter-node communications may not function correctly in a workgroup based Application Server installation. Examples of this type operation include connecting to a remote IDE, or viewing the status of a remote platform.

If you must use workgroup based networking, you can avoid communications issues by enabling "everyone permissions" for anonymous users. To enable these permissions, go to:

Local Security Policy > Local Policies > Security Options > Network Access: Let everyone permissions apply to anonymous.

Or, you can enter the following command from an administrator command prompt:

```
reg add HKLM\System\CurrentControlSet\Control\Lsa /v EveryoneIncludesAnonymous /t REG_DWORD
/d 0
```

Application configuration overview

The following material includes system-wide recommendations in a System Platform environment.

Acquire and store timestamps for event data

For RTU protocols where data timestamps may be retrieved from the remote site, it is necessary that a communication driver acquire this timestamp and make it available as a parameter of the data.

Given the data's value and its timestamp, it is possible to transfer the data's value and timestamp into the Historian database or process the data with event analysis algorithms. Such data transfer and processing operations require the development of specific Application Server objects designed for this task.

Acquire and store RTU event information

In cases where RTU protocols and event information may be retrieved, particularly as structured blocks with point ID, value, timestamp, and description, a communication driver must acquire the structured information and make it available for processing.

Typically this requires special programming for the server that transfers the data to a database. In particular the data may be transferred directly into the Historian Server node.

Disaster recovery

It is important to establish a site, physically separated from the central one, that has replication capability. Doing so ensures the integrity of an operational system where the central site is at risk from fire, tornado, hurricane or other catastrophe. The replication capability includes having duplicated hardware, and requires that software configuration and key state information is periodically propagated from the central site to the recovery site.

Each disaster recovery scenario will be unique, thus it is important to consult with system integration experts regarding the design of communications equipment, hardware and the configuration of the software.

Platform and engine tuning

Scan rate and I/O count affect the performance of an AppEngine. Faster scan rates result in a lower I/O threshold. In other words, the higher scan rate, the lower the I/O count that can be supported.

Decreasing the engine scan rate reduces the amount of event data that must be transmitted for the historian (if enabled), as well as system resources, such as CPU and memory.

For information about platform and engine tuning in redundant Application Server systems, see Tuning Recommendations for Redundancy in Large Systems.

Tuning the Historian primitive in platforms and engines

- Set deadbands for parameters of data to historize.
- Increasing the deadband decreases the network traffic.
- Do not enable the Historian component in WinPlatforms and AppEngines unless they will actually be hosting objects with attributes to be historized.

- Do not enable the Historian feature in the highest level template for any object because this forces historization of every instance. Selectively apply the Historian feature to some templates and to specific instances of objects.
- Modify the Historian tuning constants which are attributes of the Engine component found in the WinPlatform and AppEngine objects.

  Note: The following attribute default values are designed for a non-intermittent network environment and are especially important in a widely-distributed, redundant system.

You can increase the StoreForwardMinDuration value to force the host computer to function in "Store" mode for a longer time. This prevents the computer from trying to re-connect to the Historian prematurely. The longer duration allows time for any network issues to resolve themselves.

For the occasions that communications with the Historian are interrupted, local storage and recovery of historical data is provided. It is important to configure the historization parameters of the AppEngine object using the IDE to accommodate the number of packets that will be transmitted over the network when data is being restored.

Be sure enough disk space is on the local node to temporarily store data until it can be transferred to the historian.

Inter-node communications

The following section considers platform communication when deployed across a widely-distributed and/or intermittent network (SCADA). A brief summary is included for context and is not intended as a recommendation, but rather as a pointer for the developer to begin tuning the communications to accommodate the needs, and mitigate the effects, of a SCADA system.

The information assumes multiple platforms are deployed on multiple nodes in a SCADA topology.

Communication summary

Communication between distributed platforms occurs at two levels: Heartbeats and messages (data change requests and replies, subscriptions, status updates/replies, etc.). Messages are handled by Message Exchange (MX) services.

Application Server monitors heartbeats and messages (sends/receives) on a regular, configurable basis. Several attributes can be used to monitor and tune the system to avoid problems in a SCADA environment; for example, heartbeats missed because of an intermittent network may cause all subscriptions to be dropped and re-initiated, saturating the network and preventing successful reconnection with remote nodes.

The actual settings depend on the particular network environment.

Tune the following attributes when implementing Redundant Platforms/Engines within a SCADA environment:

NmxSvc Attributes	Primitive	Default Value	Remarks
NMXMsgMxTimeout	WinPlatform	30,000 ms (30 seconds)	Can set at config-time and run-time if platform is Off Scan. Specifies how long engine waits for response from another egine before declaring timeout.
NetNMXHeartbeatPeriod	WinPlatform	2000 ms (2 seconds)	Can set at config-time and run-time. Specifies how frequently the NmxSvc sends heartbeats to remote Nmx services connected to it.
NetNMXHeartbeatsMissedConsecMax	WinPlatform	3	Can set at config-time and run-time. Specifies how many heartbeats are allowed to be missed before remote NmxSvc declares the connection broken.
DataNotifyFailureConsecMax	Engine	0	Determines the number consecutive Data Change Notification failures that will be allowed before the subscription is torn down by the publisher engine.

These attributes can be set to balance correct and timely error notification with a stable system performance. For example, the DataNotifyFailureConsecMax value of 0 means that the system will begin tearing down subscriptions (and rebuilding them) if a Data Change Notification failure occurs at any time. Initiating this action means that the network is then flooded with subscription messages both when tearing them down and rebuilding them.

This action may not be realistic in an environment in certain connections are sporadically intermittent.

Using NetNMXHeartbeatsMissedConsecMax and NetNMXHeartbeatPeriod together provides the total time elapsed since the last heartbeat before the connection is declared broken. The formula is:

(NetMNXHeartbeatsMissedConsecMax + 1) * NetNMXHeartbeatPeriod

Setting the values to smaller numbers should discover broken connections faster, but may also provide "false" broken connections because the Nmxsvc doesn’t get enough CPU time to process incoming messages.

Note: These attributes do not directly affect failover. They specify when Message Exchange will declare communication errors.

Note that recovery time on a distributed network or from an outside disaster is longer on a redundant system.

Note: The redundant pair must be at the same physical location; they cannot be geographically separate.

Redundancy for Application Object Server engines may be applied as needed at remote sites. The primary and

backup nodes must include individual NICs for their RMC channels and must use a simple crossover cable between them. The only impact upon network traffic will be some amount of additional packets during deployment from the central GR node to both the primary and backup nodes.

Load balancing

Load balancing is relevant only in the central supervisory setting. This is because load balancing implies moving traffic to another CPU at the same location; SCADA systems have a physically distributed architecture. In a central location, use a cluster of Application Servers to distribute processing activities.

Diagnostics

The following information is applicable within the SCADA environment. Use a cmd prompt to run the following commands:

Ping

Ping is a basic command that helps you check out the basics of your network. When pinging another machine, a sequence of special ICMP (Internet Control Message Protocol) Echo Request packets are sent. The receiving machine responds with an echo reply.

The ping program reports a number of items including: the number of milliseconds it took to get a reply to each Echo Request packet, the maximum, minimum and average round trip times, the number of dropped packets and a TTL (Time To Live) value.

- The average round trip time provides an indication of the speed of your network. In general, it's best if round-trip times are under 200 milliseconds. The maximum and minimum round-trip times give you an idea of the variance ('jitter'). When large variance is present, you may experience poor response in communications.
- The number of dropped packets may be an indication of network problems.
- The TTL value helps you find out how many routers (or "hops") the packet goes through in order to get to its destination. Every packet sent has a TTL field set to an initial number (for example 128). As the packet traverses the network, the TTL field is decremented by one, by each router. If the TTL field in successive pings is different, it could indicate that the reply packets are traveling through different routes.

Tracert

Tracert traces the path followed by a packet from one machine to another. The results of this command also provide the IP address of each router the information goes through and how long it took on each hop.

Reviewing the time between hops enables identification of slow or heavy traffic segments: If tracert is unsuccessful, you can use the command output to help determine at which intermediate router forwarding failed or was slowed. Looking for hops that have excessive times or dropped packets in the report from a tracert command can find potential trouble spots between two machines.

Time synchronization

When the network cable is reconnected, the system event viewer may contain a message that the time provider

NtpClient is currently receiving valid time data.

This message does NOT mean the computer clock time sync happens. It means the internal clock is adjusted and will act as described in the bullets above. In other words, this message is sent every time the computer is reconnected, but only in certain cases is the actual computer clock also updated to the current Server time.

In other cases, only the internal clock is adjusted and the computer time is gradually synced with server time according to the algorithm.

System integrator checklist

This section includes tasks that may be overlooked or omitted from a scope document or a bid.

The list items are compiled from integrator comments, technical support sources, and documentation. The items include internal and external links to supporting information.

Time master

The System Platform IDE contains a configuration option to use time syncronization. This ensures that the computers in a Galaxy regularly synchronize time. This is particularly important for alarm and data historization. While any System Platform node can be used as the time master, we recommend using the node that is configured as the System Management Server. See Synchronizing time across a galaxy for additional information.

Communication

Configure IP addressing

Make sure the DNS server is properly configured. All nodes in your Galaxy must be able to communicate with each other by using both IP address and Node Name, as configured in the Network Address option of the WinPlatform object.

If PCs in the Galaxy are using fixed IP addresses, then create a hosts file with the host name to IP Address mapping.

WinPlatform connection problems may result if computers cannot be accessed by both Hostname and IP address.

This is true no matter which type of Network Address you choose to use.

For example, assume two nodes in your Galaxy (host name: NodeA, IP address: 10.2.69.1; host name: NodeB, IP address: 10.2.69.2). NodeA must be able to ping NodeB with both "NodeB" and "10.2.69.2".

The reverse must also be true for NodeB pinging NodeA. Failure in either case, may result in the following: you may not be able to connect to a remote Galaxy Repository node from the IDE or deployment operations may fail.

Configure dual NICs

Use two Network Cards on a computer that hosts I/O Communication Drivers. Doing so provides load balancing and supports redundancy.

It is also good practice to place PLC communication on a dedicated (Control) network.

Security

Configure the System Management Server

The System Management Server (SMS) is used to manage secure communication across all System Platform nodes. Refer to the *System Platform Installation Guide* for configuration details.

Confirm User Name and Password

All nodes in the Galaxy must use the same user Network Account name and password. This is configured during installation.

Configure Antivirus Software

Refer to current antivirus recommendations, available on the AVEVA Global Customer Support website.

https://softwaresupportsp.aveva.com/#/searchresults?q=antivirus

Administration (Local and Remote)

Install Application Server Components

- Install the IDE and Bootstrap on any PC that will browse the galaxy. This includes WindowMaker and SCADAlarm Event Notification Software nodes.
- Install the Bootstrap and deploy a platform to any node that will be an AOS (Application Object Server) or will be doing I/O with a Galaxy (includes WindowViewer and SCADAlarm Event Notification Software nodes).

Connection Requirements for Remote IDE (from a Client Machine to a Galaxy)

1. The System Platform Network User (accessed via the Change Network Account utility) must have the same username and password on both the client machine and the GR Node.
2. The GR node must have a user account with the same username and password as the logged-in user on the client machine. This GR node user does not have to be logged in.

Migration

Verify Version and Patches

Verify that all galaxy nodes are running the same version and patch level of Application Server.

Upgrade Correctly

Follow the upgrade/migration instructions provided in the *System Platform Installation Guide* (for full releases), or in the *Patch Installation Guide* (for patch releases), as applicable. The installation guides are located on the installation media.

Historizing Data

Historical data retention in System Platform is handled by the Historian. The individual historized data points are defined as attributes of ApplicationObjects.

General Considerations

When designing a System Platform application, consider the following data storage concerns:

- Data Point Volumes: The volume is the number of points in the application that will be historized. In Application Server, data points are attributes that have history enabled.
- Data Storage Rate: At what rate will the data be changing and how quickly will that data need to be stored?
- Data Loss Prevention: What are the possible scenarios that would result in a loss of data?
- Storing System Data: In a system with multiple historians, how does changing system topology affect the location of stored data?
- Client Locations: Keep in mind the location of historization clients as you plan the network topology for your application.
- User Account: The System Platform Network account under which services run must be the same for all applications. Also, if you specify a local computer user, then the Historian Node must be in the same network domain or workgroup as the ApplicationObject Server node

Data Point Volumes

The data storage rate of a single historized attribute (i.e. point) is a function of the scan period of its hosting AppEngine and its rate of change. The attribute can be stored no faster than the AppEngine scan rate, and is only sent to the historian if the attribute changes.

For analog attributes, a deadband can also be configured to prevent storage of small changes that are not significant.

Finally, attributes can be configured to ensure that slow-changing or non-changing attributes are always stored at a minimum interval (such as once per hour) via the force storage period.

The volume of data points that must be stored is a determining factor in deciding how many historians will be required in an application. Most applications will only require one historian. Very large applications may require multiple historians.

Data Storage Rate

When configuring a data point for storage, it is possible to set a rate for that point to be stored, but keep in mind that the data cannot be stored any faster than the AppEngine scan rate. Data storage cannot occur between AppEngine scans. The data storage rate should always be whole number multiples of the AppEngine scan rate that hosts the associated object.

Data Loss Prevention

The System Platform framework protects historized data from loss. This is done by storing the data locally to disk

when connection is lost to the Historian. This operation is called "Store Forward." Each AppEngine and Platform is capable of storing all of the associated acquired historical information to disk when a connection to the Historian cannot be established. Once the connection is reestablished, the data will be sent to the Historian. It is important to ensure that the AppEngine responsible for sending the data to the historian is not interrupted. To prevent data loss:

- Separate the Galaxy Repository from running AppEngines. As applications grow large, the configuration changes to that application will become CPU-intensive. It may be possible to have the configuration services on a computer starve the AppEngine from the required CPU cycles to perform the data storage. To prevent this, place the Galaxy Repository on a remote computer from running AppEngines.
- Do not deploy to running AppEngines. If an AppEngine is running and gathering information for the Historian, deploying additional objects to the AppEngine will cause a momentary interruption of the AppEngine execution. During that time incoming data changes may be missed.

  This effect can be prevented by only deploying new objects during non-critical data storage periods. Deploying large numbers of objects can have a large effect on system resources.

Area and Data Storage Relocation

The System topology will evolve throughout the application's life cycle. As this evolution takes place, you must determine the data storage implications. The System Platform framework designed to be extremely flexible. It is very easy to move an entire area of objects from one AppEngine to another. But it is important to remember that the AppEngine defines where the historical data for hosted objects will be sent. If using multiple Historians, and the topology is modified, there may be an impact on the location of stored data.

When designing the system topology, note that the locations of the historical data clients may impact the end design. This is particularly true when portions of the application are separated by low bandwidth or intermittent network connectivity. Client applications should not be required to access the historian over these poor connections. One solution to this is to have local historians that service the computers that are locally situated with good network connections.

Non-Historian Data Storage Considerations

As a system is put into service, it is normal to maintain the Historian node to ensure enough space is available for continued data storage. This is a requirement for any historian. However, the Historian is not the only storage mechanism that is used in System Platform.

Nodes other than the Historian Node are capable of storing large amounts of information, so it is important to assess the impact of the following settings on data storage:

- Alarm Buffer Size: If the network connection to the alarm database is lost, the alarms will begin to be stored in a local buffer. This buffer is a direct reflection of the page file size. An average alarm record is 1400 bytes of data. If the buffer fills up, storage will stop. However, a 10 MB page file can store over 3500 alarms, so using the proper precaution can easily prevent an issue.
- Log Viewer Event Storage: By default, the Log Viewer event storage mechanism (which is installed on all computers) is set to use a maximum of 5 GB of storage. You can adjust this value. The Log Viewer event storage must be considered in the total disk space requirements.
- Store-and-Forward Deletion Threshold: The amount of free space that is reserved on the local HCAL store-and-forward disk. This reserved space is used for storing the historical data until the network connection is

restored, if the network connection to the Historian is lost. The default circular deletion threshold is 100 MB. You should consider your requirements for this setting, and adjust it, if necessary, in the WinPlatform or the AppEngine object configuration.

When evaluating system configuration, it is worth spending a little time up front to consider disk space availability.

Implementing Alarms and Events

The alarm and event subsystem consists of both alarm consumers and alarm providers. When determining the topology for your application, be aware of how alarm and event messages are processed within the system and how different configurations can affect system performance.

Note: The event messages produced by alarm providers are not the same as events generated by the Historian system.

Determining the Alarm Topology

When you are determining the alarm topology, it is important to take into consideration the overall topology of the system. Alarming can be implemented using Distributed Local Network or Client/Server topologies.

Best Practice

Parent alarm areas must be on the same node as their subareas. Alarms are aggregated by the area object.

Alarming in a Distributed Local Network Topology

In a distributed local network topology, the nodes that serve data (Application Object Servers) are not separated from the clients that consume data (Visualization nodes). That is, these workstation nodes combine Application Object Server functionality with Visualization node functionality, and each node hosts both components locally. A platform is deployed to each workstation node,.

It is more likely that every platform in this topology is configured as an alarm provider, and each of the alarm consumers queries the local platforms for alarms.

Consider the scope of interest of each platform. when configured as an alarm provider, the platform requests all alarms in the galaxy by default. If a workstation does not need to view all of the alarms in the galaxy, the platform on that computer should be configured to only subscribe to alarms that are within the scope of interest.

Best Practice

The following list summarizes the key points in setting up an optimized alarm distribution system in a distributed local network topology:

- All of the platforms on workstations will be alarm providers.
- If operators at a workstation will need to view all alarms in an application, you use the default scope for the platform alarm provider on that node, which is to subscribe to all alarms in the galaxy.
- If operators at a workstation do not need to view all alarms in the galaxy, configure the platform alarm

provider scope of that node to subscribe only to alarms that are of interest to the operators at that node.

Alarming in a Client/Server Topology

The client/server topology separates nodes that serve data (Application Object Server nodes) and the clients that consume data (visualization nodes). There more clients than servers.

A platform object must be deployed on each client and each server. One or more of the platforms on the Application Object Server should be set as an alarm provider and each alarm consumer should query one of the Application Object Server platforms directly. This deployment minimizes the network traffic by channeling the alarm traffic to specific alarm providers.

If the platforms on the visualization node(s) are set as alarm providers, each of those platforms requests all alarms continuously, loading the network with unnecessary traffic. While a platform can be configured to subscribe only to alarms of particular areas, the platform still requests the alarms for the configured areas on a continual basis.

By configuring the Application Object Server platforms as alarm providers, only one node requests alarm updates. The visualization node(s) only request alarms when a window containing the alarm display is active. Alarm consumers only request the alarms that are required to satisfy the alarm query.

As stated previously, each platform is capable of providing all alarms in the galaxy. However, if all of the consumers are using a single platform as the sole alarm provider in the galaxy, there is a single point of failure for all alarm consumers. Also, the single platform would constantly be receiving the alarms from all of the other Application Object Servers, which would cause a heavy traffic load on the network.

If your client/server architecture consists of more than one Application Object Server node, take the following measures to ensure the highest availability of alarm information to the alarm consumers:

1. Configure the alarm consumer queries to query each Application Object Server platform for the areas that are hosted on that platform.
2. Configure the Application Object Server platform alarm providers to provide only alarms for the areas hosted by that platform.

These two configurations lower network traffic between Application Object Servers due to alarm distribution and ensure that no one Application Object Server is a single point of failure for alarm delivery to the consumers on the visualization nodes.

Best Practice

The following list summarizes the key points for setting up an optimized alarm distribution system in a client/server architecture. The list also applies to a widely-distributed SCADA system environment:

- The platforms on the visualization nodes should not be alarm providers.
- The alarm consumers on the visualization nodes should query each Application Object Server individually for the areas hosted by that platform.
- The Application Object Server Platform Alarm Providers should be configured to only be providers for the Areas that are hosted by that Platform.

Configuring InTouch HMI Alarm Queries

For an alarm consumer to obtain the alarms from an alarm provider, it must query the alarm provider. A typical alarm query is configured as follows:

```
\\ProviderNodeName\Provider!AlarmGroup
```

For an InTouch application these translate as follows:

- ProviderNodeName - This is the host name of the node where the alarm provider resides.
- Provider - This is the word "Galaxy." There can only be one WinPlatform per computer, and this keyword represents the platform alarm provider.
- AlarmGroup - The area objects in the IDE serve as the alarm groups. When building the application in the Model View of the IDE, you can nest the areas within each other. If an area named "Tanks" hosts another area named "Clearwell," then subscribing to the alarms in "Tanks" will automatically include the alarms in "Clearwell."
- Multiple Queries - An alarm consumer query can be used to query multiple alarm providers by adding a space between the individual query strings.

Templates

A template object represents common functional requirements of a field device (valves, pumps), a group of field devices (skids, stations), or a user function (algorithms). These requirements reflect information such as number of Inputs and Outputs, alarm conditions, history needs, and security.

An Object Wizard can be added to any derived template, and provides a simple choice-drive interface for configuring instances (assets) from the template. Object wizards allow a single template to provide the basis for a variety similar objects, such as single speed vs multi-speed motors or 2,- 3-, or 4-way valves, without requiring a template for each object subtype.

A template-centric development practice that leverages object wizards enables re-use of existing engineering and allows you to implement standards at both the enterprise and local levels.

Before Creating Templates

Before building templates, identify and document the functional requirements of the target field device.

- Identify all required field device properties (attributes). They include names, data types, and interaction requirements (that is, none, input, output, or input/output). For each attribute, determine if:
  - The attribute requires scaling or uses raw values.
  - The attribute requires alarms, and the alarming model to be used by each. This model can include where the alarms will be generated (locally or in the control system), any alarm priority assignment, and alarm messaging needs.
  - The attribute requires security and the security control.
  - The attribute requires historical logging. For example, is forced data storage required? For a variable data type, do you need to define the trend limits and a deadband?
- Identify any required scripting, such as algorithms, interaction between devices, and so on.
- Determine if field devices are grouped either into a common template or into a containment template model.

It is not necessary to know all requirements before building a template model. Extra functionality can be easily implemented in the template when new requirements are determined.

Creating a Template Model

After generating and documenting field device requirements, decide on a template model that fits those requirements. Begin by reviewing the field devices and their requirements while looking for commonality across similar field device types. Determining this commonality is the basis for developing the template model. After you develop the template model, you can derive instances (run-time objects) from the templates. A template exists only as configuration-time object.

In most cases, you can use the $UserDefined template as the logical foundation for the device type. Other templates can also be used. For example, valves, pumps, and motors that have multiple states based on discrete

limit switches can use the $DiscreteDevice base template. Process variable transmitters and controllers can use the $AnalogDevice base template.

The IDE includes a set of master templates in the **_Default Templates** folder as defaults. While these are writeable and can be configured, best practice is to create a derived template from these master copies, and configure the derived copy. A set of base templates, from which the master templates are derived is also provided. Base templates are read-only, and are located in the **System** folder. Never create instances directly from base templates, since you will not be able to take advantage of advanced configuration and maintenance capabilities.

Each template object contains three or more tabs when you open the template in the IDE Object Editor.

- The **Object Information** tab contains basic configuration and derivation information, object execution order, and a link to add a custom help file.
- The **Attributes** tab allows you to add unique, user-defined attributes. You can also configure an object wizard in the **Attributes** tab and link to graphics, scripts, and other content types, such as OMI layouts.
- The **Scripts** tab allows you add and configure scripts of different trigger types (Startup, Onscan, Offscan, Execute, and Shutdown) for the object.

Use the derived templates you build to create sets run-time instances. The derived templates become the basis for all other instances. This derivation practice is called containment. Note that changes made at the template level propagate to all objects that are derived from it, including instances and child templates.

Containment vs. Attributes

Using containment or attributes as the best configuration approach depends on actual field device requirements.

For example, template containment works best when the lower level object also has many components and may contain even lower-level objects.

Similar functionality can be achieved using attributes. However, when the lowest level object is added to a template, it may be done using either template containment or attributes. Both support an external I/O point link and history.

Use the following general guidelines to expedite an appropriate development strategy:

- Use template containment when more functionality is required, such as complex alarms, setpoints, I/O points, or other features readily available in a template.
- Use attributes when the lower-level object is very basic.
- Use attributes for memory or calculated values.

Decide the appropriate approach in advance of implementation.

Best Practice

- Ensure the container incorporates functionality. Otherwise, place it as an attribute in another object. Do not use excessive empty containers simply as placeholders to host objects. Empty containers impact engine scan resources and time.
- Up to 5,000 attributes can be added to an object. If you add a description to the object, however, this number is cut in half owing to the fundamental structure of objects.

- Deeply-nested template/container structures slow down object check-in and propagation.
- Complex objects should be built using attributes, scripts and containment.

  The form supported by Application Server is a .NET Library and can be created using Visual Studio .NET with either Visual Basic .NET or Visual C# .NET.

  Note: The Application Object Toolkit can be used to create custom templates.

Base Template Functional Summary

The following information describes each base template object and recommendations for use.

$UserDefined Template

The $UserDefined object provides an empty starting point for creating custom built objects that include attributes, scripts, and feature extensions, such as alarms, historization, and I/O. It is the most flexible and most commonly-used template for modeling your assets. Unlike other asset object templates, the UserDefined template contain only the basic editor tabs (Attributes, Scripts, and Object Information), while other object templates contain custom tabs that are applicable only to that object type.

The $UserDefined template should be used in most cases as the basis for creating the objects needed to model the physical plant environment.

$AnalogDevice Template

The $AnalogDevice Template object can be used to model more complex analog inputs and control loops.

The General tab of the Object Editor includes a field for setting the type of analog device. The Analog option type enables configuring a Process Variable (PV) input source and (optionally) a different output destination. The PV can be scaled, multiple alarm points defined, and history collected for the PV. The Analog regulator option type allows for a PV input (no separate output), a setpoint, an optional different setpoint feedback address, setpoint high and low limits, and optional control tracking. It also supports scaling, alarms, and history.

$DiscreteDevice Template

The $DiscreteDevice Template object can be used to model instruments or equipment that have two or more discrete states.

The Process Variable (PV) attribute of the discrete device is a string representing the state. This can also be read as an enumerated integer value. The object supports up to five distinct states based on one- to four inputs. Up to six discrete outputs are available from the template.

The Passive state is provided to represent the state when the field device is not energized. For example, a valve that fails to the closed state when it loses power would have a passive state of Closed. A valve that requires power to command it to open and to close may only use the two active states and not have a passive state.

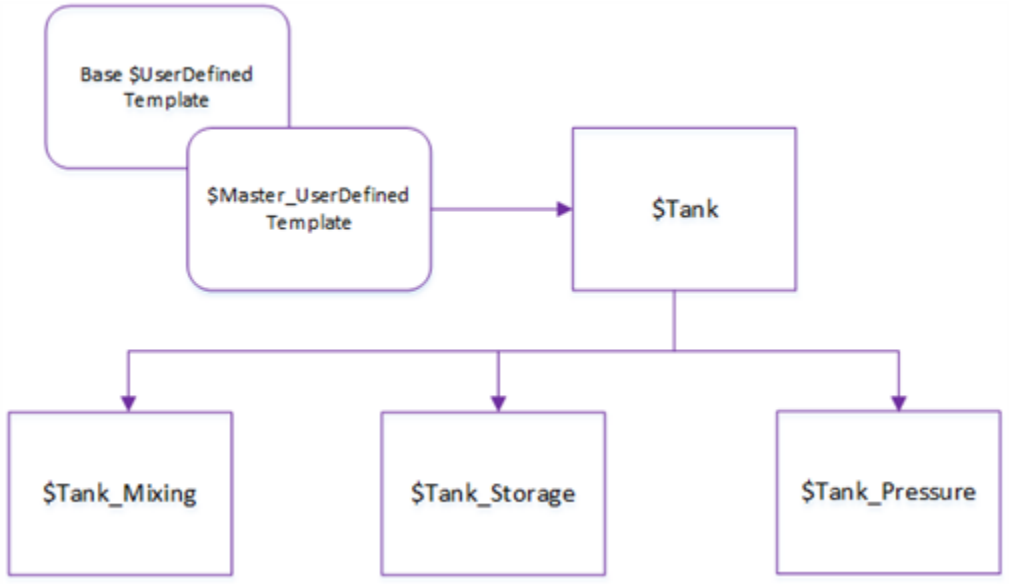
Template Modeling Examples

The following information describes two template modeling examples:

Example 1

This example describes the model of a process that contains thee types of tanks. The types are determined from device requirements. To optimize engineering development, a common derived template called $Tank that utilizes an object wizard is developed from the $UserDefined base template. The contains all device requirements and multiple configurations of tanks can be built from the template by selecting different object wizard options. If some tank configurations cannot be easily accommodated by the $Tank template, additional templates can be derived from it.

The following figure illustrates how these templates can be developed:



A base tank template contains the fields and settings common to all tanks within the facility. A new template is then derived from the base template for category of tank and contains settings that are specific to that type.

Object instances are then derived from the applicable template to represent the actual tanks.

Example 2

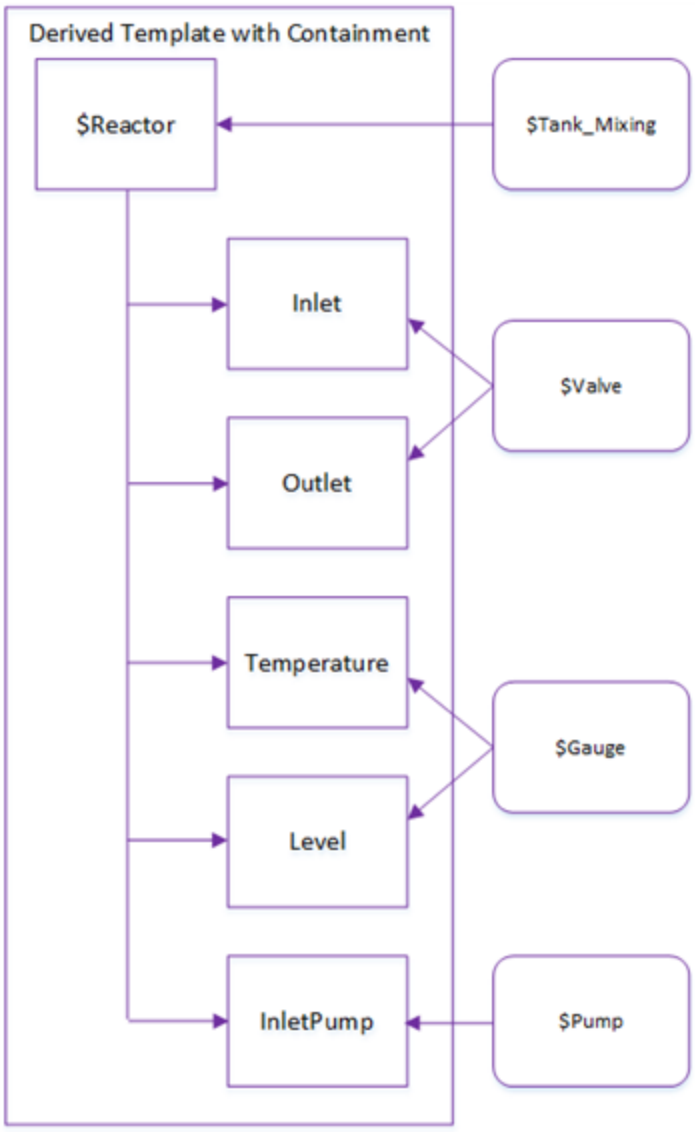
This example describes a common, complex relationship called Reactor. Reactor is based upon an interaction of five field devices. Multiple instances of this relationship are used within the plant model.

The relationship can easily be developed using containment.

Create a derived template called $Reactor from the $Tank_Mixing template shown in the previous example.

Then, create template instances representing each of the five field devices from a derived template (again, based on the $UserDefined template). The complex relationship can now be developed (using scripting) in the container object ($Reactor) using the hierarchical names given to each field device. When instances of $Reactor are created as field devices, each has two names: the containment name (hierarchical name); and the physical name.

The following figure illustrates this practice:



Using Attributes and Features

Attributes enable data type additions to the template or instances. Attributes can be further enhanced by enabling features. Features extend the functionality of an attribute by adding input, output, history, and/or alarm characteristics.

Attributes are categorized as follows:

- Calculated: The attribute is only modifiable by the instance. It will have no initial value until the object writes to it. Calculated attributes are typically used for totals, averages, and so on.
- Object Writable: The attribute is writable only by instances within the Galaxy.
- User Writable: At run time, the attribute is writable by a user (subject to security restrictions), other instances, and the configuration program.

  Note: Locking an attribute makes it a constant. Constants are not writable at run-time.

Best Practice

- When defining attributes with input or output extensions, never lock their source or destination within a template, since it will be unique to each instance and defined later.
- Use three dashes (---) to represent an unknown reference. This prevents the "could not resolve reference" warning when instances are created.
- Arrayed attributes cannot be extended.
- For a Boolean or Analog alarm that comes from the field device control logic and requires a corresponding Acknowledge, the "Acked" attribute should take on an output extension with the destination being the "Ack" point in the control logic.
- If an underscore (_) is the first character of an attribute name, that attribute is hidden from users at run time. Use this hidden attribute when you need variables to support certain functionality, but want to hide them from users in order to prevent confusion. A hidden attribute cannot be extended.
- Any attributes and its extension created within a template is inherited by all derived objects. However, attributes and extensions are only propagated when the instance is created or when that particular attribute is locked.
- Most attributes are checkpointed. This means that all data necessary to support automatic restart of a running Application Object is saved periodically. The restarted object has the same configuration, state, and associated data as the last checkpointed value.

  Unlike other attributes, calculated attributes are not generally checkpointed unless they are configured as "Retentive Enabled," in which case they will be checkpointed with the other attributes in that object.

Deriving Templates and Instances

The following information describes effective implementation practices for template and instance derivation:

Best Practice

- Derive a new template from the Master Template before deriving any instances. The IDE contains a folder called **Default Templates** that contains a set of master templates. Each of these are prefixed with "$Master_." Derive your templates from these master templates. You can create and move templates to any folder. ; your templates can exist in whichever folder you prefer.

  If you modify one of the Master Templates, you can recreate it by deriving a new Master Template from a Base Template. These are located in the System folder. Base Templates are read-only.

  When a template is derived from another template, the derived template inherits all of the characteristics of the parent template. If the parent template is modified, only the attributes and extensions that are locked will propagate to child templates. However, if you use object wizards, all changes to the object wizard propagate; there is no need to manually lock attributes.
- Propagate Security: Changes made to the security control of any attribute or extension will not propagate. If new attributes, scripts, and extension are added, they always will propagate. The derived template can then take on additional functionality.

  When an instance is derived from a template, changes made to the security control of an attribute will propagate, but changes made to the security for an extension will not propagate. If you deploy instances of a

template and then modify the template, you will then need to re-deploy the instances. Before deploying changes, you may want to perform an upload of run\-time changes. The upload overwrites the initial attribute values with current run-time data.

- Lock Attributes: Use Locking to propagate functionality when modifying a template. You can then unlock the attributes and extensions when the propagation is complete. There is no need to lock attributes if you are using object wizards.
- Name the Object: When an instance of an object is first created, it is given a default name based on the parent template name and an incremental number (_XXX).

  The name should be changed to meet the naming convention established for the project. If the instance is contained by another object instance, it will also have a hierarchical name. There are certain naming restrictions

Note: The instance name, and not the hierarchical name, is used by the Historian to store historical data.

It is important to properly name the object before it is deployed. The object hierarchy can be up to 10 levels deep for a maximum hierarchical name of 329 characters.

Object Name Limitations

Object names must be unique within each namespace, not within the Galaxy.

- Template names can be up to 32 alphanumeric characters, including the required $ as the first character. The second character cannot be $ and the name must include at least one letter. You cannot use spaces in an object name.
- Instance names can be up to 32 alphanumeric characters. You cannot use $ as the first character. The name must include at least one letter. You cannot use spaces.
- The object name extension "_New" cannot be used if another object in the Galaxy uses the same name without the "_New" extension. For example, if you have an object named "Pump," you cannot have a second object named "Pump_New." You could, however, use "Pump_Old" and "Pump_New."

  Note that this restriction applies to templates as well as instances. For example, you cannot have a derived template named "$Area_New" or "$ViewApp_New" since these add the "_New" extension to a base template name.
- The following names are reserved and cannot be used for objects:
  - Me
  - MyArea
  - MyContainer
  - MyEngine
  - MyHost
  - MyPlatform
  - System

Re-Using Templates in Different Galaxies

The following recommendations apply in an environment where separate, unique galaxies exist at different

production sites, and where standardization is required across the sites.

Templates can be reused in different galaxies (just as they can be reused in the same galaxy) to create multiple instances. The important difference in this scenario is that change propagation in a single galaxy is a simple process where locked features are automatically propagated to instances, whereas a multi-galaxy environment requires formal procedures be defined and implemented to manually update and maintain templates.

The following recommendations describe performing updates/synchronization in a structured and repeatable manner:

Best Practice

- Create a Master Galaxy for development purposes. This Galaxy contains the master template library with the latest revisions for all of them. When new galaxies must use any of these templates, ensure they include the latest revisions.
- Create a "staging" engine for testing purposes. Whenever possible, deploy this engine on a machine that has no impact on a real production system. Once an object design has been tested and its functionality verified, it can be distributed to other sites.
- Export .aaPkg packages (cab files) that contain the necessary templates out of the Master Galaxy, then import them into the new production Galaxies.
- Create local templates that derive from the master at each production galaxy. Any changes or specialization required should be implemented in the local template. Make use of toolsets to separate the master templates from the local templates; it is even recommended to hide the master template toolset in the production galaxies and treat them as if they were Read Only templates.
- Packages with exported templates do not include any logs documenting changes to the base templates, so all changes must be done in the master galaxy and properly documented upon check-in after editing objects.
- The Import Preferences dialog box (opened when importing a package with templates) includes options to handle version mismatches and name conflicts. In a multi-management environment where a package from the Master Galaxy is updating templates that already exist in production galaxies, select the Overwrite option to handle version mismatch.

  The Overwrite option will only work if the version of the object being imported is higher than the current version. The object version is stored in the ConfigVersion attribute that is present in all templates and instances.
- Changes made on the local templates should be verified and validated before they are merged into the master template on the development Galaxy.

  If it is determined that the features in local templates should be implemented in the master templates, any new functionality (i.e. attributes or scripts) must be manually implemented on the master templates and properly documented.

  After manual implementation and documenting, a new package can be exported in order to update all production galaxies with the new version of the templates.

Export/Import Templates and Instances

Application Server provides export and import functions that let you transfer templates and instances between Galaxies.

Export Automation Objects

When you export objects, the following are saved:

- The selected instances and templates.
- The templates from which the instances and templates were derived.
- The toolset used to display the templates. An import simply imports the contents of the export file.

Galaxy Dump

A Galaxy Dump exports template instances to a .csv file for editing or for adding an instance of a template. Modifications can be loaded back to the Galaxy. To perform a galaxy dump, select the objects you wish to export, the choose "Selected as CSV" from the Export Objects screen.

Note: You cannot perform a Galaxy Dump if the Galaxy object is selected.

When a dump is performed, any script, attribute, or attribute extension that is not locked at the template level will be dumped, each in its own column. A reference to the parent template is also contained in the file, in order to bring in all of the locked scripts, attributes and extensions. Attributes that are calculated or writable at run time are not dumped.

The dump and load functions are useful for quickly creating multiple instances of a template, instead of using the IDE.

To prepare a file for a Galaxy Load operation

1. Create one instance of the required template and dump this into a .csv file.
2. Open the .csv file using a text editor. Perform a search and replace to change all occurrences of object instance names as needed.

   Note: Microsoft Excel is not recommended for use in editing exported .csv files. Excel does not recognize the file as a .csv formatted file and all the data is displayed in the first column.
3. If using Excel, select the first column and "Convert Columns To Text" using delimited and comma as parameters. This places each column from the the .csv file into a different column.

   WARNING! NEVER click the Save button in the toolbar or the Save option in the File menu. Excel will save the spreadsheet as a .xlsx file and destroy the formatting conversions. See the following step for saving to a .csv format. Also, Excel may add extra commas to the .csv file, which will need to be deleted.
4. After modifying the file, select Save As from the File menu and make sure that the File Type is .csv.

   The file now has the valid format to successfully apply a Galaxy Load operation from within the IDE.

   In the following example, five additional instances were created from the $Boolean template. For three of the derived instances, the Area is not known, and for three other instances, the Area is **HomeArea**:

```
;Created on: 6/10/2023 2:01:12 PM from Galaxy:Test
:TEMPLATE=$VSD
:Tagname Area, Area
VSD1
VSD2
VSD3
VSD4, HomeArea
VSD5, HomeArea
VSD6, HomeArea
```

5. Save the changes to the .csv file.
6. Load the .csv file into the Galaxy.

For this example, the following events occur when the load is performed:

- The first three instances are created with all template functionality. If no Area is set as the default, the instances appear in the Unassigned folder in the IDE.
- The next three instances are placed in the area "HomeArea" (if HomeArea already exists. Otherwise, the objects use the Area settings for the first three instances).

The advantages of using the Galaxy Dump and Load over creating instances within the IDE are evident when conforming to a naming strategy.

For example, when a contained object has three levels and hundreds of instances, it is much easier to perform a search and replace operation to rename all the instances instead of naming each instance individually within the IDE.

Best Practice

When backing up the Galaxy database, use the Backup functionality available within the Galaxy Database Manager of the Operations Control Management Console. The backup contains all Galaxy information (including security configuration), whereas the simple export of application objects only includes the object structure and template toolsets.

Scripting at the Template Level

Add additional functionality to a template by creating scripts. Scripts are written using the QuickScript .NET language. Refer to the *Application Server Scripting Guide* for detailed information about scripting.

Best Practice

The following best practice recommendations are cross-referenced to practical examples described in the following chapter.

- Segregate functionality by creating unique scripts for each segment required.
- When functionality within the script will require an extended amount of time to execute, set the script to run asynchronously with a timeout limit. Such functionality is sometimes required by COM objects and .NET objects (for example, file operations, SQL queries, and so on).
- When creating scripts at the template level, you may want to lock them. You can then make changes to the template script, and the changes will propagate to the next level. When a script is locked and there are no declarations or aliases, these sections should also be locked for improved propagation and deployment performance.
- When the field devices have a structured containment and addressing convention, create a script that will populate all attributes with an I/O extension at the time of deployment. If instances have been configured using this method, you can use the "Upload Runtime Changes" functionality to synchronize the changes back to the Galaxy Repository.

- When adding scripts to templates, use relative names (Me, MyContainer, MyEngine, MyArea, MyPlatform, and MyHost). Using relative names saves you from having to edit absolute references in every instance.
- Lock all inherited scripts, otherwise a copy of each assembly/script will be created for each object derived from the template and multiple copies of the same script will be running on the AppEngine object where they are deployed.

  Locking the scripts generates one single assembly to be deployed to the AppEngine. The assembly is then shared by all instances derived from that template.

Note: For more information on scripting, methods, and practical examples, refer to the *Application Server Scripting Guide*.

Script Execution Types

The script execution types and when they should be used are as follows:

- Startup: Called when the object is loaded into memory. Primarily used to instantiate COM objects and .NET objects.
- OnScan: Called the first time an AppEngine calls this object to execute after the object scan state is changed to onscan. Primarily used to initiate attribute values.
- Execute: Called every time the AppEngine performs a scan and the object is onscan. Supports conditional trigger types of On True, On False, While True, While False, Periodic, and Data Change. Most run-time functionality is added here.
- OffScan: Called when the object is taken offscan. Primarily used to clean up the object and account for any needs that should be addressed as a result of the object no longer executing.
- Shutdown: Called when the object is about to be taken out of memory, usually as a result of the AppEngine stopping. Primarily used to destroy COM objects and .NET objects and clean up memory.

Determining Object and Script Execution Order

Application Server enables control of the object scan order within an Engine, and control of the script execution order within an object. Use this functionality instead of "data handshake bits" to ensure the delivery order of data from script to script and from object to object.

Before considering script execution order, it is necessary to review how objects execute. The following information reviews object execution events at a basic level.

Note: For details on object execution, see the object's help files.

AppEngine Execution

The AppEngine is the only engine that hosts more than one object. Object execution is handled by the scheduler primitive, which is single-threaded. It executes objects registered on the host engine repeatedly, and in a sequential order, during the scan interval.

The scan interval is the desired rate of execution of each Automation Object the AppEngine hosts. The following tasks execute engine-to-engine in the following order during the scan interval:

Execution Phase: Individual OnScan objects execute their functionality according to there configuration (defined at Config Time).

- Output Processing Phase: All pending output requests (SetAttributes, subscription packets, publish notifications) must be sent. Pending requests intended for the same engine should be sent as one block request so that the receiving engine can process them atomically (and in order).
- Checkpoint Snapshot Phase: This task is configured separately and may not occur during every scan interval. If a checkpoint occurs within a given scan interval, it occurs immediately after the Output Processing Phase. The checkpointer status is checked to see if it is still busy from a previous checkpoint. If not, a new asynchrounous checkpoint is initiated with a checkpoint snapshot.
- Input Processing Phase: The goal of this phase is to process all input requests (SetAttributes, subscription packets, publish notifications). Input requests are retrieved one at a time. If any input requests are left in the queue, they are processed during the idle period before the next scan interval. At least one queued input request is processed following the Output Processing Phase, and before the Execution Phase.

Scan Overruns are a boolean condition that becomes true when the Execution Phase crosses from one scan interval to the next. When a Scan Overrun occurs, a new Execution Phase is delayed until the next scan interval begins. Any of the phases in the above list can cause a Scan Overrun when they extend beyond the scan interval.

All objects deployed on the AppEngine are processed in the following order: DIObjects (multiple DIObjects are processed alphabetically by tagname), hosted ApplicationObjects, then their Areas (numerically).

Common Object Execution Order

The execution order of object instances running on an engine is configured within the Object Information tab of the object editor. As an engine executes its scan, it will process the objects in the order specified.

Each named script within an object can be specified to run as either just after inputs or just before outputs. The order in which the scripts are listed in either category is the order in which the scripts will be executed.

Each object executes its functionality in the following order:

1. Read inputs.
2. Execute "just after inputs" scripts.
3. Execute object native functionality (the UserDefined object has none).
4. Execute "just before outputs" scripts.
5. Write outputs.
6. Test alarms.

Each script is executed in its entirety before the next script is executed.

This behavior is different from InTouch, where a script can trigger another script, such as a data change script. Within InTouch, the calling script halts while the data change script is run.

Within Application Server, each script completes before the next script is run. If a user-defined attribute of a second object instance is set during the execution of the script, and that attribute triggers a script on the second object, the script of the second object may or may not run during the same scan of the engine. If the second object is configured to run after the first object, the script on the second object will run during the same scan.

If the second object has already been serviced during the scan, the script on the second object will run during the following scan.

Since the Engine manages each object, a script runs only as fast as the engine's scan period or some multiple of that period. If the engine's scan period is one second, and an object script is set to periodic for every 1.5 seconds, the script will run every other scan (that is, every two seconds).

Data requested or sent to objects residing on another engine/platform are updated on the next scan. This is also true for Application Objects on the same AppEngine if an Application Object needs data from another object but it executes before it on the scan.

For example, when Object A executes, it needs the output values from Object B.

The values received are from the previous scan, because Object B has not executed yet in the current scan. You must wait one scan if you want to verify a write of this type. Alternatively, you can change the execution order in the Object Editor so that Object B executes first.

Asynchronous Scripts

An asynchronous script runs in a separate thread and is not directly tied into the engine's scan process. Therefore, reading and writing to any object attributes (including the calling object) is a slow process.

An asynchronous script should not Read or Write within a long FOR-NEXT loop to an attribute or other external source. Since the asynchronous script runs in a separate thread, it must wait until the next scan of the engine for all the Read or Write transactions to occur. If the scripts have not all been completed at the next scan, the system must wait for another scan.

A single-system test with an engine scan period of one second achieved approximately 70 attribute writes per second and 35 attribute reads per second.

Asynchronous Timeout Limit Settings

The asynchronous timeout limit must also be set appropriately. If the script times out, the script is halted in an indeterminate state. There is no mechanism for determining what line the script was executing when it was halted.

Therefore, another script should be checking for asynchronous script time outs and cleaning up any remaining inconsistencies.

Script Editing Styles and Syntax

Application Server supports two types of scripts:

- Simple scripts can perform assignments, comparisons, simple math functions, and similar actions. Simple scripts are described in this section.
- Complex scripts can perform logical operations using conditional branching with IF-THEN-ELSE type control structures. For more information about complex control structures, see "QuickScript.NET Control Structures" in the Industrial Graphic Editor help.

Both single and multi-line comments are supported. Single-line comments start with a single quotation mark (’) at the beginning the line but do not require a single quotation mark at the end of the line. Multi-line comments are enclosed within brace characters ({ }), and can span multiple lines.

White space rules apply for space and indention. Indent using spaces, or the TAB key. Individual statements are indicated by a semicolon marking the end of the statement.

Required Syntax for Expressions and Scripts

The syntax in scripts is similar to the algebraic syntax of a calculator. Most statements are presented using the following form:

```
a = (b - c) / (2 + x) * xyz;
```

This statement places the value of the expression to the right of the equal sign (=) in the variable location named "a."

- A single entity must appear to the left of the assignment operator =.
- The operands in an expression can be constants or variables.
- Statements must end with a semicolon (;).

Entities can be concatenated by using the plus (+) operator. For example, if a data change script such as the one below is created, each time the value of "Number" changes, the indirect entity "Setpoint" changes accordingly:

```
Number=1;
Setpoint = "Setpoint" + Text(Number, "#");
```

Where the result is "`Setpoint1`."

Simple scripts

Simple scripts implement logic such as assignments, math, and functions. An example of this type of scripting is:

```
React_temp = 150;
ResultTag = (Sample1 + Sample2)/2;
{this is a comment}
```

Script Execution Types

This section describes the script execution types supported by Application Server and OMI.

- Startup Scripts
- OnScan Scripts
- Execute Scripts
- OffScan Scripts
- Shutdown Scripts
- Deployment Scripts

Scripting Redundant AppEngines

There are certain considerations that you must take into account when writing a script that will run on redundant AppEngines. This section outlines whether or not a script will run under various scenarios, including deploy, forced failover, system failure, system startup, and undeploy operations. Redundant engines can be set to run in either **Legacy Mode** or **Run Warm Mode**. The selected mode may change the circumstances under which

a script will execute.

Run Warm Mode provides much faster failover performance, but there are internal differences between the redundancy modes in how the engines start and stop. Therefore, Startup and Shutdown scripts for redundant engines may operate differently, depending on which redundancy mode is selected.

Note: New redundant engines default to Run Warm Mode. Redundant engines in migrated galaxies default to Legacy Mode.

The following tables summarize the circumstances under which each script execution type runs when redundancy is set to **Legacy Mode** (differences between the two modes are highlighted):

Legacy Mode	**Primary Engine (on Server 1)**		**Executes on Server 1**				
Action	**Initial State**	**End State**	**Startup**	**OnScan**	**Execute**	**OffScan**	**Shutdown**
Deploy	Down	Active OnScan	Y	Y	Y	N	N
Forced Failover	Active OnScan	Standby	N	N	N	Y	Y
Server 1 Failure (hard shutdown)	Active OnScan	Down	N	N	N	N	N
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	N	N	Y	N	N
Graceful shutdown of Server 1	Active OnScan	Down	N	N	N	Y	Y
Graceful shutdown of Server 2	Active OnScan	Active OnScan	N	N	Y	N	N
Start Server 1	Down (was OnScan)	Active OnScan	Y	Y	Y	N	N
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	N	N	Y	N	N
Undeploy	Active OnScan	Down	N	N	N	Y	Y

Legacy Mode	Backup Engine (on Server 2)		Executes on Server 2				
Action	**Initial State**	**End State**	**Startup**	**OnScan**	**Execute**	**OffScan**	**Shutdown**
Deploy	Down	Standby	N	N	N	N	N
Forced Failover	Standby	Active OnScan	Y	Y	Y	N	N
Server 1 Failure (hard shutdown)	Standby	Active OnScan	Y	Y	Y	N	N
Server 2 Failure (hard shutdown)	Standby	Down	N	N	N	N	N
Graceful shutdown of Server 1	Standby	Active Offscan	Y	N	N	N	N
Graceful shutdown of Server 2	Standby	Down	N	N	N	N	N
Start Server 1	Down	Down	N	N	N	N	N
Start Server 2 (Server 1 running)	Down	Standby	N	N	N	N	N
Undeploy	Standby	Down	N	N	N	N	N

The following tables summarize the circumstances under which each script execution type runs when redundancy is set to **Run Warm Mode** (differences between the two modes are highlighted):

Run Warm Mode	Primary Engine (on Server 1)		Executes on Server 1				
Action	**Initial State**	**End State**	**Startup**	**OnScan**	**Execute**	**OffScan**	**Shutdown**
Deploy	Down	Active OnScan	Y	Y	Y	N	N
Forced Failover	Active OnScan	Standby	Y	N	N	Y	Y

Server 1 Failure	Active OnScan	Down	N	N	N	N	N
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	N	N	Y	N	N
Graceful shutdown of Server 1	Active OnScan	Down	N	N	N	Y	Y
Graceful shutdown of Server 2	Active OnScan	Active OnScan	N	N	Y	N	N
Start Server 1	Down (previously OnScan)	Active OnScan	Y	Y	Y	N	N
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	N	N	Y	N	N
Undeploy	Active OnScan	Down	N	N	N	Y	Y

Run Warm Mode	**Backup Engine (on Server 2)**		**Executes on Server 2**				
Action	**Initial State**	**End State**	**Startup**	**OnScan**	**Execute**	**OffScan**	**Shutdown**
Deploy	Down	Standby	Y	N	N	N	N
Forced Failover	Standby	Active OnScan	N	Y	Y	N	N
Server 1 Failure	Standby	Active OnScan	N	Y	Y	N	N
Server 2 Failure (hard shutdown)	Standby	Down	N	N	N	N	N
Graceful shutdown of Server 1	Standby	Active Offscan	N	N	N	N	N

Graceful shutdown of Server 2	Standby	Down	N	N	N	N	Y
Start Server 1	Down	Down	N	N	N	N	N
Start Server 2 (Server 1 running)	Down	Standby	Y	N	N	N	N
Undeploy	Standby	Down	N	N	N	N	Y

Startup Scripts

Startup scripts are called when an object containing the script is loaded into memory, such as during deployment, platform, or engine start.

Startup instantiates COM objects and .NET objects. Depending on load and other factors, assignments to object attributes from the Startup method may fail. Attributes that reside off-object are not available to the Startup method.

Startup Scripts for Redundant AppEngines

There are certain considerations that you must take into account when writing a Startup script that will run on redundant AppEngines. This section outlines whether or not a script will be executed under various scenarios, including deploy, forced failover, system failure, system startup, and undeploy operations.

Redundant engines can be set to run in either **Legacy Mode** or **Run Warm Mode**. Startup scripts for redundant engines may operate differently, depending on the selected redundancy mode.

Note: New redundant engines default to Run Warm Mode. Redundant engines in migrated galaxies default to Legacy Mode.

- **Legacy mode (RunWarm attribute is disabled):** In Legacy mode, the Standby engine does not start until failover occurs. This will result in longer failover times when compared with Run Warm Mode. Highlighted text indicates where there is a difference in script execution between Legacy mode and Warm Redundancy mode.

Legacy Mode	**Primary Engine (Server 1)**		**Backup Engine (Server 2)**		**Startup Script**
Action	**Initial State**	**End State**	**Initial State**	**End State**	**Script Execution**
Deploy	Down	Active OnScan	Down	Standby	Startup scripts execute when the Primary Engine starts.

Legacy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Startup Script
					The Backup Engine does not start.
Forced Failover	Active OnScan	Standby	Standby	Active OnScan	Startup scripts execute when the Backup Engine starts.
Server 1 Failure (hard shutdown)	Active OnScan	Down	Standby	Active OnScan	Startup scripts execute when the Backup Engine starts.
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	Standby	Down	Startup scripts do not execute in the event of a Server 2 failure.
Graceful shutdown of Server 1 platform or engine using OCMC	Active OnScan	Down	Standby	Active Offscan	Startup scripts execute when the Backup Engine starts. The Primary engine shuts down, Standby engine is started and goes to active but remains OFFscan.
Graceful shutdown of Server 2 platform or engine using OCMC	Active OnScan	Active OnScan	Standby	Down	Startup scripts do not execute. Shutdown of Server 2 has no affect on operations. Server 1 continues running OnScan.
Start Server 1 only	Down	Active OnScan	Down	Down	Startup scripts execute when the Primary Engine on Server 1 starts.

Legacy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Startup Script
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	Down	Standby	Startup scripts do not execute when the Backup Engine starts. Server 1 continues running OnScan.
Undeploy	Active OnScan	Down	Standby	Down	Startup scripts do not execute during an undeploy operation.

- **Run Warm mode (RunWarm attribute is enabled):** In Run Warm mode, Startup scripts do not execute during a failover in most circumstances, since both the Primary and Backup engines start concurrently. The Backup Engine on Server 2 will only execute Startup scripts if the Primary Engine on Server 1 is down. Highlighted text indicates where there is a difference in script execution between Legacy mode and Warm Redundancy mode.

Warm Redundancy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Startup Script
Action	**Initial State**	**End State**	**Initial State**	**End State**	**Script Execution**
Deploy	Down	Active OnScan	Down	Standby	Startup scripts execute when the Engines start (both Engines start with warm redundancy).
Forced Failover	Active OnScan	Standby	Standby	Active OnScan	Startup scripts do not execute. On Server 1, the Active engine shuts down and a Standby engine is created. On Server 2, the Standby engine previously started and now goes to Active

Warm Redundancy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Startup Script
					OnScan.
Server 1 Failure (hard shutdown)	Active OnScan	Down	Standby	Active OnScan	Startup scripts do not execute since the Backup engine has already started.
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	Standby	Down	Startup scripts do not execute in the event of a Server 2 failure.
Graceful shutdown of Server 1 platform or engine using OCMC	Active OnScan	Down	Standby	Active OFFscan	Startup scripts do not execute. The Active engine shuts down, Standby engine previously started and now goes to active but remains OFFscan.
Graceful shutdown of Server 2 platform or engine using OCMC	Active OnScan	Active OnScan	Standby	Down	Startup scripts do not execute. Shutdown of Server 2 has no affect on operations. Server 1 continues running OnScan.
Start Server 1 only	Down	Active OnScan	Down	Down	Startup script executes when Primary Engine starts.
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	Down	Standby	Startup scripts execute when the Backup engine starts on

Warm Redundancy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Startup Script
					Server 2 (runs as Standby). Server 1 continues running OnScan.
Undeploy	Active OnScan	Down	Standby	Down	Startup scripts do not execute during an undeploy operation.

OnScan Scripts

OnScan scripts are called the first time an AppEngine calls this object to execute after the object's scan state changes to OnScan. The OnScan method initiates local object attribute values and provides more flexibility in the creation of .NET or COM objects.

Attributes that are off-engine are not available to the OnScan method.

OnScan Scripts for Redundant AppEngines

This section outlines whether or not the script will run under various scenarios, including including deploy, forced failover, system failure, system startup, and undeploy operations.

The selected redundancy mode, Legacy or Run Warm, does not change the behavior of OnScan scripts for redundant engines. For both Legacy and Warm Redundancy modes:

- When failover occurs, OnScan scripts are triggered when the **Active** engine goes OnScan.
- OnScan scripts are NOT triggered when the **Standby** engine goes OnScan.

Both Redundancy Modes	Primary Engine (Server 1)		Backup Engine (Server 2)		OnScan Script
Action	**Initial State**	**End State**	**Initial State**	**End State**	**Script Execution**
Deploy	Down	Active OnScan	Down	Standby	OnScan scripts execute when the Active Engine transitions to OnScan.

Both Redundancy Modes	Primary Engine (Server 1)		Backup Engine (Server 2)		OnScan Script
Forced Failover	Active OnScan	Standby	Standby	Active OnScan	OnScan scripts execute when the Backup Engine transitions to OnScan.
Server 1 Failure (hard shutdown)	Active OnScan	Down	Standby	Active OnScan	OnScan scripts execute when the Backup Engine transitions to OnScan.
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	Standby	Down	OnScan scripts do not execute in the event of a Server 2 failure (no state change for Server 1).
Graceful shutdown of Server 1 platform or engine on Server 1 using OCMC	Active OnScan	Down	Standby	Active OffScan	OnScan scripts do not execute when the OCMC shuts down a platform or Active Engine on Server 1. The Standby Engine on Server 2 remains OFFscan.
Graceful shutdown of Server 2 platform or engine using OCMC	Active OnScan	Active OnScan	Standby	Down	OnScan scripts do not execute when the OCMC shuts down a platform or Standby Engine on Backup Server 2. The Active Engine on Server 1 remains running OnScan.

Both Redundancy Modes	Primary Engine (Server 1)		Backup Engine (Server 2)		OnScan Script
Start Server 1 only	Down	Active OnScan	Down	Down	OnScan scripts execute when the primary engine on Server 1 starts and transitions to its prior state of Active OnScan.
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	Down	Standby	OnScan scripts do not execute when the Backup Engine starts (state of the active engine running on Server 1 does not change).
Undeploy	Active OnScan	Down	Standby	Down	OnScan scripts do not execute during an undeploy operation.

Execute Scripts

Execute scripts are called each time the AppEngine performs a scan and the object is OnScan.

The Execute script method is the workhorse of the scripting execution types. Use the Execute method for your run-time scripting to ensure that all attributes and values are available to the script.

If the **Quality** check-box is checked, the Execute method is similar to InTouch HMI scripts with the following conditional trigger types:

- Periodic: When going OnScan, a script with a periodic trigger executes immediately (at the next scheduled scan period of the AppEngine). It then executes periodically whenever the elapsed time evaluates as true.
- Data Change: Executes when a data value or quality changes between scans.

For the following trigger types, data changes between each scan are not evaluated, only the value at the beginning of each script is used for evaluation purposes. For example, if a Boolean attribute changes from True to False to True again during a scan cycle, this change is not evaluated as a data change as the value is True at the beginning of each scan cycle.

- OnTrue: Executes if the expression validates from a false on one scan to a true on the next scan.

- OnFalse: Executes if the expression validates from a true on one scan to a false on the next scan.

These scripts also have time-based considerations. A trigger period of 0 means that the script executes every scan.

Time-based scripts, WhileTrue, WhileFalse, and Periodic are evaluated and executed based on the elapsed time from a timestamp generated from the previous execution, not on an elapsed time counter. It is possible that a change in the system clock can change the interval between execution of these scripts.

- WhileTrue: Executes scan to scan as long as the expression validates as true at the beginning of the scan.
- WhileFalse: Executes scan to scan as long as the expression validates as false at the beginning of the scan.

For example, a periodic script is set to run every 60 minutes. The script executes at 11:13 AM. We expect it to execute 60 minutes later at 12:13 PM. However, a time synchronization event occurred and the node's time is adjusted from 11:33 AM to 11:30 AM.

The script still executes when the system time reaches 12:13 PM. But because of the time change, the actual (True) time period that elapsed between executions is 63 minutes.

Execute Scripts for Redundant AppEngines

This section outlines whether or not the script will run under various scenarios, including including deploy, forced failover, system failure, system startup, and undeploy operations.

The selected redundancy mode, Legacy or Run Warm, does not change the behavior of Execute scripts for redundant engines. For both Legacy and Warm Redundancy modes:

- The Active engine is triggered on execute.
- The Standby engine is NOT triggered on execute.

Both Redundancy Modes	**Primary Engine (Server 1)**		**Backup Engine (Server 2)**		**Execute Script**
Action	**Initial State**	**End State**	**Initial State**	**End State**	**Script Execution**
Deploy	Down	Active OnScan	Down	Standby	Execute scripts run after the Active Engine transitions to OnScan, at the next scheduled scan period of the AppEngine.
Force Failover	Active OnScan	Standby	Standby	Active OnScan	Execute scripts run after the Standby Engine transitions to

Both Redundancy Modes	Primary Engine (Server 1)		Backup Engine (Server 2)		Execute Script
					OnScan, at the next scheduled scan period of the AppEngine.
Server 1 Failure (hard shutdown)	Active OnScan	Down	Standby	Active OnScan	Execute scripts run after the Standby Engine transitions to OnScan, at the next scheduled scan period of the AppEngine.
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	Standby	Down	Execute scripts run at the next scheduled scan period of the AppEngine.
Graceful shutdown of Server 1 platform or engine using OCMC	Active OnScan	Down	Standby	Active OffScan	Execute scripts do not run when the OCMC shuts down a platform or Active Engine on Server 1. The Standby Engine on Server 2 remains OFFscan.
Graceful shutdown of Server 2 platform or engine using OCMC	Active OnScan	Active OnScan	Standby	Down	Execute scripts do not run when the OCMC shuts down a platform or standby engine on Backup Server 2. The Active Engine on Server 1 remains running OnScan.
Start Server 1 only	Down	Active OnScan	Down	Down	Execute scripts run after the

Both Redundancy Modes	Primary Engine (Server 1)		Backup Engine (Server 2)		Execute Script
					Active Engine transitions to OnScan, at the next scheduled scan period of the AppEngine.
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	Down	Standby	Execute scripts run at the next scheduled scan period of the AppEngine.
Undeploy	Active OnScan	Down	Standby	Down	Execute scripts do not run during an undeploy operation.

OffScan Scripts

OffScan scripts are called when the object is taken OffScan. This script type is primarily used to clean up the object and account for any needs to address as a result of the object no longer executing.

If an object is taken OffScan, either directly, or indirectly because its engine is taken OffScan, all in-progress asynchronous scripts for that object are requested to shut down by setting a Boolean shutdown attribute for the script to true. A well-written script checks this attribute before and after time-consuming operations. If the script takes more than 30 seconds to complete, a warning appears in the logger that the script is not responding to the shutdown command. However, the script is allowed to complete and is not terminated by force. This all takes place on the engine's main thread and could potentially hang the engine. During this time, the script might also time out and as a result exit before executing all its logic.

OffScan Scripts for Redundant AppEngines

This section outlines whether or not the script will run under various scenarios, including including deploy, forced failover, system failure, system startup, and undeploy operations.

The selected redundancy mode, Legacy or Run Warm, does not change the behavior of OffScan scripts for redundant engines. For both Legacy and Warm Redundancy modes:

- When failover occurs, OffScan scripts are triggered when the **Active** engine goes OffScan.
- OffScan scripts are NOT triggered when the **Standby** engine goes OffScan.

Both Redundancy Modes	**Primary Engine (Server 1)**		**Backup Engine (Server 2)**		**OffScan Script**
Action	**Initial State**	**End State**	**Initial State**	**End State**	**Script Execution**
Deploy	Down	Active OnScan	Down	Standby	OffScan scripts do not execute during a deploy operation.
Forced Failover	Active OnScan	Standby	Standby	Active OnScan	OffScan scripts execute when the Backup engine transitions to OffScan.
Server 1 Failure (hard shutdown)	Active OnScan	Down	Standby	Active OnScan	OffScan scripts do not execute when Server 1 has a hard shutdown.
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	Standby	Down	OffScan scripts do not execute in the event of a Server 2 failure (no state change for Server 1).
Graceful shutdown of Server 1 platform or engine using OCMC	Active OnScan	Down	Standby	Active OffScan	OffScan scripts execute when the OCMC shuts down a platform or active engine on Server 1. The Standby engine on Server 2 transitions from Standby to Active Offscan.
Graceful shutdown of Server 2 platform or engine using OCMC	Active OnScan	Active OnScan	Standby	Down	OffScan scripts do not execute when the OCMC shuts down a platform or standby engine

Both Redundancy Modes	Primary Engine (Server 1)		Backup Engine (Server 2)		OffScan Script
					on Backup Server 2.
Start Server 1 only	Down	Active OnScan	Down	Down	OffScan scripts do not execute when the Primary Engine on Server 1 starts and transitions to its prior state of Active OnScan.
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	Down	Standby	OffScan scripts do not execute when the Backup engine starts (state of the active engine running on Server 1 does not change).
Undeploy	Active OnScan	Down	Standby	Down	OffScan scripts execute during an undeploy operation when the Active Engine goes OffScan before it shuts down.

Shutdown Scripts

Shutdown scripts are called when the object is about to be removed from memory, usually as a result of the AppEngine stopping. Shutdown scripts are primarily used to destroy COM objects and .NET objects and to free memory.

Shutdown Scripts for Redundant AppEngines

There are certain considerations that you must take into account when writing a Shutdown script that will run on redundant AppEngines. This section outlines whether or not a script will be executed under various scenarios, including including deploy, forced failover, system failure, system startup, and undeploy operations.

Redundant engines can be set to run in either **Legacy Mode** or **Run Warm Mode**. Shutdown scripts for redundant engines may operate differently, depending on the selected redundancy mode.

Note: The warm redundancy feature was introduced in System Platform 2020 R2 SP1. New redundant engines default to Run Warm Mode. Galaxies with redundant engines that were created prior to System Platform 2020 R2 SP1 and migrated to the current System Platform version default to Legacy Mode.

Shutdown scripts for redundant engines may operate differently, depending on which redundancy mode is selected.

- **Legacy mode (RunWarm attribute is disabled):** In Legacy mode, the Standby Engine does not start until failover occurs. This will result in longer failover times when compared with Run Warm Mode. Highlighted text indicates where there is a difference in script execution between Legacy mode and Warm Redundancy mode.

Legacy Mode	**Primary Engine (Server 1)**		**Backup Engine (Server 2)**		**Shutdown Script**
Action	**Initial State**	**End State**	**Initial State**	**End State**	**Script Execution**
Deploy	Down	Active OnScan	Down	Standby	Shutdown scripts do not execute during a deploy operation.
Forced Failover	Active OnScan	Standby	Standby	Active OnScan	Shutdown scripts execute when the Primary Engine shuts down.
Server 1 Failure (hard shutdown)	Active OnScan	Down	Standby	Active OnScan	Shutdown scripts do not execute in the event of a hard shutdown.
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	Standby	Down	Shutdown scripts do not execute in the event of a hard shutdown.
Graceful shutdown of Server 1 platform or engine using OCMC	Active OnScan	Down	Standby	Active OFFscan	Shutdown scripts execute when the Primary Engine shuts down.

Legacy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Shutdown Script
Graceful shutdown of Server 2 platform or engine using OCMC	Active OnScan	Active OnScan	Standby	Down	Shutdown scripts do not execute when Server 2 is shut down. because the Backup Engine was never started.
Start Server 1 only	Down	Active OnScan	Down	Down	Shutdown scripts do not execute.
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	Down	Standby	Shutdown scripts do not execute.
Undeploy	Active OnScan	Down	Standby	Down	Shutdown scripts execute during an undeploy operation as the Active engine is shut down. In Legacy mode, the Backup Engine was never started.

- **Run Warm mode (RunWarm attribute is enabled):** In Run Warm mode, the Standby Engine executes Startup scripts until it becomes active. Highlighted text indicates where there is a difference in script execution between Legacy mode and Warm Redundancy mode.

Warm Redundancy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Shutdown Script
Action	**Initial State**	**End State**	**Initial State**	**End State**	**Script Execution**
Deploy	Down	Active OnScan	Down	Standby	Shutdown scripts do not execute during a deploy operation.

Warm Redundancy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Shutdown Script
Forced Failover	Active OnScan	Standby	Standby	Active OnScan	Shutdown scripts execute when the Primary Engine shuts down.
Server 1 Failure (hard shutdown)	Active OnScan	Down	Standby	Active OnScan	Shutdown scripts do not execute in the event of a hard shutdown.
Server 2 Failure (hard shutdown)	Active OnScan	Active OnScan	Standby	Down	Shutdown scripts do not execute in the event of a hard shutdown.
Graceful shutdown of Server 1 platform or engine using OCMC	Active OnScan	Down	Standby	Active Offscan	Shutdown scripts execute when the Primary Engine shuts down.
Graceful shutdown of Server 2 platform or engine using OCMC	Active OnScan	Active OnScan	Standby	Down	Shutdown scripts execute when Server 2 is shut down because in Run Warm mode, the Backup Engine was started previously.
Start Server 1 only	Down	Active OnScan	Down	Down	Shutdown scripts do not execute.
Start Server 2 (Server 1 running)	Active OnScan	Active OnScan	Down	Standby	Shutdown scripts do not execute.
Undeploy	Active OnScan	Down	Standby	Down	Shutdown scripts execute during an

Warm Redundancy Mode	Primary Engine (Server 1)		Backup Engine (Server 2)		Shutdown Script
					undeploy operation as the servers shut down. In Run Warm mode, the both engines were started and are now shut down.

Deployment Scripts

Deploying objects is both a critical and a load-intensive process for a Galaxy. Implementing scripting in the Startup and OnScan methods can adversely affect a Galaxy's deployment and redundancy performance.

While objects are being deployed, their Startup and, if deployed OnScan scripts are executed. These scripts must complete within the deployment time-out period for the deployment to be successful.

Placing large numbers of scripts, or scripts that require heavy processing power into the Startup or OnScan script methods can slow or cause a deployment or failover to fail. In addition to the load that is placed on the system at deployment time, the type of scripting done in the Startup and OnScan methods is also important because these scripts execute in a sequence.

Deployment Scripts for Redundant AppEngines

When writing a deployment script that will run on redundant AppEngines, be sure to account for the mode in which the redundant engines will run. Redundant engines can be set to run in either **Legacy Mode** or Run **Warm Mode**.

- The selected redundancy mode, Legacy or Run Warm, does not change the behavior of OnScan scripts for redundant engines.
- The selected redundancy mode may change the behavior of Startup scripts. See Startup Scripts for more information about the differences in script execution between modes.

  Note: The warm redundancy feature was introduced in System Platform 2020 R2 SP1. New redundant engines default to Run Warm Mode. Galaxies with redundant engines that were created prior to System Platform 2020 R2 SP1 and migrated to the current System Platform version default to Legacy Mode.

During deployment and restart, the Startup and OnScan script methods do not execute objects based on execution order. Objects are started up and placed on scan based on their alphanumeric tag name within their hosting Area.

Follow the recommendation below for each type of script method to help determine what scripting practices to follow in each script method.

Do not place the following types of scripting in the Startup or OnScan methods:

- Database access
- File system access to .csv, .xml, .txt, and other file types
- Off-object referencing
- Dynamic referencing

Security

AVEVA works closely with Microsoft and industry standards organizations like the OPC Foundation to involve multiple vendors in an industry-wide approach to solving security problems.

The success of a security solution is enhanced by pooling IT expertise and SCADA operations groups during the implementation and integration phases of a System Platform project.

This section provides a high-level security perspective, and specific recommendations within the System Platform environment.

For additional information about implementing security for System Platform, see the *AVEVA Cybersecurity Deployment Guide*.

AVEVA security perspective

Information systems in manufacturing facilities are evolving rapidly. The evolution of information systems is driven by the need of manufacturers to integrate with business/ERP and production systems, provide access to production data across the enterprise (both from inside and outside the environment), and reduce system maintenance costs.

Security risks also evolve as new vulnerabilities and targets are discovered in the various systems.

Numerous incentives exist to protect a control system:

- The technical knowledge, skills and tools required to penetrate your IT and plant systems are widely available.
- Regulatory mandates and government guidelines
- Guidelines and best practices for securing plant control systems from advisory groups, such as the ISA SP99 committee, IEC 62433, NIST Process Control Security Requirements Forum (PCSRF), North American Electric Reliability Corporation (NERC), etc.

The AVEVA approach to site networks and control system security is driven by the following principles:

- View security from both Management and Technical perspectives.
- Ensure security is addressed from both IT/IS and Control System perspectives.
- Design and develop multiple network, system, and application security layers.
- Ensure industry, regulatory and international standards are taken into account.
- Prevent security breakdowns and intrusions in critical in plant control systems, and detect these issues if and when they occur.

Realizing these principals is accomplished by implementing the following security recommendations:

- Maintain a prevention philosophy to support security policies and procedure/s using the following security components:
  - Firewalls
  - Network-based intrusion prevention/detection

    - Host-based intrusion prevention/detection
- Include a clearly defined and clearly communicated change management policy, for example, firewall configuration changes.
- Converge IT and plant networks.
- Maintain secure and insecure protocols on the same network.
- Enforce monitoring, alerting and diagnostics of plant network control systems and their integration with the corporate network.
- Move to an off-platform data collector in a DMZ.
- Retain forensic information to support investigation/legal litigation.
- Enable secure connectivity to wireless devices.

Common control system security considerations

When securing a control system, the number one criteria is defining and understanding the information/data that needs to be secured. In doing so, potential vulnerabilities are identified. The vulnerabilities may be the result of practices adopted primarily for convenience.

Once identified, vulnerabilities may be removed or altered to increase security with no impact on production operation performance. Areas of focus include:

- Multiple remote access points.
- Information queries that can be deferred or accessed though a DMZ/off control network, etc.

Common security evaluation topics

The following security topics are critical parts of an effective security strategy.

Policies and procedures

Security policies and procedures are the foundation of a solid security strategy. Many automation, control, and access areas must have well-defined security policies and procedures. The security policies and procedures (and their enforcement) will have a profound effect on enhancing automation and control system security.

Accounts

Types and uses of security accounts need to be defined by strong security policies and must comprise useful account creation and maintenance procedures. The policies that govern system accounts should be fully developed, documented and communicated by IT, automation engineering, and management in a collaborative environment.

The following items must be considered when developing or reviewing account policies:

- Only validated users have accounts.
- User IDs must have unique names with strong passwords.
- Individuals are accountable for the use of their User ID.

- User access should be restricted as much as possible.
- Make sure that account lockout duration is well defined.
- Groups should be defined by user access needs and roles.
- Guest accounts and default vendor accounts should be removed or reset as applicable.
- Process operator station accounts should be limited and defined by operational area.
- Service accounts should exist on the local domain or local machine and should not be used to logon to a server.

Passwords

Passwords are one of the most vulnerable security components. Define a solid password policy and configure your system to enforce the policy.

Using complex passwords and changing them regularly lessens the likelihood of unauthorized access to the control system.

The following list provides guidelines for effective password management:

- Enforce password history to limit the reuse of old passwords.
- Enforce password aging to force periodic changing of passwords.
- Enforce minimum password length and complexity requirements to reduce the chances of successful password guessing.
- Ensure passwords are not stored using reversible encryption.

Remote access

The need for access to process information, configuration information and system information from outside of the systems domain is common. Well-defined policies and procedures to manage remote access to the system by other company business units and or suppliers and venders greatly reduces the possibility of security threats penetrating the system.

The following list contains guidelines for remote access:

- Limit access as much as possible by defining different access levels based on need (job function).
- Enforce mandatory PC checkups of any equipment that is brought onsite.
- Configure a separate role-based user group for temporary accounts and review this user list often.
- Define and document all outside system access routes and accounts.

Physical access

Most production facilities have physical security plans in place. These plans should be an integral part of an overall security program. By not allowing unchecked computers and unauthorized users to have access to critical infrastructure components, many security threats can be eliminated.

Critical process control components such as servers, routers, switches, PLCs, and controllers should be protected under lock and key and have personnel assigned who are directly responsible for the components.

Backup and recovery plan

The backup and recovery plan is a critical security component. Recovery from any level of failure due to either a security or natural interruption of the system must be included in the security policy.

The following items must be considered when defining a backup and recovery plan:

- Define and document how each part of the system will or can be backed up.
- Ensure backups are included in routine system maintenance plan and when improvements or other changes to the system occur.
- Document backup procedures for all system configurations and assign administrative responsibility to appropriate personnel.
- Document and keep current all versioning of system software and hardware.
- Provide a protected off-site repository for copies of all system backups
- Provide a documented escalation plan for recovery and documented processes assigned to qualified personnel for implementing a recovery.

Virus protection

Add an additional security level at each access point of the system by defining where and what virus protection is to be implemented. Document the proper configurations for the virus protection software.

Mandatory virus definition file updates are essential.

Note: For more information about configuring anti-virus software, see Tech Note TN10567, "AVEVA System Platform 2020 AntiVirus exclusions."

Security patch implementation

Security patch management is a critical evaluation topic that has the largest impact on Microsoft operating system-based supervisory and control systems.

Careful planning and attention to detail is required when developing and documenting your procedures and policies for implementing security patches. Request a detailed support plan from each automation vendor and security software vendor, and review them with the goal of inclusion as part of any security patch management procedure or policy.

General information about security infrastructure

The security infrastructure comprises many components that support the supervisory and control system. Each component needs to be reviewed and defined by critical value and attack vulnerability. Policies and procedures must then be defined that provide auditing and maintaining the security levels of each component.

Redundancy

Each component should also be reviewed for possible redundant configuration to improve availability and protect against the system becoming unavailable due to a single failure. For detailed information on Application Server redundancy configuration, see the *Application Server User Guide*.

Authenticators

System users could be actual operators and engineers, or other systems or services that run internally or externally to the supervisory control system.

All known users must be accounted for and defined authentication methods and procedures should be developed to reduce the risk of unauthorized access to critical systems or protected information.

Security policy enforcement components

Each device or software package that is deployed for security policy enforcement must be defined by enforcement type and its impact to system on failure. These components include, but are not limited to, firewalls, routers, switches, and operating system services.

Any enforcement component that is defined as critical should be deployed in a redundant configuration if possible.

Firewalls, routers, switches

Firewalls, routers, and switches are an integral part of supervisory and control systems.

Firewalls provide for a way to isolate and control communication between segments of a network and between operational units. A detailed understanding of communication ports, IP addresses, and protocols needed for the supervisory and control system to function properly is critical for the success of the security policy.

By defining solid policies and procedures for firewall configuration, operation, and auditing, you can limit your communication to specific ports and IP addresses that allow only authorized communication between systems.

Additionally, defining solid policies and procedures for router and switch configuration ensures management of where information and access is permitted along with control over bandwidth. Optimal network utilization can then be achieved.

Although firewalls, routers, and switches have overlapping capabilities, each device should be used for its base functionality: firewalls should be used to control communication types, routers should be used to forward communication by routing protocols along a proper route, and switches should be used to manage bandwidth by controlling communication flow between ports and avoiding packet collisions.

Domain controllers

The use of services such as Microsoft Active Directory provides management and enforcement of access security for users, groups, and organizational units.

Not all software supports domain-level security. For example, some automation software will require local PC or even package- or ApplicationObject-level security to be defined and implemented. Check the product documentation carefully before deployment.

Physical networks

The basic building block of a supervisory and control system is the physical network itself. Special attention should be given to the design, selection of media, and installation of the network. A careful review of any installed network segment should be undertaken before extending or adding components.

By making sure redundant paths and proper distances are observed, slow and unreliable communication can be

avoided. All networks should be reviewed for live unsecure ports and exposed segments that could be tapped. With the complete network layout documented, recovery plans can be defined to improve system availability in the event that an incident that takes down part of the network.

Remote access devices

Policies and procedures should be developed to control the installation and use of modems for remote access. A very good alternative to allowing modem access is to implement Virtual Private Network (VPN) access. If a modem has to be used for remote access a good rule is to require dial back connections.

Wireless access

Wireless technologies are often used with supervisory and control systems. The following topics should be considered when defining a wireless implementation:

- Access can be limited to exclude unwanted areas through the use of directional antennas.
- Utilize more than the industry-standard WEP ("Wired Equivalent Privacy") protocol.
- Use a solution based on 802.1X, Extensible Authentication Protocol (EAP), and Wi-Fi encryption.
- Review implementation guides from your wireless device vendor and from your operating system vendor.

Software

The software components of a supervisory and control system can have a large impact on the security of the overall system. When reviewing the security features of the software that will be deployed within a production facility, each component should be evaluated as an integrated part of the complete system.

All software components should leverage the capabilities of the infrastructure and support configurations that meet the policies and procedures that are defined as need to secure the system. By reviewing all software from a security standpoint, policies and procedures can be established to audit the system and maintain high levels of security.

Virus and malicious software protection

With the many host-based protection system options available on the market today, ensure that all supervisory and control system software is compatible and that the vendor provides timely updates. Host-based protection software should also protect against other malicious software such as spyware, malware, and adware.

Intrusion protection and prevention

Intrusion protection and prevention has become a viable way of raising the security level within a TCP-IP LAN or WAN infrastructure.

Intrusion detection systems monitor network traffic and generate alerts when malicious traffic or repeated password guessing is detected. These tools have been employed by IT department for many years.

Intrusion prevention technology has become the preferred method to detect and alert when hacking or virus/worm attacks are present, as well as block such attempts by managing firewall policy, switch ports, router paths, and trapping emails before damage can be done.

The implementation of an intrusion detection or prevention system on a supervisory and control network does include risk. The following list explains some considerations when evaluating their use in a particular environment:

- The system should provide centralized reporting and management.
- The system should provide multiple ways to deliver alerts.
- Evaluate the supported level of signature-based identification of malicious or anomalous traffic.
- Connection Flood (denial of service) controls should be included.
- The system should support alert-only mode for tuning.
- The system should support the software and application that you have installed or going to deploy.
- The system should enable creation of your own policies.
- Evaluate supported bandwidth and connections.

Because intrusion detection and prevention systems can present a risk to functionality and operation of a supervisory and control system, a well-developed design with strong policies and procedures should accompany any implementation plan.

Operating systems

Review the base operating system that hosts all of your supervisory and control applications for proper deployment, configuration, and security patches. The initial focus should be reviewing installed components and configured users.

Microsoft provides detailed guidance for locking down your operating system to mitigate security threats. By defining what supervisory and control software is to be deployed to a system, you can define the level of lock-down, and at the same time ensure full functionality of manufacturing applications.

Databases

Database applications such as Microsoft SQL Server have become a common component of all manufacturing systems. Because of the need to allow access to database information, and the need to update and append the information, you must be very deliberate in the approach to locking down a database.

Provide a detailed mapping of users (people and services) which require access and define usable database security policies.

Securing System Platform

The ability to secure System Platform is directly related to the infrastructure (servers, workstations, ethernet cables, fiber optics, switches, routers, firewalls, etc.) as well as the hosting software, including operating systems, virus protection, intrusion protection, etc.

This section is designed primarily for process control engineers and IT professionals who are familiar with standard IT practices, Windows domain engineering and administration, and process control or SCADA environment requirements. Content is derived from AVEVA testing documentation.

Security considerations

The following section summarizes the security considerations within a production environment, and describes recommendations as applied to a process control network (PCN) or SCADA system (WAN).

Secure layers

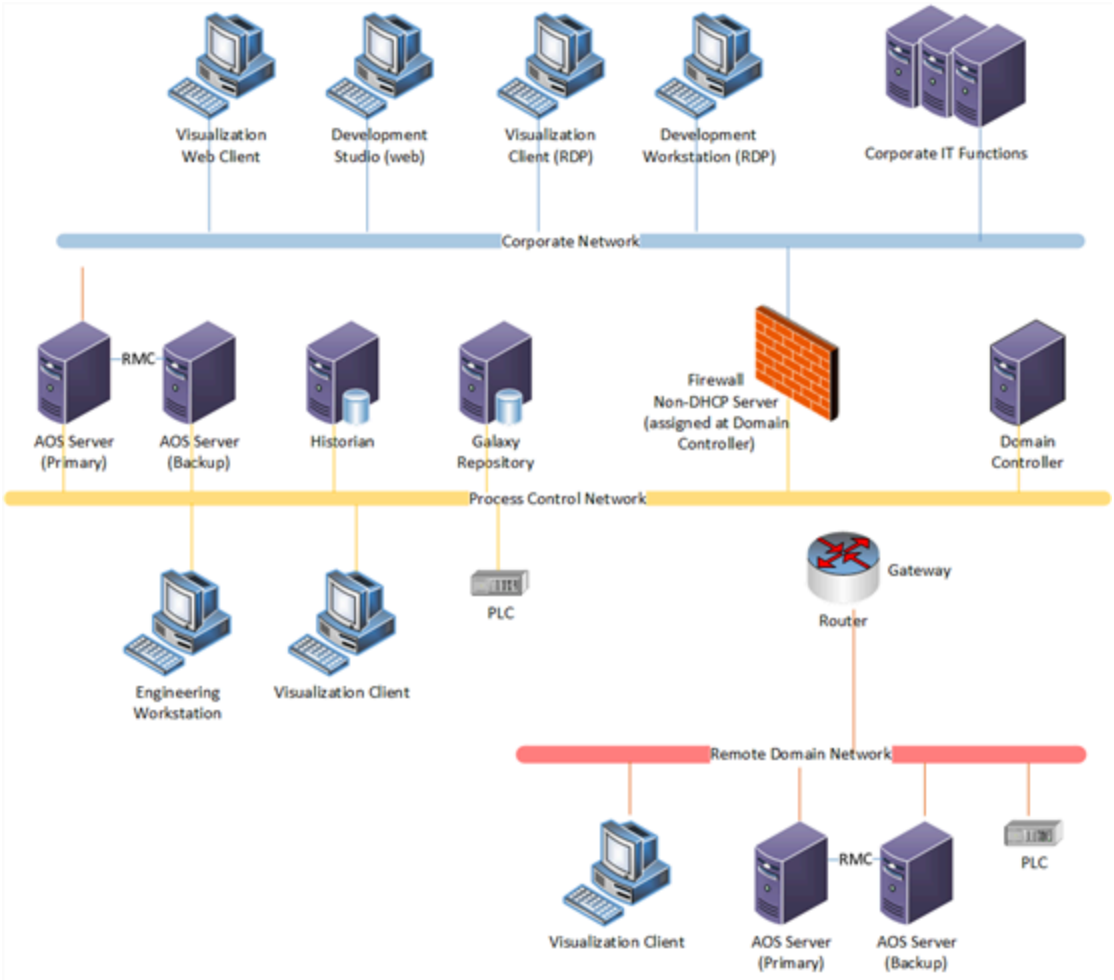
Divide the system into secure layers. In the security context, a layer is defined as a division of a network model, through which messages pass as they are prepared for transmission. All layers are separated by a router or smart switch device.

A secure layer is further defined by the need to allow or restrict access and the criticality of the sub-system. An intrusion detection system is deployed in higher-risk layers.

The following figure is designed to show a representative topology. This is not intended to depict an actual plant system topology. It includes the following named layers:

- Corporate network infrastructure
- Process control network (PCN)
- Remote domain network (adjunct to PCN)

Note that all layers (represented by the main backbone) are separated by a firewall or router:



System Platform Software Applications

System Platform applications have been tested in a wide variety of security-related implementations similar to the previous figure. The figure represents the widest usage scenarios of product combinations, which were tested in various scenarios involving limited- or no DCOM connectivity, limited port ranges, narrow firewall settings, and highly routed environments.

Some System Platform software applications, such as AVEVA Batch Management System and MES, utilize a high degree of connectivity to the corporate ERP System and process control enterprise, along with the associated distributed computational and remote services requirements. These applications could be adversely affected if unlimited DCOM connectivity is not available.

The layers and port listings are detailed below.

Object security

The System Platform IDE lets you create security groups and place the different application objects that are contained in a galaxy into separate security groups. Until you configure the security groups, all objects are contained in a single security group named "Default."

At a minimum, device integration (DI) objects and redundant DI (RDI) objects should be aggregated into a

separate group. All attributes on these objects are in the "operate" category. The security group that you place these DI and RDI objects into should be accessible only to admin-level users.

Corporate network infrastructure layer

The majority of communication between computers on a corporate (business) layer is accomplished using viewers, proxies, or interfaces such as an internet browser. These engines use HTTP or HTTPS (secure http) protocols to transmit and receive data. This data can be secured, filtered, and carefully monitored. For the most part, only traffic with proper credentials or limited functionality is allowed to pass.

RPC traffic, or Remote Procedure Calling (required with DCOM) is rare between business nodes. Closing DCOM ports for added security at this level can be effective, since the most desktop applications do not use many, if any, DCOM objects, and therefore do not require ports to transport information.

Corporate Network firewall ports

The following table lists the default firewall ports necessary for successful communication between business nodes. HTTP and HTTP ports can be changed through the System Platform Configurator.

Function	Port
HTTP	TCP 80
HTTPS	TCP 443
RDP (listening)	TCP 3389

Process control network (PCN) layer

The Process Control Network (PCN) Layer contains the production nodes (Data Servers, AppServers, etc.) that process and store all production data.

This layer requires that all nodes have unrestricted access to each other, in order to process the data in real-time.

Data sources are represented in the previous figure by the Legacy and Remote Domain sub-layers. These layers may also represent geographically distant sites which may be leased to other enterprises, and whose data is required by the parent company.

Securing visualization

System Platform includes two visualization clients: InTouch HMI and AVEVA Operations Management Interface (OMI). AVEVA OMI is an advanced visualization client built into Application Server, and directly leverages the Application Server security model and settings.

Users with different roles require different user interface experiences. Typical interface experiences include window-to-window navigation, data visibility on a specific window, and restrictions on visible actions. InTouch HMI easily supports these actions through animation links. The animation links (typically) test the InTouch Software system tag $AccessLevel. While this implementation works, it provides a very linear security model.

Application Server roles offer more flexibility and can be leveraged from InTouch HMI by using the

IsAssignedRole ("RoleName") script function. When executed, this function determines if the currently logged-in user is assigned to the role that was entered into the script call. This function allows the InTouch application to access the role-based security set in the System Platform IDE.

To implement this, add a Data Change script to InTouch Software that executes any time the InTouch system tag $Operator changes. For example, the following script could be called when the $Operator tag changes:

```
AdministrativeAccess = IsAssignedRole("Administrator");
SetpointAccess = IsAssignedRole("Engineer");
ManualAccess = IsAssignedRole("Operator");
```

In this example, AdministrativeAccess, SetpointAccess, and ManualAccess are Discrete InTouch Software memory tags. Users can possess multiple roles and more than one of these discrete tags could be set. You can animate the InTouch Software application by using these tags in the expression statements of the animation links.

This implementation has the following advantages:

- First, the scripts execute only when the user changes. Instead of running the same script for every animation, it only runs as needed, which improves overall application performance. This also improves the draw times of the screen are also improved, since it is not necessary to evaluate the user rights for each associated animation.
- The second advantage is in maintenance. By having the script appear in one location, there is only one place to go to make required changes. If, in the previous example, all Manual Control was no longer allowed by Operators, but instead this permission was only going to be given to Engineers, this can be achieved by a simple change. You would change the third line of the script to read:

  ```
  ManualAccess = IsAssignedRole("Engineer");
  ```

This change would only be made in the $Operator Data Change script, instead of every Manual Control animation.

OS group based security mode notes

The OS Group Based security mode enables user authorization based on OS Groups; in other words, this mode leverages the operating systems' user authentication system on a Group basis. This means that the user is a member of a particular group and has certain permissions within the context of that group.

Two settings are available in this mode: Login Time, and Role Update. The default value for the Login Time setting is 1,000 ms. The user will experience a 1 second delay while the system validates the login permissions. The default setting for Role Update setting is 0 ms., which means the system does not pause between validating user membership and groups. This setting is independent of the Login Time.

System Considerations

The first time a user logs on to a system, and the OS Group security mode is set, the login is validated at a domain controller. After the login is validated, a cache is created on the local machine and propagated to other nodes in the system. The user then has specific permissions to interact with the system (operator, administrator, etc.) on any node.

This scenario has several implications:

- The first time a user logs on to the system, they may experience delays while the system validates their permissions and creates the cache. This is especially relevant if the system includes a large number of OS

groups and/or network nodes. This delay may be exacerbated by widely-distributed networks (see the last bullet).

- Subsequent logins in the system use the (local) cache created at the previous login. This means that if login permissions are modified, the user can still log on, but uses the "old" cache until the update occurs. This update operation takes place "under the hood" and does not prevent the user from logging in with the old permissions.
- If the Login Time is set to 0, the system validates permissions and creates a new cache at each login. When the security mode has a large number of groups, and the system is widely-distributed (SCADA) with slow or intermittent network components, lengthy login delays may occur.

To mitigate login time delays

- Provide additional Domain Controllers on "this" side of potential network bottlenecks.
- Ensure the Login Time and Role Update settings are set correctly for the local environment. For example, setting the Login Time to 10,000 ms means that the user cannot interact with the system for 10 seconds, regardless of the use of the validation cache. In this case, 1,000 ms (default) is usually acceptable.

Securing the configuration environment

The System Platform IDE extends the security models of traditional industrial automation applications to incorporate the configuration actions as well as the run-time actions. When a galaxy is created, the administrator can authorize only those who need configuration permissions. Additionally, the administrator can limit the permissions for IDE users.

This capability can be used in an efficient way to manage a team of engineers who will be building an application together. For example, if a senior engineer is the person responsible for creating the project standards, a role could be created that has the capability of editing the templates. Another role could be created that does not have the permission to edit the templates, but can use the templates to build the application. A junior engineer would be given the latter role.

Each object maintains an audit trail of user actions performed against that object. This helps to track the versions being used and the progress of building the application, as well as provides a record of changes. This audit trail can be viewed in the IDE by accessing an object's properties.

Distributed COM (DCOM)

DCOM enables communication between objects on different computers—on a LAN, a WAN, or the Internet. It uses most transport protocols (TCP/IP, UDP, etc.).

Limiting the DCOM port range

Closing or limiting DCOM port ranges is a common IT recommendation and practice at the process control network layer. The intent of closing or limiting port ranges is to obscure them from attackers and various worms and malware by hiding which ports actually are open.

This simple solution is a perfectly appropriate and highly-effective security practice in some select cases, where a specific process control environment requires a specific number and type of highly-screened agents to interact

with the system.

For example, public utility production plants (like water and sewage treatment plants) often have relatively small staffs, who have limited interaction with a relatively slow-changing process control environment. The environment may also include out-of-spec process alarming and historization of data and interaction. This environment provides a secured environment that is extremely long-lived, relatively static, and highly effective.

Note: Such systems are usually designed to meet exacting specifications at the beginning of the project, and reviewed/updated to current standards when major engineering changes are integrated into the system.

However, in most production environments, closing DCOM ports may limit or stop necessary communication and telemetry needed for the parallel computing environment within a process control NETWORK or SCADA System.

Such limited communication has the potential to create conditions within the plant that can cause intermittent/ permanent loss of process data, unexpected operation, or loss of control of the environment resulting in dangerous or deadly conditions.

Further, the application of certain registry settings and the creation of specific ActiveDirectory rules and secure zones can create conditions within specific operating systems which will require reinstallation of the operating system in order to successfully roll back a previously undefined key, policy, or setting to the previous state of non-definition.

Establishing definitions for undefined registry values may render an operating system unusable in a process control environment. The only correction is to reinstall the operating system and reset the software benchmark for the affected machine(s).

The majority of port numbers between 1024 and 49151 are assigned to various individuals and organizations. A certain percentage of these ports will not be encountered on a Windows operating system. However, certain third-party vendors applications may be affected by limiting the access to those ports by using dcomcnfg.exe or specific registry entries in an untested and non-QA'd security profile (in addition to System Platform software products).

In most production environments, blocking ports will completely break the functionality, intelligence, and operational ability of the distributed computing environment.

Security recommendations summary

Modern industrial automation requires direct interaction between machines performing specific functionality within an intelligent enterprise.

Limiting the communications between production nodes across an enterprise will also limit the functionality and performance of the enterprise and in some cases make the enterprise unsupportable.

Extreme care should be exercised when modifying any communication protocols, communication channels and ports, or operating system processes and services within a process control environment. proposed changes to the process control environment should be tested and modeled on a shadow system or plant model system before being implemented within the operating PCN or SCADA System.

Redundancy

Redundancy within Application Server is achieved by deploying combinations of AppEngines and DI client objects on separate nodes (platforms). In its most basic configuration and and the one most generally used, there is one primary and one secondary node. This two-node primary and secondary configuration is natively supported by an device integration object dedicated to the task, the RedundantDIObject. Each node of the redundant pair has dual, dedicated NICs. At run time, the nodes will function as either active and standby. Note both the primary and secondary platforms can function as either active or standby. Active and standby status is set by the RedundantDIObject that links to the primary and secondary DI client objects running on the redundant platforms. If there is a failure that affects communication with the primary DI client object, the RedundantDIObject performs an automatic failover to the secondary object.

Implementing redundancy ensures continuous operation by providing an AppEngine that remains active in the event of a single system component failure. This configuration operates on the premise that one engine is in an Active State while the other is in a Standby State waiting to take control.

The following information describes redundancy in the context of Application Server.

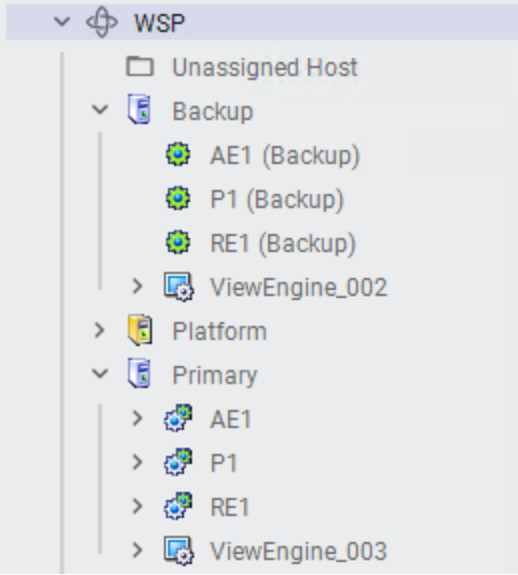
Redundant System Requirements

In a system configured for redundancy, a redundant AppEngine pair consists of one primary and one backup AppEngine. The redundant pair is configured in the AppEngine editor. The AppEngine enabled for redundancy is considered the primary engine of the redundant pair.

When the primary engine is configured for redundancy, a backup engine is automatically created on a separate platform.

Use the Operations Control Management Console, Object Viewer, or the IDE to work with the redundant engine pair. The IDE provides visualization of the redundant pair in the Deployment view pane.

The following figure shows a pair of platforms, each configured with three redundant AppEngines. Note the AppEngine icons and names. The name (Backup) is appended to the redundant AppEngine that was created automatically from the primary AppEngine.



Redundant Engine configuration requires the following:

- Redundant pair AppEngines must be deployed to different platforms.
- Both nodes hosting the redundant AppEngine pair should run the same version and service pack levels of supported operating systems.
- The Redundancy Message Channel (RMC) of each platform must be configured by assigning the corresponding IP address in the platform editor.

Best Practice

Platforms hosting primary and backup AppEngines must have identical configurations for the following elements:

- Software providing or getting data from/to the ApplicationObject Server, i.e. SuiteLink, DDE, OPC Servers, etc.
- Store and Forward directories
- Common user-defined attributes
- Common scripting
- Warm redundancy setting

  Note: For more information on attributes and scripting, see Templates.

Changing the default platform and AppEngine settings depends on the size of the system, the number of I/O points, and other variables.

Detailed information on tuning the Platform and Engine settings is included in Support Article 22407: Fine-Tuning AppEngine Redundancy Settings.

AppEngine Redundancy States

The deployment sequence (Cascade, Primary First, or Backup First) of the AppEngine pair determines which AppEngine takes the Active State.

When AppEngines are deployed individually, the first engine deployed takes the Active state while the second engine deployed takes the Standby state. The engines maintain their states until a failure occurs or there is forced failover event.

Beginning with System Platform version 2020 R2 SP1, AppEngines include an option for enabling warm redundancy. This option is located on the **Redundancy** page of the AppEngine object. Enabling warm redundancy is recommended. There are, however, some cases in which you should not enable warm redundancy. For existing redundant engines that use startup scripts, test before enabling warm redundancy to ensure that the scripts work without impacting system behavior and performance.

- When warm redundancy is enabled, the Standby engine is initialized and started at the same time as the Active engine. The Standby engine runs offscan, while the Active engine runs onscan. This allows much quicker failover from the Active to the Standby engine.
- When warm redundancy is disabled, the Standby engine does not perform the initialization and start up steps until failover occurs. When failover occurs, the Standby engine then goes through initialization and startup states. Once the engine starts, it can then transition to onscan and become active.

If either engine is deployed by itself, it assumes the Active Engine state. In a cascade deploy from the galaxy object, when the primary AppEngine is available it becomes active while the backup AppEngine goes to standby.

If a network communication problem or a failure (such as computer hardware loss or failure) occurs, the standby AppEngine assumes the active state and the engine that was in the active state may assume the standby state. When the cause of the failure has been remedied, this engine assumes the standby - ready state.

For more information on redundancy, see the *Application Server User Guide*.

Redundancy Configuration

Redundant AppEngines

WinPlatforms hosting redundancy-enabled AppEngines must run on the same operating system.

For redundancy to function properly, WinPlatforms hosting redundancy-enabled AppEngines must be deployed to computers running the same operating system.

Multiple NICs

In general, multiple NIC configuration is recommended only for redundancy purposes. Using 1GB network cards in combination with managed switches should be sufficient for most process network throughput needs.

If any nodes in the System Platform environment, other than redundant Application Server nodes, have multiple NICs, be aware that proper configuration of those computers is essential to successful communication between System Platform nodes.

In other words, if a PC has two network cards and will have a platform deployed to it, then the network binding order must have the System Platform network as the first network even if one of the network cards is disabled.

Information about configuring multiple NIC computers is included in *Application Server User Guide*.

The settings for network binding order are described in the following TechNote: Network Setup for AppEngine Redundancy.

NIC Configuration: Redundant Message Channel (RMC)

Redundant AppEngine functionality requires two computers, each with two Network Interface Cards (NIC). The first network card is for the Supervisory network; the second card is for the Redundancy Message Channel (RMC).

The RMC is a dedicated ethernet connection between the platforms hosting redundant engines. The RMC is vital to keep both engines synchronized with alarms, history, and checkpoint items from the Active engine. Each engine also uses this Message Channel to provide its health and status information to the other.

Note: Access Network Connections properties from the Windows Control Panel.

Primary Network Connection

The NIC cards require the following configuration on both nodes:

To configure the Primary network connection

1. In the **Network Connections** window, right-click **Primary Network** and select **Properties**.
2. Select **TCP/IP** and configure the **Properties** to obtain either dynamic or static IP address.
3. Configure the remaining parameters as appropriate, i.e. DNS, WNS, etc.

RMC Network Connection

To configure the RMC connection

1. In the **Network Connections** window, right-click **RMC Network** and select **Properties**.
2. Select **TCP/IP** and click the **Properties** button.
3. Select **Use the following IP address**. See your network administrator for IP address and subnet mask. The IP address must be fixed and unique.
4. In the **TCP/IP Properties** dialog box, click the **Advanced** button, then select the **DNS** tab. Be sure that the **Register this connection's address in the DNS** checkbox is not checked.

Best Practice

Assign a descriptive name to each network connection to easily identify its functionality. From the Network Connections window, rename the Local Area Connections, for example, as "Primary Network" and "RMC Network."

To assign Network Services primary connections

1. In the Network Connections window, select **Advanced/Advanced Settings** from the main menu.
2. Select the **Adapters and Bindings** tab.
3. Set the **Primary Network** as the first connection to be accessed by network services. Use the Up/Down Arrow buttons to re-order the list.
4. Verify that the normal connection between the redundant pair uses the primary network. This is done using the PING command (from the DOS Command Prompt) with the redundant partner's node name. Verify that the node name resolves to the IP Address of the partner's primary network card.

Redundant DIObjects

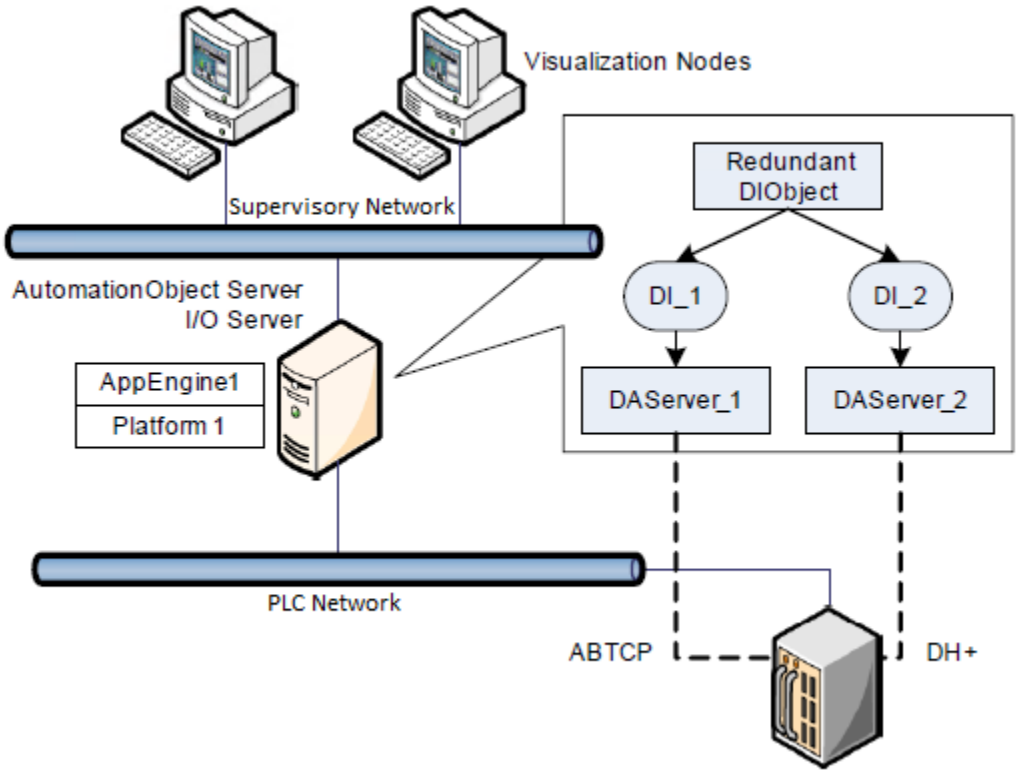
The following section explains implementing redundant DIObjects.

Configuration

AppEngines can host redundant Device Integration Objects (DIObjects). The Redundant DIObject is a DINetwork Object used to enable continuity of I/O information from field devices.

The redundant DIObject provides the ability to configure a single object with connections to two different data sources. If the primary data source fails, the redundant DIObject automatically switches to the backup data source for its information.

There is a one-to-two relationship between an instance of the redundant DIObject and the running instances of the source DIObjects. That is, for each redundant DIObject, a pair of source DIObjects is deployed.

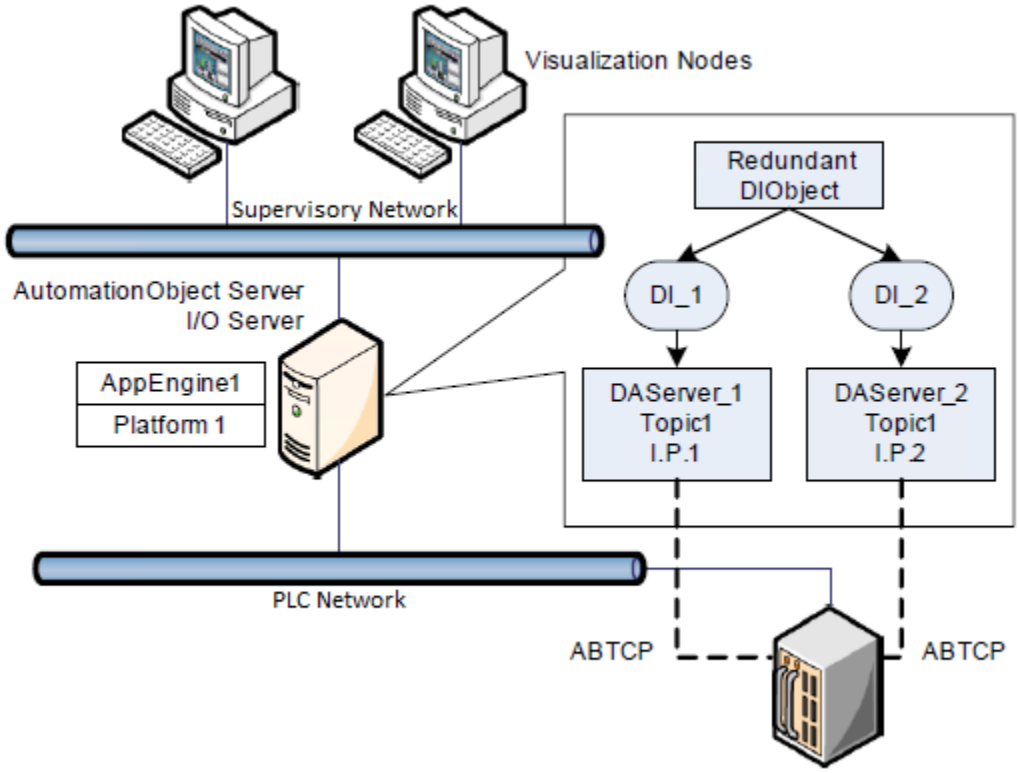


The following naming practices are recommended for implementing redundant DI client objects. The OPCClient object is typically used as the DI client object, since it includes an attribute (Hierarchy Path) that allows it to be easily configured to emulate the naming structure of many commonly-used PLCs.

- If the Communication Driver resides on the same node as the AppEngine hosting the DI client object, configure the server node name in the General tab as <Blank> or <Localhost>.
- If the Communication Driver resides on a remote node, any node name is acceptable as it refers to the same remote node regardless of where the DI client object is located.

In the previous example, the PLC sent data using two unique protocols. It is also common for the PLC to send data through two ethernet ports using the same protocol, via different IP addresses. In this case, the following

redundant DIObject configuration is recommended:



The figure shows two unique DAServer instances, each using the same topic and pointing to a unique IP address. The DAServers in this scenario can be also be deployed on different machines.

Common Configuration Requirements

Redundant DIObject configuration requires the following:

- Source DIObjects do not have to be of the same type, but must support the same type of Scan Group and have the same items address space. The Scan Groups configured in the Redundant DIObject must also be configured in both the Primary and Backup DIObjects.
- The configuration must include at least one Scan Group.
- The names of the Primary and Backup DIObject must be different. The Primary DIObject attribute refers to the name of the DIObject that will be used as the primary source of I/O attributes.

The Redundant DIObject supports creating and configuring Scan Groups, BlockReads, and BlockWrites. The Redundant DIObject can have configurable I/O points (Tag dictionary), which are periodically scanned for their value. The redundant DIObject supports Subscription, Read Transaction, and Write Transaction on I/O points.

Redundant DIObject Behavior at Run-Time

After the redundant DIObject is initialized, its state changes to Startup. The object opens MX communications and registers a reference to ScanState to track whether the DIObject is deployed. If the DIObject is off scan, the redundant DIObject treats it as a bad data source.

The ProtocolFailureCode and ConnectionStatus attributes provide status of the source device. During run-time,

the redundant DIObject performs the following tasks:

1. Adds newly activated attributes to the Active DI source.
2. Updates attributes with new values from the Active DI source.
3. Monitors the connection with the Active and Standby DI source. If the connection to the Active DI source is lost, the object switches to the Standby DI source.

If both DI sources are in bad state, the object raises the Connection Alarm.

Redundant Configuration Combinations

Multiple redundant configuration combinations are possible. The combinations include redundant AppEngines and Redundant DIObjects.

It is important to select the configuration that provides the best performance and robustness.

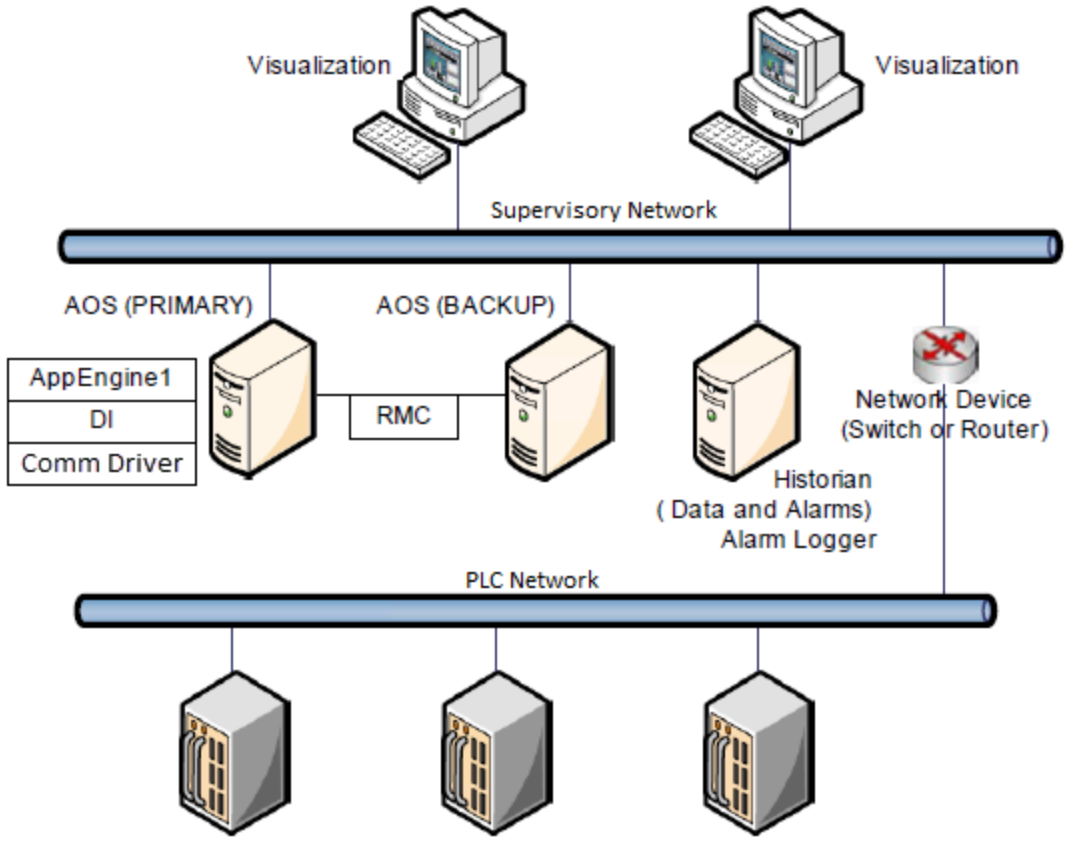
The following examples present recommended configurations for the Dedicated Standby and Load Shared scenarios.

Dedicated Standby Server - No Redundant I/O Server

This configuration includes a dedicated standby node ready to take control of the system when the active engine is off-line (refer to the following figure).

AppEngine1 hosts all AppObjects as well as DI client objects. The I/O Server is installed on both nodes but the DI client object collects data from the node where the active engine resides.

To provide a higher degree of reliability to the system, you can implement a script in the DI client object to set the Redundancy.ForceFailoverCmd attribute in the AppEngine object to True when the connection with the PLC fails.



Load Sharing Configurations

Load sharing distributes the system's processing load between two nodes. The two nodes can also be configured to backup each other.

The primary benefit of load sharing is to reduce failover time: only one half of the objects must fail over.

A system with redundant AppEngines can host Active and Standby Engines of different redundant pairs on the same node. This configuration enables efficient use of system resources while providing high availability for both nodes.

However, configuration of redundant engines on the same node requires thorough evaluation of the following critical (Performance Monitor) counters: CPU and memory use on each node.

Ensure each AOS node runs at a maximum of 25-30% CPU load. This CPU load represents the resources used by each AOS in steady condition considering Active and Standby engines running in each node as well as any other applications. When a failover occurs one of the AOS nodes will host all the Active engines in the pair. Hosting implies a CPU load increase to a total of 50-60%.

It is important to note that although the computers hosting the redundant AppEngines do not have to be identical, the total load of the application must not exceed 60% of CPU usage of the smallest computer.

Load Shared - Non Redundant I/O Data Source - Using DIObjects

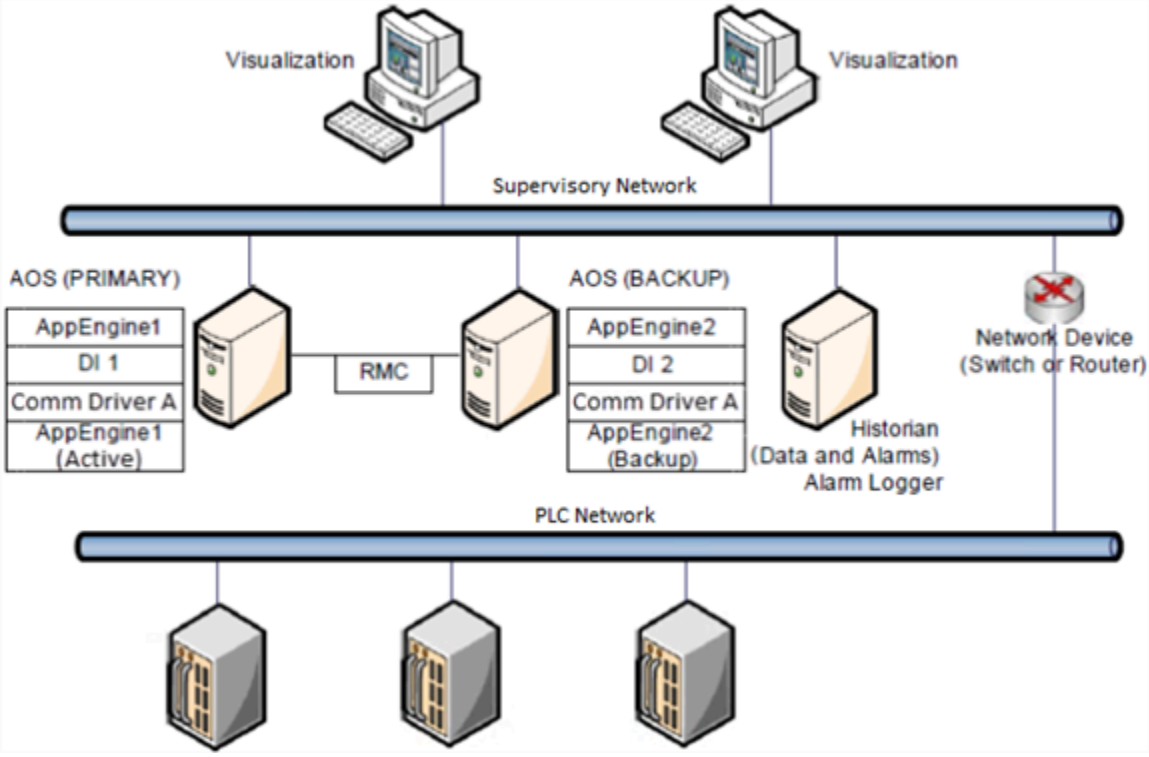
Both AOS nodes host active and backup engines for each other. If one node fails, the remaining one hosts all active engines for both nodes. The following figure shows System Platform communication drivers with OPC

Client or DDE/Suitelink client objects (DI client objects).

When both AOS nodes use the same communication driver to communicate with the PLC, the DI client object in the new active engine refers to the server running in that node when the failover occurs. The DI client objects on each node are configured to point to the local communication driver, leaving the node name blank in the DI client object editor. When a failover occurs, the DI client object in the new active engine will refer to the local existing instance of the communication driver that is currently running in that node.

When the AOS nodes use different communication drivers, both drivers must be installed on each node. One of the communication drivers will provide data to the local active DI client object (i.e. DI1), while the other server feeds data to the other DI client object (i.e. DI2) after the failover.

Use the Redundancy.ForceFailoverCmd AppEngine attribute in a script (in the corresponding DI client object) to trigger the failover in the event of a communication failure with the PLC Network.



Load Shared - Redundant I/O Data Source

This variation of the load shared configuration includes a set of redundant nodes running the I/O Servers.

AOS1 and AOS2 are configured as redundant pairs in a load shared scenario. In the event of a failure, one of the nodes hosts all AppObjects in both servers. This set up provides high availability for the execution of AppObjects.

Alternatively, the I/O data level is protected by implementing a pair of remote I/O Server nodes. Each server hosts the corresponding DI client object and I/O Server (DDE/Suitelink or OPC servers) and communication drives.

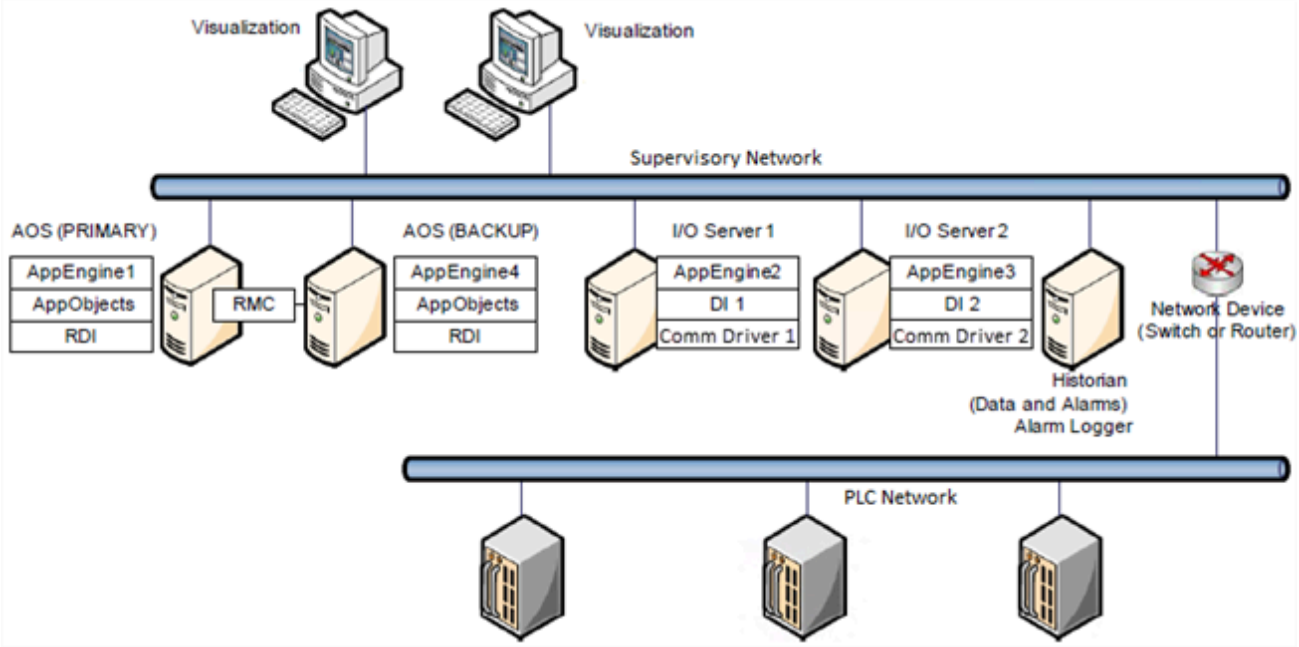
AppObjects on each engine reference I/O points in the local RDIObject. The RDIObjects switch between I/O Server nodes if a failure in any of those nodes occurs.

Using redundant I/O data sources provides the following benefits:

- The communication protocol between the I/O Server node and the AOS is MX. This protocol is optimized for

data transfer over the network, with special emphasis on slow and intermittent networks.

- As the I/O Server uses MX protocol to transfer data, it simplifies configuring OPC communication over the network, and overcomes the deficiencies of DCOM communications.
- Using a platform and an AppEngine on those nodes provides additional diagnostic of conditions associated to the system. These can be historized and alarmed in the same way as with AppObjects.



Run-Time Considerations

The following information summarizes run-time behaviors between redundant engines.

Establishing RMC Communication

The active and standby engines communicate with each other during run-time and use the RMC to monitor each other's status. The redundant engines use the Remote Partner Address (RPA) attribute to locate each other and communicate. The RPA attribute contains the IP address or host name of the platform hosting the partner engine.

At startup, each redundant AppEngine establishes communication with its partner. When the failover service receives a connection across the RMC, it updates the RPA attribute of the receiving engine if it is different than the current configured value.

Note: The value of the RPA may be different if a partner engine has been relocated to a different platform.

Checkpointing

AppEngines store specific attributes in memory, then write them to disk in both single- or redundant engine configurations. The frequency of the write operation is determined by the Checkpoint period setting in the AppEngine editor.

Note: Checkpoint period configuration details are included in Tuning Redundant Engine Attributes.

The checkpointed attribute types include:

- Scan rate
- Checkpoint directory location (default is blank but can be modified)
- StartUp attributes
- StartUp type (automatic, semi-automatic, manual)
- StartUp Reason

Redundant AppEngines maintain data synchronization through the RMC. Data is synchronized by reading checkpointed attribute values written to disk on each node, at each scan.

The checkpoint operations occurs at a pre-defined rate in the local node (Scheduler.ScanPeriod). The same operations (write to memory, then to disk) occur on the backup AppEngine at every scan (via the RMC).

When the Standby AppEngine becomes active, it reads the checkpointed values from the designated file in the local (backup) node. The system updates the Standby Engine with the values that were sent and written to disk in the last scan before the active engine failed.

The following attribute types are checkpointed:

- Attributes with Category User Writable or Object Writable that are not extended as Input/Output and Input extensions.
- Calculated Retentive category attributes are always checkpointed.

Complete a thorough evaluation and selection of checkpointed attributes. Unnecessary checkpointing may degrade the performance of the system by writing extra values to the local/backup disks, and increase data traffic over the RMC.

Deployment Considerations

Objects are always deployed to the active engine:

- If the primary engine is the currently-active engine of a redundant pair, objects are deployed to the primary engine.
- If the backup engine is the currently-active engine, the objects are deployed to the backup engine.

When an active engine becomes the standby, the engine sets all objects off scan, shuts down all features that make up the object and stops executing all deployed objects. All objects are unregistered on the previously active engine.

When a standby engine becomes active, the engine calls startup on all features that make up the objects. The call-up includes a method that shows the objects are starting up as part of a failover. The newly active engine calls SetScanState on all features and begins executing all objects that are on scan.

Best Practice

To deploy objects in a Load Shared configuration

1. Deploy the platforms individually rather than in a cascade.

2. Cascade deploy the primary engines.
3. Finally, cascade deploy the backup engines. Always deploy the primary engine first.

Scripting Considerations

- When failover occurs, attribute values keep their initial states.
- Scripts and SQL connections to databases that were interrupted by the failover must be restarted.
- OffScan scripts are executed in the event of a forced failover.
- Any state, such as local variables or calculated attributes, that is not kept in checkpointed attributes is not passed to the objects started on the newly-active engine.
- If an attribute value is being passed to the database when failover occurs, the attribute returns to its initial value when the object goes On Scan in the new active Engine.
- Before the attributes can be updated, the database connection must be restored.

Script Behavior When the Standby Engine Becomes Active

When a standby engine becomes active, it sets the Engine.StartupReason attribute to indicate the startup cause. The attribute string can be accessed in a script to determine the startup reason. The following reasons are possible:

- Starting_AfterDeploy: Engine starts from standard deploy.
- Starting_From Standby: Engine starts from failover.
- Starting_FromCheckpoint: Engine starts from reboot or AutoStart configuration.

These attributes can also be used to execute scripts that re-initialize variables and COM objects.

After a failover occurs, scripts in the new Active engine are executed based on the trigger type they use i.e. Startup, On Scan, Periodic and Data Change.

Use Startup and OnScan scripts to initialize conditions used later in the script. In many cases the initialization is required only when the object is deployed/redeployed, or when the AppEngine and or platforms are restarted. In the case of a failover, the requirement may be to continue operating using values from the checkpoint rather than re-initializing the conditions.

The Redundancy.FailoverOccurred attribute is set to "True" for the first scan right after the failover occurs; after the first scan the attribute is automatically reset to "False." Using this attribute as a script condition initializing the variables prevents the script from running when the system recovers from a failover.

Similarly, Data Change scripts execute when the object is deployed, the engine re-started and the Standby engine becomes active after a failover. Using the Redundancy.FailoverOccurred in an "If-then-else" statement will prevent the script from executing after the failover.

Any script that is set with an Execution type of Execute and a trigger type of Periodic will have the following behaviors after an AppEngine failover. The situation is described using a period of 60 minutes as an example time period:

- The script executes the first time when the engine is deployed. E.g. T0.
- The next execution time will be 60 minutes later. E.g. T60.
- The redundant engine fails over and the Standby engine is fully running in the Active state at T30.

- The Periodic script(s) will restart with the period reset to T0.

The period for the execution of the Periodic script(s) will be shorter than planned for, or possibly longer if an engine failover occurs shortly before the time period elapses.

Some applications may have critical data generated by a Periodic script. Do not use Periodic type scripts where the time period could be shorter, or longer, and this is critical information being managed by this script. Instead, set up the script to run using a condition and set up an attribute for the trigger.

Then for a time base, use System Time and calculate the time period from this to set the condition. The attribute current value is maintained in the failover and the System Time is real time versus an expiring timer.

Shutdown and OffScan scripts will execute after an orderly completed failover, such as using ForceFailoverCmd, or in the event of Primary Network failure.

Asynchronous Scripts

QuickScripts must be evaluated to anticipate likely delays (SQL Query completion, calling COM or .NET objects, etc.) due to network transport or intensive database processing. When a delay in script completion is likely, set the QuickScript to run asynchronously.

If not set to run asynchronously, it is possible that the non-asynchronous QuickScript could cause the engine to miss the following scan while waiting for the script to finish executing.

Note: The "Runs asynchronously" option must be manually selected within the Scripts tab page of the Object Editor; it is not set by default.

Once set to run asynchronously, the QuickScript will not be cut off when the scan is completed. When a problem occurs, the script could "hang" if the process never completes, as in the case of a SQL query that never returns a rowset or even an error message. When the QuickScript's ExecuteTimeout.Limit value is reached, the ExecuteError.Alarmed and ExecuteError.Condition attributes are set.

In this context, it is useful to monitor these attributes and log a message when the maximum timeout threshold is exceeded.

History

The Historian receives data only from the active engine. The active engine processes historical data and sends it to the Historian when the Historian is available. If the Historian becomes unavailable, the active engine stores the data locally (in Store Forward History Blocks) and forwards it when the Historian becomes available.

In the meantime, local Store Forward data is transferred to the standby engine via the RMC. When an engine enters Store Forward mode, it synchronizes its data with its partner engine. Store Forward data is transferred (and synchronized) every 30 seconds, so no more than 30 seconds can be lost in the event of an engine failure.

Note: Attributes and tags which were not configured in the Historian before failover are not stored.

History: Redundancy Diagnostics

In order to facilitate management of Store Forward data collected across multiple failures and to improve diagnostics, the active engine has attributes which show the status of Store Forward.

The following information is available:

- Store Forward data has been collected for engine: Engine.Historian.InStoreForward_Standby

- Store Forward data lost: Engine.Historian.StoreForwardDataLost_Standby
- Store Forward data cannot be stored on active engine in store forward mode: Engine.Historian.StoreForwardProblem_Standby

Failover Causes in Redundant AppEngines

This section describes failover triggers in a redundant AppEngine pair.

Forcing Failover

It is possible to force a failover in a pair of redundant AppEngines by simply setting the attribute ForceFailoverCmd in the active engine to "true." This can be accomplished using the ObjectViewer, an object script or any other application that has access to this attribute.

Use this attribute in a script (with any set of conditions) to trigger a failover. For example, you can monitor the status of other applications on the same machine, hardware devices, etc. and based on that status, trigger a failover to the standby engine.

When a failover occurs, the Standby engine becomes Active and stays in that status unless the system is forced to fail back when the new Standby engine becomes available. In this case, the ForceFailoverCmd can be used to take the Active engine back to the original node.

For details on the attributes associated to a Redundant AppEngine please refer to the AppEngine Help files.

Communication Failure in the Supervisory (Primary) Network

To understand the effects of a communication failure at the Supervisory Network level, refer to the matrix on the following page. It presents the original status of the redundant pair before the failure as well as the cause of the problem and the final condition.

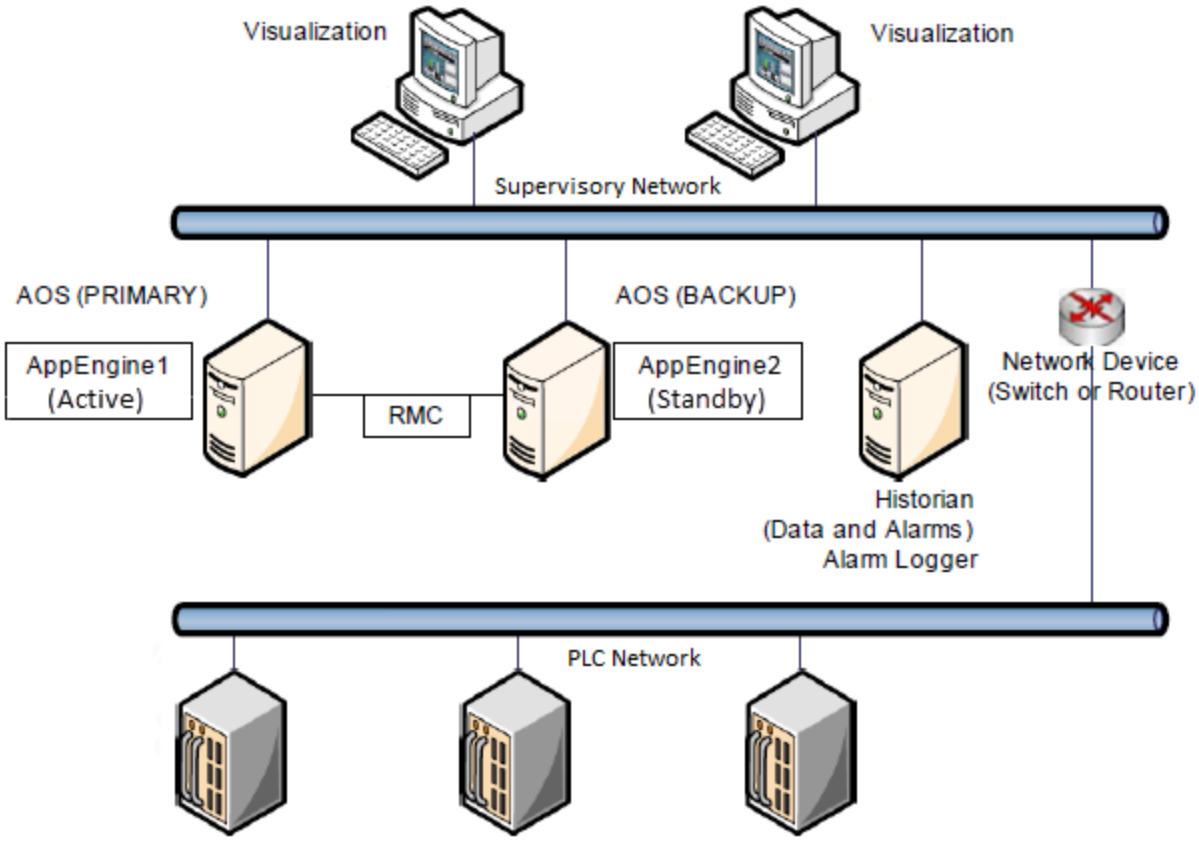
The matrix shows the values of two attributes in the AppEngine used to monitor the status of the redundant pair. They are RedundancyPartnerStatus and RedundancyStatus. The attributes (and other key attributes) are included.

Considerations

The failover scenarios described in this section refer to topologies where there is at least one more platform besides the two hosting the redundant pair, i.e. client/server configuration.

If the topology consists of just two platforms hosting the redundant pair (peer-to-peer configuration) a failover does not occur in the event of a communication failure in the supervisory network. Instead, the Redundancy.PartnerStatus attribute is set to missed heartbeats while both partners synchronize data through the RMC.

In this case, the user can execute the failover either manually or via scripting, if required.



	Initial Condition		**Transition**		**Final Condition**	
Primary	**Backup**	**Primary**	**Backup**	**Primary**	**Backup**	
Scenario 1a						
Primary Network	Connected	Connected	Disconnected	Connected	Disconnected	Connected
Red.Partner Status	Standby Ready					Missed Heartbeats
Red. Status	Active					Active
Scenario 1b						
Primary Network	Disconnected	Connected	Connected	Connected	Disconnected	Connected
Red.Partner Status		Missed Heartbeats				Stanby Ready
Red. Status		Active				Active

	Initial Condition		Transition		Final Condition	
Scenario 2						
Primary Network	Connected	Connected	Connected	Disconnected	Connected	Disconnected
Red.Partner Status	Standby Ready				Missed Heartbeats	
Red. Status	Active				Active	
Scenario 2b						
Primary Network	Connected	Disconnected	Connected	Connected	Connected	Connected
Red. Partner Status	Missed Heartbeats				Standby Ready	
Red. Status	Active				Active	
Scenario 3						
Primary Network	Connected	Connected	Disconnected	Disconnected	Disconnected	Disconnected
Red.Partner Status	Standby Ready				Missed Heartbeats	
Red. Status	Active				Active	

RMC Communication Failure

Even though an RMC communication failure does not trigger failover, it is important to know the system behavior in this event:

	Initial Condition		Transition		Final Condition	
Primary	**Backup**	**Primary**	**Backup**	**Primary**	**Backup**	
Scenario 1						
Failure in RMC	Connected	Connected	Connected	Disconnected	Connected	Disconnected

	Initial Condition		Transition		Final Condition	
Red. Partner Status	Standby Ready				Unknown	
Red. Status	Active				Active - Standby not Available	
Scenario 2						
Failure in RMC	Connected	Connected	Disconnected	Connected	Disconnected	Connected
Red. Partner Status	Standby Ready				Unknown	
Red. Status	Active				Active - Standby not Available	

PC Failure

If a power failure occurs on the Active Engine node, the Standby node takes control of the system. The following matrix shows the corresponding status and AppEngine attribute values under different conditions:

	Initial Condition		Transition		Final Condition	
Primary	**Backup**	**Primary**	**Backup**	**Primary**	**Backup**	
Scenario 1						
PC Failure	PC Available	PC Available	PC Available	PC Not Available	PC Available	PC Not Available
Red. Partner Status	Standby Ready				Unknown	
Red. Status	Active				Active - Standby not Available	
Scenario 1b						
PC Failure	PC Available	PC Not Available	PC Available	PC Available	PC Available	PC Available
Red. Partner Status	Unknown				Standby Ready	

	Initial Condition		Transition		Final Condition	
Red. Status	Active - Standby Not Available				Active	
Scenario 2						
PC Failure	PC Available	PC Available	PC Not Available	PC Available	PC Not Available	PC Not Available
Red. Partner Status	Standby Ready					Unknown
Red. Status	Active					Active - Standby not Available
Scenario 2b						
PC Failure	PC Not Available	PC Available	PC Available	PC Available	PC Available	PC Available
Red. Partner Status		Unknown				Standby Ready
Red. Status		Active - Standby Not Available				Active

Undeploying AppEngines

Undeploying AppEngines in a redundant pair may trigger failover. The following description refers to non-cascade Undeploy operation of the AppEngine.

Executing a cascade Undeploy operation of the Primary AppEngine undeploys all objects from both engines.

The table below describes the expected behavior under the non-cascade condition:

	Initial Condition		Transition		Final Condition	
	Primary	Backup	Primary	Backup	Primary	Backup
Scenario 1						
Undeploy Backup AppEngine	Deployed	Deployed	Deployed	Undeployed	Deployed	Undeployed
Red. Partner	Standby				Unknown	

Status	Ready					
Red. Status	Active				Active - Standby not Available	
Scenario 2						
Undeploy Primary Engine	Deployed	Deployed	Undeployed	Deployed	Undeployed	Deployed
Red. Partner Status	Standby Ready					Unknown
Red. Status	Active					Active - Standby not available
Scenario 3						
Deploy Primary Engine	Undeployed	Deployed	Deployed	Deployed	Deployed	Deployed
Red. Partner Status		Unknown				Unknown
Red. Status		Active - Standby not available				Active

Note: It may become necessary to relocate a Primary or Backup AppEngine. Ensure relocation is performed at a non-critical time (scheduled maintenance, plant shutdown, etc.), and that Store Forward is not in operation during the relocation process (undeploy, relocate, redeploy). Data loss may occur if Store Forward is operational during the redeploy operation.

Dual Communications Channel Failure Consideration

A redundant AppEngine system configuration provides high availability in the event of a single communication channel failure. This means that the system will gracefully handle situations where only one communication channel fails at a given time; for example, PC failure, Primary network failure, etc.

The role of the RMC as a dedicated link to synchronize data and monitor the status of the redundant pair minimizes the chances of failures within the system, since it is a dedicated cross-over cable between two computers and does not carry external data traffic.

If the active AppEngine simultaneously loses the connection to the pPrimary network and to the RMC, the system reverts to an ACTIVE-ACTIVE state.

To recover from an ACTIVE-ACTIVE condition, both connections must be reestablished. The system arbitrates assigning the Active state, so that the Primary server will become Active, regardless of its state before the

ACTIVE-ACTIVE condition.

It is possible to design the application to handle the ACTIVE-ACTIVE condition. The solution includes scripting and a third component to monitor the status of both engines. For example, a condition script could be implemented in the PLC to monitor the status of both engines and arbitrate which one stays active if a simultaneous failure occurs in the RMC and primary network.

Redundant System Checklist

Refer to the following checklist when planning your redundant system:

Determine the Redundant System Configuration

Evaluate what type of redundant configuration is a better fit for your system. Both the Dedicated Standby and Load Shared configurations provide a reliable and robust solution, but depending on the process requirements and system architecture, one of the configurations may be more efficient than the other.

Analyze the Expected System Behavior After Failover

It is very important to understand how scripts are executed after a failover. Scripting Considerations explains how scripts behave in a failover condition.

Identify which attributes need to be retentive after a failover. See Checkpointing Attributes for additional information.

Distribute Data Traffic

To make the best use of network bandwidth, distribute the traffic coming in and out of the ApplicationObject Servers over different networks. For example, the need for data from the control network may be more critical than the data requirements of the supervisory system. Distributing data traffic through separate networks requires the use of multiple network cards in a server.

Note: Although distributing data traffic through separate networks is not required, multiple NICs are used for optimal communication in the network figures that appear in this chapter.

Rename Each Local Area Connection When Using Multiple Network Adapters

The operating system detects network adapters and automatically creates a local area connection in the Network Connections folder for each network adapter.

Renaming each local area connection to reflect its network eliminates confusion. Add or enable the network clients, services, and protocols required for each connection. When doing so, the client, service, or protocol is added or enabled in all other network and dial-up connections.

Tuning Recommendations for Redundancy in Large Systems

To ensure seamless failover in "Large" systems (high I/O, high CPU utilization, SCADA System) performance, modify several default Platform attribute values. See Tuning Redundant Engine Attributes for more informaton.

Note: The following information applies to a large, locally distributed topology. SCADA-specific settings are described in Inter-node communications.

Tuning Redundant Engine Attributes

Multiple variables (I/O points, number of objects, number of historized attributes, DIObject distribution, etc.) are involved in the detection and execution of a Redundant AppEngine Failover. The following tables describes some key Engine attribute values that can be modified to ensure proper failover performance.

AppEngine Object Settings

Parameter	Forced failover timeout
Editor Tab	**Redundancy**
Attribute	Redundancy.ForcedFailoverTimeout
Description	The maximum allowed time, in milliseconds, for a standby engine to become active after a forced failover has been initiated using the ForceFailoverCmd attribute. If the standby engine does not become active within this time period, the engine reverts to the active engine.
Default	90,000 ms (90 seconds)
Tuning	30,000 ms (less than 3,000 I/O) 45,000 to 240,000 ms (from 3,000 I/O at to 40,000 I/O) 300,000 ms (more than 40,000 I/O)
Notes	I/O values represent the load on the individual AppEngine, not the Galaxy size. If setting is too small, forced failover will not succeed. If setting is too large, failure will not be detected in a timely manner. Tuning values represent a range that can be adjusted as required.

Parameter	Maximum checkpoint deltas buffered
Editor Tab	Not shown, edit **Attribute** value if necessary
Attribute	Redundancy.CheckpointDeltasBufferedMax
Description	The maximum number of checkpoint deltas that can be buffered before a full checkpoint synchronization is

	performed.
Default	0
Tuning	N/A
Notes	N/A

Parameter	Maximum alarm state changes buffered
Editor Tab	Parameter not shown, edit **Attribute** value if necessary
Attribute	Redundancy.AlarmStateChangesBufferedMax
Description	The maximum number of alarm state changes that can be buffered before a full snapshot of the alarm state changes for the engine is performed.
Default	0
Tuning	N/A
Notes	N/A

Parameter	Active engine heartbeat period
Editor Tab	**Redundancy**
Attribute	Redundancy.ActiveHeartbeatPeriod
Description	The time interval, in milliseconds, at which heartbeats are sent by the failover service on the active engine to the failover service on the standby engine via RMC.
Default	1000 ms (1 second)
Tuning	May be increased to avoid false failovers.
Notes	N/A

Parameter	Standby engine heartbeat period
Editor Tab	**Redundancy**
Attribute	Redundancy.StandbyHeartbeatPeriod
Description	The time interval, in milliseconds, at which heartbeats are sent by the failover service on the standby engine to the failover service on the active engine via RMC.

Default	1000 ms (1 second)
Tuning	May be increased to avoid false failovers.
Notes	N/A

Parameter	Maximum consecutive heartbeats missed from Active engine
Editor Tab	**Redundancy**
Attribute	Redundancy.ActiveHeartbeatsMissedConsecMax
Description	The maximum number of heartbeats from the active engine that can be missed before a bad connection is assumed by the standby engine via RMC. For example, if the maximum consecutive heartbeats missed from active engine is configured as 5, and the active engine heartbeat period is configured as 1000 milliseconds, then the standby engine will assume a bad connection from the active engine if no heartbeats are received within five seconds.
Default	5
Tuning	5 (less than 3,000 I/O) 10 to 30 (from 3,000 I/O to 40,000 I/O) ~60 (more than 40,000 I/O)
Notes	I/O values represent the load on the individual AppEngine, not the Galaxy size. Setting this value too low produces false failovers. Setting this value too high results in slow detection of a required failover.

Parameter	Maximum consecutive heartbeats missed from Standby engine
Editor Tab	**Redundancy**
Attribute	Redundancy.StandbyHeartbeatsMissedConsecMax
Description	The maximum number of heartbeats from the standby engine that can be missed before a bad connection is assumed by the active engine. If a bad connection is detected, the active engine will switch to the "Active - Standby Not Available" state via RMC. For example, if the maximum consecutive heartbeats missed from the standby engine configured as 5, and the standby

	engine heartbeat period is configured as 1000 milliseconds, then the active engine assumes a bad connection from the standby engine if no heartbeats are received within five seconds.
Default	5
Tuning	5 (less than 3,000 I/O) 10 to 30 (from 3,000 I/O to 40,000 I/O) ~60 (more than 40,000 I/O)
Notes	I/O values represent the load on the individual AppEngine, not the Galaxy size. Setting this value too low produces false failovers. Setting this value too high results in slow detection of a required failover.

Parameter	Maximum time to maintain good quality after failure
Editor Tab	**Redundancy**
Attribute	Redundancy.StandbyActivateTimeout
Description	The maximum time period, in milliseconds, after the active engine fails before subscribed references to it are set to "uncertain."
Default	15,000 ms (15 seconds)
Tuning	15,000 ms (less than 3,000 I/O) 120,000 ms (from 3,000 I/O to 40,000 I/O) 150,000 ms (more than 40,000 I/O)
Notes	I/O values represent the load on the individual AppEngine, not the Galaxy size. Assuming remote I/O, setting the value too low causes all I/O references to unsubscribe, then resubscribe on failover. The optimum setting ensures that remote I/O references are preserved for failover. This behavior also applies in the RDI Object context.

Parameter	Maximum time to discover partner
Editor Tab	**Redundancy**
Attribute	Redundancy.PartnerConnectTimeout

Description	The maximum time period, in milliseconds, allowed for the connection to the failover partner to be established before the failover partner state is set to "unknown."
Default	15,000 ms (15 seconds)
Tuning	N/A
Notes	N/A

Parameter	Restart engine when it fails
Editor Tab	Parameter not shown, can be viewed in **Attribute** tab
Attribute	Engine.RestartOnFailure
Description	The AppEngine object automatically attempts to restart if a failure occurs.
Default	True
Tuning	N/A
Notes	This behavior cannot be changed, even if the attribute is set to false.

Parameter	Checkpoint period
Editor Tab	**General**
Attribute	Scheduler.CheckpointPeriod
Description	Checkpointing saves run-time attribute values. The checkpoint period is the time, in milliseconds, at which checkpointing is performed. The default checkpoint period is 10,000 ms. If set to 0, the checkpoint period defaults to the scan period, but may occur at a slower rate (it is done as fast as possible as a background task). The minimum checkpoint interval for retentive attributes is 10,000 ms. Retentive attributes are defined as those attributes configured as calculated retentive, or object- or user-writeable. If the checkpoint period is set to less than 10,000 ms, retentive attributes will not be saved at every checkpoint. For example, if the checkpoint period is set to 4,000 ms, retentive attribute values will only be saved at every third checkpoint (4,000 x 3 = 12,000 ms).

	Retentive attributes retain the last value set during run time, and the run-time value is saved across redeployments. Non-retentive attributes revert to their configured values at redeployment.
Default	10,000 ms (10 seconds)
Tuning	10,000 ms (up to 3,000 I/O 20,000 ms (up to 20,000 I/O) 60,000 ms (more than 20,000 I/O)
Notes	I/O values represent the load on the individual AppEngine, not the Galaxy size. Setting this value too low results in high resource usage. Setting this value too high means that if both partners fail, checkpointed data may not be current.

WinPlatform Object Settings

Parameter	NMX heartbeat period
Editor Tab	**General**
Attribute	NetNMXHeartbeatPeriod
Description	The time interval, in milliseconds, at which heartbeats are sent to other platforms. Heartbeats will only be established between platforms if a publish/subscribe relationship exists between engines on the platforms. For example, if an engine on WinPlatformA is subscribed to data from an engine on WinPlatformB, then heartbeats will be sent between WinPlatformA and WinPlatformB. WinPlatformA will send heartbeats to WinPlatformB at the rate specified by the WinPlatformA NetNMXHeartbeatPeriod attribute. WinPlatformB will send heartbeats to WinPlatformA at the rate specified by the WinPlatformB NetNMXHeartbeatPeriod attribute.
Default	2,000 ms (2 seconds)
Tuning	Use the default value a platform object with a low

	I/O count (up to 3,000).
Notes	I/O values represent the load on individual AppEngines, not the Galaxy size

Parameter	Consecutive number of missed NMX heartbeats allowed
Editor Tab	**General**
Attribute	NetNMXHeartbeatsMissedConsecMax
Description	The maximum number of consecutive heartbeats that are allowed to be missed from a platform before a platform communication error is generated for that platform. For example, assume an engine on WinPlatformA is subscribed to data from an engine on WinPlatformB. If the NetNMXHeartbeatsMissedConsecMax attribute on WinPlatformB has a value of 5, then WinPlatformA will generate a platform communication error when it misses six consecutive heartbeats from WinPlatformB. If the NetNMXHeartbeatsMissedConsecMax attribute on WinPlatformA has a value of 2, then WinPlatformB will generate a platform communication error when it misses three consecutive heartbeats from WinPlatformA.
Default	3
Tuning	Small configuration (up to 10,000 I/O per engine): 3 Larger configurations (more than 10,000 I/O per engine): 6
Notes	I/O values represent the load on individual AppEngines, not the Galaxy size. Missed consecutive heartbeats determines the number of missed heartbeats that will trigger the redundant engine to act. Setting the values smaller makes the engines more sensitive to network failure. Setting the values larger makes the engines more tolerant of high CPU loads that can cause missed heartbeats. Specifying a value of 0 is not recommended, as this may trigger false communication errors that can deteriorate the system performance.

Failover services talk between themselves using the RMC and determine the communication status between the

two nodes. The status is provided by monitoring Heartbeat attributes.

Message Channel Heartbeat settings control the heartbeat intervals; i.e., how often the redundant platforms send each heartbeat through the RMC.

AppEngine Monitoring

The following information describes how the failover service monitors the redundant engines.

In general, an engine has the following states:

- Start Up: Measured as the time required for all engine objects to be created, initialized and started.
- Execution: Measured as the time required for all engine objects to be executed in one scan cycle.
- Shut Down: Measured as the time required for all engine objects to be stopped.

The following parameters determine how much time the engine can be unresponsive during each of the above states.

Start Up and Shut Down

If you shut down the platform from the Operations Control Management Console and the platform shuts down improperly, you may need to increase the amount of system RAM.

Execution

The EngineFailureTimeout attribute determines how long that engine has to inform the bootstrap that it is executing. If the timeout period elapses, a failover to the backup engine occurs.

Setting this attribute value too low causes the redundant partner to overreact when CPU usage is high. Setting the value too high can delay a failover to the backup engine.

The timeout period should be set to a long enough interval to accommodate the completion of any engine-related actions that might prevent or delay the bootstrap notification (for example, cascade deployment of objects with a large number of scripts). The minimum timeout the system allows at run time is 55,000 ms (default value is 30,000 ms). If a shorter timeout is configured, the configured value is ignored and the object will use the 55,000 ms minimum when it is deployed.

Maintenance

System Platform allows users to develop applications that have built-in diagnostics and maintenance functionality. For example, an Application Server platform in can provide information about system resources such as CPU load, memory, network traffic, or disk usage. System operators and supervisors can access both process data and system health information from the alarm and event database, Historian Server, OMI ViewApp, or InTouch HMI windows with links to various attributes in galaxy objects.

System Platform also accommodates system administrators, who require the ability to back up system files periodically, and to perform more in-depth diagnostics if problems occur.

This section presents diagnostic and maintenance tools available to System Platform users. For information on other resources, refer to the AVEVA Global Customer Support web site.

System Platform Diagnostic and Maintenance Tools

The following section describes diagnostic tools available for System Platform.

Object Viewer

The Object Viewer monitors the status of the objects and their attributes and can be used to modify an attribute value for testing purposes.

To add an object to the Object Viewer Watch list, you can manually type the object and attribute names into the attribute reference box in the menu bar and select Go. When prompted to enter the Attribute Type, select OK.

You can save a list of items being monitored. Once you have a list of attributes in the Watch Window, you can select all or some of them and save them to an XML file. Right-click on the Watch window to save the selection or load an existing one. You can also add a second Watch window that shows as a separate tab in the bottom of the Viewer.

Note: Do not use Ctrl + S to save the Watch window. Instead of saving the Watch window, this key combination will instead create an empty file.

Refer to the platform and engine object help for information about attributes that may indicate system health. These attributes provide alarm and statistics on how much load a platform or engine may have when executing application objects or communicating with I/O servers and other platforms.

Testing the Quality Value of Attributes

When you use Attribute Data Quality for diagnostic purposes, observe the following tips:

Best Practice - To test for the Attribute Quality Value

The actual value of a Bad and Good quality is 0 and 192, respectively. Past methods for testing the Quality value have resulted in code such as:

```
If MyObject.PV.Quality == 192 then
```

A more appropriate way to code such tests is to call one of the quality test functions available within the QuickScript language. The previous example for testing for a GOOD quality condition would be coded as:

```
If IsGood(MyObject.PV.Quality) then
```

The available functions for testing the Quality value of an attribute are as follows. The functions return a Boolean (True) value for success and a Boolean (False) for failure of the test.

Test Condition:

- IsBad
- IsGood
- IsInitializing
- IsUncertain
- IsUsable

As in the above example, the coding syntax requires the desired function (the specific attribute to test). Note that the parenthesis around the quality is required.

Best Practice - To use the Set Condition Function

The available functions for setting the Quality value of an attribute are as follows. The functions return a Boolean (True) value for success and a Boolean (False) for failure of the test.

Set Condition:

- SetBad
- SetGood
- SetInitializing
- SetUncertain

The syntax for the Set Condition functions is the same as the Test Condition functions except that the attribute to be SET must be an attribute within the object that the script is attached to.

Example:

```
SetInitializing(me.PV)
```

For more information on Attribute Data Quality, see Templates.

Operations Control Management Console (OCMC)

Galaxy Database Manager

The Galaxy Database Manager is a System Platform utility you can use to manage your Galaxies. Use it to back up your Galaxies so that if a Galaxy becomes corrupted, you can restore the Galaxy. You can also use a backup to reproduce a Galaxy on another computer

The Galaxy Database Manager is automatically installed on every Galaxy Repository node. It lets you view all the galaxies in the Galaxy Repository, as well as the nodes they reside on.

Operations Integration Server Manager

The Operations Integration Server Manager allows local or remote configuration of the Operations Integration Server and its device groups and items, and can monitor and perform diagnostics on communication driver communication with PLCs and other devices.

Important! Like the LogViewer and Platform Manager, the Operations Integration Server Manager is a Microsoft Management Console (MMC) snap-in. Many high-level functions and user-interface elements of the Operations Integration Server Manager are universal to all Operations Integration Servers. It is critical to read the documentation for both the MMC and the Operations Integration Server Manager.

To read the documentation about the MMC and Operations Integration Server Manager, select the Help topics on the OCMC Help menu. Both the MMC Help and the Operations Integration Server Manager Help are displayed.

Log Viewer

An important troubleshooting tool, the Log Viewer records messages from machine execution. The Log Viewer can:

- Monitor messages on any machine in the system
- Send a portion of the log to notepad or E-mail
- Filter messages on a flag Security for the log viewer is set at the galaxy level.

Log Flag Editor

The Log Flag Editor is a utility of the Log Viewer. You can use the Log Flag Editor to assign a flag value to each category of messages issued by an System Platform component. By switching flags on or off, you control which categories of messages are saved to the Logger.

You can use the Log Flag Editor to assign a flag value to each category of messages issued by an System Platform component. By switching flags on or off, you control which categories of messages are saved to the Logger. Also, you can use log flags to assign message categories to other System Platform components.

Most categories of messages are not logged. Typically, a component issues only error, warning, and informational messages. Most other message categories remain inactive.

The Log Flag Editor is primarily a troubleshooting tool. When a component begins logging error messages, you can use log flags to activate other message categories and begin saving additional diagnostic messages to the Logger.

Platform Manager

Using Platform Manager, you can access platforms and engines deployed to any PC node in the galaxy.

After highlighting a platform, you can use the Action menu to start or stop a platform, or set it OnScan/OffScan. If the platform has security implemented, you must be logged on as a user configured with the proper OCMC permissions to start the OCMC, Start/Stop engines and platforms, or write from the Object Viewer.

AVEVA System Monitor

AVEVA System Monitor is an application that monitors and manages the performance and availability of the AVEVA System Platform system including the core software, engineered software application(s), and the related hardware and network infrastructure. It is included with System Platform.

System Monitor detects log and reports on system performance issues/errors/trends and monitors key system attributes, and then generates alerts when those attributes exceed defined operational limits.

When an attribute is out of operational limits, the support team (internal, systems integrator, and/or AVEVA Knowledge and Support Center) is notified, and can respond proactively to prevent production interruption or downtime. The goal is to check that your solutions' performance meets and/or exceeds expectations.

System Monitor runs in one of two modes:

- Basic mode (unlicensed)
- Full mode (licensed)

System Monitor, when licensed, uses activation-based licensing. Therefore, it requires an AVEVA License Server and an AVEVA License Manager from which a System Monitor license can be acquired.

- In licensed mode, System Monitor provides FULL monitoring of an unlimited number of machines.
- If an activated license is not detected (BASIC mode), System Monitor provides license server and license acquisition monitoring for all machines using software that requires activation-based licensing and FULL monitoring for one machine (user-selected).

Note: The System Monitor SSRS reports are not available in the BASIC mode.

OS Diagnostic Tools

The following information describes tools packaged with the Microsoft operating system.

Performance Monitor

The Windows Performance Monitor can be accessed from the `Windows Administrative Tools` app. The Performance tool includes the System Monitor and Performance Logs. System Monitor allows you to collect and view real-time data about memory, disk, processor, network, and other activity. Performance Logs and Alerts allows you to configure logs to record performance data and set system alarms. For information about using Performance tools, click the Action menu in Performance, and then click Help.

Event Viewer

The Event Viewer can be accessed from the `Windows Administrative Tools` app. It maintains logs about program, security, and system events. You can use the Event Viewer to view and manage the event logs, gather information about hardware and software problems, and monitor Windows operating system security events.

Virtualization

This section describes the implementation of System Platform in a virtualized environment that utilizes VMware, Microsoft Hyper-V, and SIOS DataKeeper clustering software.

Also discussed are strategies for creating High Availability, Disaster Recovery, and High Availability configurations with Disaster Recovery capabilities that leverage the virtualized environment.

Getting Started with Virtualization

Mission-critical operations in both small- and large-scale organizations demand availability—defined as the ability of the user community to access the system—along with dependable recovery from natural or man-made disasters. Virtualization technologies provide a platform for High Availability and Disaster Recovery solutions.

Using this Guide

Note: This guide has been updated to include support for Hyper-V 3.0 and vSphere 6.0.

The purpose of this guide is to help you to implement System Platform in a virtualized environment, including:

- Implementing some of the new features in Microsoft Windows Server 2012 and higher
- Implementing High Availability, Disaster Recovery, or High Availability with Disaster Recovery utilizing Windows Server virtualization technologies such as Hyper-V
- Implementing High Availability and Disaster Recovery using VMware technology

This chapter introduces and defines virtualization concepts in general, as well as in a System Platform context. This chapter also defines a basic workflow and planning framework for your virtualization implementation.

Subsequent chapters describe features of Windows Server and Hyper-V, and provide an outline of how to use them. Among the topics discussed are how to configure High Availability, Disaster Recovery, High Availability with Disaster Recover, how create virtual images, and how to implement a virtualized backup strategy.

Subsequent chapters also provide test and performance metrics for a wide variety of system configurations, including Recovery Time Objective (RTO) and Recovery Point Objective (RPO).

Understanding Virtualization

Virtualization is the creation of an abstracted or simulated—virtual, rather than actual—version of something, such as an operating system, server, network resource, or storage device. Virtualization technology abstracts the hardware from the software, extending the life cycle of a software platform.

In virtualization, a single piece of hardware, such as a server, hosts and coordinates multiple guest operating systems. No guest operating system is aware that it is sharing resources and running on a layer of virtualization software rather than directly on the host hardware. Each guest operating system appears as a complete, hardware-based OS to the applications running on it.

Definitions

This implementation guide assumes that you and your organization have done the necessary research and analysis and have made the decision to implement System Platform in a virtualized environment. Such an environment can take advantage of advanced virtualization features including better utilization of hardware resources, High Availability and Disaster Recovery. In that context, we'll define the terms as follows:

- Virtualization can be defined as **creating a virtual, rather than real, version of System Platform or one of its components, including servers, nodes, databases, storage devices, and network resources**.
- High Availability (HA) can be defined as a **primarily automated System Platform design and associated services implementation which ensures that a pre-defined level of operational performance will be met during a specified, limited time frame**.
- Disaster Recovery (DR) can be defined as **the organizational, hardware and software preparations for System Platform recovery or continuation of critical System Platform infrastructure after a natural or human-induced disaster**.

While these definitions are general and allow for a variety of HA and DR designs, this implementation guide focuses on virtualization, an indispensable element in creating the redundancy necessary for HA and DR solutions.

The virtualized environment described in this guide is based on Microsoft Hyper-V technology incorporated in the Windows Server 2012 and higher operating systems, and on VMware technology.

Types of Virtualization

There are eight types of virtualization:

Hardware	A software execution environment separated from underlying hardware resources. Includes hardware-assisted virtualization, full and partial virtualization and paravirtualization.
Memory	An application operates as though it has sole access to memory resources, which have been virtualized and aggregated into one memory pool. Includes virtual memory and memory virtualization.
Storage	Complete abstraction of logical storage from physical storage
Software	Multiple virtualized environments hosted within a single operating system instance. Related is a virtual machine (VM) which is a software implementation of a computer, possibly hardware-assisted, which behaves like a real computer.

Mobile	Uses virtualization technology in mobile phones and other types of wireless devices.
Data	Presentation of data as an abstract layer, independent of underlying databases, structures, and storage. Related is database virtualization, which is the decoupling of the database layer within the application stack.
Desktop	Remote display, hosting, or management of a graphical computer environment—a desktop.
Network	Implementation of a virtualized network address space within or across network subnets.

Virtualization Using a Hypervisor

Virtualization technology implements a type of hardware virtualization using a hypervisor, permitting a number of guest operating systems (virtual machines) to run concurrently on a host computer. The hypervisor is so named because it exists above the usual supervisory portion of the operating system.

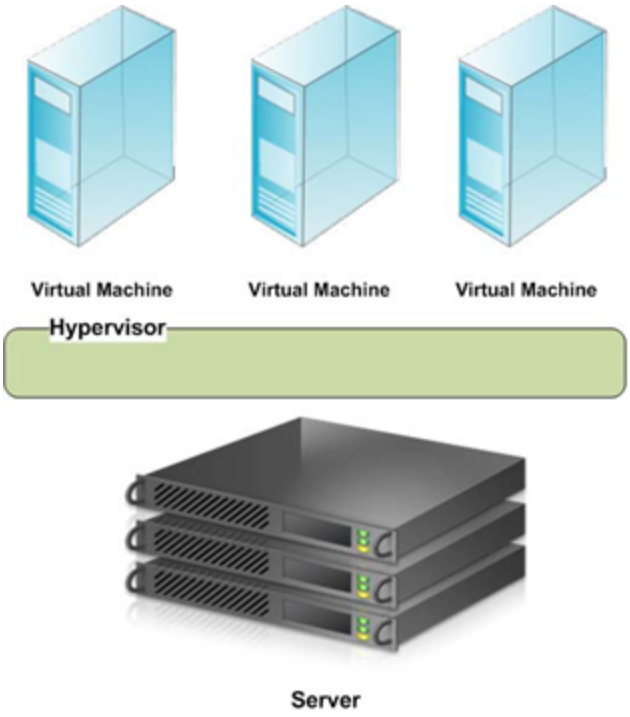
Hypervisor Classifications

There are two classifications of hypervisor:

- **Type 1**: Also known as a bare metal hypervisor, runs directly on the host hardware to control the hardware and to monitor the guest operating systems. Guest operating systems run as a second level above the hypervisor. ESXi for VMware vSphere, and Hyper-V for Windows Server are examples of type 1 hypervisors.
- **Type 2**: Also known as a hosted hypervisor, runs within a conventional operating system environment as a second software level. Guest operating systems run as a third level above the hypervisor. VMWorkstation is an examples of a type 2 hypervisor.

Hypervisor Architecture

Hyper-V and VMware implement Type 1 hypervisor virtualization, in which the hypervisor primarily is responsible for managing the physical CPU and memory resources among the virtual machines. This basic architecture is illustrated in the following diagram.



Virtualizing System Platform

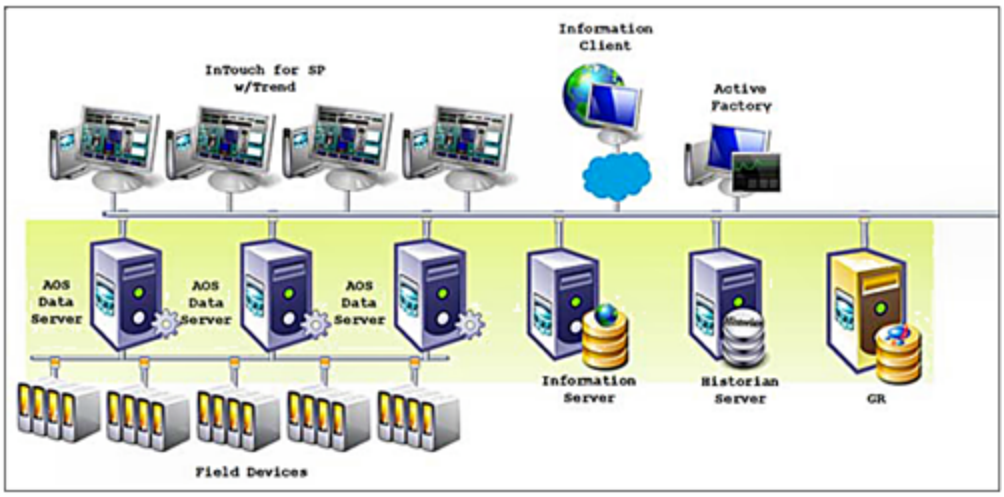
Abstraction Versus Isolation

With the release of InTouch 10.0, supporting the VMWare ESX platform, we became one of the first companies to support virtual machine operation of industrial software. VMware ESX is referred to as a "bare metal" virtualization system. The virtualization is run in an **abstraction layer**, rather than in a standard operating system.

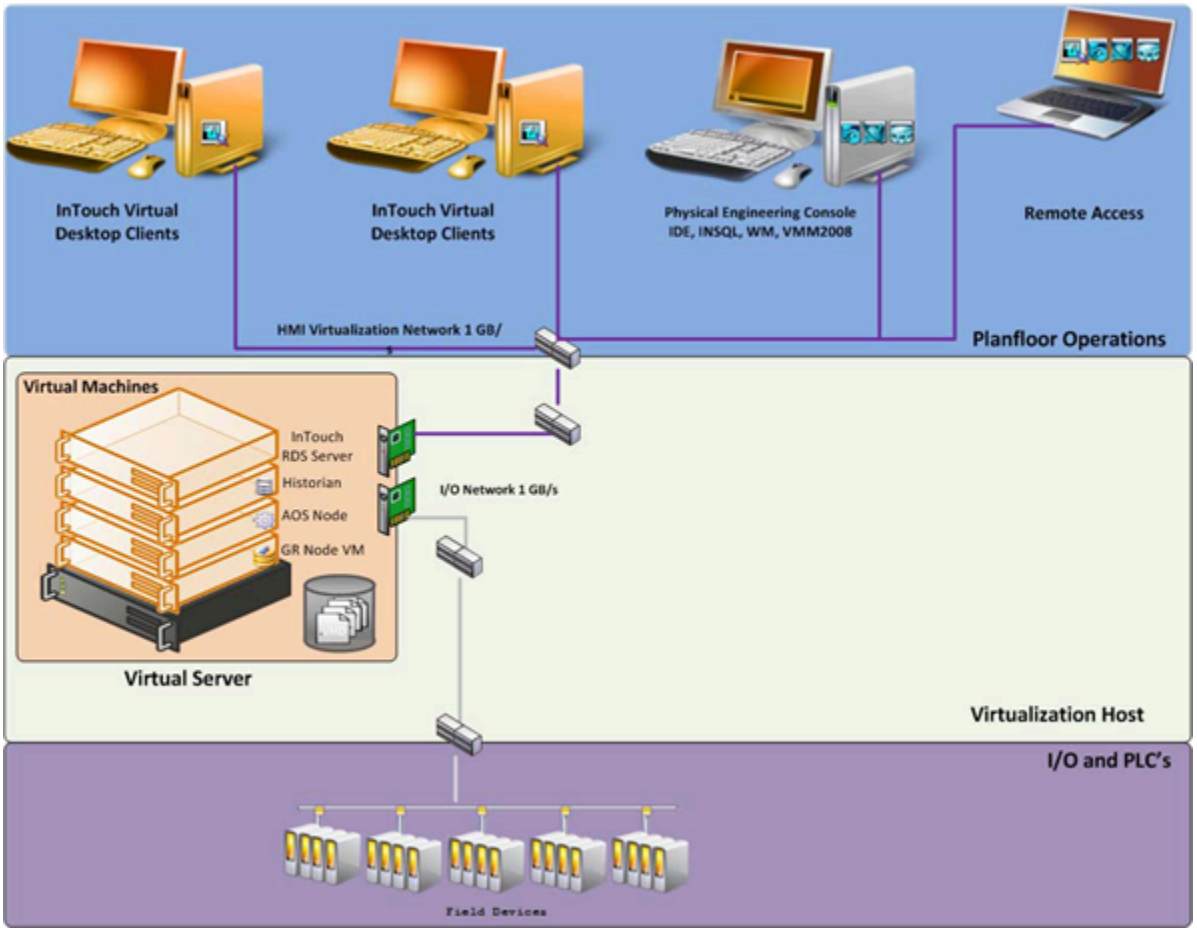
Microsoft takes a different approach to virtualization. Microsoft Hyper-V is a hypervisor-based virtualization system. The hypervisor is essentially an **isolation layer** between the hardware and partitions which contain guest systems. This requires at least one parent partition, which runs Windows Server 2012 or higher.

Note: An abstraction layer is a layer with drivers that make it possible for the virtual machine (VM) to communicate with hardware (VMware). In this scenario the drivers need to be present for proper communication with the hardware. With an isolation layer, the VM uses the operating system, its functionality, and its installed drivers. This scenario does not require special drivers. As a comparison, the abstraction layer in VMware is 32MB and in Hyper-V it is 256kb.

The following diagram shows a common System Platform topology, non-virtualized:



The following diagram shows the same environment virtualized:



Levels of Availability

When planning a virtualization implementation—for High Availability, Disaster Recovery, Fault Tolerance, and Redundancy—it is helpful to consider levels or degrees of redundancy and availability, described in the following table.

Level	Description	Comments
Level 0 Redundancy	No redundancy built into the architecture for safeguarding critical architectural components	Expected failover: None
Level 1 Cold Stand-by Redundancy	Redundancy at the Application Object level Safeguards single points of failure at the OI Server level or AOS redundancy.	Expected failover: 10 to 60 seconds Availability 99%: Annual uptime impact is approximately four days down per year
Level 2 High Availability (HA)	• With provision to synchronize in real-time • Uses virtualization techniques • Can be 1-n levels of hot standby • Can be geographically diverse (DR) • Uses standard OS and nonproprietary hardware	Expected failover: Uncontrolled 30 seconds to 2 minutes, DR 2 - 7 minutes Availability 99.9%: Annual uptime impact is approximately 8 hrs down per year
Level 3 Hot Redundancy:	Redundancy at the application level typically provided by Schneider Electric controllers. For example, hot backup of Schneider Electric software such as Alarm System.	Expected failover: Next cycle or single digit seconds Availability 99.99%: Annual uptime impact is approximately 52 minutes down per year.
Level 4 Lock-step Fault Tolerance (FT)	Provides lock-step failover	Expected failover: Next cycle or without loss of data. Availability 99.999%: Annual uptime impact is considered as "continuous availability" with downtime less than 5 minutes per year. A 99.999% availability is considered the "gold standard." For System Platform, this would be a Marathon-type solution, which also can be a virtualized system.

A typical system without virtualization, using a High Availability implementation, might attain Level 1 availability with a good server. With a good infrastructure, you can achieve Level 3 availability by using virtualized High

Availability.

A typical system could reach Level 4 availability by using virtualization with more than two possible hosts, RAID storage options, dual power supplies, teamed NICs, and by implementing application monitoring. This allows the application to restart on another host if a crash occurs.

Performance of failover can vary and is dependent on the quality and implementation of the HA architecture.

About RTO and RPO

The Recovery Time Objective (RTO) is the duration of time within which a business process must be restored to its service level after a disaster or other disruption in order to avoid a break in business continuity.

A Recovery Point Objective (RPO), is defined by business continuity planning. It is the maximum tolerable period in which data might be lost from an IT Service due to a major incident.

For System Platform in a normal, non-virtualized, implementation, depending on the system size, RTO could be hours or days on a complete loss of the system. The RPO would be 45 seconds or more for Application Server redundancy, or more—in terms of hours— for non-redundant components such as Terminal Servers for InTouch HMI or Information Server.

For System Platform in a virtualized High Availability implementation that uses double-host configuration, the measured recovery time is as follows:

- RTO is less than 2 minutes for the complete system. Controlled RTO is seconds, with un-controlled RTO less than 2 minutes.
- RPO is within 2 minutes.

High Availability

About HA

High Availability refers to the availability of resources in a computer system following the failure or shutdown of one or more components of that system.

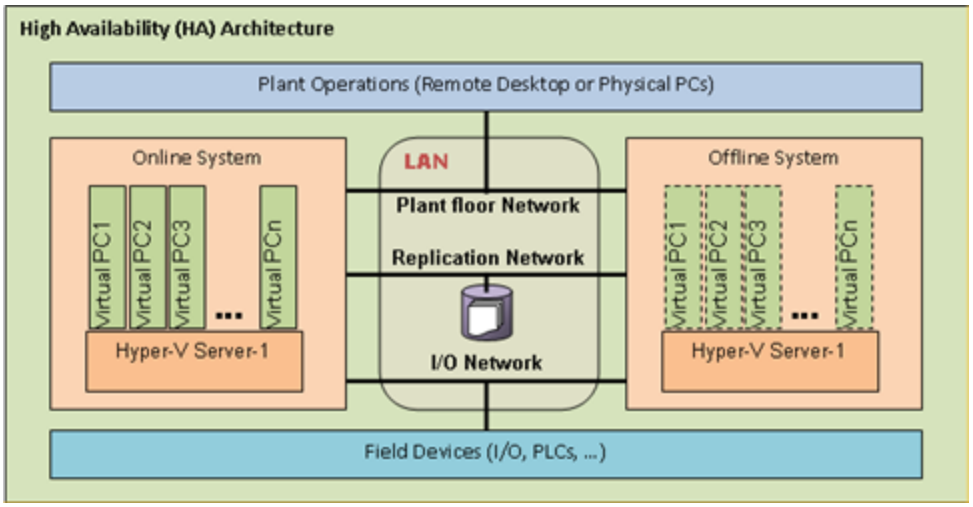
At one end of the spectrum, traditional HA has been achieved through custom-designed and redundant hardware. This solution produces High Availability, but has proven to be very expensive.

At the other end of the spectrum are software solutions designed to function with off-the-shelf hardware. This type of solution typically results in significant cost reduction, and has proven to survive single points of failure in the system.

High Availability Scenarios

The basic HA architecture implementation described in this guide consists of an online system including a Hyper-V or VMware Server and a number of virtual PCs, linked by a LAN to an offline duplicate system. The LAN accommodates a number of networks including a plant floor network linked to plant operations, an I/O network linked to field devices, and a replication network linked to storage.

The following example shows Hyper-V implementation.



This basic architecture permits a number of common scenarios.

IT maintains a virtual server	• A system engineer fails over all virtual nodes hosting System Platform software to back up the virtualization server over the LAN. • For a distributed system, the system engineer fails over all virtual nodes to back up the virtualization server over a WAN. • IT performs the required maintenance, requiring a restart of the primary virtualization server.
Virtualization server hardware fails	• The primary virtualization server hardware fails with a backup virtualization server on the same LAN. • For a distributed system, the virtualization server hardware fails with a backup virtualization server over WAN. **Note:** This scenario is a hardware failure, not software. A program that crashes or hangs is a failure of software within a given OS.
A network fails on a virtual server	• Any of the primary virtualization server network components fail with a backup virtualization server on the same LAN, triggering a backup of virtual nodes to the backup virtualization server. • Any of the primary virtualization server network components fail with a backup virtualization server connected via WAN, triggering a backup of virtual nodes to the backup virtualization server over WAN.

For these scenarios, the following expectations apply:

- For the maintenance scenario, all virtual images are up and running from the last state of execution prior to failover.
- For the hardware and network failure scenarios, the virtual images restart following failover.
- For LAN operations, you should see operational disruptions for approximately 2-15 seconds (LAN operations assumes recommended speeds and bandwidth. For more information refer to "Networks").
- For WAN operations, you should see operational disruptions for approximately 2 minutes (WAN operations assumes recommended speeds and bandwidth. For more information refer to "Networks").

Note: The disruption spans described here are general and approximate. For specific metrics under a variety of scenarios, see the relevant Recovery Time Objective (RTO) and Recovery Point Objective (RPO) sections in chapters 2, 3, and 4.

Disaster Recovery

About DR

Disaster Recovery planning typically involves policies, processes, and planning at the enterprise level, which is well outside the scope of this implementation guide.

DR, at its most basic, is all about data protection. The most common strategies for data protection include the following:

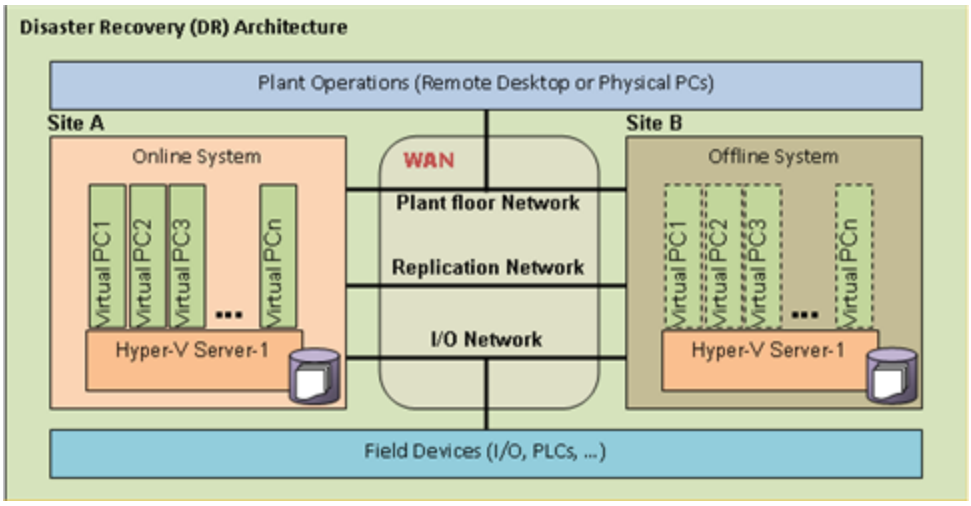
- Backups made to tape and sent off-site at regular intervals, typically daily.
- For the hardware and network failure scenarios, the virtual images restart following failover
- For the hardware and network failure scenarios, the virtual images restart following failover
- Backups made to disk on-site, automatically copied to an off-site disk, or made directly to an off-site disk.
- Replication of data to an off-site location, making use of storage area network (SAN) technology. This strategy eliminates the need to restore the data. Only the systems need to be restored or synced.
- High availability systems which replicate both data and system off-site. This strategy enables continuous access to systems and data.

The System Platform virtualized environment implements the fourth strategy—building DR on an HA implementation.

Disaster Recovery Scenarios

The basic DR architecture implementation described in this guide builds on the HA architecture by moving storage to each Hyper-V or VMware server, and moving the offline system to an off-site location.

The following example shows Hyper-V implementation.



The DR scenarios duplicate those described in High Availability Scenarios, with the variation that all fail-overs and backups occur over WAN.

High Availability with Disaster Recovery

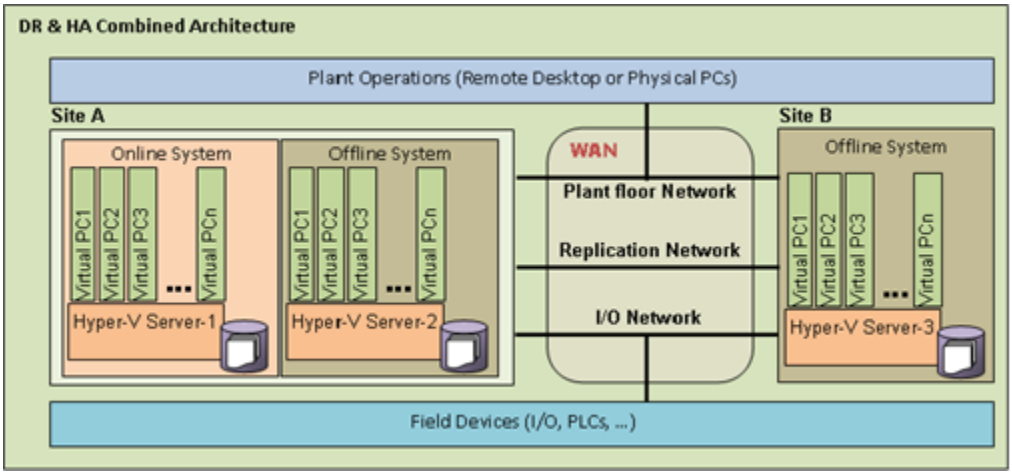
About HADR

The goal of a High Availability and Disaster Recovery (HADR) solution is to provide a means to shift data processing and retrieval to a standby system in the event of a primary system failure.

Typically, HA and DR are considered as individual architectures. HA and DR combined treat these concepts as a continuum. If your system is geographically distributed, for example, HA combined with DR can make it both highly available and quickly able to recover from a disaster.

HADR Scenarios

The basic HADR architecture implementation described in this guide builds on both the HA and DR architectures adding an offline system plus storage at "Site A". This creates a complete basic HA implementation at "Site A" plus a DR implementation at "Site B" when combined with distributed storage.



The scenarios and basic performance metrics described in High Availability Scenarios also apply to HADR.

Planning the Virtualized System

Planning an System Platform virtualization implementation is a three-step process—based upon an understanding of the available technology:

Step 1: Assess your existing System Platform installation

Step 2: Assess virtualization requirements

Step 3: Extend your assessment to define HA, DR, or HADR

For more information about configuring HA, DR, and HADR, see the following chapters in this guide:

- Implementing High Availability Using Hyper-V
- Implementing High Availability Using vSphere
- Implementing Disaster Recovery Using Hyper-V
- Implementing Disaster Recovery Using vSphere
- Implementing High Availability and Disaster Recovery Using Virtualization

Planning Information for a Hyper-V Implementation

About Hyper-V

The following table summarizes key Hyper-V features:

Dynamic Memory	Dynamic Memory enables better utilization of Hyper-V host memory resources by balancing how memory is distributed between running virtual machines. Memory can be dynamically reallocated between different virtual machines in response to the changing workloads of these machines.
Live Migration	Data-centers with multiple Hyper-V physical hosts can move running virtual machines to the best physical computer for performance, scaling, or optimal consolidation without affecting users.
Hardware Support for Hyper-V Virtual Machines	Depending on the operating system, Hyper-V supports up to 320 logical processors in the host processor pool, allowing greater VM density per host, and more flexibility in assigning CPU resources to VMs, and enabling migration across a broader range of server host hardware.
Cluster Shared Volumes	Hyper-V uses Cluster Shared Volumes (CSV) storage to simplify and enhance shared storage usage. CSV enables multiple Windows Servers to access SAN storage using a single consistent namespace for all volumes on all hosts.

Cluster Node Connectivity Fault Tolerance	CSV architecture improves cluster node connectivity fault tolerance that directly affects VMs running on the cluster. The CSV architecture implements a mechanism, known as dynamic I/O redirection, where I/O can be rerouted within the failover cluster based on connection availability.
Enhanced Cluster Validation Tool	A Best Practices Analyzer (BPA) is included for all major server roles, including Failover Clustering. This analyzer examines the best practices configuration settings for a cluster and cluster nodes.
Management of Virtual Datacenters	The number of VMs tends to proliferate much faster than physical computers because machines typically do not require a hardware acquisition. This makes efficient management of virtual data centers more imperative than ever.
Virtual Networking Performance	Hyper-V leverages networking technologies to improve overall VM networking performance.
Performance & Power Consumption	Enhancements have been added that reduce virtual machine power consumption.
Networking Support	Jumbo Frames, previously available in non-virtual environments, has been extended to work with VMs. The Virtual Machine Queue (VMQ) feature allows physical computer network interface cards (NICs) to use direct memory access (DMA) to place the contents of packets directly into VM memory, increasing I/O performance.
Dynamic VM storage	Hyper-V supports hot plug-in and hot removal of storage. This allows the addition and removal of both VHD files and pass-through disks to existing SCSI controllers for VMs.
Broad OS Support	Broad support for simultaneously running different types of operating systems, including 32-bit and 64-bit systems across different server platforms, such as Windows, Linux, and others.
Network Load Balancing	Hyper-V includes virtual switch capabilities. This means virtual machines can be easily configured to run with Windows Network Load Balancing (NLB) Service to balance load across virtual machines on different servers.

Hardware Sharing Architecture	With virtual service provider/virtual service client (VSP/VSC) architecture, Hyper-V provides improved access and utilization of core resources, such as disk, networking, and video.
Virtual Machine Snapshot	Hyper-V provides the ability to take snapshots of a running virtual machine so you can easily revert to a previous state, and improve the overall backup and recoverability solution.
Extensibility	Standards-based Windows Management Instrumentation (WMI) interfaces and APIs in Hyper-V enable independent software vendors and developers to quickly build custom tools, utilities, and enhancements for the virtualization platform.

VM and Hyper-V Limits in Windows Server

Refer to Microsoft documentation for maximum values that apply to Hyper-V. These values can vary significantly between releases and versions of Windows Server. For Windows Server 2012 and Windows Server 2012 R2, these limits are 64 virtual processors and 1 TB memory per virtual machine.

For more information on Hyper-V and its maximums, refer to the Microsoft TechNet resources on Hyper-V, available via the following link:

https://technet.microsoft.com/en-us/windowsserver/dd448604.aspx

Planning Information for a VMware Implementation

About vCenter Server and vSphere

VMware vCenter Server is a simple and efficient way to manage multiple VMware vSpheres. It provides unified management of all the hosts and VMs in your datacenter from a single console monitoring the performance of clusters, hosts, and VMs.One administrator can manage 100 or more workloads.

VMware vCenter Servers allow you to provide VMs and hosts using standardized templates. Use of templates helps to ensure compliance with vSphere host configurations and host and VM patch levels with automated remediation. With proactive management, VMware vCenter Server allows you to dynamically provide new services, allocate resources, and automate high availability.

VMware vCenter Server enables management of a large scale enterprise, more than 1,000 hosts and up to 10,000 VMs, from a single console.

Extensibility

VMware vCenter Server's open plug-in architecture supports a broad range of additional capabilities that can directly integrate with vCenter Server, allowing you to easily extend the platform for more advanced management capability in areas such as:

- Capacity management
- Compliance management

- Business continuity
- Storage monitoring
- Integration of physical and virtual management tools

VMware vSphere 6.0 and Newer Editions

VMware vSphere 6 includes scalability improvements and security enhancements over previous releases. It is available in three editions: Standard, Enterprise, and Enterprise Plus. One instance of VMware vCenter Server, sold separately, is required for all VMware vSphere deployments. ESXi is the type 1 (bare metal) hypervisor included in the vSphere suite of products.

For information about each edition's features and capabilities, refer to the VMware website:

https://www.vmware.com/products/vSphere/compare

VM and Virtual Server Limits in VMware

The following tables show maximum values for VMs and for a server running vSphere 6.0. By understanding the limits of the hardware, software, and virtual machines, you can better plan your System Platform virtualized environment.

vSphere Virtual Machine Maximums (ESXi 6.0)

Component	Maximum	Notes
Virtual CPUs per VM	128	32 previously
RAM per VM	4 TB	1 TB previously
IDE controllers per VM	1	Supports two channels (primary and secondary) each with a master and slave device.
SCSI adapters per VM	4	Any combination of supported SCSI virtual storage controllers. Four Paravirtual SCSI adapters may be used only if the virtual machine boots from a device attached to an IDE controller, or from the network.
Virtual SCSI targets per virtual SCSI adapter	15	Any combination of disk, CD-ROM, or VMDirectPath SCSI target
Virtual hard disk capacity	62 TB	2TB previously
Size of physical disks attached to a VM	Varies	Maximum size is determined by the guest operating system.

Component	Maximum	Notes
Checkpoints (Snapshots)	32	The actual number depends on the available storage and may be lower. Each snapshot is stored as a file that consumes physical storage.
Virtual network adapters	10	Any combination of supported virtual NICs.
Virtual floppy controllers	1	
Virtual floppy devices	2	BIOS is configured for 1 floppy device.
USB controllers	1	Supports USB 1.x, 2.x,and 3.x devices
USB devices connected to a virtual machine	20	
Parallel ports	3	
Serial (COM) ports	32	4 previously

vSphere ESXi 6.0 Host Maximums

Component	Maximum	Notes
Logical CPUs per host	480	160 previously
Virtual machines per host	1024	512 previously
Virtual CPUs per host	4096	2048 previously
Memory	6 TB	12 TB is supported on specific OEM certified platforms. Refer to the VMware Compatibility Guide for additional information.
Virtual disks per host	2048	

Component	Maximum	Notes
Physical network adapters	32	tg3 1 Gb Ethernet ports (Broadcom) • 16 with NetQueue enabled • 32 with NetQueue disabled NetQueue is enabled by default in vSphere 6.0
Maximum active ports per host	1016	
Virtual network switch ports per host	4096	vSphere Standard and Distributed Switch

VMware Requirements

VMware ESXi 6.0 Installation Requirements

The minimum requirements to install the vSphere Hypervisor (ESXi 6.0) are listed in the following table. For complete installation requirements and additional information, refer to vSphere Installation and Setup instructions:

https://pubs.vmware.com/vSphere-60/topic/com.vmware.ICbase/PDF/vSphere-esxi-vcenter-server-60-installation-setup-guide.pdf

For additional information refer to the VMWare Compatibility Guide:

http://www.vmware.com/resources/compatibility

Component	Requirement
64-bit Processor	ESXi 6.0 installs and run only on servers with 64-bit x86 CPUs. ESXi 6.0 requires a host machine with at least two cores. ESXi 6.0 requires the NX/XD bit to be enabled for the CPU in the BIOS.
RAM	4GB RAM minimum; 8 GB RAM minimum to fully leverage ESXI features and run machines in typical production environments.
Network Adapters	One or more Gigabit or faster Ethernet controllers.

Component	Requirement
Installation and Storage	SCSI disk or a local, non-network, RAID LUN with unpartitioned space for the virtual machines. For Serial ATA (SATA), a disk connected through supported SAS controllers or supported on-board SATA controllers. SATA disks will be considered remote, not local. These disks will not be used as a scratch partition by default because they are seen as remote. **Note:** You cannot connect a SATA CD-ROM device to a virtual machine on an ESXi 6.0 host. IDE emulation mode must be used to enable a SATA CD-ROM device.
Support for 64-bit Virtual Machines	Support for hardware virtualization (Intel VT-x or AMD RVI) must be enabled.
Storage Systems	Refer to the VMWare Compatibility Guide: http://www.vmware.com/resources/compatibility

VMware Disaster Recovery Requirements

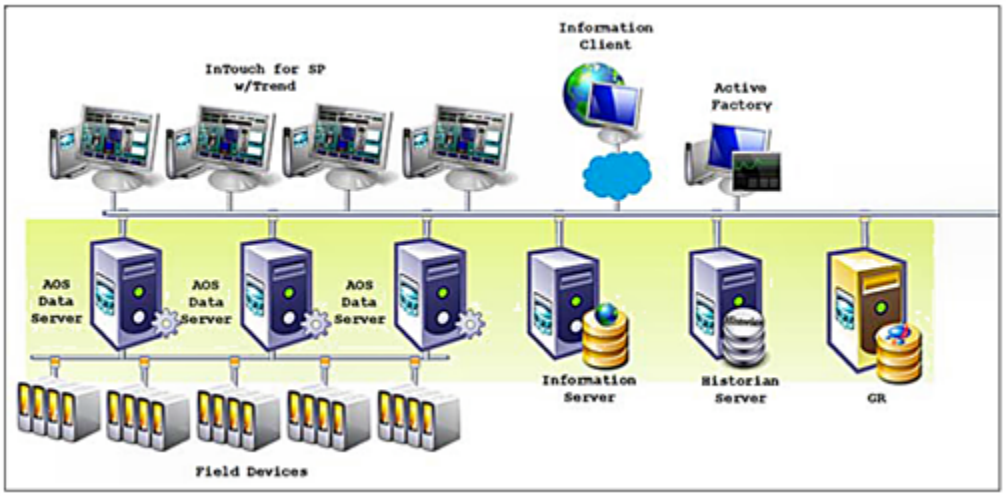
VMware Disaster Recovery (DR) implementations require installation of VMware vCenter Site Recovery Manager, Standard or Enterprise edition.

Scalability limits of the vCenter Site Recovery Manager editions are:

- Standard Edition: 75 virtual machines
- Enterprise Edition: Unlimited, subject to the product's technical scalability limits.

Assessing Your System Platform Installation

In most cases, a System Platform installation already exists. You will need to create an assessment of the current architecture. You can start with a basic topology diagram, similar to the following:



Once you have diagramed your topology, you can build a detailed inventory of the system hardware and software.

Microsoft Planning Tools

Microsoft tools to assist with virtualization assessment and planning:

- Microsoft Assessment and Planning Toolkit (MAP)

  The MAP toolkit is useful for a variety of migration projects, including virtualization. The component package for this automated tool is available for download from Microsoft at the following address:

  http://www.microsoft.com/en-us/download/details.aspx?id=7826
- Infrastructure Planning and Design Guides for Virtualization (IPD)

  The IPD Guides from Microsoft provide a series of guides specifically geared to assist with virtualization planning. They are available for download from Microsoft at the following address:

  http://technet.microsoft.com/en-us/solutionaccelerators/ee395429

VMware Planning Tools

VMware tools to assist with virtualization assessment and planning:

- VMware Capacity Planner

  The VMware Capacity Planner is a business and IT tool for datacenter and desktop capacity planning.

  http://www.vmware.com/products/capacity-planner/overview.html
- VMware SAN System Design and Deployment Guide

  This guide describes how to design and deploy virtual infrastructures using VMware technology.

  http://www.vmware.com/files/pdf/techpaper/SAN_Design_and_Deployment_Guide.pdf
- VMware Infrastructure 3 Planning

  This guide is specific to planning virtualization using Hewlett-Packard computer equipment. It offers considerable insight into planning, architecture, and deployment.

  http://www.vmware.com/support/pubs/vi_pubs.html

Sizing Recommendations for Virtualization

This section provides sizing guidelines and recommended minimums for System Platform installations.

For a only implementation, you can use these minimums and guidelines to size the virtualization server or servers that will host your System Platform configuration.

Cores and Memory

Spare Resources

The host server should always have spare resources of 25% above what the guest machines require.

For example, if a configuration with five nodes requires 20GB of RAM and 10 CPUs, the host system should have 25GB of RAM and 13 CPUs. If this is not feasible, choose the alternative closest to the 25% figure, but round up

so the host server has 32GB of RAM and 16 cores.

Hyper-Threading

Hyper-Threading Technology can be used to extend the amount of cores, but it does impact performance. An 8-core CPU will perform better than a 4-core CPU that is Hyper-Threading.

Storage

It is always important to plan for proper Storage. A best practice is to dedicate a local drive or virtual drive on a Logical Unit Number (LUN) to each of the VMs being hosted. We recommend SATA or higher interfaces.

Recommended Storage Topology

To gain maximum performance, the host OS also should have a dedicated storage drive. A basic storage topology would include:

- Host storage
- VM storage for each VM
- A general disk

  This disk should be large enough to hold snapshots, backups, and other content. It should not be used by the host or by a VM.

Recommended Storage Speed

Boot times and VM performance are impacted both by storage bandwidth and storage speed. Faster is always better. Drives rated at 7200 rpm perform better than those rated at 5400 rpm. Solid-state drives (SSDs) perform better than 7200-rpm drives.

Keep in mind that multiple VMs attempting to boot from one hard drive will be slow, and your performance will experience a significant degrade. Attempting to save on storage could well become more costly in the end.

Networks

Networking is as important as any other component for the overall performance of the system.

Recommended Networking for Virtualization

If virtualization is your only requirement, your network topology could include the following elements:

- Plant network
- Storage network
- Virtualization network.

A best practice is to establish, on every node, an internal-only Static Virtual Network. In the event that the host and the guest VMs become disconnected from the outside world, you will still be able to communicate through an RDP session independent of external network connectivity.

Recommended Networking for HA

If HA is your requirement, then we recommend using fast, dedicated drives for local use. In the case of a Storage Area Network (SAN), we recommend using iSCSI 1GB/s as a minimum configuration.

A higher-performance configuration would be an FO connection to the storage at 10GB/s. For HA, we recommend a dedicated network for virtualization at 1GB/s. This will ensure fast transfers under different

migration scenarios.

Recommended Minimums for System Platform

Following are approximate numbers of nodes to define small, medium, and large systems.

- Small: 1–3 nodes
- Medium: 4–8 nodes
- Large: More than 8 nodes

The following table provides recommended minimums for System Platform configurations.

	Cores	RAM	Storage
Small Systems			
GR Node	2	2	100
Historian	2	2	250
Application Server	2	2	100
RDS Servers	2	2	100
Information Servers	2	2	100
Historian Clients	2	2	100
Medium and Large Systems			
GR Node	4	4	250
Historian	4	4	500
Application Server	2–4	4	100
RDS Servers	4–8	4–8	100
Information Server	4	4	100
Historian Clients	2	4	100

After installation of the server, you will start from scratch, or you can use the existing installation. A free tool on Microsoft TechNet called Disk2vhd supports extracting a physical machine to a VHD file. The Disk2vhd tool is available for download from Microsoft at the following address:

http://technet.microsoft.com/en-us/sysinternals/ee656415

Another tool you can use to migrate physical machines into to a virtual environment is Virtual Machine Manager. This tool is available for purchase from Microsoft. For more information, see the following Microsoft address:

https://www.microsoft.com/en-us/download/details.aspx?id=10712

A VMware tool for disk conversion is the vCenter Converter Standalone for P2V Conversion, available from VMware as a free download at the following address:

https://www.vmware.com/tryvmware/?p=converter&rct=j&q=vmware%20converter&source=web&cd=6&sqi=2&ved=0CEoQFjAF&url=http://www.vmware.com/go/getconverter&ei=4XIPT7ePB7CPigLR0OzSDQ&usg=AFQjCNH3Et0HlSZPzkw2VZxLVZoNZ_yY5g

Defining High Availability

To define a High Availability implementation, you need to plan for the following requirements:

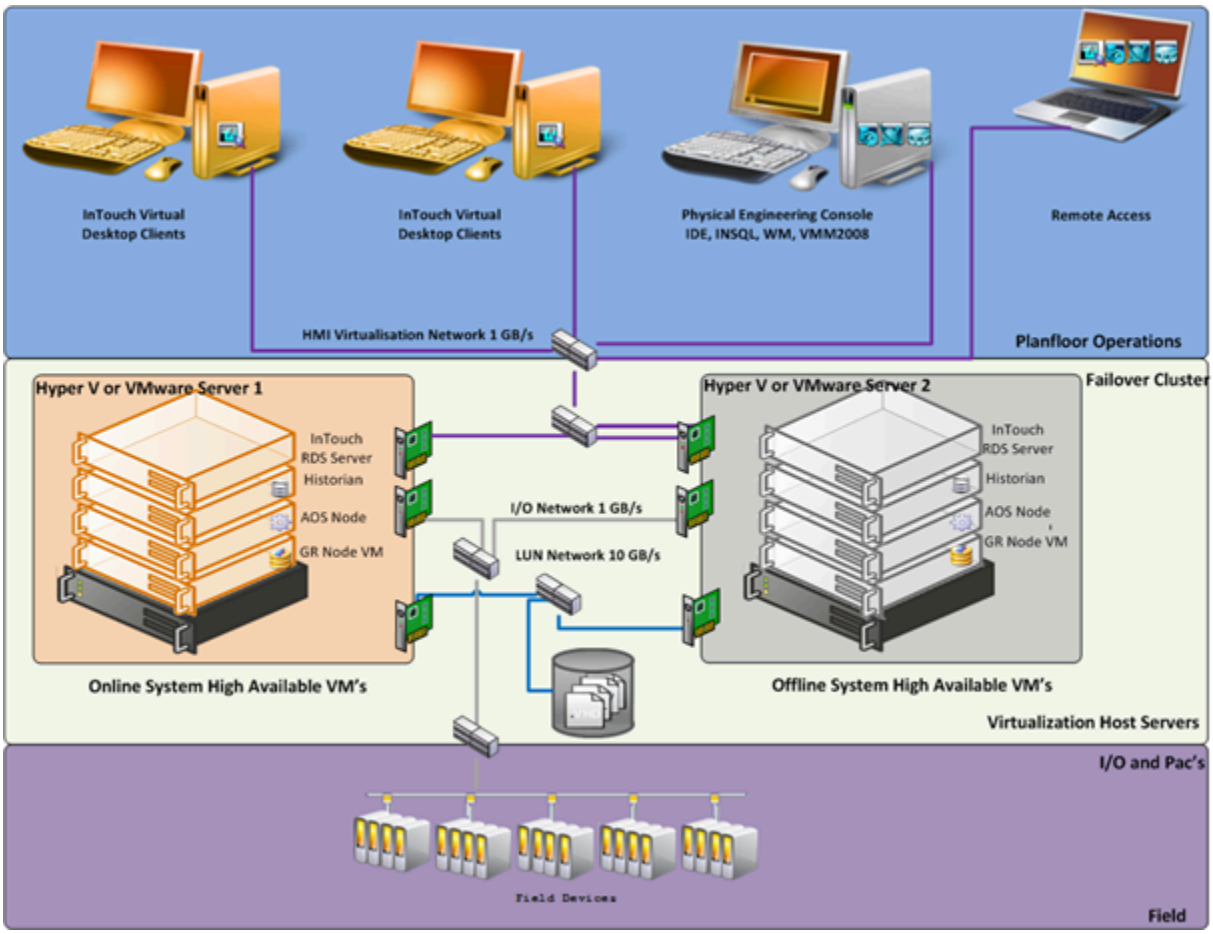
- Server specification doubles

  Double the baseline configuration is required for shadow nodes in the Failover Cluster.
- Minimum OS requirements increase

  Hyper-V failover is supported only on Windows Server 2012 and higher Enterprise operating system editions.

  Also, Hyper-V live migration, remote applications, and other features are available only if the host machines are Windows Server 2012 and higher editions.

The following shows a System Platform HA implementation:



To implement HA, we strongly recommend the use of a SAN configured with the sizing guidelines and

recommendations outlined in the preceding section.

Defining Disaster Recovery

To define a Disaster Recovery implementation, you need to plan for the following requirements:

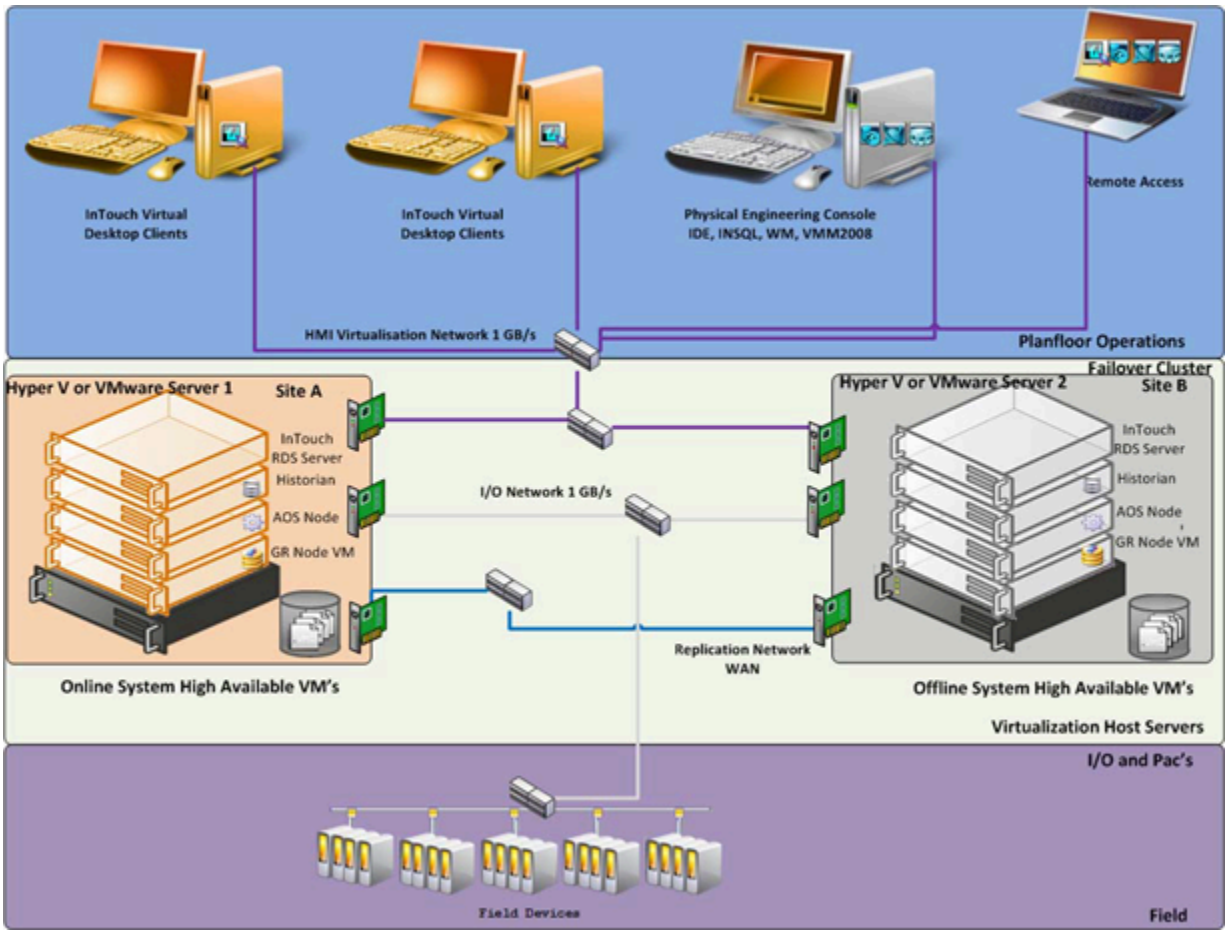
- Adding a second server set with the same specifications as the first

  The second server set moves to the off-site location and connects over LAN or (more likely) WAN. Hyper-V Replica provides asynchronous replication of Hyper-V VMs on a second server.
- Configuring minimum bandwidth

  The minimum network bandwidth is 100MB/sec. Recovery times improve with higher network speeds.
- Installing and configuring third-party software with Hyper-V virtualization

  Third party software from SIOS (SteelEye) mirrors the drives from site A to site B. The replication can be done on a SAN system or as shown in the illustration, with regular local hard drives.

  Important: Mirrored partitions must have identical drive letters and sizes.



Defining High Availability and Disaster Recovery Combined

An important advantage from implementing HA and DR in the same scenario is that a local HA set can quickly

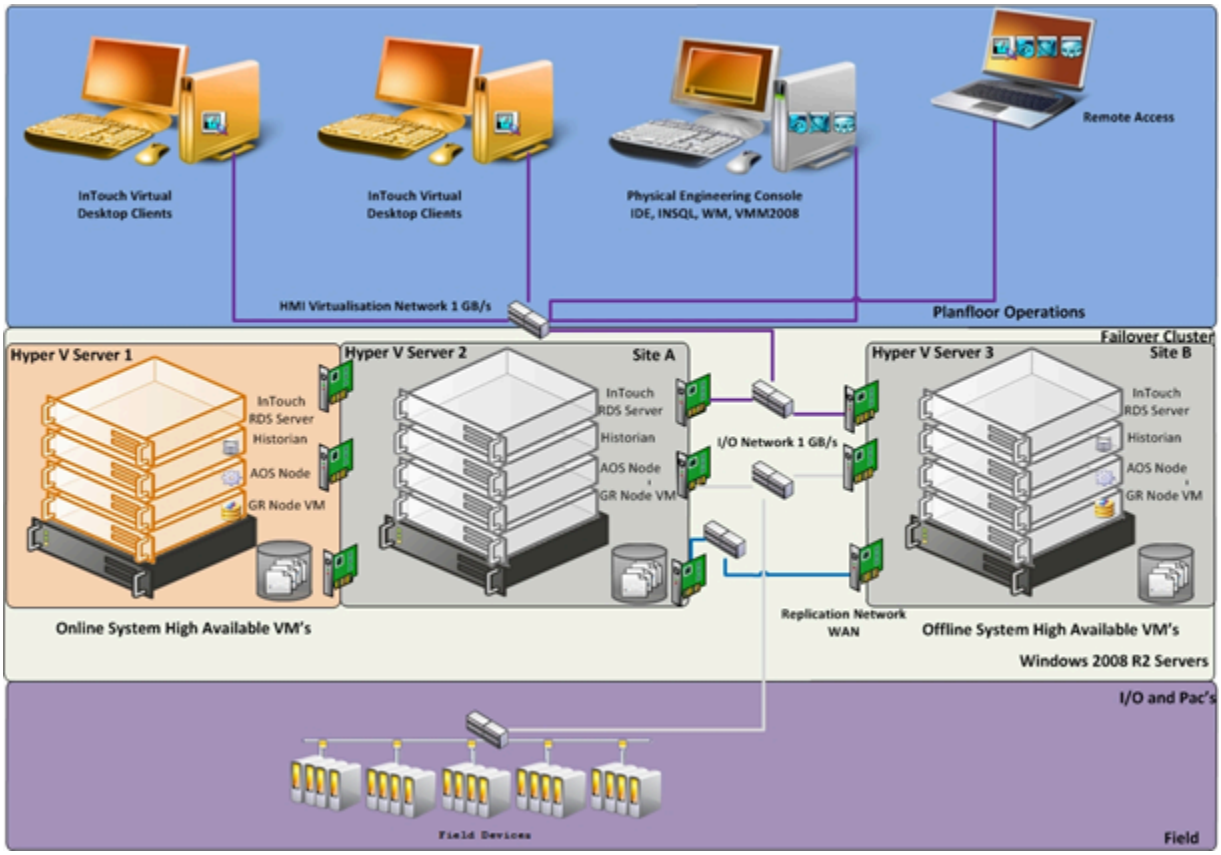
resume functionality upon failure. In the event that site A is offline, the system can resume at site B without intervention from site A.

To define a HADR implementation, you need to plan for the following requirements:

- Sizing

  You'll need to triple the size of the estimated baseline server.
- SANs

  Two SANs are required—one local and one remote—to host the storage. In HADR implementation, the local configuration uses the failover cluster configuration and the set of VMs are replicated to a remote site.



Recommendations and Best Practices

The following recommendations and best practices apply to all for High Availability (HA), Disaster Recovery (DR), and HADR implementations, with guidelines specific to System Platform products.

- Ensure that auto log on is set up for all virtual machines running the System Platform products. This is to ensure that these virtual machines start automatically after the failover.
- Ensure the time on all the host servers, the virtual machines, and all other nodes which are part of the High Availability Environment are continuously synchronized. Otherwise, the virtual machines running on the host experience time drifts and results in discarding of data.You can add the time synchronization utility in the Start Up programs so that this utility starts automatically whenever the machine reboots.
- On the host servers disable all the network cards which are not utilized by the System Platform Environment.

This is to avoid any confusion during the network selections while setting up the cluster.

- Ensure the Virtual Networks have the same name across all the nodes which are participating in the Cluster. Otherwise, migration/failover of virtual machines will fail.

System Platform Product-specific Recommendations and Observations

- During the preparation for Live and Quick migrations it is observed that the network freezes intermittently and then at the time of actual migration connectivity to the VM is lost. As a result, the System Platform node under migration experiences intermittent data loss during the preparation for Live and Quick migrations, and then has a data gap for the duration of actual migration.

The Historian

- In case of Live and Quick migration of the Historian, you may notice that the Historian logs values with quality detail 448 and there may be values logged twice with same timestamps. This is because the suspended Historian VM starts on the other cluster node with the system time it was suspended at before the migration. As a result, some of the data points it is receiving with the current time seem to be in the future to the Historian. This results in the Historian modifying the timestamps to its system time and updating the QD to 448. This happens until the system time of the Historian node catches up with the real current time using the TimeSync utility, after which the problem goes away. So, it is recommended to stop the historian before the migration and restart it after the VM is migrated and its system time is synced up with the current time.
- Live and Quick migration of the Historian should not be done when the block change over is in progress on the Historian node.
- If a failover happens (for example, due to a network disconnect on the source Host Virtualization Server) while the Historian status is still "Starting", the Historian node fails over to the target Host Virtualization Server. In the target host, the Historian fails to start. To recover from this state, kill the Historian services that failed to start and then start the Historian by launching the SMC.

InTouch HMI

- Ensure that InTouch Window Viewer is added to the Start Up programs so that the view is started automatically when the virtual machine reboots.

Application Server

- If a failover happens (for example, due to a network disconnect on the source Host Virtualization Server) while the Galaxy Migration is in progress, the GR node fails over to the target Host Virtualization Server. In the target host, on opening the IDE for the galaxy, the templates do not appear in the Template toolbox and in Graphic toolbox. To recover from this state, delete the Galaxy and create a new Galaxy. Initiate the migration process once again.
- If a failover happens (for example, due to an abrupt power-off on the source Host Virtualization Server) while a platform deploy is in progress, the Platform node fails over to the target Host Virtualization Server. In the target host, some objects will be in deployed state and the rest will be in undeployed state. To recover from this state, redeploy the whole Platform once again.
- If a failover happens (for example, due to an abrupt power-off on the source Host Virtualization Server) while

a platform undeploy is in progress, the Platform node fails over to the target Host Virtualization Server. In the target host, some objects will be in undeployed state and the rest will be in deployed state. To recover from this state, undeploy the whole Platform once again.

Operations Integration Server

In case of Live and Quick migration of an I/O Server node (for example, DASSIDirect), InTouch I/O Tags acquiring data from that I/O server need to be reinitialized after the I/O server node is migrated. To automatically acquire the data for these tags from the I/O server after migration, it is recommended to have an InTouch script which monitors the quality status of any of those tags and triggers reinitialize I/O once the quality goes to bad. Execute this script every 3 to 5 seconds until the tag quality becomes good.

Additional Guidelines for DR and HADR Implementations (only)

The following guidelines apply to DR and HADR implementation only, and are in addition to all of the guidelines and recommendations listed under "Recommendations and Best Practices" and System Platform Product-specific Recommendations and Observations.

- As per the topology described earlier for the Disaster Recovery environment, only one network is used for all communications. If multiple networks are being used, then make sure only the primary network which is used for the communication between the Nodes is enabled for the Failover Cluster Communication. Disable the remaining cluster networks in Failover Cluster Manager.

Best Practices for SIOSIQ Mirroring

- While creating the SIOS IQ mirroring job, ensure the drive letters of the source and target drives to be mirrored are the same.
- We suggest that you have zero latency in the network when SIOS IQ mirroring, failover/migration of virtual machines between host servers take place. In the case of networks with latency, refer to the SIOS documentation on network requirements.
- While designing the network architecture, particularly with regard to bandwidth between the hosts in the Disaster Recovery environment, make sure to select the bandwidth based on the rate of data change captured from Disk Write Bytes/Sec on the host server for all the mirrored volumes. To verify that you have sufficient network bandwidth to successfully replicate your volume, use the Windows Performance Monitoring and Alerts tool to collect Write Bytes/sec on the replicated volumes to calculate the rate of data change. Collect this counter every 10 seconds and use your own data analysis program to estimate your rate of data change. For more details, refer to SIOS documentation on network requirements.

SIOS IQ can handle the following approximate average rates of change:

Network Bandwidth	**Rate of Change**
1.5 Mbps(T1)	182,000 Bytes/sec (1.45 Mbps)
10 Mbps	1,175,000 Bytes/sec (9.4 Mbps)
45 Mbps (T3)	5,250,000 Bytes/sec (41.75 Mbps)

Network Bandwidth	Rate of Change
100 Mbps	12,000,000 Bytes/sec (96 Mbps)
1000 Mbps (Gigabit)	65,000,000 Bytes/sec (520 Mbps)

The following table lists the impact on CPU utilization and bandwidth with various compression levels.

- Medium Configuration Load: Approx. 50000 IO Points with Approx. 20000 attributes being historized
- Network: Bandwidth controller with bandwidth: 45Mbps and No Latency

  These readings are when the mirroring is continuously happening between the source and destination storage SANs when all the VM are running on the source host server. The data captured shows that the % CPU utilization of the SIOS mirroring process increases with increasing compression levels. Based on these findings we recommend Compression Level 2 in the Medium scale virtualization environment.

	Impact on CPU of Source Host Server		Impact on Bandwidth
	% Processor Time (ExtMirrSvc) - SIOS Mirroring process	**% Processor Time (CPU) - Overall CPU**	**Total Bytes / Sec**
Compression 0	Min: 0 Max:4.679 Avg: 0.157	Min: 0 Max:28.333 Avg: 1.882	Min: 0 Max: 11,042,788 Avg: 2,686,598
Compression 1	Min: 0 Max: 4.680 Avg: 0.254	Min: 0 Max: 31.900 Avg: 1.895	Min: 0 Max: 10,157,373 Avg: 1,871,426
Compression 2	Min: 0 Max:6.239 Avg: 0.402	Min: 0 Max:37.861 Avg: 2.622	Min: 791.970 Max: 10,327,221 Avg: 1,199,242
Compression 9	Min: 0 Max:13.525 Avg: 0.308	Min: 0 Max:42.094 Avg: 3.244	Min: 0 Max: 7,066,439 Avg: 649,822

Additional Guidelines for HADR Implementations (only)

The following guidelines apply to HADR implementation only, and are in addition to all of the guidelines and recommendations listed under "Recommendations and Best Practices", including:

- System Platform Product-specific Recommendations and Observations
- Best Practices for SIOSIQ Mirroring

- Additional Guidelines for DR and HADR Implementations (only)

Though this is a three-node failover topology, to achieve the required failover order, a fourth node is required for setting up the Node Majority in the failover cluster. The three nodes are used for virtual machine services and the fourth node is used for Quorum witness. The fourth node is not meant for failover of virtual machines running on the cluster. This fourth node should not be marked as the preferred owner while setting up the preferred owners for the virtual machines running on the cluster.

The following scenario is a description of the failover order. Node 1 and Node 2 are in High Available site and Node 3 is in Disaster site. The failover sequence is Node 1 > Node 2 > Node 3.

- When all VMs are running on Node 1:
  - All three nodes are up. Now Node 1 goes down. The VMs running on Node 1 move to Node 2.
  - Node 1 and Node 3 are up and Node 2 is down. Now Node 1 goes down. The VMs running on Node 1 move to Node 3.
- When all VMs are running on Node 2:
  - Node 2 and Node 3 are up and Node 1 is down. Now Node 2 goes down. The VMs running on Node 2 move to Node 3.
  - All three nodes are up. Now Node 2 goes down. The VMs running on Node 2 move to Node 3.

Implementing High Availability Using Hyper-V

This section introduces virtualization high-availability solutions that improve the availability of System Platform Products. A high-availability solution masks the effects of a hardware or software failure, and maintains the availability of applications so that the perceived downtime for users is minimized.

The set-up and configuration procedures, expected Recovery Time Objective (RTO) observations, Recovery Point Objective (RPO) observations, and data trend snapshots are presented first for small-scale virtualization environment, and are then repeated for medium-scale virtualization environment.

Small Scale Virtualization Environments

This section contains the following topics:

- Set Up Small Scale Virtualization Environment
- Configuration of System Platform Products in a Typical Small Scale Virtualization
- Expected Recovery Time Objective and Recovery Point Objective

Set Up Small Scale Virtualization Environment

The following procedures help you to set up and implement a small scale virtualization environment.

Note: In the event that the private network becomes disabled, you may need to add a script to enable a failover. For more information, see Add Script to Force Failover of the Virtual Machine.

Plan for Small Scale Virtualization Environment

The following table lists the minimum and recommended hardware and software requirements for the machines used for a small scale virtualization environment:

Important: The following information is provided as an example of this kind of configuration, and is not intended to be used as instructions to set up and configure your environment.

Hyper-V Hosts

Processor:	Two - 2.66 GHz Intel Xeon with - 8 Cores
Operating System	Windows Server 2012 Data Center or higher with Hyper-V Enabled
Memory	12GB
Storage	Local Volume with Capacity of 500 GB

Note: For the Hyper-V Host to function optimally, the server should have the same processor, RAM, storage and service pack level. Preferably the servers should be purchased in pairs to avoid hardware discrepancies. Though the differences are supported, it will impact the performance during failovers.

Virtual Machines

Using the above Specified Hyper-V Host, three virtual machines can be created with the following Configuration.

Virtual Machine 1: DAS SI, Historian, and Application Server (GR) node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Historian, ArchestrA, DAS SI

Virtual Machine 2: Application Server Runtime node 1

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	2 GB
Storage	40 GB
System Platform Products Installed	Application Server Runtime only, and InTouch

Virtual Machine 3: Information Server node, InTouch, and Historian Client

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	40 GB
System Platform Products Installed	Information Server, InTouch, Historian Client

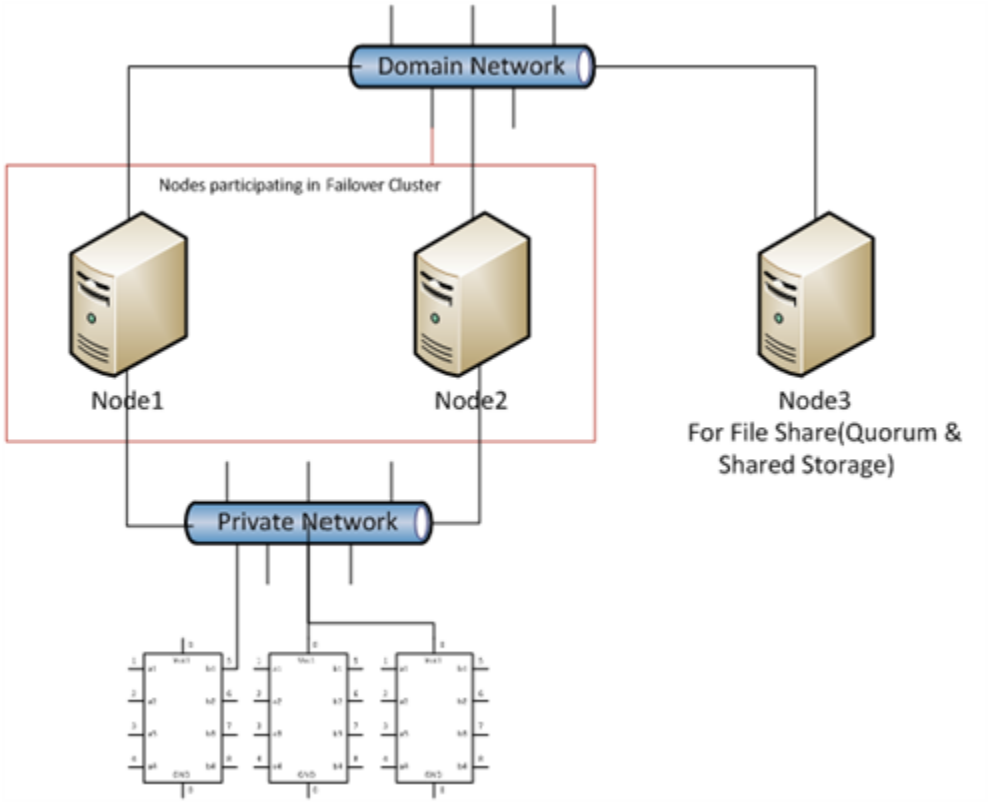
Note: There should be a minimum of two Hyper-V hosts to configure the failover cluster.

Network Requirements

For this high availability architecture, you can use two physical network cards that need to be installed on a host computer and configured, to separate the domain network and the process network.

Configure Failover Cluster

The following is the recommended topology of the failover cluster for a small scale virtualization high availability environment.



This setup requires a minimum of two host servers and one storage server shared across two hosts. Another independent node is used for configuring the quorum. For more information on configuring the quorum, refer to "Configure Cluster Quorum Settings".

The workflow for installing and configuring a failover cluster with two nodes is outlined in the following section.

This workflow is applicable to setting up a small scale virtualization high availability environment.

Install Failover Cluster

To install the failover cluster feature, you need to run Windows Server 2012 or higher Data Center edition on your server. Refer to Microsoft's server documentation for information about installing the failover cluster feature and step-by-step instructions.

Microsoft TechNet Library: Using Hyper-V and Failover Clustering

https://technet.microsoft.com/en-us/library/cc732181%28v=ws.10%29.aspx

Validate Failover Cluster Configuration

You must validate your configuration before you create a cluster. Validation helps you confirm the configuration of your servers, network, and storage meets the specific requirements for failover clusters. Refer to the Microsoft TechNet Library: Using Hyper-V and Failover Clustering for additional information

Create a Cluster

To create a cluster, you need to run the Create Cluster wizard. Refer to Microsoft Windows Server TechNet for information about creating a cluster and step-by-step instructions.

Create a failover cluster:

https://docs.microsoft.com/en-us/windows-server/failover-clustering/create-failover-cluster

Disable the Plant Network for Cluster Communication

After creating the Failover cluster using two or more Network Cards enabled, Make sure only Primary Network card which is used for the Communication between the Hyper-V nodes is enabled for the Failover Communication. You must disable the remaining custer networks.

Configure Cluster Quorum Settings

After both nodes have been added to the cluster, and the cluster networking components have been configured, you must configure the failover cluster quorum. Refer to Microsoft Windows Server TechNet for information about configuring the cluster quorum.

https://technet.microsoft.com/en-us/library/cc731739.aspx

The file share to be used for the node and File Share Majority quorum must be created and secured before configuring the failover cluster quorum. If the file share has not been created or correctly secured, the following procedure to configure a cluster quorum will fail. The file share can be hosted on any computer running a Windows operating system.

To configure the cluster quorum, you need to perform the following procedures:

- Create and secure a file share for the node and file share majority quorum
- Use the failover cluster management tool to configure a node and file share majority quorum

After you configure the cluster quorum, you must validate the cluster. For more information, refer to http://technet.microsoft.com/en-us/library/bb676379(EXCHG.80).aspx.

Configure Storage

For a smaller virtualization environment, storage is one of the central barriers to implementing a good virtualization strategy. But with Hyper-V, VM storage is kept on a Windows file system. Users can put VMs on any file system that a Hyper-V server can access. As a result, HA can be built into the virtualization platform and storage for the virtual machines. This configuration can accommodate a host failure by making storage accessible to all Hyper-V hosts so that any host can run VMs from the same path on the shared folder. The back-end part of this storage can be a local or storage area network, iSCSI or whatever is available to fit the implementation. For this architecture, the Shared Folder is used.

The following table lists the minimum storage recommendations to configure storage for each VM:

System	Processor
Historian and Application Server (GR node) Virtual Machine	80 GB
Application Engine (Runtime node) Virtual Machine	40 GB
InTouch and Information Server Virtual Machine	40 GB

The recommended total storage capacity for a high availability virtual environment should be minimum 1TB.

Configure Hyper-V

With Microsoft Hyper-V, you can create a virtual environment that improves server utilization. It enhances patching, provisioning, management, support tools, processes, and skills. Microsoft Hyper-V provides live migration, cluster shared volume support, expanded processor, and memory support for host systems. Refer to Microsoft Technet library for Hyper-V installation prerequisites and other considerations.

The pre-requisites to set up Hyper-V include:

- x64-based processor
- Hardware-assisted virtualization
- Hardware Data Execution Prevention (DEP)

Configure Virtual Machines

After installing Hyper-V, you need to create a virtual machine. For more information, refer to https://technet.microsoft.com/en-us/library/cc772480.aspx

Add Script to Force Failover of the Virtual Machine

As part of configuration of the Virtual Machine, add a script to force failover of the Virtual Machine if the Domain/ Private Network is disabled.

Whenever public network is disconnected on the node where the virtual machines are running, Failover Cluster Manager forces failover of all the Virtual Machine Services and applications to the other host node in the cluster. If the private network which is not participating in the cluster communication fails, Failover Cluster Manager does not failover any Cluster Service or Application.

To overcome this, we need to add a script which detects the private network failure as a dependency to the Virtual Machine. This results in failover of the Virtual Machine when the script fails.

Follow the process mentioned in the following URL to add the script: http://gallery.technet.microsoft.com/ScriptCenter/5f7b4df3-af02-47bf-b275-154e5edf17e6/

The recommended setting for maximum failures in a period is 15, and the period should be set to 1 hour.

Configuration of System Platform Products in a Typical Small Scale Virtualization

To record the expected Recovery Time Objective (RTO) and Recovery Point Objective (RPO) trends and various observations in a small scale virtualization environment, tests are performed with System Platform Product configuration shown below.

The virtualization host server used for small scale configuration consists of three virtual machines listed below.

Node 1: GR, Historian and DAS SI Direct - Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit

Node 2 (AppEngine): Bootstrap, IDE and InTouch (Managed App) - Windows 2012 Data Center edition (64bit)

Node 3: Information Server, Bootstrap and IDE, InTouch Terminal Service and Historian Client - Windows Server 2012 Data Center edition (64bit) with SQL Server 2012 SP2 and Microsoft Office

Important: The following information is provided as an example of this kind of configuration, and is not intended to be used as instructions to set up and configure your environment.

Virtual Node	IO tags (Approx.)	Historized tags (Approx.)
GR	10000	2500
AppEngine	10000	5000

Expected Recovery Time Objective and Recovery Point Objective (Small Scale)

The following table shows the indicative Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) for the load of IO and Attributes historized shown above and with the configuration of Host Virtualization Servers and Hyper-V Virtual Machines explained in the Setup instructions of Small Scale Virtualization. In addition to these factors, the exact RTO and RPO depend on factors like the CPU utilization, memory usage and network usage at the time of failover/migration activity.

RTO and RPO Observations-HA Small Configuration

Important: The following sample data are provided only as guidelines for establishing testing specific to your needs.

Scenarios and observations in this section:

Scenario	Observations
Scenario 1: IT provides maintenance on Virtualization Server	Live Migration
	Quick Migration
	Quick Migration of all Nodes Simulatenously
	Shut down
Scenario 2: Virtualization Server hardware fails	Failover occurs due to hardware failure, simulated with power-off on the host server.

Scenario 3: Network fails on Virtualization Server	Failover due to public network disconnect
	Failover due to private network disconnect
Scenario 4: Virtualization Server becomes unresponsive	No failover of VMs occurs when CPU utilization is 100%.

The following tables display RTO and RPO observations with approximately 20000 IO points with approximately 7500 attributes being historized:

Scenario 1: IT provides maintenance on Virtualization Server

Live Migration

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	2 sec	Application Server tag (script)	8 sec
			Application Server IO tag (SiDirect)	13 sec
	Historian Client	2 sec	Historian local tag	0 sec
			InTouch Tag $Second	4 sec
			Application Server IO tag (SiDirect)	20 sec
			Application Server tag (script)	0 sec
	Communication Driver	5 sec	N/A	N/A
Information Server	InTouch HMI	5 sec		5 sec
	Information Server		N/A	N/A
	Historian Client		N/A	N/A
AppEngine	AppEngine	1 sec	Application Server IO tag (SiDirect)	3 sec
			Application Server tag (script)	6 sec

Quick Migration

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	134 sec	Application Server tag (script)	183 sec

			Application Server IO tag (SiDirect)	184 sec
	Historian Client	145 sec	Historian Local tag	148 sec
			InTouch Tag $Second	152 sec
			Application Server IO tag (SiDirect)	165 sec
			Application Server tag (script)	0 sec
	Communication Driver	146 sec	N/A	N/A
Information Server Node	InTouch	79 sec		89 sec
	Information Server	79 sec	N/A	N/A
	Historian Client	79 sec	N/A	N/A
AppEngine	AppEngine	59 sec	Application Server IO tag (SiDirect)	105 sec
			Application Server tag (script)	104 sec

Quick Migration of all nodes simultaneously

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	188 sec	Application Server tag (script)	222 sec
			Application Server IO tag (SiDirect)	227 sec
	Historian Client	220 sec	Historian Local tag	221 sec
			InTouch Tag $Second	228 sec
			Application Server IO tag (SiDirect)	238 sec
			Application Server tag (script)	135 sec
	OI Server	221 sec	N/A	N/A

Information Server Node	InTouch	183 sec		228 sec
	Information Server	183 sec	N/A	N/A
	Historian Client	183 sec	N/A	N/A
AppEngine	AppEngine	100 sec	Application Server IO tag (SiDirect)	238 sec
			Application Server tag (script)	135 sec

Shut down

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	160 sec	Application Server tag (script)	3 min 36 sec
			Application Server IO tag (SiDirect)	3 min 43 sec
	Historian Client	211 sec	Historian Local tag	3 min 25 sec
			InTouch Tag $Second	3 min 32 sec
			Application Server IO tag (SiDirect)	3 min 50 sec
			Application Server tag (script)	2 min 46 sec
	OI Server	212 sec	N/A	N/A
Information Server Node	InTouch	202 sec		212 sec
	Information Server	202 sec	N/A	N/A
	Historian Client	202 sec	N/A	N/A
AppEngine	AppEngine	114 sec	Application Server IO tag (SiDirect)	3 min 50 sec
			Application Server tag (script)	2 min 46 sec

Scenario 2: Virtualization Server hardware fails

The failover occurs due to hardware failure, and it is simulated with power-off on the host server.

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	497 sec	Application Server tag (script)	9 min
			Application Server IO tag (SiDirect)	9 min
	Historian Client	532 sec	Historian Local tag	9 min 23 sec
			InTouch Tag $Second	10 min + time taken to start viewer
			Note: RPO is dependent on the time taken by the user to start the InTouchViewApp on the InTouch node and the RTO of the Historian node, which historizes this tag.	
			Application Server IO tag (SiDirect)	8 min 23 sec
			Application Server tag (script)	7 min 1 sec
	OI Server	269 sec	N/A	N/A
Information Server Node	InTouch	10 min 1 sec + time taken by the user to start the InTouchViewApp		10 min 11 sec
			Note: RPO is dependent on the time taken by the user to start the InTouchViewApp on the InTouch node and the RTO of the Historian node, which historizes this tag.	
	Information Server	10 min 1 sec + time taken by the user to start the Information Server	N/A	N/A
	Historian Client	10 min 1 sec + time taken by the user to start the Hist Client	N/A	N/A
AppEngine	AppEngine	6 min 6 sec	Application Server IO tag (SiDirect)	8 min 23 sec
			Application Server tag (script)	7 min 1 sec

Scenario 3: Network fails on Virtualization Server

The failover occurs due to public network disconnect. In this case, the VMs restart after moving to the other host server.

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	535 sec	Application Server tag (script)	9 min 8 sec
			Application Server IO tag (SiDirect)	8 min 53 sec
	Historian Client	544 sec	Historian Local tag	9 min 35 sec
			InTouch Tag $Second	9 min 16 sec
			Note: RPO is dependent on the time taken by the user to start the InTouchViewApp on the InTouch node and the RTO of the Historian node, which historizes this tag.	
			Application Server IO tag (SiDirect)	8 min 57 sec
			Application Server tag (script)	7 min 52 sec
	OI Server	457 sec	N/A	N/A
Information Server Node	InTouch	6 min 55 sec + time taken by the user to start the InTouchViewApp		9 min 54 sec
			Note: RPO is dependent on the time taken by the user to start the InTouchViewApp on the InTouch node and the RTO of the Historian node, which historizes this tag.	
	Information Server	6 min 45 sec + time taken by the user to start the Information Server	N/A	N/A
	Historian Client	6 min 45 sec + time taken by the user to start the Hist Client	N/A	N/A
AppEngine	AppEngine	7 min 43 sec	Application Server IO tag (SiDirect)	8 min 57 sec
			Application Server tag (script)	7 min 52 sec

Failover due to private network disconnect

In this case, the private network disconnects on GR, VM will be moved to the other host server.

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	118 sec	Application Server tag (script)	132 sec
			Application Server IO tag (SiDirect)	140 sec
	Historian Client	2 sec	Historian Local tag	132 sec
			InTouch Tag $Second	147 sec
			Note: RPO is dependent on the time taken by the user to start the InTouchViewApp on the InTouch node and the RTO of the Historian node, which historizes this tag.	
			Application Server IO tag (SiDirect)	145 sec
			Application Server tag (script)	0 (Sfed)
	OI Server	134 sec	N/A	N/A
Information Server Node	InTouch	N/A	N/A	/A
	Information Server	N/A	N/A	N/A
	Historian Client	N/A	N/A	N/A
AppEngine	AppEngine	N/A	Application Server IO tag (SiDirect)	
			Application Server tag (script)	

Scenario 4: Virtualization Server becomes unresponsive

There is no failover of VMs to the other host server when the CPU utilization on the host server is 100%.

Primary Node	Products	RTO	RPO (Tag/Data Loss Duration)	
GR	Application Server	N/A	N/A	N/A
	Historian Client	N/A	N/A	N/A
Information Server Node	InTouch HMI	N/A	N/A	N/A
	Information Server	N/A	N/A	N/A
	Historian Client	N/A	N/A	N/A

AppEngine	AppEngine	N/A	N/A	N/A
	InTouch HMI	N/A	N/A	N/A

Medium Scale Virtualization Environments

This section contains the following topics:

Set Up Medium Scale Virtualization Environment

The following procedures help you to set up and implement the medium scale virtualization high availability environment.

Note: In the event that the private network becomes disabled, you may need to add a script to enable a failover. For more information, see Add Script to Force Failover of the Virtual Machine.

Plan for Medium Scale Virtualization Environment

The minimum recommended hardware and software requirements for the Host and Virtual machines used for medium virtualization environment are provided in the table below:

Hyper-V Host

Processor	Two 2.79 GHz Intel Xeon with 24 Cores
Operating System	Windows Server 2012 Data Center or higher with Hyper-V enabled
Memory	48 GB
Storage	SAN with 1TB storage disk

Note: For the Hyper-V Host to function optimally, the server should have the same processor, RAM, storage and service pack level. Preferably the servers should be purchased in pairs to avoid hardware discrepancies. Though the differences are supported, it will impact the performance during failovers.

Virtual Machines

Using the Hyper-V host specified above, six virtual machines can be created in the environment with the configuration given below.

Virtual Machine 1: Historian node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher

Memory	8 GB
Storage	200 GB
System Platform Products Installed	Historian

Virtual Machine 2: Application Server node, DAS SI

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	100 GB
System Platform Products Installed	ArchestrA-Runtime, DAS SI

Virtual Machine 3: InTouch TS node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	InTouch HMI

Virtual Machine 4: Application Server Runtime node 1

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 or higher Standard
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Application Server Runtime only and InTouch

Virtual Machine 5: Application Server Runtime node 2

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Application Server Runtime only

Virtual Machine 6: Historian Client node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows 8.1 or higher Enterprise
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Historian Client

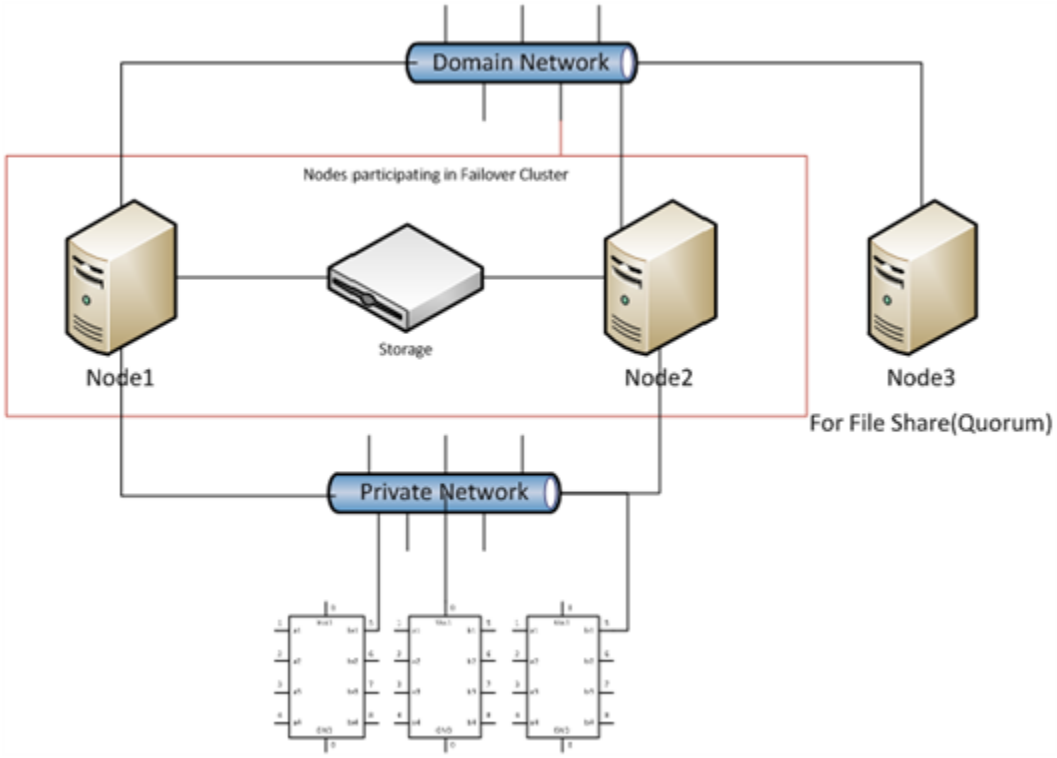
Note: There should be a minimum of two Hyper-V hosts to configure the failover cluster.

Network Requirements

For this high availability architecture, you can use two physical network cards that need to be installed on a host computer and configured to separate the domain network and the process network.

Configure Failover Cluster

The following is the recommended topology of the failover cluster for a medium scale virtualization high availability environment.



This setup requires a minimum of two host servers and one storage server shared across two hosts. Another independent node is used for configuring the quorum. For more information on configuring the quorum, refer to "Configure Cluster Quorum Settings".

The workflow for installing and configuring a failover cluster with two nodes is outlined in the following section. This workflow is applicable to setting up a medium scale virtualization high availability environment.

Install Failover Cluster

To install the failover cluster feature, you need to run Windows Server 2012 or higher Enterprise Edition on your server. Refer to Microsoft's server documentation for information about installing the failover cluster feature and step-by-step instructions.

Microsoft TechNet Library: Using Hyper-V and Failover Clustering

https://technet.microsoft.com/en-us/library/cc732181%28v=ws.10%29.aspx

Validate Failover Cluster Configuration

You must validate your configuration before you create a cluster. Validation helps you confirm the configuration of your servers, network, and storage meets the specific requirements for failover clusters.

Create a Cluster

To create a cluster, you need to run the Create Cluster wizard. Refer to Microsoft Windows Server TechNet for information about creating a cluster and step-by-step instructions.

Create a failover cluster:

https://docs.microsoft.com/en-us/windows-server/failover-clustering/create-failover-cluster

Disable the Plant Network for Cluster Communication

After creating the Failover cluster using two or more Network Cards enabled, Make sure only Primary Network card which is used for the Communication between the Hyper-V nodes is enabled for the Failover Communication Disable the remaining Cluster Networks

Configure Cluster Quorum Settings

Quorum is the number of elements that need to be online to enable continuous running of a cluster. In most instances, the elements are nodes. In some cases, the elements also consist of disk or file share witnesses. Each of these elements determines whether the cluster should continue to run. Refer to Microsoft Windows Server TechNet for information about configuring the cluster quorum.

https://technet.microsoft.com/en-us/library/cc731739.aspx

All elements, except the file share witnesses, have a copy of the cluster configuration. The cluster service ensures that the copies are always synchronized. The cluster should stop running if there are multiple failures or if there is a communication error between the cluster nodes.

After both nodes have been added to the cluster, and the cluster networking components have been configured, you must configure the failover cluster quorum.

The file share to be used for the node and File Share Majority quorum must be created and secured before configuring the failover cluster quorum. If the file share has not been created or correctly secured, the following procedure to configure a cluster quorum will fail. The file share can be hosted on any computer running a Windows operating system.

To configure the cluster quorum, you need to perform the following procedures:

- Create and secure a file share for the node and file share majority quorum
- Use the failover cluster management tool to configure a node and file share majority quorum

After you configure the cluster quorum, you must validate the cluster. For more information, refer to http://technet.microsoft.com/en-us/library/bb676379(EXCHG.80).aspx.

Configure Storage

For any virtualization environment, storage is one of the central barriers to implementing a good virtualization strategy. But with Hyper-V, VM storage is kept on a Windows file system. Users can put VMs on any file system that a Hyper-V server can access. As a result, you can build HA into the virtualization platform and storage for the virtual machines. This configuration can accommodate a host failure by making storage accessible to all Hyper-V

hosts so that any host can run VMs from the same path on the shared folder. The back-end part of this storage can be a local storage area network, iSCSI or whatever is available to fit the implementation.

The following table lists the minimum storage recommendations for each VM:

System	Processor
Historian Virtual Machine	200 GB
Application Server (GR node) Virtual Machine	100 GB
Application Engine 1(Runtime node) Virtual Machine	80 GB
Application Engine 2 (Runtime node) Virtual Machine	80 GB
InTouch Virtual Machine	80 GB
Information Server Virtual Machine	80 GB
Historian Client	80 GB

The recommended total storage capacity for a high availability virtual environment should be minimum 1TB.

Configure Hyper-V

With Microsoft Hyper-V, you can create a virtual environment that improves server utilization. It enhances patching, provisioning, management, support tools, processes, and skills. Microsoft Hyper-V provides live migration, cluster shared volume support, expanded processor, and memory support for host systems. Refer to Microsoft Technet library for Hyper-V installation prerequisites and other considerations.

The pre-requisites to set up Hyper-V include:

- x64-based processor
- Hardware-assisted virtualization
- Hardware Data Execution Prevention (DEP)

Configure Virtual Machines

After installing Hyper-V, you need to create a virtual machine. For more information, refer to https://technet.microsoft.com/en-us/library/cc772480.aspx

Add Script to Force Failover of the Virtual Machine

As part of configuration of the Virtual Machine, add a script to force failover of the Virtual Machine if the Domain/ Private Network is disabled.

Whenever public network is disconnected on the node where the virtual machines are running, Failover Cluster Manager forces failover of all the Virtual Machine Services and applications to the other host node in the cluster.

If the private network which is not participating in the cluster communication fails, Failover Cluster Manager does not failover any Cluster Service or Application.

To overcome this, we need to add a script which detects the private network failure as a dependency to the Virtual Machine. This results in failover of the Virtual Machine when the script fails.

Follow the process mentioned in the following URL to add the script: http://gallery.technet.microsoft.com/ScriptCenter/5f7b4df3-af02-47bf-b275-154e5edf17e6/

Configuration of System Platform Products in a Typical Medium Scale Virtualization

To record the expected Recovery Time Objective (RPO) and Recovery Point Objective (RPO), trends and various observations in a medium scale virtualization environment, tests are performed with System Platform Product configuration shown below.

The virtualization host server used for medium scale configuration consists of six virtual machines listed below.

Important: The following information is provided as an example of this kind of configuration, and is not intended to be used as instructions to set up and configure your environment.

Node 1 (GR): GR, InTouch and DAS SI Direct – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit

Node 2 (AppEngine1): Bootstrap, IDE and InTouch (Managed App) – Windows 2012 Data Center edition (64bit)

Node 3 (AppEngine2): Bootstrap, IDE – Windows 2012 Data Center edition (64bit)

Node 4: Historian – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit

Node 5: InTouch – Windows 2012 Data Center edition (64bit) with RDS enabled

Node 6: Historian Client and InTouch – Windows 8.1 Professional Edition (64bit) with SQL Server 2012 SP2 32 bit

Virtual Node	IO tags (Approx.)	Historized tags (Approx.)
AppEngine1	25000	10000
AppEngine2	25000	10000

Expected Recovery Time Objective and Recovery Point Objective (Medium Scale)

This section provides the indicative Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) for the load of IO and Attributes historized shown above and with the configuration of Host Virtualization Servers and Hyper-V virtual machines explained in the Setup instructions of Medium Scale Virtualization. In addition to these factors, the exact RTO and RPO depend on factors like storage I/O performance, CPU utilization, memory usage, and network usage at the time of failover/migration activity.

RTO and RPO Observations—HA Medium Configuration

Important: The following sample data are provided only as guidelines for establishing testing specific to your needs.

Scenarios and observations in this section:

Scenario	Observation
Scenario 1: IT provides maintenance on Virtualization Server	"Scenario 1: IT provides maintenance on Virtualization Server"
	"Quick Migration"
	"Quick Migration of all nodes simultaneously"
Scenario 2: Virtualization Server hardware fails	"Scenario 2: Virtualization Server hardware fails"
Scenario 3: Network fails on Virtualization Server	"Scenario 3: Network fails on Virtualization Server"
Scenario 4: Virtualization Server becomes unresponsive	"Scenario 4: Virtualization Server becomes unresponsive"

The following tables display RTO and RPO observations with approximately 50000 IO points with approximately 20000 attributes being historized:

Scenario 1: IT provides maintenance on Virtualization Server

Live Migration

Products	RTO	RPO (Tag/Data Loss Duration)	
InTouch HMI	13 sec	Data Loss for $Second tag (Imported to Historian)	13 sec
GR	10 sec	Application Server Tag (Script)	12 sec
		Application Server IO Tag (DASSiDirect)	59 sec
AppEngine1	15 sec	Application Tag (Script)	22 sec
		Application Server IO Tag (DASSiDirect)	57 sec
AppEngine2	7 sec	Application Server Tag (Script)	11 sec
		Application Server IO Tag (DASSiDirect)	57 sec
Historian Client	9 sec	SysTimeSec (Historian)	0 sec
		$Second (InTouch)	2 sec
		Application Server Tag	0 (Data is SFed)

		(Script)	
		Application Server IO Tag (DASSiDirect)	0 (Data is SFed)
OI Server SIDirect	14 sec	N/A	N/A
Historian Client	0 sec	N/A	N/A
Information Server	5 sec	N/A	N/A

Quick Migration

Products	RTO	RPO (Tag/Data Loss Duration)	
InTouch HMI	31 sec	Data Loss for $Second tag (Imported to Historian)	27 sec
GR	50 sec	IAS Tag (Script)	50 sec
		Application Server IO Tag (DASSiDirect)	1 Min 51 Sec
AppEngine1	35 sec	Application Server Tag (Script)	35 sec
		Application Server IO Tag (DASSiDirect)	54 sec
AppEngine2	41 sec	Application Server Tag (Script)	44 sec
		Application Server IO Tag (DASSiDirect)	1 Min 14 Sec
Historian Client	84 sec	SysTimeSec (Historian)	1 Min 25 Sec
		$Second (InTouch)	1 Min 51 Sec
		Application Server Tag (Script)	0 (data is SFed)
		Application Server IO Tag (DASSiDirect)	0 (data is SFed)
OI Server SIDirect	50 sec	N/A	N/A

Historian Client	1 Min 32 Sec	N/A	N/A
Information Server	33 sec	N/A	N/A

Quick Migration of all nodes simultaneously

The following table displays the data for Quick Migration of all nodes.

Products	RTO	RPO (Tag/Data Loss Duration)	
InTouch HMI	28 Sec	Data Loss for $Second tag (Imported to Historian)	1 Min 40 Sec
GR	04 Sec	Application Server Tag (Script)	1 Min 36 Sec
		Application Server IO Tag (DASSiDirect)	4 Min 14 Sec
AppEngine1	67 Sec	Application Server Tag (Script)	1 Min 20 Sec
		Application Server IO Tag (DASSiDirect)	4 Min 11 Sec
AppEngine2	54 Sec	Application Server Tag (Script)	52 Sec
		Application Server IO Tag (DASSiDirect)	4 Min 28 Sec
Historian Client	73 Sec	SysTimeSec (Historian)	1 Min 14 Sec
		$Second (InTouch)	1 Min 40 Sec
		Application ServerTag (Script)	1 Min 36 Sec
		Application Server IO Tag (DASSiDirect)	4 Min 14 Sec
OI Server SIDirect	107 Sec	N/A	
Historian Client	38 Sec	N/A	
Information Server	36 Sec	N/A	

Scenario 2: Virtualization Server hardware fails

The Virtualization Server hardware failure results in failover that is simulated with power-off on the host server. In this case, the VMs restart, after moving to the other host server.

Products	RTO	RPO (Tag/Data Loss Duration)	
InTouch HMI	335 Sec + time taken by the user to start the InTouchView	Data Loss for $Second tag (Imported to Historian)	6 Min 47 Sec.
		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
GR	313 Sec	Application Server Tag (Script)	5 Min 44 Sec
		Application Server IO Tag (DASSiDirect)	7 Min 28 Sec
AppEngine1	365 Sec	Application Server Tag (Script)	6 Min 35 Sec
		Application Server IO Tag (DASSiDirect)	7 Min 29 Sec
AppEngine2	372 Sec	Application Server Tag (Script)	6 Min 41 Sec
		Application Server IO Tag (DASSiDirect)	7 Min 20 Sec
Historian Client	381 Sec	SysTimeSec (Historian)	6 Min 33 Sec
		$Second (InTouch)	6 Min 47 Sec
		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
		Application Server Tag (Script)	5 Min 45 Sec
		Application Server IO Tag (DASSiDirect)	7 Min 30 Sec
OI Server	265 Sec	N/A	N/A

SIDirect			
Historian Client	214 Sec + time taken by the user to start the Historian Client	N/A	N/A
Information Server	255 Sec + time taken by the user to start the Information Server	N/A	N/A

Scenario 3: Network fails on Virtualization Server

Failover due to Network Disconnect (Public)

In this case, after the VMs move to the other host server, the VMs restart.

Products	RTO	RPO (Tag/Data Loss Duration)	
InTouch HMI	150 sec + time taken by the user to start the InTouchView	Data Loss for $Second tag (Imported to Historian)	4 Min 14 Sec
		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
GR	197 sec	Application Server Tag (Script)	3 Min 41 Sec
		Application Server IO Tag (DASSiDirect)	3 Min 50 Sec

Products	RTO	RPO (Tag/Data Loss Duration)	
AppEngine1	188 sec	Application Server Tag (Script)	3 Min 31 Sec
		Application Server IO Tag (SiDirect)	4 Min 2 Sec
AppEngine2	200 sec	Application Server Tag (Script)	3 Min 41 Sec
		Application Server IO Tag (SiDirect)	4 Min 08 Sec

Historian Client	236 sec	SysTimeSec (Historian)	3 Min 55 Sec
		$Second (InTouch)	4 Min 14 Sec
		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
		Application Server Tag (Script)	3 Min 41 Sec
		Application Server IO Tag (SiDirect)	3 Min 50 Sec
OI Server SIDirect	174 sec	N/A	N/A
Historian Client	163 sec + time taken by the user to start the Historian Client	N/A	N/A
Information Server	66 sec + time taken by the user to start the Information Server	N/A	N/A

Failover due to network disconnect (plant)

In this case, only the GR Node moves to other host server and restarts. Only GR has data acquisition through Plant network and disconnected Plant network results in failover of GR alone.

Products	RTO	RPO (Tag/Data Loss Duration)	
InTouch HMI	N/A	Data Loss for $Second tag (Imported to Historian)	N/A
GR	97 Sec	IAS Tag (Script)	1 Min 43 Sec
		Application Server IO Tag (SiDirect)	1 Min 46 Sec
AppEngine1	N/A	Application Server Tag (Script)	N/A
		Application Server IO Tag (SiDirect)	1 Min 50 Sec
AppEngine2	N/A	Application Server Tag (Script)	N/A

		Application Server IO Tag (DASSiDirect)	1 Min 58 Sec
Historian Client	N/A	SysTimeSec (Historian)	N/A
		$Second (InTouch)	N/A
		Application Server Tag (Script)	1 Min 43 Sec
		Application Server IO Tag (SiDirect)	1 Min 46 Sec
OI Server SIDirect	111 Sec	N/A	N/A
Historian Client	N/A	N/A	N/A
Information Server	N/A	N/A	N/A

Scenario 4: Virtualization Server becomes unresponsive

There is no failover of VMs to the other host server when the CPU utilization on the host server is 100%.

Products	RTO	RPO (Tag/Data Loss Duration)	
InTouch HMI	N/A	N/A	N/A
GR	N/A	N/A	N/A
	N/A	N/A	N/A
AppEngine1	N/A	N/A	N/A
	N/A	N/A	N/A
AppEngine2	N/A	N/A	N/A
Historian Client	N/A	N/A	N/A
	N/A	N/A	N/A
	N/A	N/A	N/A
	N/A	N/A	N/A
OI Server SIDirect	N/A	N/A	N/A
Historian	N/A	N/A	N/A

Client			
Information Server	N/A	N/A	N/A

Implementing High Availability Using vSphere

The following procedures are designed to help you set up and implement High Availability using VMware vSphere. These procedures assume that you have VMware ESXi™ 5.0 or above, vCenter Server™, and vSphere Client already installed.

For basic procedures to install these and other VMware products, see product support and user documentation at http://www.vmware.com/.

The High Availability vSphere implementation assumes that you are implementing a a medium-scale system.

This section contains the following topics:

- Plan the Virtualization Environment
- Configuration of System Platform Products in a Typical Virtualization Environment
- Set up the Virtualization Environment
- Expected Recovery Time Objective and Recovery Point Objective

Plan the Virtualization Environment

The minimum recommended hardware and software requirements for the Host and Virtual machines used for virtualization environment are provided in the following table:

ESXi Host

Processor	Two 2.79 GHz Intel Xeon with 8 cores (Hyper-threaded)
Operating System	ESXi 5.0 or higher
Memory	48 GB
Storage	SAN with 1TB storage disk

Note: For the ESXi Host to function optimally, the server should have the same processor, RAM, storage and service pack level. Preferably the servers should be purchased in pairs to avoid hardware discrepancies. Though the differences are supported, it will impact the performance during failovers.

Virtual Machines

Using the ESXi host specified above, six virtual machines can be created in the environment with the configuration given below.

Virtual Machine 1: Historian node

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	4 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	200 GB
System Platform Products Installed	Historian

Virtual Machine 2: Application Server node, DAS SI

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	4 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	100 GB
System Platform Products Installed	ArchestrA-Runtime, DAS SI

Virtual Machine 3: InTouch HMI node

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	2 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	InTouch HMI

Virtual Machine 4: Application Server Runtime node 1

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	2 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB

System Platform Products Installed	Application Server Runtime only and InTouch

Virtual Machine 5: Application Server Runtime node 2

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	2 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Application Server Runtime only

Virtual Machine 6: Historian Client node

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	1 vCPUs
Operating System	Windows 8.1 or higher Enterprise
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Historian Client

Note: There should be a minimum of two vSphere hosts to configure the failover cluster.

Network Requirements

For this high availability architecture, you can use two physical network cards that need to be installed on a host computer and configured to separate the domain network and the process network.

Configuration of System Platform Products in a Typical Virtualization Environment

To record the expected Recovery Time Objective (RPO) and Recovery Point Objective (RPO), trends and various observations in a virtualization environment, tests are performed with System Platform Product configuration shown below.

The virtualization host server used for configuration consists of seven virtual machines listed below.

Node 1 (GR): GR, InTouch and DAS SI Direct – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit

Node 2 (AppEngine1): Bootstrap, IDE and InTouch (Managed App) – Windows 2012 Data Center edition (64bit)

Node 3 (AppEngine2): Bootstrap, IDE – Windows 2012 Data Center edition (64bit)

Node 4: Historian – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit

Node 5: Information Server, Bootstrap and IDE – Windows Server 2012 Data Center edition (64bit) with SQL

Server 2023 SP2 and Microsoft Office

Node 6: InTouch Terminal Service – Windows 2012 Data Center edition (64bit) with RDS enabled

Node 7: Historian Client and InTouch – Windows 8.1 Professional Edition (64bit) with SQL Server 2012 SP2 32 bit

Virtual Node	IO tags (Approx.)	Historized tags(Approx.)
AppEngine1	25000	10000
AppEngine2	25000	10000

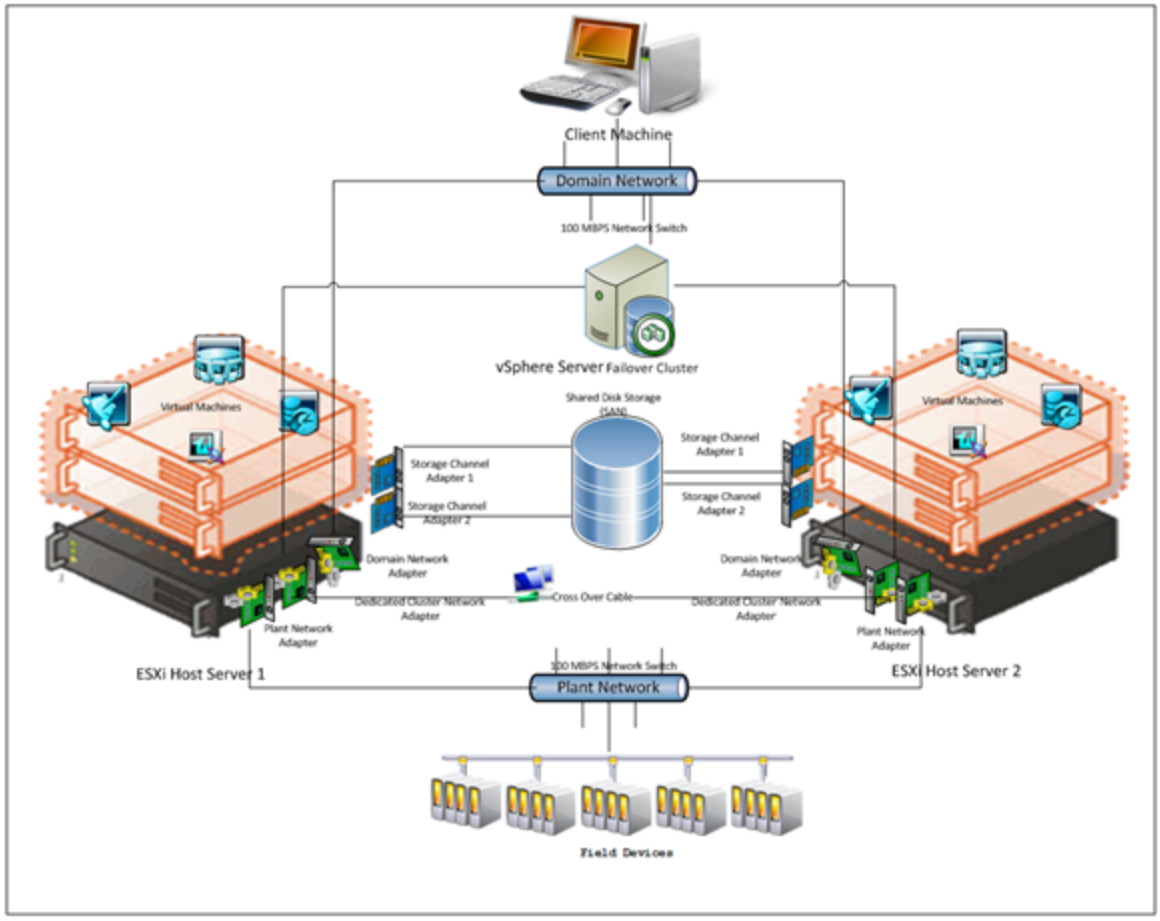
Set up the Virtualization Environment

The following procedures help you to set up and implement the high availability virtualization environment using vSphere technology.

Note: In the event that the private network becomes disabled, you may need to add a script to enable a failover.

Create a Datacenter

The vSphere Datacenter virtualizes an infrastructure that includes servers, storage, networks. It provides for end-to-end connectivity between client machines and field devices. The following is the recommended topology of the Datacenter, with a vSphere Failover Cluster, for a High Availability environment.



The following workflow outlines how to configure a Datacenter as a virtualized High Availability environment with a failover cluster consisting of two nodes. This setup requires a minimum of two host servers and one storage server shared across two hosts.

Create the Datacenter

Use the vSphere Client to create the DataCenter. Refer to your vSphere documentation for additional information.

Add hosts to the Datacenter

Refer to your vSphere documentation or the VMware knowledge base for additional information and add an ESXi host.

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2032896

Repeat this procedure to add another ESXi host.

Create a Failover Cluster

A cluster in vSphere is a group of hosts. Resources of a host added to a cluster, also known as a failover cluster, become part of the cluster's resources, and are managed by the cluster. In a vSphere High Availability environment, virtual machines automatically restart on a different physical server in a cluster if a host fails.

Refer to your vSphere documentation or the VMware knowledge base for information about adding the failover cluster.

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2032896

To create a failover cluster:

1. Select the new cluster option.
2. Select vSphere HA.
3. Use the New Cluster Wizard to complete configuration.

Configure Storage

VMware Virtual Machine File System (VMFS) datastores serve as repositories for virtual machines. You can set up VMFS data stores on any SCSI-based storage devices that the host discovers, including Fibre Channel, iSCSI, and local storage devices.

Use the following workflow to create a datastore. Your new datastore is added to all hosts if you use the vCenter Server system to manage your hosts.

Important: Install and configure any adapter that your storage requires before creating datastores. After you create a datastore, rescan the adapters to discover the new storage device.

Create a datastore

1. Log on to vSphere Client and select a host.
2. Configure storage for the host.
3. Configure the file system version.
4. Check the parameters you have selected and create the datastore.

Configure Networks

After you create a datacenter, use the following the workflow to configure one or more networks on the ESXi host.

Configure networks on the ESXi host

1. Log on to vSphere Client and select a host
2. Add networking through the Add Network Wizard.
3. Click **Add Networking.** The **Add Network Wizard** appears.
4. Use the Wizard to complete network configuration.

Create a Virtual Machine in vSphere Client

You can populate your virtualization environment by creating virtual machines, which are the key components in a virtual infrastructure.

When you create a virtual machine, you associate it with a particular datacenter, host, cluster or resource pool, and a datastore. The virtual machine consumes resources dynamically as the workload increases, or it returns resources dynamically as the workload decreases.

Every virtual machine has virtual devices that provide the same function as the physical hardware. A virtual machine derives the following attributes from the host with which it is associated:

- A CPU and memory space
- Access to storage
- Network connectivity

User vSphere Client to create a Virtual Machine and configure its properties.

Enable vMotion for Migration

VMware vMotion enables migration of a running virtual machine from one server to another, including the VM's associated storage, network identity, and network connections. Access to the VM's storage switches to the new physical host. Access to the VM continues with its same virtualized network identity.

Following are typical migration scenarios:

- Removing VMs from underperforming or problematic servers
- Performing hardware maintenance and upgrades
- Optimizing VMs within resource pools

Use vSphere Client to enable vMotion for migration.

Implementing Disaster Recovery Using Hyper-V

This section introduces several Disaster Recovery (DR) virtualization solutions that improve the availability of System Platform Products. For more information refer to Chapter 1 Getting Started with High Availability and Disaster Recovery.

The set-up and configuration procedures, expected Recovery Time Objective (RTO) observations, Recovery Point Objective (RPO) observations, and data trend snapshots are presented first for small-scale virtualization environment, and are then repeated for medium-scale virtualization environment.

Small Scale Virtualization Environments

This chapter contains the following topics:

Set Up Small Scale Virtualization Environment

The following procedures help you to set up small scale virtualization disaster recovery environment.

Plan for Disaster Recovery

The minimum and recommended hardware and software requirements for the Host and Virtual machines used for small scale virtualization disaster recovery environment.

Hyper-V Hosts

Processor	Two 2.66 GHz Intel Xeon with 8 Cores
Operating System	Windows Server 2012 or higher Enterprise with Hyper-V enabled
Memory	12 GB
Storage	Local Volume with Capacity 500 GB

Note: For the Hyper-V Host to function optimally, the server should have the same processor, RAM, storage and service pack level. Preferably the servers should be purchased in pairs to avoid hardware discrepancies. Though the differences are supported, it will impact the performance during failovers.

Virtual Machines

Using the above specified Hyper-V host, three virtual machines can be created with below configuration.

Virtual Machine 1: DAS SI, Historian, and Application Server (GR) Node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 or higher Standard
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Historian, ArchestrA, DAS SI

Virtual Machine 2: Application Server Runtime Node 1

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 or higher Standard
Memory	2 GB
Storage	40 GB
System Platform Products Installed	Application Server Runtime only and InTouch

Virtual Machine 3: Information Server Node, InTouch, Historian Client

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 or higher Standard
Memory	4 GB

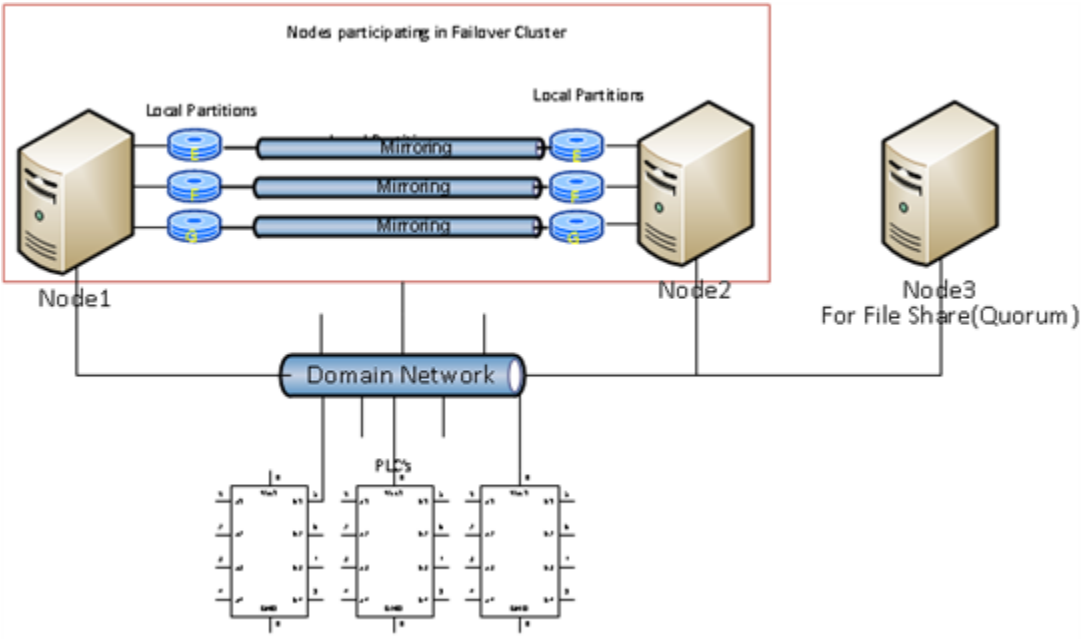
Storage	40 GB
System Platform Products Installed	InTouch, Historian Client

Network Requirements

For this architecture, you can use one physical network card that needs to be installed on a host computer for the domain network and the process network.

Configure Failover Cluster

The recommended topology of the failover cluster for disaster recovery process for small scale virtualization environment is given below:



This setup requires a minimum of two host servers with sufficient local disk space on each server to create logical drives for the virtual machines. Each logical drive is replicated to the two hosts for disaster recovery. Another independent node is used for configuring the quorum. For more information on configuring the quorum, refer to "Configure Cluster Quorum Settings".

The workflow for setting up the small virtualization disaster recovery environment is outlined in the following section.

Install Failover Cluster

To install the failover cluster feature, you need to run Windows Server 2012 or higher on your server. Refer to Microsoft's server documentation for information about installing the failover cluster feature and step-by-step instructions.

Microsoft TechNet Library: Using Hyper-V and Failover Clustering

https://technet.microsoft.com/en-us/library/cc732181%28v=ws.10%29.aspx

Validate Cluster Configuration

Before creating a cluster, you must validate your configuration. Validation helps you to confirm the configuration of your servers, network, and to storage meets the specific requirements for failover clusters. Refer to the

Microsoft TechNet Library: Using Hyper-V and Failover Clustering for additional information.

Create a Cluster

To create a cluster, run the Create Cluster wizard. Refer to Microsoft Windows Server TechNet for information about creating a cluster and step-by-step instructions.

Create a failover cluster:

https://docs.microsoft.com/en-us/windows-server/failover-clustering/create-failover-cluster

Configure Cluster Quorum Settings

Quorum is the number of elements that need to be online to enable continuous running of a cluster. In most instances, the elements are nodes. In some cases, the elements also consist of disk or file share witnesses. Each of these elements determines whether the cluster should continue to run.

All elements, except the file share witnesses, have a copy of the cluster configuration. The cluster service ensures that the copies are always synchronized. The cluster should stop running if there are multiple failures or if there is a communication error between the cluster nodes.

After both nodes have been added to the cluster, and the cluster networking components have been configured, you must configure the failover cluster quorum.

The file share to be used for the node and File Share Majority quorum must be created and secured before configuring the failover cluster quorum. If the file share has not been created or correctly secured, the following procedure to configure a cluster quorum will fail. The file share can be hosted on any computer running a Windows operating system.

To configure the cluster quorum, you need to perform the following procedures:

- Create and secure a file share for the node and file share majority quorum
- Use the failover cluster management tool to configure a node and file share majority quorum

Refer to Microsoft Windows Server TechNet for information about configuring the cluster quorum:

https://technet.microsoft.com/en-us/library/cc731739.aspx

Validate the cluster quorum after you have configured it. For more information, refer to:

http://technet.microsoft.com/en-us/library/bb676379(EXCHG.80).aspx

Configure Storage

For a smaller virtualization environment, storage is one of the central considerations in implementing a good virtualization strategy. But with Hyper-V, VM storage is kept on a Windows file system. You can put VMs on any file system that a Hyper-V server can access. As a result, HA can be built into the virtualization platform and storage for the virtual machines. This configuration can accommodate a host failure by making storage accessible to all Hyper-V hosts so that any host can run VMs from the same path on the shared folder. The back-end part of this storage can be a local, storage area network, iSCSI, or whatever is available to fit the implementation.

For this architecture, local partitions are used.

The following table lists the minimum storage recommendations to configure storage for each VM:

System	Storage Capacity
Historian and Application Server (GR node) Virtual Machine	80 GB
Application Engine (Runtime node) Virtual Machine	40 GB

System	Storage Capacity
InTouch and Information Server Virtual Machine	40 GB

The total storage capacity should be minimum recommended 1 TB.

Configure Hyper-V

Microsoft® Hyper-V™ helps in creating a virtual environment that improves server utilization. It enhances patching, provisioning, management, support tools, processes, and skills. Microsoft Hyper-V provides live migration, cluster shared volume support, expanded processor, and memory support for host systems.

Hyper-V is available in x64-based versions of Windows Server 2012 operating systems and higher.

The following are the pre-requisites to set up Hyper-V:

- x64-based processor
- Hardware-assisted virtualization
- Hardware Data Execution Prevention (DEP)

Configure SIOS (SteelEye) DataKeeper and Hyper-V Replica

SIOS (SteelEye) DataKeeper and Hyper-V Replica are replication software for real-time Windows data. Both can be used to replicate all data types, including the following:

- Open files
- SQL and Exchange Server databases
- Hyper-V .vhd files

Hyper-V Replica is a built-in replication mechanism that was introduced in the Windows Server 2012 Hyper-V Role.

The ability of both SteelEye DataKeeper and Hyper-V Replica to replicate live Hyper-V virtual machines ensures that a duplicate copy is available in case the primary storage array fails. This helps in disaster recovery (DR) without impacting production.

SteelEye DataKeeper Cluster Edition is a host-based replication solution, which extends Microsoft Windows Server Failover Clustering (WSFC) and Microsoft Cluster Server (MSCS) features such as cross-subnet failover and tunable heartbeat parameters. These features make it possible to deploy geographically distributed clusters.

You can replicate a virtual machine across LAN, WAN, or any Windows server through SIOS Microsoft Management Console (MMC) interface. You can run the DataKeeper MMC snap-in from any server. The DataKeeper MMC snap-in interface is similar to the existing Microsoft Management tools.

Note: For information on installing the SteelEye DataKeeper, refer to http://www.sios.com. Ensure that the local security policies, firewall, and port settings are configured as per the details provided in the SteelEye DataKeeper documents. For information on using Hyper-V Replica, refer to the Hyper-V Replica Overview at https://technet.microsoft.com/en-us/library/jj134172.aspx

The following sections outline the workflow for using SteelEye DataKeeper.

Configure Virtual Machines

Create a SteelEye mirroring job and then create a virtual machine in the disk.

After creating the virtual machine, you need to add the dependency between the virtual machine and the datakeeper volume in the cluster.This dependency triggers the switching of the source and target Servers of the SteelEye DataKeeper Volume resource when failover of the virtual machines occurs in the Failover Cluster Manager.

You can create multiple virtual machines.

Configuration of System Platform Products in a Typical Small Scale Virtualization

To record the expected Recovery Time Objective (RTO) and Recovery Point Objective (RPO), trends and various observations in a small scale virtualization environment, tests are performed with System Platform Product configuration shown below.

The virtualization host server used for small scale configuration consists of three virtual machines listed below.

Node 1: GR, Historian and DAS SI Direct – Windows Server 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit

Node 2 (AppEngine): Bootstrap, IDE and InTouch (Managed App) – Windows 2012 Data Center edition (64bit)

Node 3:Bootstrap and IDE, InTouch RDS and Historian Client – Windows Server 2012 (32bit) with SQL Server 2012 SP2 and Microsoft Office

Virtual Node	IO tags (Approx.)	Historized tags (Approx.)
GR	10000	2500
AppEngine	10000	5000

Expected Recovery Time Objective and Recovery Point Objective

This section provides the indicative Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) for the load of IO and Attributes historized shown above and with the configuration of Host Virtualization Servers and Hyper-V virtual machines explained in the Setup instructions of Medium Scale Virtualization. For more information refer to "Set Up Medium Scale Virtualization Environment". In addition to these factors, the exact RTO and RPO depend on factors like storage I/O performance, CPU utilization, memory usage, and network usage at the time of failover/migration activity.

RTO and RPO Observations - DR Small Configuration

Important: The following sample data are provided only as guidelines for establishing testing specific to your needs.

Scenarios and observations in this section:

Scenario	Observation
Scenario 1: IT provides maintenance on Virtualization Server	"Live Migration"
	"Quick Migration"
	"Quick Migration of All Nodes Simultaneously"
Scenario 2: Virtualization Server hardware fails	"Scenario 2: Virtualization Server hardware fails"
Scenario 3: Network fails on Virtualization Server	"Scenario 3: Network fails on Virtualization server"
Scenario 4: Virtualization Server becomes unresponsive	"Scenario 4: Virtualization Server becomes unresponsive"

The following tables display RTO and RPO Observations with approximately 20000 IO points with approximately 7500 attributes being historized:

Scenario 1: IT provides maintenance on Virtualization Server

Live Migration

Primary Node	Products	RTO	RPO	
			Tags	Data Loss Duration
GR	IAS	14 sec	IAS tag (Script)	20 sec
			IAS IO tag (DASSiDirect)	26 sec
	Historian Client	19 sec	Historian Local tag	22 sec
			InTouch Tag $Second	27 sec
			IAS IO Tag (DASSiDirect)	32 sec
			IAS tag (Script)	0 (data is SFed)
	DAServer	21 sec	N/A	N/A

WIS	InTouch HMI	12 sec	$Second	12 sec
	Information Server	12 sec	N/A	N/A
	Historian Client	12 sec	N/A	N/A
AppEngine	AppEngine	12 sec	IAS IO tag (DASSiDirect)	26 sec
			IAS tag Script)	13 sec
	InTouch HMI	12 sec	$Second	12 sec

Quick Migration

Node Name	Products	RTO	RPO	
			Tags	Data Loss Duration
GR	IAS	147 sec	IAS tag (Script)	160 sec
			IAS IO Tag (DASSiDirect)	167 sec
	Historian Client	156 sec	Historian Local tag	164 sec
			InTouch tag $Second	171 sec
			IAS IO Tag (DASSiDirect)	170 sec
			IAS tag (Script)	0 (data is SFed)
	DAServer	156 sec	N/A	N/A
WIS	InTouch HMI	91 sec	$Second	91 sec
	Information Server	91 sec	N/A	N/A
	Historian Client	91 sec	N/A	N/A
AppEngine	AppEngine	59 sec	IAS IO tag (DASSiDirect)	80 sec
			IAS Tag (Script)	73 sec

	InTouch HMI	68 sec	$Second	68 sec

Quick Migration of All Nodes Simultaneously

Primary node	Products	RTO	RPO	
			Tags	Data Loss Duration
GR	IAS	221 sec	IAS tag (Script)	229 sec
			IAS IO tag (DASSiDirect)	234 sec
	Historian Client	225 sec	Historian Local tag	226 sec
			InTouch tag $Second	238 sec
			IAS IO tag (DASSiDirect)	242 sec
			IAS tag (Script)	160 sec
	DAServer	225 sec	N/A	
WIS	InTouch HMI	225 sec	$Second	255 sec
	Information Server	225 sec	N/AS	
	Historian Client	225 sec	N/A	
AppEngine	AppEngine	150 sec	IAS IO tag (DASSiDirect)	242 sec
			IAS tag (Script)	160 sec
	InTouch HMI	149 sec	$Second	149 sec

Scenario 2: Virtualization Server hardware fails

The Virtualization Server hardware failure results in failover that is simulated with power-off on the host server. In this case, the VMs restart, after moving to the other host server.

Primary node	Products	RTO	RPO	
			Tags	Data Loss Duration

GR	IAS	270 sec	IAS tag (Script)	5 Min 22 sec
			IAS IO tag (DASSiDirect)	5 Min 12 sec
	Historian Client	362 sec	Historian Local tag	6 Min 40 sec
			InTouch tag $Second	6 Min 58 sec **Note:** RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of he Historian node, which historizes this tag.
			IAS IO tag (DASSiDirect)	5 Min 16 sec
			IAS tag (Script)	4 Min 55 sec
	DAServer	196 sec	N/A	N/A

Primary Node	Products	RTO	RPO	
			Tags	Data Loss Duration
WIS	InTouch HMI	240 sec + time taken by the user to start the InTouchView	$Second	6 Min 58 sec **Note:** RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.

	Information Server	240 sec + time taken by the user to start the Information Server	N/A	N/A
	Historian Client	240 sec + time taken by the user to start the Historian Client	N/A	N/A
AppEngine	AppEngine	267 sec	IAS IO tag (DASSiDirect)	5 Min 16 sec
			IAS tag (Script)	4 Min 55 sec
	InTouch HMI	267 sec + time taken by the user to start the ITView	$Second	267 sec + time taken by the user to start the ITView
			Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	

Scenario 3: Network fails on Virtualization server

The failure of network on the Virtualization Server results in failover due to network disconnect (Public). Bandwidth used is 45Mbps and there is no latency. In this case, the VMs restart, after moving to the other host server.

Primary Node	Products	RTO	RPO	
			Tags	Data Loss Duration
GR	IAS	251 sec	IAS tag (Script)	4 Min 42 sec
			IAS IO tag (DASSiDirect)	4 Min 47 sec
	Historian Client	290 sec	Historian local tag	5 Min 11 sec

			InTouch tag $Second	5 Min 10 sec **Note:** RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.
			IAS IO tag (DASSiDirect)	4 Min 42 sec
			IAS tag (Script)	3 Min 58 sec
	DAServer	191 sec	N/A	N/A

Primary Node	Products	RTO	RPO	
			Tags	Data Loss Duration
WIS	InTouch HMI	215 sec + time taken by the user to start the InTouchView	$Second	5 Min 10 sec
			Note: RPO is dependent on time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node which historizes this tag	
	Information Server	215 sec + time taken by the user to start the Information Server	N/A	N/A
	Historian Client	215 sec + time taken by the user to start the Historian Client	N/A	N/A

AppEngine	AppEngine	209 sec	IAS IO Tag (DASSiDirect)	4 Min 42 sec
			IAS tag (Script)	3 Min 58 sec
	InTouch HMI	195 sec + time taken by the user to start the ITView	$Second	195 sec
			Note: RPO is dependent on time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node which historizes this tag.	

Scenario 4: Virtualization Server becomes unresponsive

There is no failover of VMs to the other host server when the CPU utilization on the host server is 100%.

Primary Node	Products	RTO	RPO	
			Tags	Data Loss Duration
GR	IAS	N/A	N/A	N/A
			N/A	N/A
	Historian Client	N/A	N/A	N/A
			N/A	N/A
			N/A	N/A
			N/A	N/A
	DAServer	N/A	N/A	N/A
WIS	InTouch HMI	N/A	N/A	N/A
	Information Server	N/A	N/A	N/A
	Historian Client	N/A	N/A	N/A
AppEngine	AppEngine	N/A	N/A	N/A

			N/A	N/A
	InTouch HMI	N/A	N/A	N/A

Medium Scale Virtualization Environments

This section contains the following topics:

- Set Up Medium Scale Virtualization Environment
- Configure System Platform Products in a Typical Medium Scale Virtualization
- Expected Recovery Time Objective and Recovery Point Objective

Set Up Medium Scale Virtualization Environment

The following procedures help you to set up small scale virtualization disaster recovery environment.

Plan for Disaster Recovery

The minimum and recommended hardware and software requirements for the Host and Virtual machines used for medium scale virtualization disaster recovery environment.

Hyper-V Hosts

Processor	Two 2.79 GHz Intel Xeon with 24 Cores
Operating System	Windows Server 2012 Data Center or higher with Hyper-V enabled
Memory	48 GB
Storage	SAN with 1TB storage disk

Note: For the Hyper-V Host to function optimally, the server should have the same processor, RAM, storage and service pack level. Preferably the servers should be purchased in pairs to avoid hardware discrepancies. Though the differences are supported, it will impact the performance during failovers.

Virtual Machines

Using the Hyper-V host specified above, six virtual machines can be created in the environment with the configuration given below.

Virtual Machine 1: Historian Node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	200 GB

System Platform Products Installed	Historian

Virtual Machine 2: Application Server Node, DAS SI

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	100 GB
System Platform Products Installed	ArchestrA-Runtime, DAS SI

Virtual Machine 3: InTouch Node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	InTouch HMI

Virtual Machine 4: Application Server Runtime Node 1

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Application Server Runtime only and InTouch

Virtual Machine 5: Application Server Runtime Node 2

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Application Server Runtime only

Virtual Machine 6: Historian Client Node

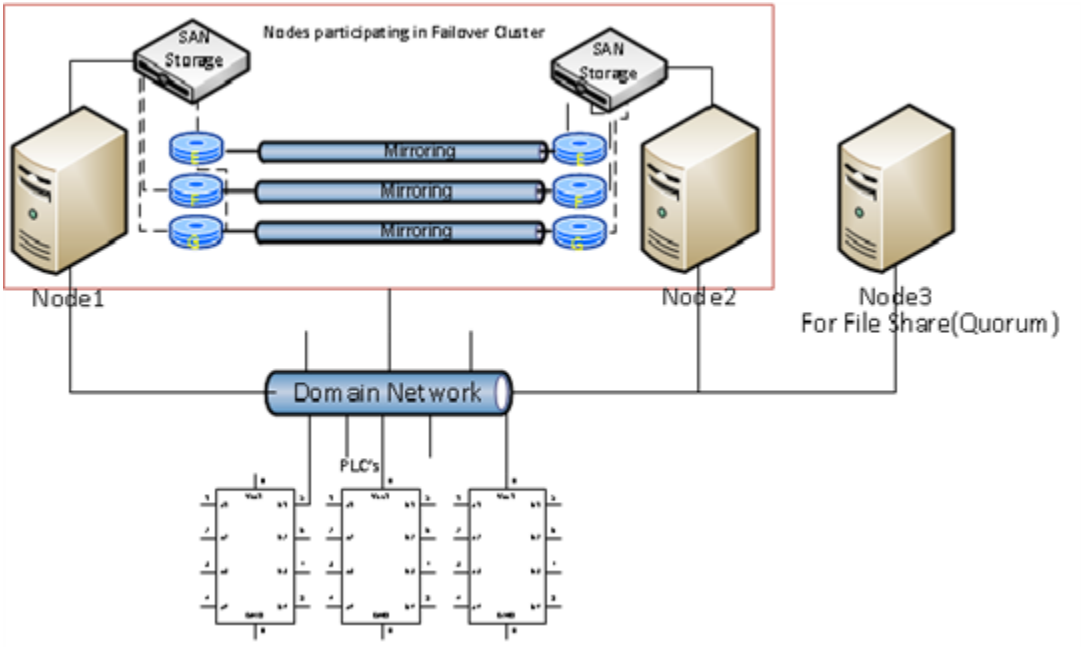
Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows 8.1 or higher Enterprise
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Historian Client

Network Requirements

For this architecture, you can use one physical network card that needs to be installed on a host computer for the domain network and the process network.

Configure Failover Cluster

The recommended topology of the failover cluster for disaster recovery process for medium scale virtualization environment is given below:



This setup requires a minimum of two host servers and two storage servers connected to each host independently. Another independent node is used for configuring the quorum. For more information on configuring the quorum, refer to "Configure Cluster Quorum Settings".

The workflow for installing and configuring a failover cluster with two nodes is outlined in the following section. This workflow is applicable to setting up a medium configuration.

Install Failover Cluster

To install the failover cluster feature, you need to run Windows Server 2012 or higher on your server. Refer to Microsoft's server documentation for information about installing the failover cluster feature and step-by-step instructions.

Microsoft TechNet Library: Using Hyper-V and Failover Clustering

https://technet.microsoft.com/en-us/library/cc732181%28v=ws.10%29.aspx

Validate Cluster Configuration

You must validate your configuration before you create a cluster. Validation helps you to confirm the configuration of your servers, network, and to storage meets the specific requirements for failover clusters. Refer to the Microsoft TechNet Library: Using Hyper-V and Failover Clustering for additional information.

Create a Cluster

To create a cluster, run the Create Cluster wizard. Refer to Microsoft Windows Server TechNet for information about creating a cluster and step-by-step instructions.

Create a failover cluster:

https://docs.microsoft.com/en-us/windows-server/failover-clustering/create-failover-cluster

Configure Cluster Quorum Settings

Quorum is the number of elements that need to be online to enable continuous running of a cluster. In most instances, the elements are nodes. In some cases, the elements also consist of disk or file share witnesses. Each of these elements determines whether the cluster should continue to run.

All elements, except the file share witnesses, have a copy of the cluster configuration. The cluster service ensures that the copies are always synchronized. The cluster should stop running if there are multiple failures or if there is a communication error between the cluster nodes.

After both nodes have been added to the cluster, and the cluster networking components have been configured, you must configure the failover cluster quorum.

You must create and secure the file share that you want to use for the node and the file share majority quorum before configuring the failover cluster quorum. If the file share has not been created or correctly secured, the following procedure to configure a cluster quorum will fail. The file share can be hosted on any computer running a Windows operating system.

To configure the cluster quorum, you need to perform the following procedures:

- Create and secure a file share for the node and file share majority quorum
- Use the failover cluster management tool to configure a node and file share majority quorum

Refer to Microsoft Windows Server TechNet for information about configuring the cluster quorum:

https://technet.microsoft.com/en-us/library/cc731739.aspx

Validate the cluster quorum after you have configured it. For more information, refer to:

http://technet.microsoft.com/en-us/library/bb676379(EXCHG.80).aspx

Create and secure a file share for the node and file share majority quorum

Create a new folder on the system that will host the share directory and allow sharing.

Then, use the failover cluster management tool to configure the node and file share majority quorum.

After you configure the cluster quorum, validate the cluster. For more information, refer to http://technet.microsoft.com/en-us/library/bb676379(EXCHG.80).aspx.

Configure Storage

For any virtualization environment, storage is one of the central barriers to implementing a good virtualization strategy. But with Hyper-V, VM storage is kept on a Windows file system. Users can put VMs on any file system that a Hyper-V server can access. As a result, you can build HA into the virtualization platform and storage for the virtual machines. This configuration can accommodate a host failure by making storage accessible to all Hyper-V hosts so that any host can run VMs from the same path on the shared folder. The back-end part of this storage

can be a local, storage area network, iSCSI or whatever is available to fit the implementation.

The following table lists the minimum storage recommendations for each VM:

System	Storage Capacity
Historian Virtual Machine	200 GB
Application Server (GR node) Virtual Machine	100 GB
Application Engine 1(Runtime node) Virtual Machine	80 GB
Application Engine 2 (Runtime node) Virtual Machine	80 GB
InTouch Virtual Machine	80 GB
Historian Client	80 GB

The total storage capacity should be minimum recommended 1 TB.

Configure Hyper-V

With Microsoft® Hyper-V™, you can create a virtual environment that improves server utilization. It enhances patching, provisioning, management, support tools, processes, and skills. Microsoft Hyper-V provides live migration, cluster shared volume support, expanded processor, and memory support for host systems. Refer to Microsoft Technet library for Hyper-V installation prerequisites and other considerations.

The following are the pre-requisites to set up Hyper-V:

- x64-based processor
- Hardware-assisted virtualization
- Hardware Data Execution Prevention (DEP)

Configuring SIOS (SteelEye) DataKeeper and Hyper-V Replica

SIOS (SteelEye) DataKeeper and Hyper-V Replica are replication software for real-time Windows data. Both can be used to replicate all data types, including the following:

- Open files
- SQL and Exchange Server databases
- Hyper-V .vhd files

Hyper-V Replica is a built-in replication mechanism that was introduced in the Windows Server 2012 Hyper-V Role.

The ability of both SteelEye DataKeeper and Hyper-V Replica to replicate live Hyper-V virtual machines ensures that a duplicate copy is available in case the primary storage array fails. This helps in disaster recovery (DR) without impacting production.

SteelEye DataKeeper Cluster Edition is a host-based replication solution, which extends Microsoft Windows Server Failover Clustering (WSFC) and Microsoft Cluster Server (MSCS) features such as cross-subnet failover and

tunable heartbeat parameters. These features make it possible to deploy geographically distributed clusters.

You can replicate a virtual machine across LAN, WAN, or any Windows server through SIOS Microsoft Management Console (MMC) interface. You can run the DataKeeper MMC snap-in from any server. The DataKeeper MMC snap-in interface is similar to the existing Microsoft Management tools.

Note: For information on installing the SteelEye DataKeeper, refer to http://www.steeleye.com. Ensure that the local security policies, firewall, and port settings are configured as per the details provided in the SteelEye DataKeeper documents. For information on using Hyper-V Replica, refer to the Hyper-V Replica Overview at https://technet.microsoft.com/en-us/library/jj134172.aspx

The following sections outline the workflow for using SteelEye DataKeeper.

You can replicate a virtual machine across LAN, WAN, or any Windows server through SIOS Microsoft Management Console (MMC) interface. You can run the DataKeeper MMC snap-in from any server. The DataKeeper MMC snap-in interface is similar to the existing Microsoft Management tools.

Note: For information on installing the SteelEye DataKeeper, refer to SteelEye DataKeeper Planning and Installation, and Administration Guides for Windows Server. Ensure that the local security policies, firewall, and port settings are configured as per the details in the SteelEye documentation.

The following procedures help you set up a virtual machine in the Disaster Recovery environment.

Configure Virtual Machines

Create a SteelEye mirroring job and then create a virtual machine in the disk.

After creating the virtual machine, you need to add the dependency between the virtual machine and the datakeeper volume in the cluster.This dependency triggers the switching of the source and target Servers of the SteelEye DataKeeper Volume resource when failover of the virtual machines occurs in the Failover Cluster Manager.

You can create multiple virtual machines.

Configure a Virtual Machine

After creating a DataKeeper mirroring job, you need to create a virtual Adding the Dependency between the Virtual Machine and the Disk in the Cluster

After creating the virtual machine, you need to add the dependency between the virtual machine and the datakeeper volume in the cluster.This dependency triggers the switching of the source and target Servers of the SteelEye DataKeeper Volume resource when failover of the virtual machines occurs in the Failover Cluster Manager.

Configure System Platform Products in a Typical Medium Scale Virtualization

The expected Recovery Time Objective (RTO) and Recovery Point Objective (RPO), trends and various observations in a medium scale virtualization environment are recorded by performing tests with System Platform Product configuration.

The virtualization host server used for medium scale configuration consists of seven virtual machines listed below.

Important: The following information is provided as an example of this kind of configuration, and is not intended to be used as instructions to set up and configure your environment.

- Node 1 (GR): GR, InTouch and DAS SI Direct – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit
- Node 2 (AppEngine1): Bootstrap, IDE and InTouch (Managed App) – Windows 2012 Data Center edition (64bit)
- Node 3 (AppEngine2): Bootstrap, IDE – Windows 2012 Data Center edition (64bit)
- Node 4: Historian – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit
- Node 5:Bootstrap and IDE – Windows Server 2012 Data Center edition (64bit) with SQL Server 2012 SP2 and Microsoft Office
- Node 6: InTouch HMI – Windows 2012 Data Center edition (64bit) with RDS enabled
- Node 7: Historian Client and InTouch – Windows 8.1 Professional Edition (64bit) with SQL Server 2012 SP2 32 bit

The following table displays the approximate data of virtual nodes, IO tags and historized tags in a medium scale virtualization environment:

Virtual Node	IO tags (Approx.)	Historized tags (Approx.)
AppEngine1	25000	10000
AppEngine2	25000	10000

Expected Recovery Time Objective and Recovery Point Objective

This section provides the indicative Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) for the load of IO and Attributes historized shown above and with the configuration of Host Virtualization Servers and Hyper-V virtual machines explained in the Setup instructions of Medium Scale Virtualization. For more information refer to "Set Up Medium Scale Virtualization Environment". In addition to these factors, the exact RTO and RPO depend on factors like storage I/O performance, CPU utilization, memory usage, and network usage at the time of failover/migration activity.

RTO and RPO Observations - DR Medium Configuration

Important: The following sample data are provided only as guidelines for establishing testing specific to your needs.

Scenario	Observation
Scenario 1: IT provides maintenance on Virtualization Server	"Live Migration"
	"Quick Migration of all nodes simultaneously"
	"Shut down of host server"
Scenario 2: Virtualization Server hardware fails	"Scenario 2: Virtualization Server hardware fails"

Scenario	Observation
Scenario 3: Network fails on Virtualization Server	"Scenario 3: Network fails on Virtualization Server"
Scenario 4: Virtualization Server becomes unresponsive	"Scenario 4: Virtualization Server becomes unresponsive"

The following tables display RTO and RPO Observations with approximately 50000 IO points with approximately 20000 attributes being historized:

Scenario 1: IT provides maintenance on Virtualization Server

Live Migration

Product	RTO	RPO	
		Tags	Data Loss Duration
InTouch HMI	9 sec	Data Loss for $Second tag (Imported to Historian)	1min 52 sec
GR	8 sec	IAS tag (Script)	13 sec
		IAS IO tag (DASSiDirect)	1 min 35 sec
AppEngine1	7 sec	IAS tag (Script)	15 sec
		IAS IO Tag (DASSiDirect)	1 min 13 sec
AppEngine2	13 sec	IAS tag (Script)	15 sec
		IAS IO tag (DASSiDirect)	1 min 14 sec
Historian Client	27 sec	SysTimeSec (Historian)	17 sec
		$Second (InTouch)	26 sec
		IAS tag (Script)	0 (data is SFed)
		IAS IO tag (DASSiDirect)	0 (data is SFed)
DAServer SIDirect	13 sec	N/A	N/A
Historian Client	12 sec	N/A	N/A

Information Server	9 sec	N/A	N/A
InTouch HMI	1 min 18 sec	Data Loss for $Second tag (Imported to Historian)	1min 23 sec
GR	1 min 55 sec	IAS tag (Script)	2 min 43 sec
		IAS IO tag (DASSiDirect)	2 min 55 sec
AppEngine1	3 min 25 sec	IAS Tag (Script)	3 min 40 sec
		IAS IO Tag (DASSiDirect)	3min 49 sec
AppEngine2	2 min 20 sec	IAS Tag (Script)	2 min 48 sec
		IAS IO tag (DASSiDirect)	2 min 54 sec
Historian Client	6 min 27 sec	SysTimeSec (Historian)	5 min 57 sec
		$Second (InTouch)	6 min 19 sec
		IAS tag (Script)	0 (data is SFed)
		IAS IO tag (DASSiDirect)	0 (data is SFed)
DAServer SIDirect	2min 1 sec	N/A	N/A
InTouch HMI	1 min 18 sec	Data Loss for $Second tag (Imported to Historian)	1min 23 sec

Product	RTO	RPO	
		Tags	Data Loss Duration
GR	1 min 55 sec	IAS tag (Script)	2 min 43 sec
		IAS IO tag (DASSiDirect)	2 min 55 sec

AppEngine1	3 min 25 sec	IAS Tag (Script)	3 min 40 sec
		IAS IO Tag (DASSiDirect)	3min 49 sec
AppEngine2	2 min 20 sec	IAS Tag (Script)	2 min 48 sec
		IAS IO tag (DASSiDirect)	2 min 54 sec
Historian Client	6 min 27 sec	SysTimeSec (Historian)	5 min 57 sec
		$Second (InTouch)	6 min 19 sec
		IAS tag (Script)	0 (data is SFed)
		IAS IO tag (DASSiDirect)	0 (data is SFed)
DAServer SIDirect	2min 1 sec	N/A	N/A

Quick Migration of all nodes simultaneously

Quick Migration of all nodes occurs simultaneously to migrate all nodes.

Product	RTO	RPO	
		Tags	Data Loss Duration
InTouch	3 min 29 sec	Data Loss for $Second tag (Imported to Historian)	12 min 8 sec
GR	6 min 11 sec	IAS tag (Script)	6 Min 35 sec
		IAS IO tag (DASSiDirect)	7 Min 26 sec
AppEngine1	8 min 12 sec	IAS tag (Script)	8 Min 6 sec
		IAS IO Tag (DASSiDirect)	8 Min 28 sec
AppEngine2	6min 6 sec	IAS tag (Script)	6 min 58 sec
		IAS IO tag (DASSiDirect)	7 min 34 sec

Historian	11 min 59 sec	SysTimeSec (Historian)	12 min 2 sec
		$Second (InTouch)	12 min 8 sec
		IAS tag (Script)	6 min 35 sec
		IAS IO tag (DASSiDirect)	7 min 26 sec
DAS SIDirect	6 min 48 sec	N/A	N/A
Historian Client	9 min 4 sec	N/A	N/A
Information Server	4 min 59 sec	N/A	N/A

Shut down of host server

Product	RTO	RPO	
		Tags	Data Loss Duration
InTouch	12 min 32 sec	Data Loss for $Second tag (Imported to Historian)	14 min
GR	11 min 41 sec	IAS tag (Script)	12 Min 58 sec
		IAS IO tag (DASSiDirect)	13 Min 11 sec
AppEngine1	11 min 38 sec	IAS tag (Script)	12 Min 6 sec
		IAS IO Tag (DASSiDirect)	13 Min 49 sec
AppEngine2	11 min 57 sec	IAS tag (Script)	12 Min 58 sec
		IAS IO tag (DASSiDirect)	13 Min 54 sec
Historian	12 Min 55 sec	SysTimeSec (Historian)	13 Min
		$Second (InTouch)	14 Min
		IAS tag (Script)	12 Min 58 sec

		IAS IO tag (DASSiDirect)	13 Min 11 sec
DAS SIDirect	6 Min 48 sec	N/A	N/A
Historian Client	9 Min 4 sec	N/A	N/A
Information Server	4 Min 59 sec	N/A	N/A

Scenario 2: Virtualization Server hardware fails

The failover occurs due to hardware failure, and it is simulated with power-off on the host server.

Product	RTO	RPO	
		Tags	Data Loss Duration
InTouch	11 Min 43 sec + time taken by the user to start the InTouchView	Data Loss for $Second tag (Imported to Historian)	12 Min 27 **Note:** RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.
GR	10 Min 51 sec	IAS tag (Script)	11 Min 16
		IAS IO tag (DASSiDirect)	11 Min 02
AppEngine1	10 min 29 sec	IAS tag (Script)	10 Min 40
		IAS IO Tag (DASSiDirect)	11 Min 16
AppEngine2	10 min 59 sec	IAS tag (Script)	9 Min 26
		IAS IO tag (DASSiDirect)	11 Min 08

Product	RTO	RPO	
		Tags	Data Loss Duration
Historian	14 Min 49 sec	SysTimeSec (Historian)	12 Min 21

		$Second (InTouch)	12 Min 27 **Note:** RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node which historizes this tag.
		IAS tag (Script)	11 Min 16
		IAS IO tag (DASSiDirect)	11 Min 02
DAS SIDirect	11 Min 20 sec	N/A	N/A
Historian Client	7 Min 16 sec + time taken by the user to start the Historian Client	N/A	N/A
Information Server	9 Min 39 sec + time taken by the user to start the Information Server	N/A	N/A
Historian Client	7 Min 16 sec + time taken by the user to start the Historian Client	N/A	N/A
Information Server	9 Min 39 sec + time taken by the user to start the Information Server	N/A	N/A

Scenario 3: Network fails on Virtualization Server

There is a failover due to network disconnect (Public). In this case, the VMs restart, after moving to the other host server.

Products	RTO	RPO	
		Tags	Data Loss Duration
InTouch	8 min 55 sec + time taken by the user to start the InTouchView	Data Loss for $Second tag (Imported to Historian)	14 min

		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
GR	11 min 32 sec	IAS Tag (Script)	12 min 01
		IAS IO Tag (DASSiDirect)	12 min
AppEngine1	10 min 52 sec	IAS Tag (Script)	11 min 26
		IAS IO Tag (DASSiDirect)	11 min 58
AppEngine2	10 min 28 sec	IAS Tag (Script)	10 min 19
		IAS IO Tag (DASSiDirect)	12 min 04

Products	RTO	RPO	
		Tags	Data Loss Duration
Historian	13 min 20 sec	SysTimeSec (Historian)	13 min 52
		$Second (InTouch)	14 min **Note:** RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.
		IAS Tag (Script)	12 min 01
		IAS IO Tag (DASSiDirect)	12 min
DAS SIDirect	9 min 9 sec	N/A	N/A
Historian Client	8 min + time taken by the user to start the Historian Client	N/A	N/A

Information Server	8 min 25 sec + time taken by the user to start the Information Server	N/A	N/A

Scenario 4: Virtualization Server becomes unresponsive

There is no failover of VMs to the other host server when the CPU utilization on the host server is 100%.

Primary Node	Products	RTO (sec)	RPO
InTouch	N/A	N/A	N/A
GR	N/A	N/A	N/A
	N/A	N/A	N/A
AppEngine1	N/A	N/A	N/A
	N/A	N/A	N/A
AppEngine2	N/A	N/A	N/A
	N/A	N/A	N/A
Historian	N/A	N/A	N/A
	N/A	N/A	N/A
	N/A	N/A	N/A
	N/A	N/A	N/A
DAS SIDirect	N/A	N/A	N/A
Historian Client	N/A	N/A	N/A
Information Server	N/A	N/A	N/A

Implementing Disaster Recovery Using vSphere

The workflows described below are designed to help you set up and implement Disaster Recovery using VMware vSphere. These workflows assume that VMware ESXi™ 5.0 or above, vCenter Server™, and vSphere Client already installed.

For basic procedures to install these and other VMware products, see product support and user documentation at http://www.vmware.com/.

The Disaster Recovery vSphere implementation assumes that you are implementing a medium-scale system.

This section contains the following topics:

Plan the Virtualization Environment

The recommended hardware and software requirements for the Host and Virtual machines used for the virtualization Disaster Recovery environment are as follows:

ESXi Hosts

Processor	Two 2.79 GHz Intel Xeon with 8 Cores (Hyper-threaded)
Operating System	ESXi 5.0 or above
Memory	48 GB
Storage	SAN with 1TB storage disk

Note: For the ESXi Host to function optimally, the server should have the same processor, RAM, storage, and service pack level. To avoid hardware discrepancies, the servers should preferably be purchased in pairs. Though differences are supported, it will impact the performance during failovers.

Virtual Machines

Using the specified ESXi host configuration, six virtual machines can be created in the environment with the following configuration.

Virtual Machine 1: Historian Node

Processor	Host Compatible Processor with 2-4 Cores
irtual CPUs	4 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	200 GB
System Platform Products Installed	Historian

Virtual Machine 2: Application Server Node, DAS SI

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	4 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB

Storage	100 GB
System Platform Products Installed	ArchestrA-Runtime, DAS SI

Virtual Machine 3: InTouch HMI Node

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	2 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	InTouch with RDS enabled

Virtual Machine 4: Application Server Runtime Node 1

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	2 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Application Server Runtime only and InTouch

Virtual Machine 5: Application Server Runtime Node 2

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	2 vCPUs
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Application Server Runtime only

Virtual Machine 6: Historian Client Node

Processor	Host Compatible Processor with 2-4 Cores
Virtual CPUs	1 vCPUs

Operating System	Windows 8.1 or higher Enterprise
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Historian Client

Network Requirements

For this architecture, you can use one physical network card that needs to be installed on a host computer for the domain network and the process network.

Configure System Platform Products in a Typical Virtualization Environment

The expected Recovery Time Objective (RTO) and Recovery Point Objective (RPO), trends, and various observations in a virtualization environment are recorded by performing tests with System Platform Product configuration.

The virtualization host server consists of the following seven virtual machines:

Important: The following information is provided as an example of this kind of configuration, and is not intended to be used as instructions to set up and configure your environment.

- Node 1 (GR): GR, InTouch and DAS SI Direct – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit
- Node 2 (AppEngine1): Bootstrap, IDE and InTouch (Managed App) – Windows 2012 Data Center edition (64bit)
- Node 3 (AppEngine2): Bootstrap, IDE – Windows 2012 Data Center edition (64bit)
- Node 4: Historian – Windows 2012 Data Center edition (64bit) with SQL Server 2012 SP2 32 bit
- Node 5: IBootstrap and IDE – Windows Server 2012 Data Center edition (64bit) with SQL Server 2012 SP2 and Microsoft Office
- Node 6: InTouch HMI – Windows 2012 Data Center edition (64bit) with RDS enabled
- Node 7: Historian Client and InTouch – Windows 8.1 Professional Edition (64bit) with SQL Server 2012 SP2 32 bit

The following table displays the approximate data of virtual nodes, IO tags and historized tags in the virtualization environment:

Virtual Node	IO tags (Approx.)	Historized tags (Approx.)
AppEngine1	25000	10000
AppEngine2	25000	10000

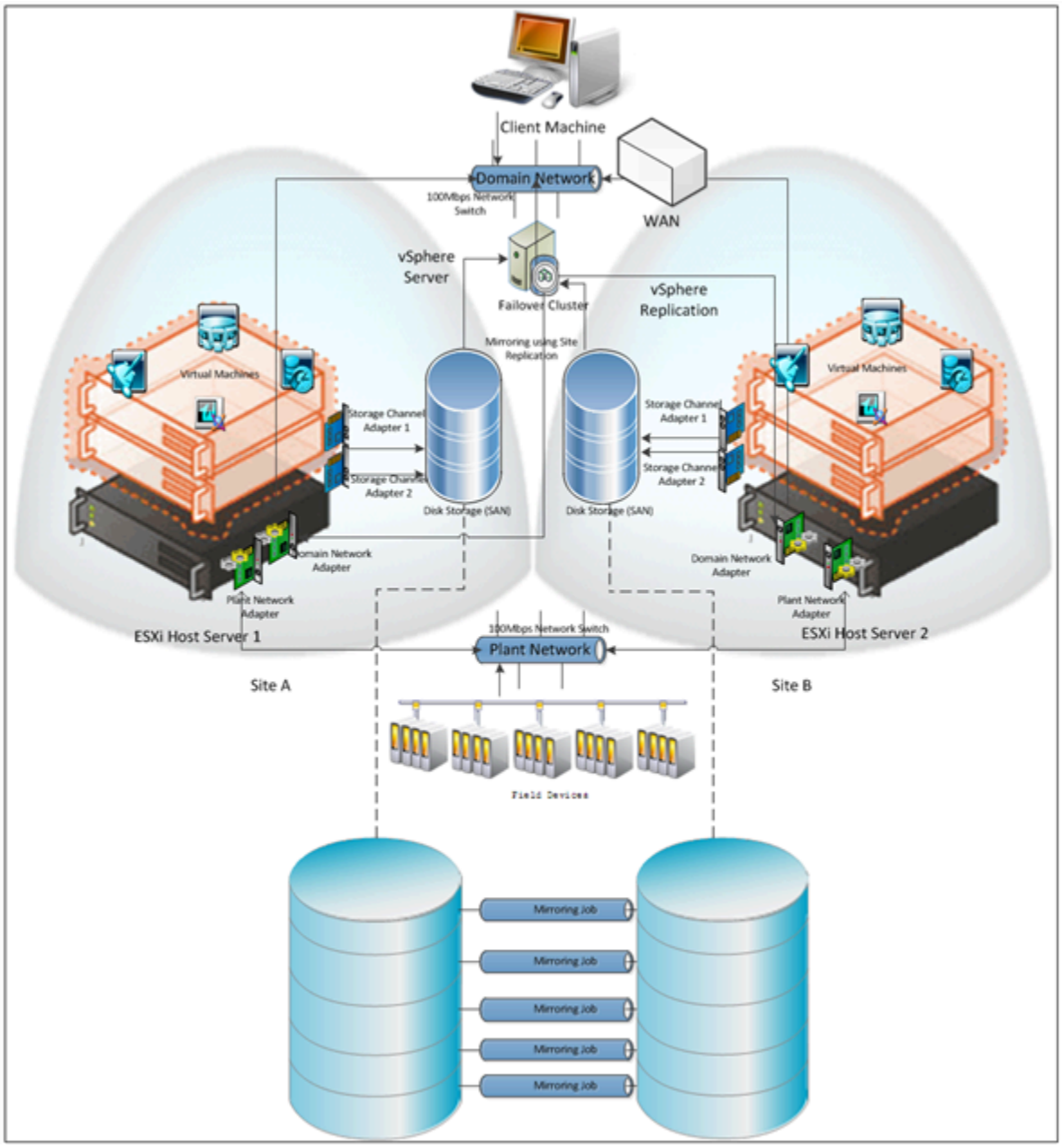
Set Up the Virtualization Environment

Use the following workflows to set up the virtualization environment for Disaster Recovery using vSphere

technology.

Create a Datacenter

The vSphere Datacenter virtualizes an infrastructure that includes servers, storage, networks, and provides for end-to-end connectivity from client machines to field devices and back. The recommended topology of the Datacenter for a Disaster Recovery is:



The following workflow requires a minimum of two host servers and two storage servers connected to each host independently. This workflow will help you configure a virtualized Disaster Recovery environment consisting of a Datacenter with a Failover Cluster that has two nodes and two Storage Area Networks (SANs).

Create the Datacenter

Use the vSphere Client to create the DataCenter. Refer to your vSphere documentation for additional information.

Add hosts to the Datacenter

Refer to your vSphere documentation or the VMware knowledge base for additional information and add an ESXi host.

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2032896

Repeat this procedure to add another ESXi host.

Create a Failover Cluster

A cluster in vSphere is a group of hosts. Resources of a host added to a cluster, also known as a failover cluster, become part of the cluster's resources, and are managed by the cluster.

Refer to your vSphere documentation or the VMware knowledge base for information about adding the failover cluster.

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2032896

To create a failover cluster:

1. Select the new cluster option.
2. Select vSphere HA.
3. Use the New Cluster Wizard to complete configuration.

Configure Storage

VMware Virtual Machine File System (VMFS) datastores serve as repositories for the virtual machines. You can set up VMFS datastores on any SCSI-based storage devices that the host discovers, including Fibre Channel, iSCSI, and local storage devices.

Use the following workflow to create a datastore. Your new datastore is added to all hosts if you use the vCenter Server system to manage your hosts.

Important: Install and configure any adapters that your storage requires before creating datastores. After you create a datastore, rescan the adapters to discover the new storage device.

Create a datastore

1. Log on to vSphere Client and select a host.
2. Configure storage for the host.
3. Configure the file system version.
4. Check the parameters you have selected and create the datastore.

Configure Networks

After you create a datacenter, use the following the workflow to configure one or more networks on the ESXi host.

Configure networks on the ESXi host

1. Log on to vSphere Client and select a host
2. Add networking through the Add Network Wizard.
3. Click **Add Networking.** The **Add Network Wizard** appears.
4. Use the Wizard to complete network configuration.

After you create a datacenter, add a host and configure storage. You can configure multiple networks on the ESXi host networks.

Create a Virtual Machine in the vSphere Client

You can populate your virtualization environment by creating virtual machines, which are the key components in a virtual infrastructure.

When you create a virtual machine, you associate it to a datastore and datacenter, host, cluster or resource pool. The virtual machine consumes resources dynamically as the workload increases, or it returns resources dynamically as the workload decreases.

Every virtual machine has virtual devices that provide the same function as physical hardware. A virtual machine gets CPU and memory, access to storage, and network connectivity from the host with which it is associated.

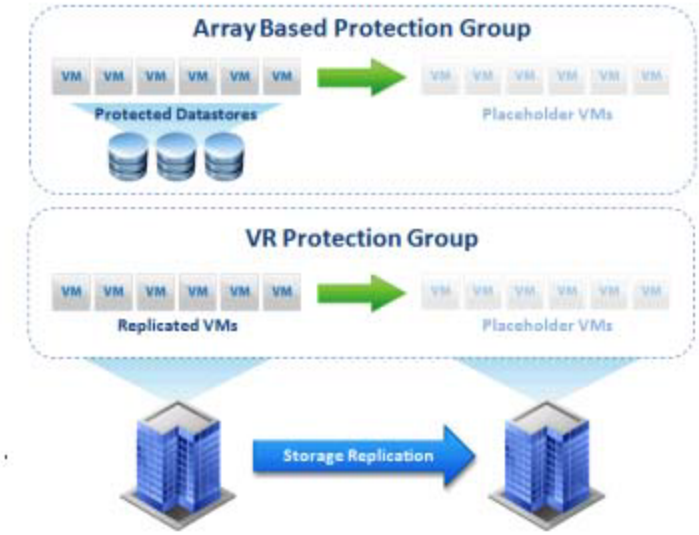
Set up Replication

Replicating live virtual machines ensures that a duplicate copy is available in case the primary storage array fails. This helps in Disaster Recovery without impacting production. Set up vSphere replication through the vSphere client.

Configure Protection Groups

Protection groups identify the virtual components that are considered to be most important for maintaining business continuity. Protection groups can define groups of associated VMs that should be recovered together, such as Infrastructure (Windows Active Directory or DNS), Mission Critical, or Business Critical.

Storage array-based protection groups include protected datastores. VR protection groups include replicated VMs. Recovery plans, detailed later in this chapter, are encapsulations of one or more protection groups stored at the recovery site to define the Disaster Recovery failover process.



Create a Recovery Plan

After creating the protection group, you must create the recovery plan for Disaster Recovery. Use vSphere Client to create the recovery plan.

Recover Virtual Machines to a Disaster Recovery Site

To recover the virtual machines in case of a disaster at a site, you must you must set up a disaster recovery site through vSphere Client.

Implementing High Availability and Disaster Recovery Using Virtualization

This section introduces several High Availability and Disaster Recovery (HADR) virtualization solutions that improve the availability of System Platform Products. A HADR solution offsets the effects of a hardware or software failure across multiple sites during a disaster. It makes sure all applications are available in order to minimize the downtime during times of crisis.

Important: The information and procedures in this chapter are specific to Hyper-V. You can implement a VMware HADR virtualization solution by following the procedures and settings in Implementing High Availability Using vSphereand in Implementing High Availability Using vSphere

Working with a Medium Scale Virtualization Environment

This section contains the following topics:

- Set Up the Virtualization Environment
- Expected Recovery Time Objective and Recovery Point Objective

Set Up the Virtualization Environment

The following procedures help you to set up and implement the high availability and disaster recovery for the medium scale virtualization environment.

Plan the Virtualization Environment

The minimum recommended hardware and software requirements for the Host and Virtual machines used for this virtualization environment are provided in the table below:

Hyper-V Hosts

Processor	Two 2.79 GHz Intel Xeon Processor with 24 Cores
Operating System	Windows Server 2012 Data Center or higher with Hyper-V enabled
Memory	48 GB
Storage	SAN with 1 TB storage disk

Note: For the Hyper-V Host to function optimally, the server should have the same processor, RAM, storage, and OS version (including service packs). Preferably, the servers should be purchased in pairs to avoid hardware discrepancies. Though the differences are supported, it impacts the performance during failovers.

Virtual Machines

Using the Hyper-V host specified above, six virtual machines can be created in the environment with the configuration given below.

Virtual Machine 1: Historian Node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	200 GB
System Platform Products Installed	Historian

Virtual Machine 2: Application Server Node and OI SI

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	8 GB
Storage	100 GB
System Platform Products Installed	ArchestrA-Runtime and DAS SI

Virtual Machine 3: InTouch Node

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	InTouch with RDS enabled

Virtual Machine 4: Application Server Runtime Node 1

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB
Storage	80 GB
System Platform Products Installed	InTouch and Application Server Runtime only

Virtual Machine 5: Application Server Runtime Node 2

Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows Server 2012 Data Center or higher
Memory	4 GB

Storage	80 GB
System Platform Products Installed	Application Server Runtime only

Virtual Machine 6: Historian Client Node

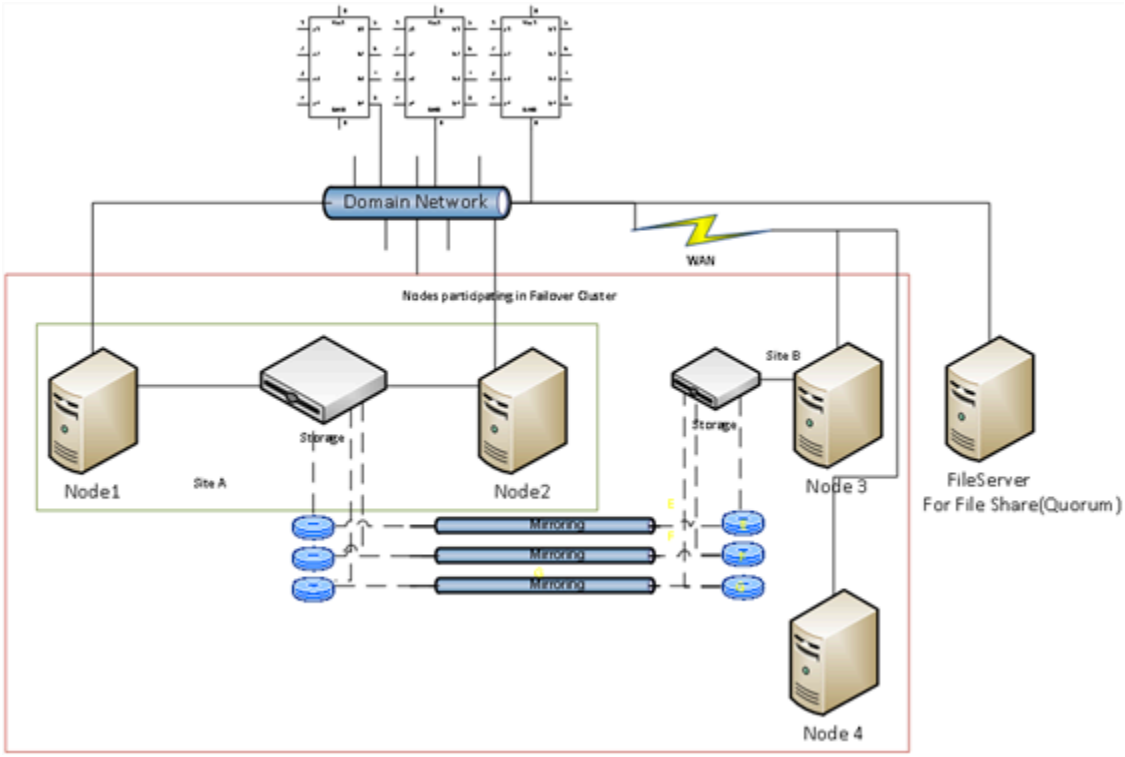
Processor	Host Compatible Processor with 2-4 Cores
Operating System	Windows 8.1 or higher Enterprise
Memory	4 GB
Storage	80 GB
System Platform Products Installed	Historian Client

Network Requirements

For this architecture, you can use one physical network card that needs to be installed on a host computer for both the domain and the process networks.

Configure a Failover Cluster

The following diagram shows the recommended topology of the failover cluster for high availability and disaster recovery for the virtualization environment:



The following workflow will guide you on how to setup high availability and disaster recovery for medium scale virtualization environment.

This setup requires a minimum of three host servers and two storage servers with sufficient disk space to host the virtual machines on each disk. One storage server is shared across two servers on one site and another

storage server is connected to the third host. Each disk created on the storage server is replicated in all the sites for disaster recovery. Node 4 is used for Node Majority in the failover cluster. Another independent node is used for configuring the quorum. For more information on configuring the quorum, refer to "Configure Cluster Quorum Settings".

Install Failover Cluster

To install the failover cluster feature, you need to run Windows Server 2012 or higher Data Center edition on your server.

To install failover cluster on a server

1. Go to Initial Configuration Tasks > Customize This Server > Add features.
2. Select Failover Clustering.

Validate Cluster Configuration

Before creating a cluster, you must validate your configuration. Validation helps you to confirm that the configuration of your servers, network, and storage meet the specific requirements for failover clusters. Use the Failover Cluster Manager to validate the configuration.

Create a Cluster

To create a cluster, run the Create Cluster wizard from the Server Manager.

To create a cluster

1. From the Failover Cluster Manager, open the Create Cluster Wizard.
2. Enter the server name to be added. Skip the validation test.
3. Enter the cluster name and close the wizard.

Configure Cluster Quorum Settings

Quorum is the number of elements that need to be online to enable continuous running of a cluster. In most instances, the elements are nodes. In some cases, the elements also consist of disk or file share witnesses. Each of these elements determines whether the cluster should continue to run or not.

All elements, except the file share witnesses, have a copy of the cluster configuration. The cluster service ensures that the copies are always synchronized. The cluster should stop running if there are multiple failures or if there is a communication error between the cluster nodes.

After both nodes have been added to the cluster, and the cluster networking components have been configured, you must configure the failover cluster quorum.

You must create and secure the file share that you want to use for the node and the file share majority quorum before configuring the failover cluster quorum. If the file share has not been created or correctly secured, the following procedure to configure a cluster quorum will fail. The file share can be hosted on any computer running a Windows operating system.

To configure the cluster quorum, you need to perform the following procedures:

- Create and secure a file share for the node and file share majority quorum
- Use the failover cluster management tool to configure a node and file share majority quorum

Validate the cluster quorum after you have configured it. For more information, refer to:

http://technet.microsoft.com/en-us/library/bb676379(EXCHG.80).aspx

Configure Storage

For any virtualization environment, storage is one of the central barriers to implementing a good virtualization strategy. However in Hyper-V, VM storage is kept on a Windows file system. You can put VMs on any file system that a Hyper-V server can access. As a result, you can build HA into the virtualization platform and storage for the virtual machines. This configuration can accommodate a host failure by making storage accessible to all Hyper-V hosts so that any host can run VMs from the same path on the shared folder. The back-end of this storage can be a local, storage area network, iSCSI or whatever is available to fit the implementation.

The following table lists the minimum storage recommendations for each VM in medium scale virtualization environment:

System	Storage Capacity
Historian Virtual Machine	200 GB
Application Server 1 (GR Node) Virtual Machine	100 GB
Application Engine 2 (Runtime Node) Virtual Machine	80 GB
InTouch Virtual Machine	80 GB
Information Server Virtual Machine	80 GB
Historian Client	80 GB

To build up High Availability and Disaster Recovery system, you must have a minimum of two SAN storage servers, each installed at different sites with the above storage recommendations.

The total storage capacity should be minimum recommended 1 TB.

Configure Hyper-V

Microsoft Hyper-V helps in creating virtual environment that improves server utilization. It enhances patching, provisioning, management, support tools, processes, and skills. Microsoft Hyper-V provides live migration, cluster shared volume support, expanded processor, and memory support for host systems.

Hyper-V is available in x64-based versions of Windows Server 2012 Data Center edition and higher operating system.

The following are the pre-requisites to set up Hyper-V:

- x64-based processor
- Hardware-assisted virtualization
- Hardware Data Execution Prevention (DEP)

Configure SIOS (SteelEye) DataKeeper and Hyper-V Replica

SteelEye DataKeeper and Hyper-V Replica are replication software for real-time Windows data. Both can be used to replicate all data types, including the following:

- Open files
- SQL and Exchange Server databases

- Hyper-V .vhd files

The ability of both SteelEye DataKeeper and Hyper-V Replica to replicate logical disk partitions hosting the .vhd files for the Hyper-V virtual machines ensures that a mirrored disk image is available on the stand-by cluster host in case the primary cluster host fails. This helps provide disaster recovery (DR) solutions.

SteelEye DataKeeper Cluster Edition is a host-based replication solution, which extends Microsoft Windows Server Failover Clustering (WSFC) and Microsoft Cluster Server (MSCS) features such as cross-subnet failover and tunable heartbeat parameters. These features make it possible to deploy geographically distributed clusters.

You can replicate .vhd files across LAN, WAN, or any Windows server through SIOS Microsoft Management Console (MMC) interface. You can run the DataKeeper MMC snap-in from any server. The DataKeeper MMC snap-in interface is similar to the existing Microsoft Management tools.

Note: For information on installing the SteelEye DataKeeper, refer to http://www.steeleye.com. Ensure that the local security policies, firewall, and port settings are configured as per the details provided in the SteelEye DataKeeper documents. For information on using Hyper-V Replica, refer to the Hyper-V Replica Overview at https://technet.microsoft.com/en-us/library/jj134172.aspx

The following workflow outlines how to set up a virtual machine in the Disaster Recovery environment, using SteelEye DataKeeper.

1. Create a SteelEye DataKeeper Mirroring Job.
2. Enter the relevant job name and description
3. Choose a source and a target.
4. Select the destination server, IP address and volume.
5. After you have completed setting up SteelEye DataKeeper Mirroring jobs and created the datakeeper, you can view the disk management topologies.

Configure Virtual Machines

After creating a steel eye mirroring job, you need to create a virtual machine in the disk.

Use the New Virtual Machine Wizard to create and configure a new virtual machine.

Add the Dependency between the Virtual Machine and the DataKeeper volume in the Cluster

After creating the virtual machine, you need to add the dependency between the virtual machine and the datakeeper volume in the cluster. This dependency triggers the switching of the source and target servers of the SteelEye DataKeeper Volume resource when failover of the virtual machines occurs in the Failover Cluster Manager.

Expected Recovery Time Objective and Recovery Point Objective

This section provides the indicative Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) for the load of IO and Attributes historized as shown in Configuring System Platform Products in a Typical Medium Scale Virtualization in Chapter 3 and with the configuration of Host Virtualization Servers and Hyper-V virtual machines explained in the setting up Medium Scale Virtualization Environment. For more information refer to, Set Up the Virtualization Environment. In addition to these factors, the exact RTO and RPO depend on factors like storage I/O performance, CPU utilization, memory usage, and network usage at the time of failover/migration activity.

RTO and RPO Observations - HADR Medium Configuration

Important: The following sample data are provided only as guidelines for establishing testing specific to your needs.

Scenarios and observations in this section:

Scenario	Observation
HA-Scenario: Virtualization Server hardware fails	"HA-Scenario: Virtualization Server hardware fails"
DR-Scenario: Network fails on Virtualization Server	"DR-Scenario: Network fails on Virtualization Server"

The following tables display RTO and RPO Observations with approximately 50000 IO points with approximately 20000 attributes being historized:

HA-Scenario: Virtualization Server hardware fails

The failover occurs due to hardware failure, and it is simulated with power-off on the host server.

Products	RTO	RPO	
		Tags	Data Loss Duration
InTouch	5 min 35 sec + time taken by the user to start the InTouchView	Data Loss for $Second tag (Imported to Historian)	6 min 47 sec
		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
GR	5 min 13 sec	IAS Tag (Script)	5 min 44 sec
		IAS IO Tag (DASSiDirect)	7 min 28 sec
AppEngine1	6 min 05 sec	IAS Tag (Script)	6 min 35 sec
		IAS IO Tag (DASSiDirect)	7 min 29 sec
AppEngine2	6 Min 12 sec	IAS Tag (Script)	6 Min 41 sec
		IAS IO Tag (DASSiDirect)	7 Min 20 sec

Products	RTO	RPO	
		Tags	Data Loss Duration
Historian	6 min 21 sec	SysTimeSec (Historian)	6 Min 33 sec
		$Second (InTouch)	6 Min 47 sec
		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
		IAS Tag (Script)	5 Min 45 sec
		IAS IO Tag (DASSiDirect)	7 Min 30 sec
DAS SIDirect	4 Min 25 sec	N/A	N/A
Historian Client	3 Min 34 sec + time taken by the user to start the Historian Client	N/A	N/A
Information Server	4 Min 15 sec + time taken by the user to start the Information Server	N/A	N/A

DR-Scenario: Network fails on Virtualization Server

There is a failover due to network disconnect (Public). In this case, the VMs restart after moving to the other host server.

Products	RTO	RPO	
		Tags	Data Loss Duration
InTouch	11 min 4 sec + time taken by the user to start the InTouchView	Data Loss for $Second tag (Imported to Historian)	15 min 32 sec

		Note: RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.	
GR	12 min 20 sec	IAS Tag (Script)	13 min 11 sec
		IAS IO Tag (DASSiDirect)	13 min 01 sec
AppEngine1	11 min 35 sec	IAS Tag (Script)	12 min 26 sec
		IAS IO Tag (DASSiDirect)	13 min 05 sec
AppEngine2	11 min 48 sec	IAS Tag (Script)	11 min 24 sec
		IAS IO Tag (DASSiDirect)	13 min 19 sec
Historian	20 min 0 sec	SysTimeSec (Historian)	15 min 16 sec
		$Second (InTouch)	15 min 32 sec RPO is dependent on the time taken by the user to start the InTouchView on the InTouch node and the RTO of the Historian node, which historizes this tag.
		IAS Tag (Script)	13 min 11 sec
		IAS IO Tag (DASSiDirect)	13 min 01 sec
DAS SIDirect	12 min 25 sec	N/A	N/A
Historian Client	5 min 32 sec + time taken by the user to start the Historian Client	N/A	N/A
Information Server	5 min 38 sec + time taken by the user to start the Information Server	N/A	N/A

Working with Windows Server

This chapter provides an overview of Windows Server features as they relate to following functions:

- Communication Between System Platform Nodes with VLAN
- RMC Communication Between Redundant Application Server Nodes with VLAN
- Access a System Platform Node with a Remote Desktop
- Access System Platform Applications as Remote Applications
- Display the System Platform Nodes on a Multi-Monitor with a Remote Desktop
- Network Load Balancing
- Hardware Licenses in a Virtualized Environment

About Microsoft Hyper-V

Hyper-V is available both as a standaone product and as part of Windows Server 2012 and higher. Choosing which one to use will generally be an issue of licensing costs. While the standalone Hyper-V is available as a free download, you will have to license the virtual machines that will run on top of the hypervisor. If you have Windows Server Datacenter, you can run unlimited numbers of VMs without having to pay for additional licenses.

A virtualized environment can run multiple virtual machines (VMs) on a single server, thereby reducing the number of physical servers required on the network. Hyper-V provides a virtualized computing environment on Windows Server. Hyper-V is a hardware-assisted virtualization platform that uses partitions to host VMs. One of the benefits that Hyper-V provides is isolation, which ensures that the child VMs execute in their individual partitions and exist on the host as separate machines. This allows multiple operating systems and conflicting applications to run on the same server.

Hyper-V provides support for using Virtual LANs (VLANs) on both parent and child partitions. By configuring VLAN, VMs can communicate over the specified VLAN using Virtual Network switch.

RemoteApp can be used to start an application seamlessly from an RDS session and make it appear as it were running locally on the client machine.

Communication Between System Platform Nodes with VLAN

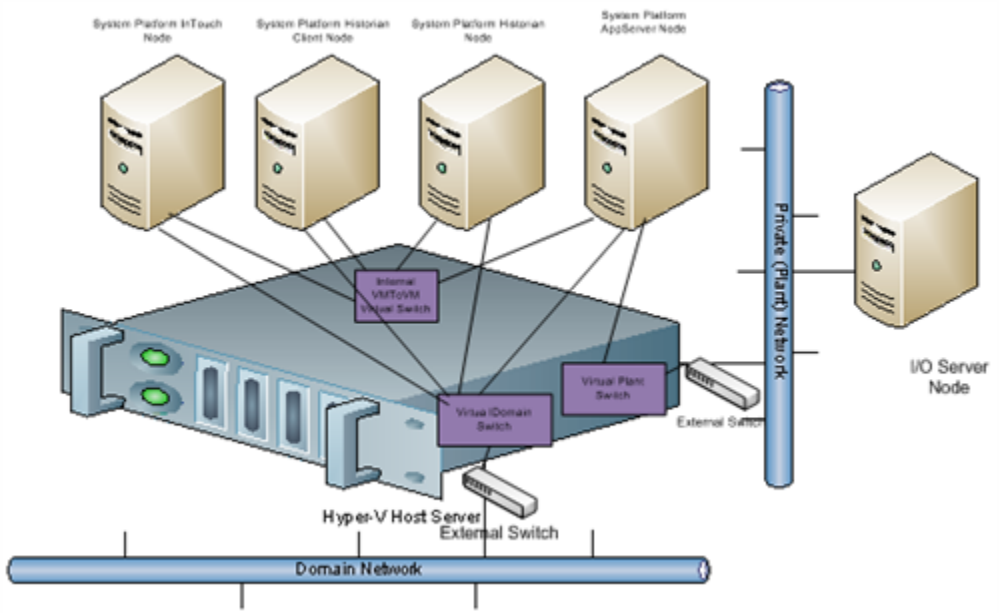
Virtual LANs perform traffic separation within a shared network environment. All released versions of Hyper-V support virtual local area networks (VLANs). Since the VLAN configuration is software-based, you can move a computer and still maintain the network configurations. For each virtual network adapter you connect to a virtual machine, you can configure a VLAN ID for the virtual machine.

You need the following network adapters to configure VLANs:

- A physical network adapter that supports VLANs
- A physical network adapter that supports network packets with VLAN IDs that are already applied

On the management operating system, you need to configure the virtual network to allow network traffic on the physical port. This enables you to use the VLAN IDs internally with the VMs. You can then configure the VM to

specify the virtual LAN that the VM will use for all network communications.



Configure Virtual Network Switches on the Hyper-V Host Server and Add Virtual Network Adapters on the VM Nodes

You can create virtual networks on a server running Hyper-V to define various networking topologies for VMs and the virtualization server. Following are the three types of virtual networks:

- Private network: Provides communication between VMs
- Internal network: Provides communication between the virtualization server and VMs
- External network: Provides communication between a VM and a physical network by associating to a physical network adapter on the virtualization server

On a Hyper-V host server, you can create the following virtual network adapter switches.

- External Network adapter switch to communicate with the external domain network.
- External Network adapter switch to communicate with the external plant network.
- Internal Network adapter switch to communicate between VM nodes created on Hyper-V host server.

For more information, refer to http://technet.microsoft.com/en-us/library/cc732470(WS.10).aspx

Create a Virtual Network Switch for Communication Between a VM Node and an External Domain or a Plant Network

A virtual network switch or a virtual switch is a virtual version of a physical network switch. A virtual network provides access to local or external network resources for one or more VMs. You need to create a virtual network switch to communicate with the external domain or plant network.

Note: A virtual network works like a physical network except that the switch is software based. After an external

virtual network is configured, all networking traffic is routed through the virtual switch.

Use the Hyper-V Manager on a Hyper-V host to create a virtual network switch for communication between a VM node and an external domain network or a plant network. Then, use the Virtual Network Manager to add a new external virtual network.

To create a virtual network switch for communication between a VM node and an external domain network or a plant network

1. Open the Hyper-V Manager on a Hyper-V host.
2. Go to the Virtual Network Manager.
3. Add a new external virtual network.
4. Enter the new virtual network details, including the network name, and the external domain or plant network to which you are connecting,

This creates the external network switch which will be used to communicate between the VM nodes and the domain or plant network.

Create a Virtual Network Switch for Communication Between Internal VM Nodes

To communicate with the other VMs hosted on the Hyper-V host server, you need to create an internal virtual network switch.

Use the Hyper-V Manager add a new internal virtual network. The internal virtual network switch is used to communicate between the VM nodes on the host server.

To create a virtual network switch for communication between internal VM nodes

1. Open the Hyper-V Manager on a Hyper-V host.
2. Go to the Virtual Network Manager.
3. Add a new internal virtual network.
4. Enter the new virtual network name. Be sure to specify that this is an internal network.

This creates the internal network switch which will be used to communicate between the VM nodes and the host server.

Add an Internal Virtual Network Adapter to a VM Node for Communication Between VM Nodes

You can configure one or more virtual network adapters for a VM by creating or modifying the hardware profile of a VM.

If you connect a virtual network adapter configured for a VM to an internal network, you can connect to the VMs deployed on the same host and communicate over that internal network.

Use the Hyper-V Manager add and enable a new internal virtual network adapter for communication between VM nodes. All traffic for the management operating system that goes through the network adapter is tagged with the VLAN ID you provide.

To add an internal virtual network adapter to a VM node for communication between VM nodes

1. Open the Hyper-V Manager on a Hyper-V host.

2. Shut down the VM node to which you want to add the network adapter.
3. Select the hardware settings for the VM node. Add a network adapter and enter virtual LAN ID.

   Note: All traffic for the management operating system that goes through the network adapter is tagged with the VLAN ID you enter.

Add a Virtual Network Adapter to a VM Node for Communication Between a VM Node and a Plant Network

If you connect a virtual network adapter configured for a VM to a physical network adapter on the host on which the VM is deployed, the VM can access the network to which the physical host computer is connected and can function on the host's local area network (LAN) in the same way that physical computers connected to the LAN can function.

Use the Hyper-V Manager add a virtual network adapter for communication between a VM node and a plant network.

To add a virtual network adapter to a VM node for communication between a VM node and a plant network

1. Open the Hyper-V Manager on a Hyper-V host.
2. Shut down the VM node to which you want to add the network adapter.
3. Select the hardware settings for the VM node and add a network adapter.

Configure Network Adapters on the System Platform Virtual Machine (VM) Nodes

By default, one network adapter is added to the VM node when you create the VM nodes on a Hyper-V host server.

Based on the requirements, you can add multiple internal or external network adapters.

For the VM System Platform node to communicate with the external domain or external plant network, it needs to have external network adapter added.

For the VM System Platform node to communicate internally to the other VM System Platform nodes hosted by the Hyper -V server, it needs to have internal network adapter added.

You can create the following VM nodes on the virtualization server for which the VLAN communication needs to be set up:

- InTouch VM node
- Historian VM node
- Application Server VM node
- Historian Client VM node
- Information Server VM node

VM nodes on Hyper-V host server have the following network adapters:

- An external network adapter to communicate with the external domain network
- An external network adapter to communicate with the external plant network. This is available if the VM node is acquiring the data from the IOServer connected to the external plant network

- An internal network adapter to communicate internally between the VM nodes configured on Hyper-V host server
- An internal network adapter to communicate between the Application Server nodes to use for Redundancy Message Channel (RMC) communication. Only the Application Server VM nodes configured for Redundant Application Engines have this network adapter.

Each System Platform node can have various combinations of the following network adapters, depending on your configuration:

Note: It is assumed that the host virtualization server is configured with one external virtual network switch to communicate with the domain network, one external virtual network switch to communicate with the plant network, and one internal virtual network switch for the internal VM to VM communication.

Product node	Network adapters
InTouch	• An external network adapter to communicate with the external domain network • An external network adapter to communicate with the external plant network (This is to acquire the data from the IOServer which is connected to the plant network.) • An internal network adapter to communicate between the other VM nodes configured on a Hyper-V host server (For example, to a Historian VM node)
Historian	• An external network adapter to communicate with the external domain network • An external network adapter to communicate with the external plant network (This is to acquire the data from the IOServer which is connected to the plant network.) • An internal network adapter to communicate between the other VM nodes configured on a Hyper-V host server (for example, an InTouch VM node).

Product node	Network adapters
Historian Client	• An external network adapter to communicate with the external domain network. • An external network adapter to communicate with the external plant network. This is to acquire the data from the IO Server which is connected to the plant network. • An internal network adapter to communicate between the other VM nodes configured on a Hyper-V host server (for example, a Historian VM node).
Information Server	• An external network adapter to communicate with the external domain network. • An internal network adapter to communicate between the other VM nodes configured on a Hyper-V host server (for example, to a Historian Client VM node).
Application Server	• An external network adapter to communicate with the external domain network. • An external network adapter to communicate with the external plant network. This is to acquire the data from the IO Server which is connected to the plant network. • An internal network adapter to communicate between the other VM nodes configured on a Hyper-V host server (for example, a Historian VM node).

You will need to create VM nodes with the specified OS installed on all the nodes. Configure one physical machine in the workgroup with an IO Server installed and connected to a plant or private network. Add one internal and one external virtual network adapter to the VM node. Use the same VLAN ID that you used for Configure the required VM node. Repeat for each VN node you are configuring. The general workflow is as follows:

To configure virtual network adapters on VM node

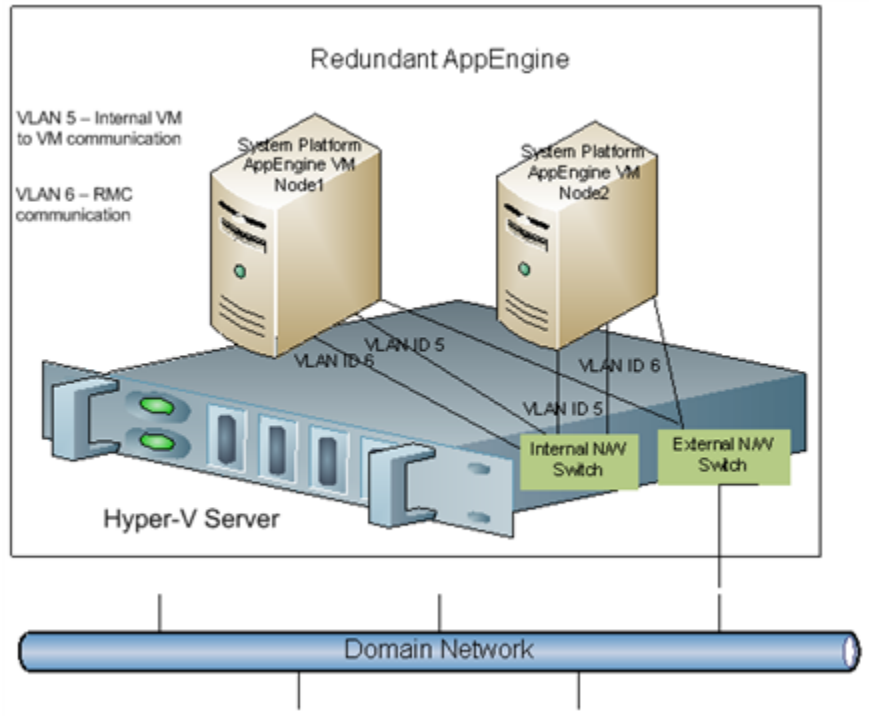
1. Add an internal virtual network adapter to the required node, for example, an InTouch node.

   Note: You must provide the same VLAN ID that you provided for the first VM node you configured.
2. Add an external virtual network adapter to the required node, for example an InTouch node.
3. Connect to the required VM node.
4. Configure the required VM node. Select IPv4.
5. Enter the IP address for the network adapter.
   - For the internal network added for communication between VM nodes, enter the required IP address.
   - For external network adapter added for communication between a VM node and an external plant network communication, enter the required static IP address.

     Note: Configure the other VM nodes following the same steps.

RMC Communication Between Redundant Application Server Nodes with VLAN

For successful communication between a redundant pair of Application Engines, each Application Engine must be assigned to a separate WinPlatform and a valid redundancy message channel (RMC) must be configured for each WinPlatform. You can configure an RMC using a virtual LAN.



Configure RMC for Redundant AppEngine over a VLAN

For a successful communication between a redundant pair of Application Engines, each Application Engine should configure a valid redundancy message channel (RMC). You can configure the RMC using Virtual LAN

(VLAN). For configuring the RMC, Application Server VM System Platform node requires the internal network adapters for communication:

- An internal network adapter to communicate between the other VM nodes configured on a Hyper-V host server, for example, a Historian VM node
- An internal network adapter to communicate with the other Application Server VM nodes configured as Redundancy Application Engine to use as a RMC

To configure RMC for a Redundant AppEngine node, you will need to add an internal virtual network adapter to a Application Server node. Use the same VLAN ID for both Application Server nodes. This allows the Application Server VM nodes to internally communicate with each other over the specified LAN ID as RMC channel. The general workflow is as follows:

Note: While installing the AVEVA products, select the **Create Local Account** check box and provide the same user name and password to use as network account user.

To configure RMC for a Redundant AppEngine node

1. Add an internal virtual network adapter to a Application Server node.

   Note: In the Settings window, enter the same VLAN ID that you entered while configuring the InTouch and Historian Client nodes. This enables the VM nodes to communicate internally over the specified LAN ID.
2. Add an internal virtual network adapter to a Application Server node to use as RMC communication.

   Note: In the Settings window, enter the same VLAN ID you entered on both the Application Server nodes for virtual network adapter. This enables the Application Server VM node to communicate internally over the specified LAN ID as an RMC channel to communicate to another Application Server VM node.
3. Add an external virtual network adapter to a Application Server node.
4. Connect to the required Application Server VM node.
5. Configure the internal/external network adapter for the node. Be sure to select IPv4, and enter the IP address. For the internal network adapter added to use as RMC, enter the required static IP address in the IP address box and subnet mask in the Subnet mask box.

   For example:

10.0.0.1

255.0.0.0

6. Follow the same steps to configure another Application Server node for Redundant Application Server.

Note: Note: While installing the AVEVA products, select the Create Local Account check box and provide the same user name and password to use as network account user.

Access a System Platform Node with a Remote Desktop

You can use Hyper-V to access a system platform node through a remote desktop. You can specify the required remote users, who will be able to access the VM running the system platform.

To access a system platform node with a remote desktop, log on to the system platform node as a member of the local administrators group. Then, modify the remote settings of the system platform node to specify the remote desktop versions to which you want to allow access, and select the user to whom you want to provide access.

To access a system platform node with a remote desktop

1. Log on to the system platform node as a member of the local administrators group.
2. Modify the remote settings of the system platform node. Specify the remote desktop option you want to use.
3. Add users to allow them to access the system.

Access System Platform Applications as Remote Applications

Remote Desktop Services (RDS) Remote Applications enables you to deploy RemoteApp programs to users. With RemoteApp, the remote session connects with a specific application rather than with the entire desktop. You can access the RemoteApp programs remotely through Remote Desktop Service. A RemoteApp program appears as if it is running on your local computer. Instead of being present on the desktop of the remote terminal server, the RemoteApp program is integrated with the client's desktop, running in its own resizable window with its own entry in the task bar.

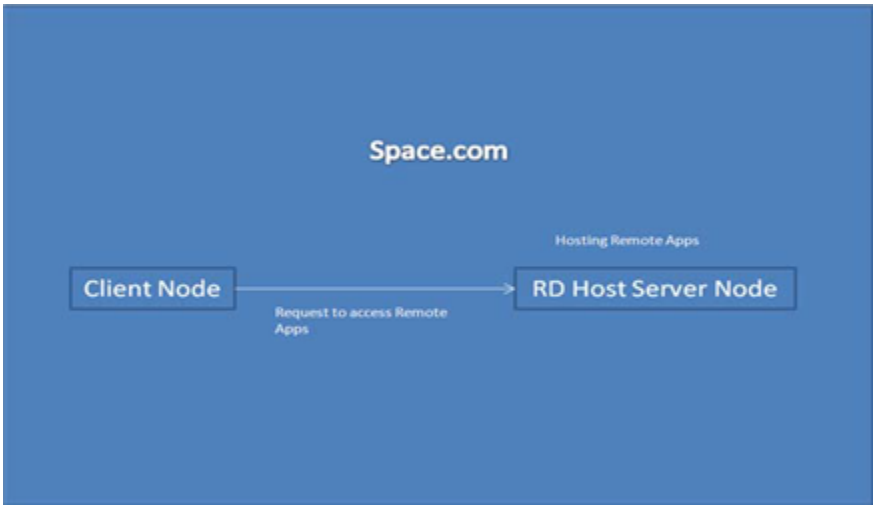
Prerequisites for accessing Remote Applications

- A virtual machine node or physical node with Windows Server 2012 which has Remote Desktop Session Host server installed.
- Remote Applications, part of the Windows Server RDS role.
- VM nodes (Remote Desktop Session Host server) running IOM Products, such as InTouch and Historian Client need to be on Windows Server 2012 or higher where Remote Desktop Services are available.
- Client node with a browser (any operating system)

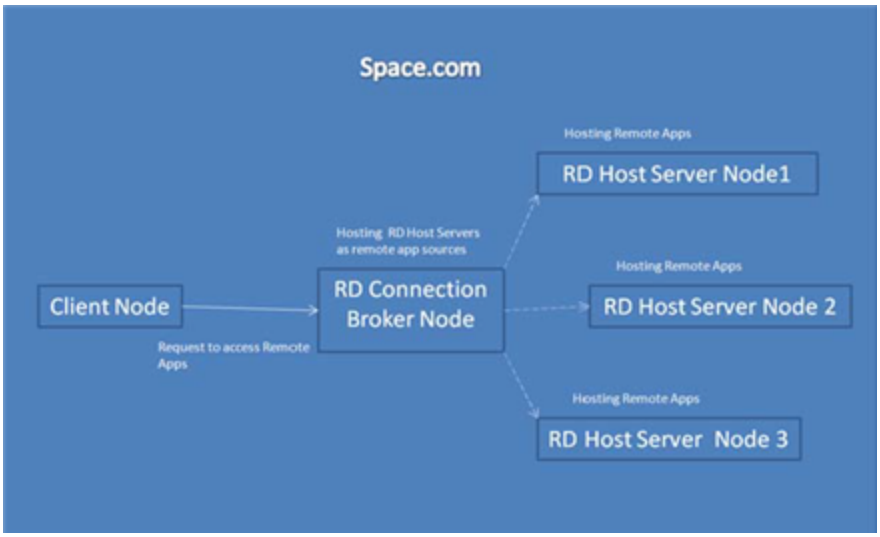
  Note: To access RemoteApp programs through Remote Desktop-Web Access, the client computer must have Remote Desktop Connection enabled.

- The client node and the Remote Desktop Session Host server should be able to communicate.

The following figure illustrates how RemoteApps configured at Remote Desktop Host Server node can be accessed:



The following figure illustrates how RemoteApps configured at multiple Remote Desktop Host Server nodes through Remote Desktop Connection Broker server can be accessed:



You need to perform the following procedures to deploy remote application programs through a remote desktop Web access:

- Install and configure the Remote Desktop Web access role service at an Remote Desktop Session Host server node installed with Windows 2012 or higher.
- Configure remote applications at a server node.
- Access the remote applications from a client node.

Install and Configure the Remote Desktop Web Access Role Service at a Remote Desktop Session Host Server Node

Remote Desktop Web Access service and Remote Desktop Host service (Remote Application) allow you to deploy a single Web site to run programs, access the full remote desktop, or connect remotely to the desktop of any computer in the internal network where you have the required permissions.

Log on to the Remote Desktop Session Host server node with local administrator privileges to install and configure the Remote Desktop web access role service at a remote Desktop Session Host server node.Use Network Level Authentication to provide a secure authentication method.

To install and configure the Remote Desktop web access role service at an Remote Desktop Session Host server node

1. Log on to the Remote Desktop Session Host server node with local administrator privileges.
2. Open the Server Manager and add roles and the required role services.
3. For Remote Desktop Services, select Remote Desktop Session Host and Remote Desktop Web Access. Add Required Role Services.
4. Specify the authentication method for the remote desktop session host.

   Note: Click the Require Network Level Authentication option for a secure authentication method.
5. Add the required user group you want to allow access to the Remote Desktop Session Host server.
6. Install the Remote Desktop Web Access role service.

You will be prompted to restart your computer once the installation is complete.

Configure Remote Applications at Remote Desktop Session Host Server Node

After the Remote Desktop Web Access role is installed and configured, you can configure the remote applications at Remote Desktop Session Host server node.

Use the Server Manager window to select the programs you want to add to the RemoteApps list. The general workflow is as follows:

1. Open the Server Manager window.
2. Add the required remote programs.

Allow Application Access to Specific Users

After the remote applications are configured, you can define users or user groups who can access the applications at the client node, if required.

Select which users or user groups you want to provide access to the application. The general workflow is as follows:

1. Configure remote applications.
2. Select the required remote application.
3. Add users.

The added users or user groups can now access the application at the client node.

Access the Remote Applications from a Client Node

At the client node, you can access the configured remote applications in the following ways:

- Access a program on a Web site using Remote Desktop Web Access.

  At the client node, open **Internet Explorer** and connect to the Remote Desktop Web Access Web site using the following URL: https://technet.microsoft.com/en-us/library/cc731508.aspx.

  Log on with a domain account of the Remote Desktop Session Host server's administrators group.

  Note: Any application launched from Remote Desktop Connection Broker appears as it were running on your local computer.

- Access a program on a Web site using Remote Desktop Web Access with Remote Desktop Connection Broker

  You can also access the configured remote applications from a client through another Remote Desktop Connection Broker Server node.

  Remote Desktop Connection Broker (RD Connection Broker), earlier known as Terminal Services Session Broker (TS Session Broker), provides access to remote applications and desktop connections. Accessing the remote applications and a desktop connection you can get a single, personalized, and aggregated view of RemoteApp programs, session-based desktops, and virtual desktops. Remote Desktop Connection Broker also supports load balancing and reconnection to existing sessions on virtual desktops, Remote Desktop sessions, and RemoteApp programs and aggregates RemoteApp sources from multiple Remote Desktop Session host (RD Session Host) servers that host different RemoteApp programs.

  Remote Desktop Connection Broker extends the TS Session Broker capabilities included in Windows Server

2012 and higher by creating a unified administrative experience for traditional session-based remote desktops and VM-based remote desktops. A VM-based remote desktop can be either a personal virtual desktop or part of a virtual desktop pool.

In case of a personal virtual desktop, there is a one-to-one mapping of VMs. You are assigned a personal virtual desktop that can be personalized and customized. These changes are available to you each time you log on to your personal virtual desktop. For a virtual desktop pool, a single image is replicated across many VMs.Virtual desktop pool is to provide users with a virtual desktop that is dynamically assigned from a pool of identically configured virtual machines. As you connect to the shared virtual desktop pool, you are dynamically assigned a virtual desktop. You may not be assigned the same virtual desktop when you connect the next time. This means that any personalization and customization made by you are not saved. If you use a virtual desktop pool and want to save any customization, you can use roaming profiles and folder redirection.

Note: The improvements to the Remote Desktop Connection Broker role service are particularly useful while implementing a Virtual Desktop Infrastructure (VDI) or deploying session-based desktops or RemoteApp programs. These improvements further enhance the Remote Desktop Services.

Add the Remote Desktop Connection Broker role service on a computer running Windows Server 2012 or higher, and then use Remote Desktop Connection Manager to identify the RemoteApp programs and virtual desktops that are available through RemoteApp and Desktop Connection.

You will need to prepare another node where Remote Desktop role service is installed and Remote Desktop Connection Broker service is enabled. For more information, refer to "Install and Configure the Remote Desktop Web Access Role Service at a Remote Desktop Session Host Server Node".

To add Remote Desktop Session Host server in RemoteApp sources of Remote Desktop connection broker server

1. Open the Server Manager window.
2. Add the RemoteApp Source.
3. Add the Remote Desktop Session Host server name.
4. Add the Remote Desktop Connection Broker Server name in the TS Web Access Computers security group.

   Note: Enable Network Discovery on the NLB Cluster nodes and RD Connection Broker node so that nodes can able to see each other and other network computers and devices and allows people on other network computers to see your computer.
5. Add the client node name in TS Web Access Computers security group on the Remote Desktop Connection Broker Server name.

To access RemoteApps configured at a Remote Desktop Session Host server from a client node

1. Connect to the Remote Desktop Web Access Web site.

   At the client node, open Internet Explorer and connect to https://technet.microsoft.com/en-us/library/cc731508.aspx.
2. Open the Enterprise Remote Access window.
3. Log on with a domain account of the local administrators group in all the nodes (Remote Desktop Connection Broker Server and Remote Desktop Session Host server).
4. Connect to the required Remote Desktop Connection Broker Server.

Note: Any application launched from the RD Connection Server Broker appears as it were running on your local computer. You can connect to the client machine through the VPN and access the RemoteApps.

The following table lists the applications which can be accessed as RemoteApp of the different System Platform nodes.

In Touch	Historian	Historian Client	Application Server	Common Utilities
Alarm DB Logger Manager	ITTagImporter	Trend	ArchestrA IDE	ArchestrA License Manager
Alarm DB Purge – Archive	Import InTouch Historical Data	Query	Object Viewer	Change Network Account
Alarm DB Restore	aahDBdump			Historian Configurator
Alarm Hot Backup Manager	ITHistImporter			License Utility
Alarm Printer	aahHistorianCfg			SMC
Alarm Suite History Migration				
InTouch				
Window Maker				
Window Viewer				

Display the System Platform Nodes on a Multi-Monitor with a Remote Desktop

Prerequisites for the client node where the remote desktop is invoked

- Graphics card that supports multi-monitor and associated drivers
- Client Machine with an operating system (OS) that has RDP 7.0 or higher

After the client machine is prepared, you can display the system platform on a multi-monitor with a remote desktop.

To display the system platform nodes on a multi-monitor with a remote desktop

1. Ensure that the client machine is able to detect plugged-in secondary monitors.
2. Use the Control Panel Modify to configure the display settings. Use the "Extend these displays" option from the multiple displays list.

Verify the Display of System Platform Nodes on a Multi-Monitor with a Remote Desktop

Prerequisites for VMs running on the host Virtualization Server:

- VM nodes with OS that has RDP 7.0 or higher
- VM nodes running products such as InTouch

Note: The host virtualization server runs on Windows 2012 or higher.

To verify system platform nodes display on a multi-monitor with a remote desktop, access any VM node installed with an IOM product from the client machine.

1. Open the Remote Desktop Connection window. Go to Run, and then enter "mstsc /admin". The Remote Desktop Connection window appears.
2. Click Display, and select the Use all my monitors for the remote session check box and then click **Connect**. The VM node opens.

   Note: If the client machine does not have RDP 7.0, this option will not be available to you.
3. Launch the IOM product and test the application. Drag and drop to move the application between the different monitors.

Use the Multi-Monitors as a Single Display

The multiple monitors configured on the client node, from where the remote desktop session is invoked, are used as independent displays when the remote session is used to connect to System Platform products installed on the VM nodes (with the exception of InTouch). In case of InTouch, the multi-monitors can be used either as independent displays or as a single display.

To use the multi-monitors as a single display, on an InTouch VM node, go to the path where win.ini exists and open win.ini. For example, the path is C:\User\<User_Name>\AppData\Local\Wonderware, where <User_Name> is the user login with which the remote session from the client connects to this VM node.

Enter the following parameters under the InTouch section and save it.

- MultiScreen – Enter "1" to enable the multi-monitor mode. Enter "0" to disable the multi-monitor mode.
- MultiScreenWidth – Enter the width of a single screen in pixels.
- MultiScreenHeight – Enter the height of a single screen in pixels. For example, if you want to show your InTouch application with a screen resolution of 2560 x 1024 on two horizontal monitors, enter the following:
  - "[InTouch]
  - MultiScreen=1
  - MultiScreenWidth=1280
  - MultiScreenHeight=1024"

Refer to the TechNote on multi-monitors for InTouch at https://gcsresource.invensys.com/support/kbcd/html/1/T001115.htm

Network Load Balancing

Network Load Balancing (NLB) distributes traffic across several servers by using the TCP/IP networking protocol. You can use NLB with a terminal server farm to scale the performance of a single terminal server by distributing sessions across multiple servers.

About the Network Load Balancing Feature

The NLB feature in Windows Server 2012 and higher enhances the availability and scalability of Internet server applications such as those used on Web, FTP, firewall, proxy, virtual private network (VPN), and other mission-critical servers. A single computer running Windows Server provides a limited level of server reliability and scalable performance. However, by combining the resources of two or more computers running one of the products in Windows Server into a single virtual cluster, an NLB can deliver the reliability and performance that Web servers and other mission-critical servers need.

About Remote Desktop Connection Broker

Remote Desktop Connection Broker keeps track of user sessions in a load-balanced Remote Desktop Session Host server farm. The Remote Desktop Connection Broker database stores session information, (including the name of the Remote Desktop Session Host server where each session resides), the session state for each session, the session ID for each session; and the user name associated with each session. Remote Desktop Connection Broker uses this information to redirect a user who has an existing session to the Remote Desktop Session Host server where the user's session resides.

Remote Desktop Connection Broker is also used to provide users with access to RemoteApp and Desktop Connection. RemoteApp and Desktop Connection provide a customized view of RemoteApp programs and virtual desktops. Remote Desktop Connection Broker supports load balancing and reconnection to existing sessions on virtual desktops accessed by using RemoteApp and Desktop Connection. To configure the Remote Desktop Connection Broker server to support RemoteApp and Desktop Connection, use the Remote Desktop Connection Manager tool. For more information, see the Remote Desktop Connection Manager Help in Windows Server.

Remote Desktop Connection Broker that is used in an NLB setup is included in Windows Server 2012 and higher. You do not require a license to use this feature.

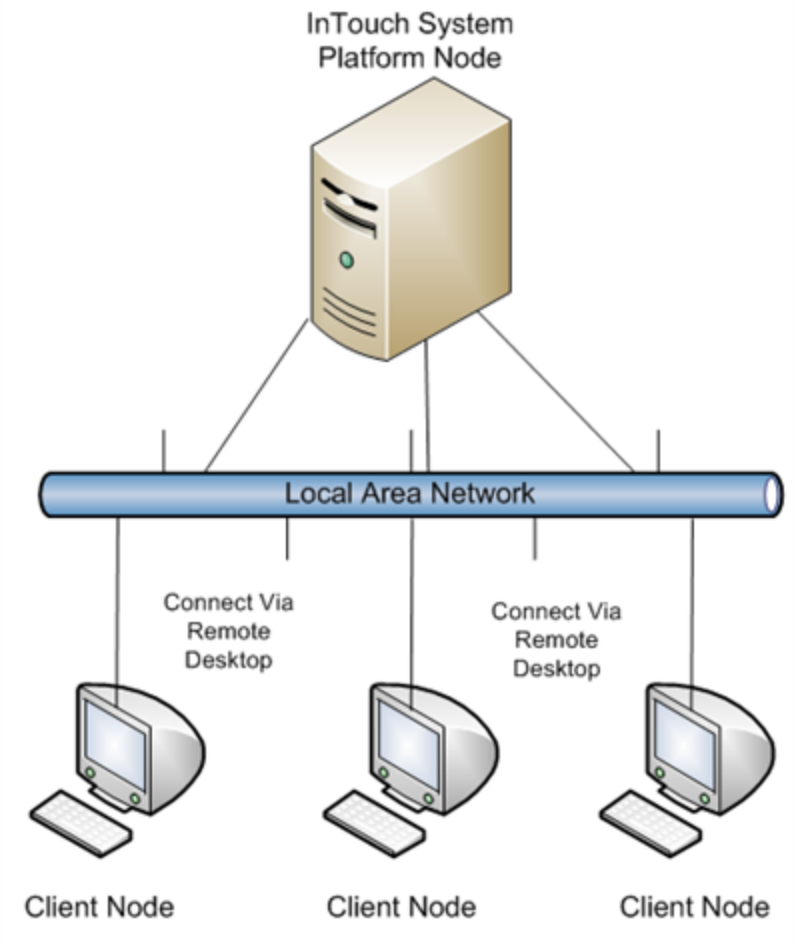
You need a Microsoft RD license for managing the remote desktop terminal server sessions.

About Managed InTouch Application with Network Load Balancing

The features provided by Remote Desktop are made available through the Remote Desktop Protocol (RDP). RDP is a presentation protocol that allows a Windows-based terminal (WBT), or other Windows-based clients, to communicate with a Windows-based Terminal Server. RDP is designed to provide remote display and input capabilities over network connections for Windows-based applications running on your Windows desktop computer.

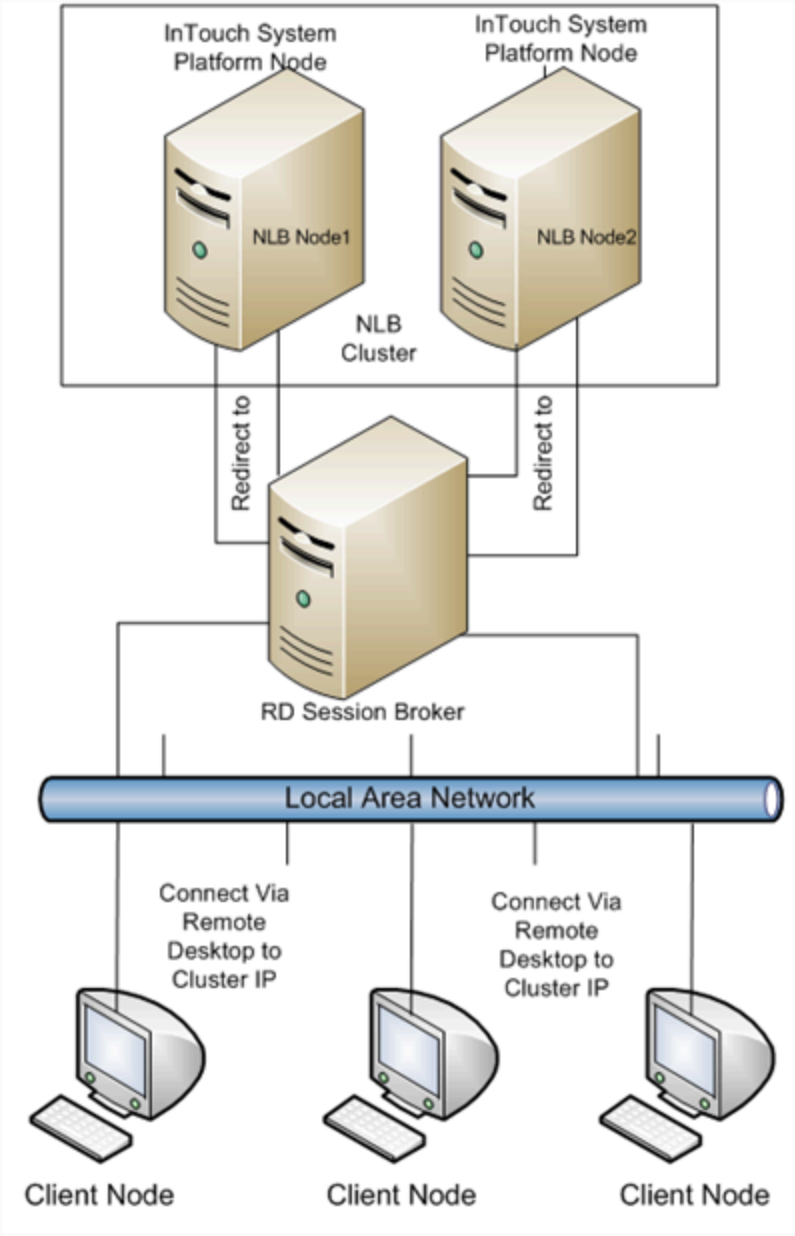
In this topology, clients can access the InTouch System Platform node via Remote Desktop. Whenever a new connection is requested to the InTouch System Platform Node, a new session is created. So all the traffic goes to the system platform node and degrades the performance of the InTouch node.

The following figure displays a topology without Network Load Balancing (NLB):



Network Load Balancing distributes IP traffic to multiple copies (or instances) of a TCP/IP service, such as a Web server, each running on a host within the cluster. Network Load Balancing transparently partitions the client requests among the hosts and enables the client to access the cluster using one or more "virtual" IP addresses. The cluster appears to be a single server that answers these client requests.

The following figure displays a topology with Networking Load Balancing:



Note: The Remote Desktop Connection Broker shown, as a separate node in the above topology, can be configured on one of the NLB cluster nodes itself.

You can leverage the load balancing for InTouch-managed applications.

To configure an NLB for managed InTouch application

1. Configure one VM or Physical machine with Application Server
2. On both the NLB cluster nodes, install InTouch TS with terminal server license.
3. Configure an NLB cluster as explained below.
4. On the Application Server node, develop managed InTouch application and deploy it on each of the NLB Cluster node.

Configuring an NLB for InTouch System Platform nodes, allows you to combine application servers to provide a level of scaling and availability that is not possible with an individual server.

NLB distributes incoming client requests to InTouch System Platform nodes among the servers in the cluster to more evenly balance the workload of each InTouch System Platform server and prevent overload on any InTouch System Platform server. To client computers, the NLB cluster appears as a single server that is highly scalable and fault tolerant.

Leveraging Network Load Balancing

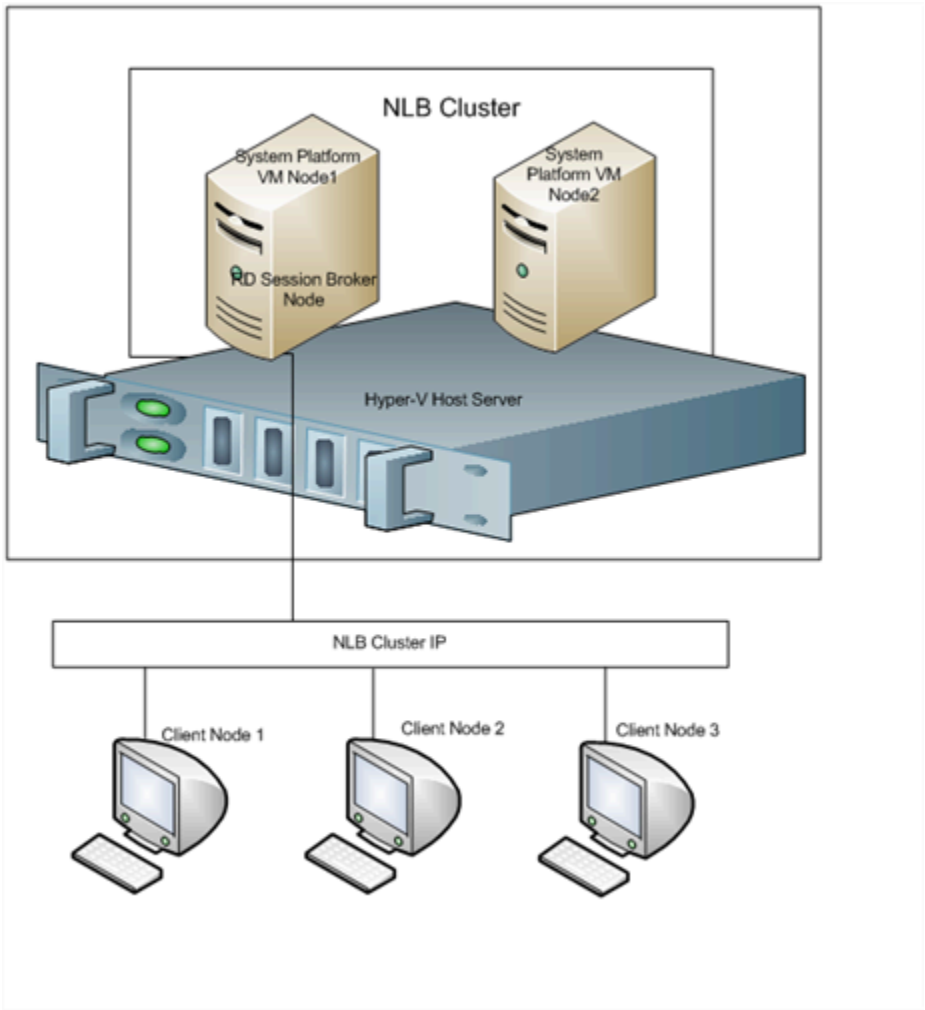
To setup an NLB:

1. Prepare two VM nodes that are remote desktop-enabled and have Windows Server 2012 or higher.
2. Assign static IPs to both nodes.

   Note: NLB disables Dynamic Host Configuration Protocol (DHCP) on each interface it configures, so the IP addresses must be static.

Example Topology 1: Configuring Remote Desktop

You can configure an NLB cluster configuring the Remote Desktop Connection Broker on one of the NLB cluster nodes.



To configure NLB with Topology 1

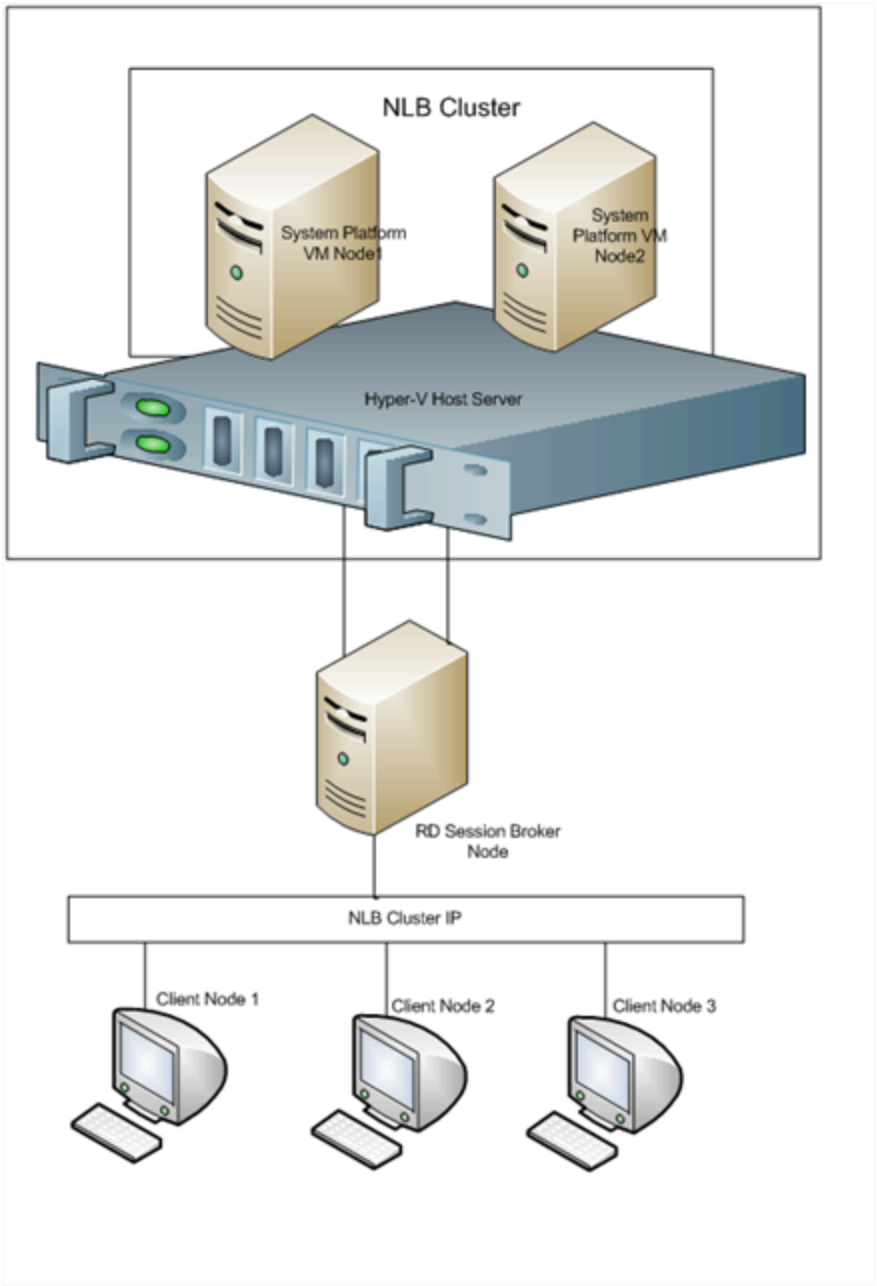
1. On each of the cluster nodes install Remote Desktop Services. For more information, refer to "Install Remote Desktop Services".

   Note: On the **Select Role Services** screen, select Remote Desktop Session Host and Remote Desktop Connection Broker on one of the Cluster Nodes to configure it as NLB Cluster node as well as RD connection broker node. On the other NLB Cluster node, select only Remote Desktop Session Host.

2. On each of the cluster nodes, install Network Load Balancing. For more information, refer to "Install Network Load Balancing".
3. On the NLB cluster node which is configured as RD connection broker as well, add a Remote Desktop Session Host Server. For more information, refer to "Add a Remote Desktop Session Host Server".
4. On each of the cluster nodes, create a Network Load Balancing Cluster. For more information, refer to "Create a Network Load Balancing Cluster".
5. On each of the cluster nodes, configure Remote Desktop Connection Broker Settings. For more information, refer to "Configure Remote Desktop Connection Broker Settings".

Example Topology 2: Configuring Remote Desktop Connection Broker on a Separate Node

Instead of configuring the Remote Desktop Connection Broker on one of the NLB cluster nodes, you can also configure the Remote Desktop Connection Broker on a separate node.



To configure NLB with Topology 2

On the NLB Cluster nodes, do the following:

1. Install Remote Desktop Services. For more information refer to "Install Remote Desktop Services".

   Note: In **Select Role Services** screen, select **Remote Desktop Session Host** on the NLB Cluster nodes.

2. Install Network Load Balancing. For more information, refer to "Install Remote Desktop Services".
3. Create a Network Load Balancing Cluster. For more information, refer to "Create a Network Load Balancing Cluster".
4. Configure remote desktop connection broker settings.For more information, refer to "Configure Remote Desktop Connection Broker Settings".

On the Remote Desktop Connection Broker Node do the following:

1. Install Remote Desktop Services. For more information, refer to "Install Remote Desktop Services".

Note: On the **Select Role Services** screen, select only Remote Desktop Connection Broker on the Remote Desktop Connection Broker Node.

1. Add a Remote Desktop Session Host Server. For more information, refer to "Add a Remote Desktop Session Host Server".

Install Remote Desktop Services

Remote Desktop Services provides technologies that enable access to session-based desktops, VM-based desktops, or applications in the datacenter from both within a corporate network and the Internet. Remote Desktop Services enables a rich-fidelity desktop or application experience, and helps to securely connect remote users from managed or unmanaged devices.

Use the Server Manager to install Remote Desktop Services. Specify the option, "Do not require network level authentication."

There are two types of Windows Client Access Licenses from which to choose: device-based or user-based, also known as Windows Device CALs or Windows User CALs. This means you can choose to acquire a Windows CAL for every device (used by any user) accessing your servers, or you can choose to acquire a Windows CAL for every named user accessing your servers (from any device).

When you complete Remote Desktop Services installation, restart the node.

To install Remote Desktop Services

1. Open the Server Manager window.
2. Add the required role services. Select Remote Desktop Session Host and Remote Desktop Connection Broker.
3. Specify Do not require Network Level Authentication.
4. Select the applicable licensing option.

   Note: There are two types of Windows Client Access Licenses from which to choose: device-based or user-based, also known as Windows Device CALs or Windows User CALs. This means you can choose to acquire a Windows CAL for every device (used by any user) accessing your servers, or you can choose to acquire a Windows CAL for every named user accessing your servers (from any device).
5. Confirm the details you entered, and install the services.

Install Network Load Balancing

You will need to install a Network Load Balancer (NLB) on the network adapter that you want to use for the Remote Desktop Protocol (RDP) connection.

Use the Server Manager to install the NLB. Select Network Load Balancing from the list of options.

To install NLB

1. Open the Server Manager window.
2. Add the Network Load Balancing feature and install it.

Add a Remote Desktop Session Host Server

A Remote Desktop Session host (RD Session Host) server hosts Windows-based programs or the full Windows desktop for Remote Desktop services client. You can connect to an Remote Desktop Session Host server to run programs, save files, and use network resources on this server. You can access an Remote Desktop Session Host server by using Remote Desktop Connection or RemoteApp.

You can add a Remote Desktop Session Host server to the connection broker computers' local group.

Use the Configuration option in the Server Manager to add an RD Session Host server. Select the required group to add to the Remote Desktop Session Host server.

To add an RD Session Host server

1. Open the Server Manager window.
2. Select the Session Broker Computers group to add to the Remote Desktop Session Host server.
3. Add the computer account for the Remote Desktop Session Host server.

Create a Network Load Balancing Cluster

To configure an NLB cluster, you need to configure the following parameters:

- Host parameters that are specific to each host in an NLB cluster.
- Cluster parameters that apply to an NLB cluster as a whole.
- Port rules

  Note: You can also use the default port rules to create an NLB cluster.

Use the Network Load Balancing Manger to connect the required host to a new cluster.

- If you select the Unicast option, NLB instructs the driver that belongs to the cluster adapter to override the adapter's unique, built-in network address and change its MAC address to the cluster's MAC address. Nodes in the cluster can communicate with addresses outside the cluster subnet. However, no communication occurs between the nodes in the cluster subnet.
- If you select the Multicast option, both network adapter and cluster MAC addresses are enabled. Nodes within the cluster are able to communicate with each other within the cluster subnet, and also with addresses outside the subnet.

Add additional hosts as needed for load balancing. Then, add users to the Remote Desktop Users group to access the Network Load Balancing Cluster.

Note: Users can be local users and need not be domain users/administrators. If the users are local users they should be added on both the NLB cluster nodes with same user name and password.

To create an NLB cluster

1. Open the Network Load Balancing Manager window.
2. Enter the name of the host for the new cluster.
3. Select an interface for the new cluster, and create the cluster.

   Note: The Priority value is the unique ID for each host. The host with the lowest numerical priority among the current members of the cluster handles the entire cluster's network traffic that is not covered by a port

rule. You can override these priorities or provide load balancing for specific ranges of ports by specifying the rules on the Port rules tab of the Network Load Balancing Properties window.

- Add a cluster IPv4 static address, and enter the subnet mask.
- Enter the internet name of the new cluster.
- Select either the unicast or multicast option.
  - If you click the Unicast option, NLB instructs the driver that belongs to the cluster adapter to override the adapter's unique, built-in network address and change its MAC address to the cluster's MAC address. Nodes in the cluster can communicate with addresses outside the cluster subnet. However, no communication occurs between the nodes in the cluster subnet.
  - If you click the Multicast option, both network adapter and cluster MAC addresses are enabled. Nodes within the cluster are able to communicate with each other within the cluster subnet, and also with addresses outside the subnet.

4. Add another host to the cluster. Enter the name of node 2 and connect to it.
5. Enter a priority value for the host.

To add users to the Remote Desktop Users group to access Network Load Balancing Cluster

1. Specify which remote desktop versions you want to allow access to.
2. Select which user you want for which you want to allow access.

Note: The users can be local users and need not be domain users/administrators. If the users are local users they should be added on both the NLB cluster nodes with same user name and password.

Configure Remote Desktop Connection Broker Settings

Remote Desktop Connection Broker, earlier called Terminal Services Session Broker (TS Session Broker), is a role service that enables you to do the following:

- Reconnect to existing sessions in a load-balanced Remote Desktop Session Host server farm. You cannot connect a different Remote Desktop Session Host server with a disconnected session and start a new session
- Evenly distribute the session load among Remote Desktop Session Host servers in a load-balanced Remote Desktop Session Host server farm.
- Access virtual desktops hosted on Remote Desktop Virtualization host servers and RemoteApp programs hosted on Remote Desktop Session Host servers through RemoteApp and Desktop Connection.

To configure Remote Desktop connection broker settings, select the Farm member option and enter the name of the node where RD Connection Broker is installed. Then, enter the name of the farm that you want to join in the Remote Desktop Session Broker, select the option to participate in Connection Broker Load Balancing and assign weight for the server. You do this for both nodes.

Note: By assigning a relative weight value, you can distribute the load between more powerful and less powerful servers in the farm.

To add users to the Remote Desktop Users group to access Network Load Balancing Cluster

1. From Control Panel > System and Security > System Remote, select System Properties.
2. Under Remote Desktop, click the relevant option to specify the remote desktop versions you want to allow

access to.

3. Select users to provide access to the system.

Note: The users can be local users and need not be domain users/administrators. If the users are local users they should be added on both the NLB cluster nodes with same user name and password.

Disconnect from and Connect to a Remote Desktop Session

If you disconnect from a session (whether intentionally or because of a network failure), the applications you were running will continue to run. When you reconnect, the Remote Desktop Connection Broker is queried to determine whether you had an existing session, and if so, on which Remote Desktop Session Host server. If there is an existing session, Remote Desktop Connection Broker redirects the client to the Remote Desktop Session Host server where the session exists.

With Remote Desktop Connection Broker Load Balancing, if you do not have an existing session and you connect to an Remote Desktop Session Host server in the load-balanced Remote Desktop Session Host server farm, you will be redirected to the Remote Desktop Session Host server with the fewest sessions. If you have an existing session and you reconnect, you will be redirected to the Remote Desktop Session Host server where your existing session resides. To distribute the session load between more powerful and less powerful servers in the farm, you can assign a relative server weight value to a server.

View Connected Sessions

You can use Remote Desktop Services Manager to view sessions connected to each node of the NLB cluster, and view information and monitor users and processes on Remote Desktop Session host (RD Session Host) servers. Open the Remote Desktop Services Manager window from any node of the NLB to view sessions connected to each node of the cluster.

To view sessions connected to each node of the cluster

1. On any node of NLB, open the Remote Desktop Services. From Administrative tools, open the Remote Desktop Services Manager.
2. Create a new group and enter the group name.

Note: The group name need not be the same as the cluster name.

3. Add the required computers to the group.

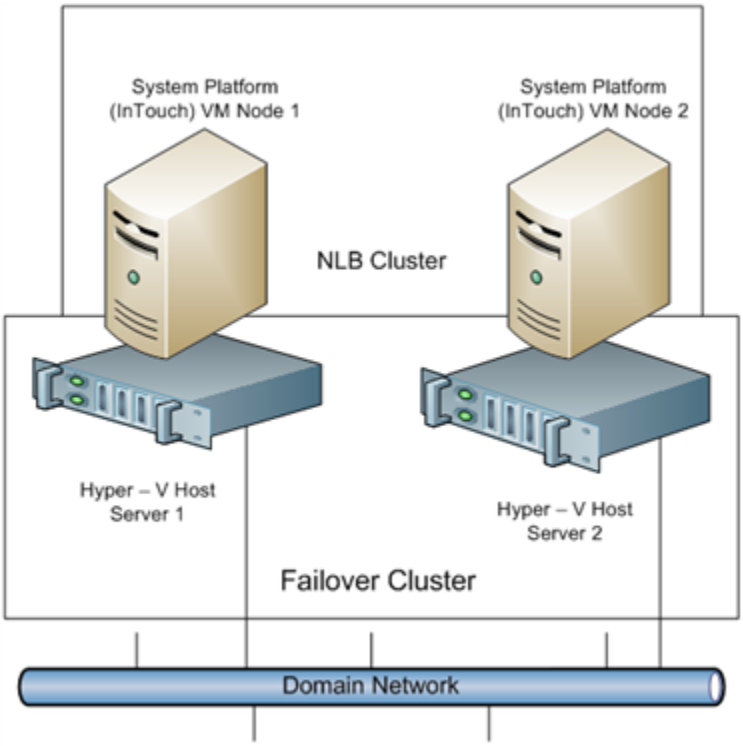
You can now select the newly-created group name and view the sessions connected to each node of the cluster.

Configure Network Load Balancing Cluster on Microsoft Failover Cluster

Windows Server 2012 and higher provide two clustering technologies: failover clusters and NLB. Failover clusters primarily provide high availability; NLB provides scalability and, at the same time, helps increase availability of Web-based services.

By using a failover cluster, you can ensure that there is nearly constant access to important server-based resources. A failover cluster is a set of independent computers that work together to increase the availability of services and applications. The clustered servers (called nodes) are connected by physical cables and by software. If one of the nodes fails, another node begins to provide service through a process known as failover.

NLB that is configured in a failover cluster offers high performance in environments in which each request from a client is stateless, and there is no in-memory application state to maintain



To configure NLB cluster on Microsoft failover cluster

1. Set up Microsoft Failover Cluster out of two Hyper-V host servers.
2. Configure two VM nodes one on each Hyper-V host server.
3. Configure the NLB cluster out of two VM nodes hosted by each Hyper-V host server following the procedures in Leveraging NLB by Configuring Remote Desktop Session Broker on a NLB Cluster Node explained in topology 1. For more information, refer to "Example Topology 1: Configuring Remote Desktop".

Understanding the Behavior of NLB Cluster in Microsoft Failover Cluster

1. During a live migration of one of the NLB cluster nodes, there are no disruptions in the active sessions connected to the cluster node. The Reconnect window will not appear on the NLB cluster node as there is no disruption of the active session. After the live migration is complete all sessions connected to the NLB cluster node are retained.
2. During a quick migration, when one of the Hyper-V host servers (Microsoft Failover Cluster Node) is shut down or switched off and the failover is completed, all active sessions on the NLB cluster node hosted by the Microsoft failover cluster node are automatically connected and all sessions on the NLB cluster node are retained.

Observations while using NLB for Managed InTouch System Platform node:

- The NLB feature is qualified for InTouch managed application. InTouch TSE license is required on each of the NLB cluster nodes.

- Local InTouch Tag Alarms are local to the session. Local InTouch Tag alarms updated in a session remain local to that session only.
- ArchestrA Alarms are common across all sessions. ArchestrA Alarms updated in one of the sessions get reflected across all the sessions.
- For IO tags poking in one session, the data reflects across all the sessions. However, while poking local InTouch tags, data does not get updated across all sessions since it is local to the session.
- When you lose the NLB cluster node with the active sessions, all the active sessions on the NLB cluster node closes. To retain all the active sessions, configure the NLB Cluster in a Microsoft Failover Cluster in a Hyper-V environment. The NLB cluster nodes are VM nodes hosted by Hyper-V host servers and Hyper-V host. For more information, refer to "Configure Network Load Balancing Cluster on Microsoft Failover Cluster".

Hardware Licenses in a Virtualized Environment

Windows Server 2012 does not support hardware licenses in the Hyper-V virtualized environment. You may want to verify support under later server editions.

Hardware licensing using AnywhereUSB are supported. For more information, visit http://www.digi.com/products/usb-and-serial-connectivity/usb-over-ip-hubs/anywhereusb

Planning Storage in a Virtualized Environment

In virtual environments, storage options require special consideration. This chapter introduces available and recommended options and provides information to guide decisions.

As a rule of thumb, instead of simply directing the traffic of four VMs to one hard disk, use one spindle (drive arm) per VM. This guideline will help you to avoid the most common resource problem.

Shared storage is one of the least-understood components because, outside of traditional large data centers, it is rarely used. If a common file storage environment is needed, the solution is often to install extra hard drives in a server, and sharing a drive or a folder.

The major advantage of a shared storage solution is high availability. While most hypervisors can utilize local storage on a server to run virtual machines, high availability and portability functions are lost if data is stored on a local machine instead of on a shared storage solution.

Choosing Connectivity

When establishing shared storage to take advantage of high availability, the first major choice involves connectivity to the device. While shared storage is more accurately referred to as network storage, the "network" component does not necessarily refer to ethernet.

The first type of network shared storage used for sared storage is Fibre Channel. It uses a proprietary protocol (as opposed to TCP/IP) transported over a fiber-optic cable. Like traditional Ethernet, Fibre Channel requires a specialized adapter in the host, along with switches to aggregate connectivity.

The second major technology choice is the more familiar Ethernet. Using Ethernet involves standard network interface cards located in the server, standard Ethernet switches, and copper cabling for up to 1GB/s speeds.

Fibre Channel

For many years, there were three primary reasons for choosing Fibre Channel.

- Fibre Channel was typically always faster than any competing Ethernet protocol. The most common top end speed of Fibre Channel today is 8 GB/s, with 16 GB/s becoming more common.
- Because Fibre Channel utilizes a proprietary protocol that does not contain the overhead of TCP/IP, the communications latency is extremely low, typically < 1 ms.
- Until recently, almost all quality storage arrays implemented Fibre Channel as their interface of choice.

Ethernet

While Fibre Channel still enjoys some advantages over Ethernet, modern implementations of Ethernet protocols are eliminating these advantages.

- The speed advantage enjoyed by Fibre Channel over Ethernet has effectively been nullified. Ethernet at 10GB/s has been in common use since 2010, and 40GB and 100GB speeds will soon be available.
- Specialized Ethernet switches can help you obtain ultra-low latency connections within your Ethernet fabric.
- You can usually connect to all but the most high-end network storage devices with Ethernet protocols.
- Specialized Ethernet cards offer TCP Offload Engine (TOE) technology, which offloads all processing of the TCPI/IP stack to the netwok controller. This becomes a major advantage when using gigabit and 10 gigabit Ethernet speeds, in which processing the network stack becomes a significant task.

Today, it is not typical to see a new installation utilize Fibre Channel, and is generally not recommended.

Instead, most new small and mid-sized environments are choosing Ethernet protocols. Of special note is the need to separate the storage and computer networks, a concept discussed in further detail later.

Choosing Protocols

If you have chosen Ethernet connectivity, you must next select the protocol, either NFS or iSCSI.

Another popular term used when discussing NFS vs. ISCSI is File vs. Block. Understanding this difference will help you understand the difference in the protocols. The following table lists the File and Block storage options.

Block Storage Options (SAN)	**File-Based Storage Options (NAS)**
iSCSI	NFS
Fibre Channel	CIFS
AoE (ATA over Ethernet)	

Both protocols are capable of acceptable performance in small- to medium-sized environments. The primary differences are reflected in setup and scalability.

Advantages: Protocol Setup and Scalability

Although there are a few more steps in setting up an iSCSI datastore, this is usually a one-time activity that only takes a few extra minutes more than configuring an NFS datastore.

Another major difference between protocols regards their scalability. While you can connect multiple network cables to an NFS controller, only one of them can be used at a time between a computer host and the storage device. This is an inherent limitation in the protocol. The primary purpose of using multiple connections to an NFS array is for redundancy, not throughput. iSCSI, on the other hand, can use as many network connections as you configure between the computer host and the network device. It is also a protocol-specific ability that allows for this functionality. The real question is determining if this difference actually matters. A single host rarely saturates a typical 1 GB/s link between the host and storage unit. Only in extreme cases (for example, when the user is running a specialized test) does saturation of the 1 GB/s link occur.

Pros and Cons: NFS vs SAN Protocols

One final consideration is the optimizations present in the VMFS file system utilized by VMWare when implementing block storage. While NFS was designed from the ground up as a multi-user protocol, it was never designed with the intent of handling really large virtual machine disk files. VMFS, on the other hand, has always been and will continue to be designed to handle extremely large files with simultaneous multi-host access.

NFS Protocol

Pros:

- Usually less expensive
- Simpler configuration
- Simpler management

Cons:

- Higher CPU overhead

SAN Protocol

Pros:

- Higher performance
- Ability to offload protocol overhead to hardware components
- Allows hypervisor to use specialized file system

Cons:

- Usually more expensive
- More complex configuration

Initializing the NFS Protocol

For NFS, the following tasks are involved in setting up a datastore:

1. Create an NFS export on your storage device.
2. Use vSphere to create a special network port over which you will connect to your NFS datastore.
3. Connect vSphere to the NFS export and create a new datastore. Once connected, the user can immediately begin to store virtual machines on the newly created datastore.

Initializing the iSCISI Protocol

For iSCSI, there are a few additional steps.

1. First, create an LUN on your storage device.
2. From vSphere create a special network connection for ISCSI data.
3. Next, add your storage device as an ISCSI target.
4. After adding the device, perform a rescan of ISCSI targets. At this point, you should see the LUN created.
5. Select the LUN and create a new datastore.
6. Once the datastore is created, format the datastore with VMFS.

Choosing Features

Once connectivity methods and protocols have been determined, select a system with components and features that match your needs.

Controllers

All disk systems have at least one disk controller through which read/write requests are passed. The following sections detail the types, attributes, and features of controllers to help you to select the configuration that best suits your needs.

Controller Attributes

The primary difference between storage arrays, and the type of storage used in standard desktops and servers, is the quality and quantity of controllers. In a storage array, the controllers are designed for high throughput, as when many (10+) computer hosts access data simultaneously. For this reason, these specialized controllers usually contain more RAM and a faster chipset than what is found in a typical Redundant Array of Independent Disks (RAID) controller in a standalone server. Arrays installed in a production environments must have a redundant configuration.

There are two reasons for this.

- When a controller fails, a backup controller takes over its duties without any system interruption. Depending on the architecture, this may cause a decrease in performance:

  - In an "Active-Active" array, both controllers are processing the workload. If one controller is lost, performance is cut in half.
  - In an "Active-Passive" array, only one controller is active at a time. In the event of a failure, no performance difference should be experienced. Depending on your performance requirements, this may not be much of a distinguishing factor.
- If using multiple controllers, ensuring that they are hot-swappable enables their replacement at the time of failure with no interruption in availability.

  Tip: Though not necessary, the ability to upgrade firmware by failing back and forth between controllers can be a useful addition.

Controller NVRAM and Cache

Another feature to consider carefully is the presence of NVRAM, or a battery-backed cache, on the controllers. These features store data as it is being written, so that if power is cut while the array is writing data, the write can be completed when power is restored.

Depending on numerous different factors, this downtime can be as long as a week or two before potential corruption becomes an issue. This is a feature that quality arrays will implement as a matter of standard configuration. If this is offered as an option rather than being automatically included, it is a red flag that this array may not be suitable for a demanding environment. While downtime is bad, corrupted data is unacceptable in a manufacturing environment.

Network Accessibility

When selecting a controller, consider the number of network ports offered. At a minimum, there should be two network ports for data and one network port for a maintenance interface. Having a separate port for maintenance allows routing to your production network for configuration and maintenance while leaving the actual storage data on separate ports on a separate network.

Expansion

To prepare for future expansion, consider whether the unit has additional "shelves" or "enclosures." Typically, these units take the form of a 2U device with nothing but hard drives and an interconnect. These enclosures are connected via an external SAS cable to the controllers in the first enclosure. On each of these shelves, there are usually IN and OUT connections, allowing more shelves to be added in a daisy chain fashion. If implemented properly, these new enclosures can be added without any disruption to the running array. It is not atypical for a modest array to support as many as 48 to 96 hard drives on a single set of controllers.

Online Maintenance

Just as modern DCS systems are engineered to run for years without downtime, a quality array should be similarly designed.

Downtime can typically have two sources:

- Component malfunction. This is mitigated via component-level redundancy and careful design of parts to maximize Mean Time Between Failure (MTBF).

- Upgrades and modifications. A quality array will allow for the creation, maintenance, and resizing of disk volumes with no downtime on the system or the volume. As mentioned earlier, a quality array will also allow for firmware updates with no downtime.

Software Features

The software features available in the modern storage array are extensive. Some of the major features include compression, deduplication, snapshots, and replication.

Compression works in a similar manner as creating ZIP files on your computer and unzipping the file when you need to access the files. Compression and decompression are performed by the array in the background as data are stored and retrieved.

Deduplication involves a system analyzing each block of data being stored and determining if an exact copy already exists. If a copy exists, then the system simply stores a pointer to the existing block instead of storing the data a second time. Typical deduplication ratios in a relatively homogenous environment (i.e. lots of Windows installs) are approximately 5x to 10x. With deduplication, what previously took 5 TB to store now only takes 1 TB.

A snapshot is a method of backup that takes place on the array itself rather than by software installed on the machine. While snapshots are efficient, some users may find that they are slightly more difficult to work with as opposed to a typical virtual machine backup software package.

Some arrays provide the ability to perform near real-time replication. Though expensive, this is a excellent method for ensuring business continuity in the event of a disaster taking out the primary array. A major drawback is that a corrupted file may be replicated, creating two corrupted files or databases. Some replication methods combat this by allowing a rollback to a previous state.

Performance

The final and most important item to consider when purchasing an array is its performance.

Disk performance takes two major forms; Input/Output Operations Per Second (IOPS) and throughput. IOPS is measured in total read / write operations per second. Throughput is typically measured in MB/second.

While both are important measures, the primary limiting factor in most environments is IOPS. An I/O operation occurs whenever data is written to, or read from, the disk.

There are three major factors under the user's control that influence IOPS.

The first is disk speed. The faster the speed of the underlying disk, the more IO operations a particular disk can support. Although seek time and rotational latency are factors, we will focus on disk speed.

Second, the total number of drives - commonly referred to as spindles - in a volume (aggregated set of disks with a particular capacity) can influence IOPS. The more spindles in a volume, the more IOPS it can support. Using multiple slower disks can sometimes provide better performance than fewer fast disks.

Finally, the RAID configuration of the volume can have a substantial effect on the IOPS performance. The easiest way to see the effect of each is to calculate the average IOPS for a particular disk arrangement while adjusting different parameters to see the effect.

RAID Impact on System Performance

When writing to a RAID array, the system must not only split the write across multiple disks, but also calculate parity bits (in all but mirrored (RAID 1), striped (RAID 0), or mirrored + striped configurations). The more parity

bits required, the more severe the hit on write performance.

This can have a dramatic effect on performance; on a typical Application Object Server on a System Platform environment, disk access consists almost entirely of write operations. Though historians also typically have a high percentage of write operations, keep in mind the number of clients that may be running trends at the same time.

In order to measure read and write operations, use Perfmon, a tool included with all Microsoft operating systems. As a best practice, run these metrics at one minute intervals for 24 hours. This frequency should account for daily activities and for backups.

In a real-world case study across multiple Application Object servers, an average of 130 write operations per second - accounting for nearly 100% of disk activity - was observed. These writes operations, in turn, were almost exclusively the result of the application engines writing checkpoint files and historical store/forward data to protect against engine failures. As a side note, the system originally distributed its entire load across three machines instead of five. When the system only had three machines, the checkpointing was slowed to once every five seconds because the machines could not keep up.

This was initially speculated to represent insufficient RAM and CPU, but a closer look at disk statistics revealed that the bottleneck was the disk subsystem. In response, a pair of 2.5" 10K RPM drives in a RAID 1 configuration was installed in each machine. According to an online calculator, this configuration was capable of supporting 140 IOPS. A quick check of the math yields (130 IOPS * 5 new machines)/(3 old machines) = 217 IOPS/old machine.

In summary, performance - rather than capacity - is the primary concern when planning an array for your environment. If a system cannot meet I/O demand, capacity is not a concern. For this reason, you will typically see that the highest quality arrays provide capacity at or under a terabyte. This is because manufacturers realize that a user will typically run out of IOPS before GBs.

SSD Performance

When shopping for arrays, you will find those using solid state disks (SSDs). Though these devices were initially created and marketed to fight vibration and shock issues in industrial PCs, this has become a secondary concern. The primary reason for inclusion of SSDs in newer arrays is their superior performance vs. traditional, mechanical hard drives.

A glance at available online hard drive benchmarks like those available at http://www.harddrivebenchmark.net/ shows that even low-end, consumer grade SSDs far outperform expensive, high-RPM, mechanical drives.

Just as critical, however, is a disk's lifetime. Users are accustomed to hard drives lasting at least five years, and it is not unusual for a drive to last even longer. However, SSDs have a shorter useful life, and typically fail suddenly and catastrophically.

Many of the more traditional vendors utilize SSDs as a tier of storage. In this scenario, the array watches the blocks of data that are most active in terms of read/write activity. The most active blocks are transferred by the array from the slowest mechanical drives up to the faster mechanical drives, and finally to a layer of SSDs providing the best performance. The purpose of this approach is economic, which allows using much more expensive devices without making the overall unit unaffordable.

A second approach involves using SSDs as a conduit, or sort of cache, to slower disks. In this configuration, all write operations are performed on the SSDs first. Once the resources are are available, this data is transferred to slower, cheaper disks in the array.

A third and riskier configuration involves packing the entire array with consumer grade SSDs, using sophisticated software to perform inline compression, deduplication, and other advanced techniques to reduce the number of writes required.

Networking

Network storage refers to a physically isolated network that manages all storage traffic communication. The storage network should be a dedicated system of cables and switches, for many of the same reasons needed to isolate the PLC network from general client-server traffic. When designing your network, plan for redundant switches. Losing your storage backend will typically allow machines to run for about 20 or 30 seconds in a frozen state until they fail.

A properly configured storage network involves a single controller that connects to multiple switches with a minimum of four network connections. This ensures that the system can continue operation in the event of a controller and switch failing simultaneously.

Cost Factors

A good general rule when budgeting for a virtualization project is that your main server(s) should account for 50% of the total hardware budget, and storage for the other 50% of the total hardware budget (including 4 switches, 2 for virtual machine traffic, and 2 for storage traffic).

Note that some vendors include only base functionality in a starting price, and allow you to select features, like those discussed earlier, in an a la carte fashion. It can sometimes double or triple the starting price of your unit.

Finally, pay close attention to warranty costs. While higher-end units will typically include three to five years of base warranty, maintenance costs after warranty expiration can become extremely expensive. This can be a driver in the refresh cycle for a typical IT organization, since while old hardware may be functioning well, the costs of maintaining the warranty for old hardware can sometimes make it more affordable to purchase new hardware. Work closely with your budget managers on this detail.

Conclusions

As regards the cost of a virtualization environment, note the following:

- A three physical host system can easily support 24-30 servers.
- An average physical server should cost approximately $5K if properly specified.
- The acquisition cost for these servers, ignoring the additional cost of networking, would be approximately $120K.

Contrasted with a $50K acquisition cost for a virtualized three-host system with storage, a virtualized system with high quality storage is much less expensive. The economic advantages are significant once you pass 8-10 servers.

Acknowledgements

The preceding information is provided with express permission, and with content created by, Avid Solutions, and Andy Robinson with Avid Solutions.

The original content was also authored by A. Robinson and R. Kambach as part of a white paper for the Developer Network resource.

Implementing Backup Strategies in a Virtualized Environment

A virtual server backup is a copy of data stored on a virtual server to prevent data loss. There are two fundamental types of backups:

- Guest-level backup
- Host-level backup

Backup and Restore Strategies

There are a number of backup and restore strategies in both virtualized and non-virtualized environments. For the guest level, the virtual machines (VMs) are backed up as if they were physical servers. Although this strategy is among the simplest, it also has several drawbacks. You need to install backup software in each virtual machine (VM) to be copied in Guest Operating Systems, and maintain separate backup jobs (or even multiple backup jobs) per VM. This approach requires additional resources to execute the backups, and can affect the performance of the virtual machines. This backup type is not suitable for restoration in the event of a disaster or granular restores within applications, such as databases or email.

Another backup strategy is to use a host-level backup. In this approach, back up the entire VM at one time. However, it can be as granular as your backup and restore application allows it to be.

We recommend using the host-level backup. It creates a complete disaster recovery image of the virtual server, which can be restored directly into the source virtual infrastructure.

Checkpointing Method

In this method you can take point-in-time checkpoints (snapshots) of the entire VM. We recommend this method as it ensures data consistency and allows for a fast and complete restore of the entire VM. One of the few disadvantages in this method is that you need to restore the entire checkpoint even if a file is lost or corrupt.

In a Microsoft virtualized environment, you can take and restore checkpoints using either System Center Virtual Machine Manager (VMM) or Microsoft® Hyper-V Manager. The following sections describe how to implement backup strategies using SCVMM.

Taking Checkpoints Using SCVMM

By creating a checkpoint, you can save all contents of a virtual machine hard disk. You can reset your machine to a previous configuration if required, without having to uninstall programs or reinstall operating systems. This also helps you test applications across various configurations.

You can checkpoint one or more VMs both in the online and offline modes. However, you can checkpoint a VM only when it is deployed on a host.

Important: Typically, there are dependencies among nodes. Taking a checkpoint of a VM and restoring it later could negatively impact those dependencies. For more information, refer to "Checkpoints of System Platform Products - Observations and Recommendations".

Take a Checkpoint of an Offline VM

It is recommended that you shut down the virtual machine before creating a checkpoint. You can also create a checkpoint of the virtual machine offline. This stops the machine temporarily while the checkpoint is created. Turning off the virtual machine prevents loss of data while the conversion takes place. The general workflow for

offline VMs is as follows:

To take a checkpoint of an offline VM

1. Open the System Center Virtual Machine Manager (SCVMM).
2. Select the VM that you want to checkpoint.
3. Shut down the selected VM you selected.
4. Make a new checkpoint.
5. Verify the checkpoint.

Take a Checkpoint of an Online VM

You can create checkpoints of a virtual machine while it is running. However, creating a checkpoint in online mode requires special application support.

Important: To avoid losing any data, do not make any configuration changes to the machine while creating a checkpoint. For more information, refer to "Checkpoints of System Platform Products - Observations and Recommendations".

If you create a checkpoint after making configuration changes when the VM is online there may be issues when you restore the VM to that checkpoint.

For example, if you create a checkpoint for an online IOM Historian Product VM state and then try to restore it, the history block that is created shows a discrepancy in the start and end time and the following errors are displayed.

Warning: aahIndexSvc Attempted to create history block ending in the future

Error: aahIndexSvc ERROR: Invalid file format

To avoid such errors, stop the Historian VM before creating a checkpoint in the online mode. The general workflow for online VMs is as follows:

To take a checkpoint of an online VM

1. Open the System Center Virtual Machine Manager (SCVMM).
2. Select the VM that you want to checkpoint.
3. Make a new checkpoint.
4. Verify the checkpoint.

Restore Checkpoints

You can revert a virtual machine to a previous state by restoring it to the required checkpoint. When you restore a virtual machine to a checkpoint, VMM stops the virtual machine and the files on the virtual machine are restored to their previous state.

Important: If the virtual machine has been in use since the checkpoint was created, take a backup of the data files before you restore the virtual machine to avoid loss of any data.

Restore Checkpoints from a Virtual System Platform Backup

You can restore a VM to its previous state by using checkpoints. You can restore checkpoints of VMs both in the online and offline modes.

Restore a Checkpoint of an Offline VM

When you restore a VM to a checkpoint taken of an offline VM, there should not be any loss of data. When checkpoints are taken from a VM that is offline, the machine temporarily stops, minimizing data loss during the conversion process. The general workflow for an offline VM is as follows:

To restore a checkpoint of an offline VM

1. Open the System Center Virtual Machine Manager (SCVMM).
2. Select the offline VM for which you want to restore a checkpoint.
3. Restore the checkpoint.

Restore a Checkpoint of an Online VM

You can restore a VM to a checkpoint that was taken when the machine was online. Restoring a VM to a checkpoint taken while online may lead to loss of data. However, if no changes to the configuration were made while creating the checkpoint, there should not be any data loss. The general workflow for an online VM is as follows:

To restore a checkpoint of an online VM

1. Open the System Center Virtual Machine Manager (SCVMM).
2. Select the VM for which you want to restore a checkpoint.
3. Restore the checkpoint.

Take and Restore Checkpoints of Products with No Dependencies

You can create and restore checkpoints of IOM products that do not have dependencies. When you restore the VM to a checkpoint, data is restored up to the point at which you took the checkpoint. Data related to all changes made after the checkpoint was taken is not captured and will not be restored.

For example, on an Application Server node, two User Defined Objects (UDOs) are created at different points in time and checkpoints taken at each point. If you restore your VM to the first checkpoint, it will be restored to the state where only the first UDO was created. The second UDO created will not be backed up or restored in your system. The general workflow is as follows:

To take and restore checkpoints of products with no dependencies

1. Open the System Center Virtual Machine Manager (SCVMM).
2. Select the VM for which you want to create and restore a checkpoints.
3. Connect to the virtual machine.
4. In **Application Server** under **Platform**, **Engine**, and **Area**, create UDO1.

5. Use Virtual Machine Manager to select the VM.
6. Make a new checkpoint.
7. Connect to the virtual machine, if not already connected.
8. In **Application Server** under **Platform**, **Engine**, and **Area**, create UDO2.
9. Restore the VM.

Checkpoints of System Platform Products - Observations and Recommendations

The following are some of the observations and recommendations to take and restore checkpoints of System Platform Products.

- Take checkpoints of System Platform Products only when there are no configuration changes. For example, some of the scenarios where the checkpoints should not be taken are as follows:

System Platform Product	Configuration Changes
Application Server	deploy, migrate, import, export, check-in, check-out
Historian	import, export, create history block

- You must be aware of the consequences and make decisions when taking and restoring checkpoints of System Platform Products that have dependencies. If the configuration of a System Platform node has a dependency on the configuration of another System Platform node, it is recommended to take and restore checkpoints on such dependent nodes together. For more information, refer to "Recommendations".

Take and Restore Checkpoints (Snapshots) in the Offline Mode

It is recommended that you take checkpoints of System Platform Products when the VMs hosting them are in the offline mode. Turn off the System Platform Product VM before taking a checkpoint.

Restoring checkpoints of VMs in the offline mode result in smooth functioning of the System Platform Products after the restoration. After restoring a checkpoint, start the VM, and then start the System Platform Product hosted in the VM.

Take and Restore Checkpoints (Snapshots) in the Online Mode

While the VM is in the online mode, the System Platform Product hosted on the VM functions in one of the following ways:

- **Scenario 1**: If the System Platform Product is not running on an online VM, it functions smoothly after the restoration of checkpoints.
- **Scenario 2**: If a checkpoint is taken while the System Platform Product is running on an online VM and there are no configuration changes in progress, the System Platform Product performs normally. However, when

checkpoints are restored, there would be issues with the System Platform Product running on that VM. Some of the issues are explained in the following table.

Recommendations

Observation	Recommendations
Historian	
	• Do not take checkpoints while a history block change is in progress. Restoring such a checkpoint leads to unpredictable behavior of the product. • In case of communication issues between the Historian and dependent System Platform Products, restart the VMs. • If a checkpoint is taken before configuring Application Server to historize attributes, re-deploy the platform after the Historian is restored.
Issue 1: When you restore a checkpoint of the Historian node taken while the Historian was running and the block change was in progress, there is a conflict in the start and end time in the history block. The following errors and warnings are logged: Warning: aahIndexSvc Attempted to create history block ending in the future. Error: aahIndexSvc ERROR: Invalid file format.	As a recovery step of Issue 1, shut down and disable the Historian, and then start and enable it.
Issue 2: While creating a checkpoint there may be an action in progress resulting from an event. The incomplete action is not saved when you restore such a checkpoint.	
Application Server	

Observation	Recommendations
If checkpoints are restored on either GR node or remote IDE node, the configurations might go out of synchronization.	Perform galaxy object component synchronization (GOCS) after opening the IDE on the remote node.
Data Acquisition Server (DAS)	
If checkpoints are restored on DAS, there may be connectivity and configuration mismatch issues for the dependent System Platform Products.	Deactivate, and then activate the DAS with appropriate configuration file. If it does not resolve the connectivity issues, restart the dependent System Platform Product VMs.
InTouch	
If the AlarmDBLogger is configured on the local SQL Server, restoring checkpoints results in expected data loss.	If the alarm data is critical, configure the AlarmDBLogger on a remote SQL Server.
Information Server (WIS)	
If checkpoints are restored on WIS, there may be connectivity issues for the dependent System Platform Products.	Log off and re-launch the WIS browser.
Historian Client	
If checkpoints are restored on Historian Client, there may be connectivity issues to access the Historian.	Log off the server connection and log on to the Historian Client Applications.

Glossary

Application Engine (AppEngine)

A scan-based engine that hosts and executes the run-time logic contained within Automation Objects.

application object

An Automation Object that represents some element of your production environment. This can include things like:
An automation process component. For example, a thermocouple, pump, motor, valve, reactor, or tank
An associated application component. For example, function block, PID loop, sequential function chart, ladder logic program, batch phase, or SPC data sheet

Application Server

It is the supervisory control platform. Application Server uses our existing products, such as InTouch for

visualization, Historian for data storage, and the device integration product line like a Operations Integration Server (OI Server) for device communications.
An Application Server can be distributed across multiple computers as part of a single Galaxy namespace.

ArchestrA

The distributed architecture for supervisory control and manufacturing information systems. It is an open and extensible technology based on a distributed, object-based design.

child partition

Child partitions are made by the hypervisor in response to a request from the parent partition. There are a couple of key differences between a child partition and a parent/root partition. Child partitions are unable to create new partitions. Child partitions do not have direct access to devices (any attempt to interact with hardware directly is routed to the parent partition). Child partitions do not have direct access to memory. When a child partition tries to access memory the hypervisor / virtualization stack re-maps the request to different memory locations.

clone

A VM clone is an exact copy of a VM at a specific moment in time. The most common use of a VM clone is for mass deployment of standardized VMs, called VM templates. VM clones also come in handy for test and development; because they allow use of a real workload without affecting the production environment. A VM clone is not appropriate for backup, disaster recovery, or other data protection methods.

clustered file system

A clustered file system organizes files, stored data, and access for multiple servers in a cluster. Clustered file systems are most useful when clusters work together and require shared access, which individual file systems do not provide. A Windows or Linux clustered file system can also identify and isolate defective nodes in a cluster. A Windows clustered file system will isolate the node logically, while a Linux clustered file system will use a utility to power down the node.

compact

To reduce the size of a dynamically expanding virtual hard disk by removing unused space from the .vhd file. See also dynamically expanding virtual hard disk

differencing disk

A virtual hard disk that is associated with another virtual hard disk in a parent-child relationship. The differencing disk is the child and the associated virtual hard disk is the parent.

differencing virtual hard disk (diffdisk)

A virtual hard disk that stores the changes or "differences" to an associated parent virtual hard disk for the purpose of keeping the parent intact. The differencing disk is a separate .vhd file (that may be stored in a separate location) that is associated with the .vhd file of the parent disk. These disks are often referred to as "children" or "child" disks to distinguish them from the "parent" disk. There can be only one parent disk in a chain of differencing disks. There can be one or more child disks in a differencing disk chain of disks that are "related" to each other. Changes continue to accumulate in the differencing disk until it is merged to the parent disk. See also virtual hard disk. A common use for differencing disks is to manage storage space on a virtualization server. For example, you can create a base parent disk- such as a Windows 2012 Data Center base image - and use it as the foundation for all other guest virtual machines and disks that will be based on Windows Server 2012 Data Center edition.

dynamically expanding virtual hard disk (dynamic VHD, DVHD)

A virtual hard disk that grows in size each time it is modified. This type of virtual hard disk starts as a 3 KB .vhd file and can grow as large as the maximum size specified when the file was created. The only way to reduce the file size is to zero out the deleted data and then compact the virtual hard disk. See also virtual hard disk, VHD.

external virtual network

A virtual network that is configured to use a physical network adapter. These networks are used to connect virtual machines to external networks. See also internal virtual network, private virtual network.

failover

In server clusters, failover is the process of taking resource groups offline on one node and bringing them online on another node.

Fibre Channel

A high-speed network technology (commonly running at 2-, 4-, 8- and 16-gigabit speeds) primarily used for storage networking.

fragmentation

The scattering of parts of the same disk file over different areas of the disk.

guest operating system

This is the operating system/runtime environment that is present inside a partition. Historically with Virtual Server / Virtual PC, in a host operating system and a guest operating system where the host ran on the physical hardware and the guest ran on the host. In Hyper-V, all operating systems on the physical computer are running on top of the hypervisor so the correct equivalent terms are parent guest operating system and child guest operating system. Many find these terms confusing and instead use physical operating system and guest operating system to refer to parent and child guest operating systems, respectively.

guests and hosts

A guest virtual machine and host server are the two main building blocks of virtualization. The guest virtual machine is a file that contains a virtualized operating system and application, and the host server is the hardware on which it runs. The other important component is the hypervisor—the software that creates the guest virtual machine and lets it interact with the host server. The hypervisor also makes the host server run multiple guest virtual machines.

historical storage system (Historian)

The time series data storage system that compresses and stores high volumes of time series data for later retrieval. The standard Historian is the Historian.

Internet Small Computer Storage Interface (iSCSI)

Takes standard SCSI disk commands and, instead of executing them over a local SCSI connection, encapsulates them in TCP/IP packets for transmission over Ethernet. These low-level commands do not directly interact with files, but rather with arbitrary blocks of data on disk. The system reading and writing the data implements a file system on top of the ISCSI share, or LUN (logical unit number), to be able to read and write data for files. In the case of vSphere, this file system is called VMFS. Hyper-V also implements a file system on top of ISCSI LUNs. This allows the Hypervisor to implement a file system (like VMFS) that is optimized for the I/O needs of the hypervisor.

hypervisor

The hypervisor is to Hyper-V what the kernel is to Windows. The hypervisor is the lowest level component that is responsible for interaction with core hardware. It is responsible for creating, managing, and destroying partitions. It directly controls access to processor resource and enforces an externally-delivered policy on memory and device access. The hypervisor is just over 100k in size and the entire Hyper-V role is around 100mb in size. A full installation of Windows Server 2012 with Hyper-V will be multiple gigabytes in size. After you have installed the Hyper-V role, the hypervisor is loaded as a boot critical device.

live migration

Virtual machine live migration is the process of moving a VM from one host server to another without shutting

down the application. The benefits of virtual machine live migration are some of the biggest selling points for virtualization, affecting business continuity, disaster recovery, and server consolidation. Virtual machine live migration is a feature in all of the major virtualization platforms, including VMware vSphere, Microsoft Hyper-V R2, and Citrix Systems XenServer.

logical processor

This is a single execution pipeline on the physical processor.Earlier, if someone told you that they had a two-processor system, you would know exactly what they had. Today, if someone told you they had a two-processor system, you do not know how many cores each processor has, or if hyperthreading is present. A two-processor computer with hyperthreading would actually have four execution pipelines, or four logical processors. A two-processor computer with quad-core processors would, in turn, have eight logical processors.

management operating system

The operating system that was originally installed on the physical machine when the Hyper-V role was enabled. After installing the Hyper-V role, this operating system is moved into the parent partition. The management operating system automatically launches when you reboot the physical machine. The management operating system actually runs in a special kind of virtual machine that can create and manage the virtual machines that are used to run workloads and/or different operating systems. These virtual machines are sometimes also called child partitions. The management operating system provides management access to the virtual machines and an execution environment for the Hyper-V services. The management operating system also provides the virtual machines with access to the hardware resources it owns.

memory overcommit

A hypervisor can let a guest VM use more memory space than that available in the host server. This feature is called memory overcommit. Memory overcommit is possible because most VMs use only a little bit of their allocated physical memory. That frees up memory for the few VMs that need more. Hypervisors with memory overcommit features can identify unused memory and reallocate it to more memory-intensive VMs as needed.

Network-Attached Storage (NAS)

Network-attached storage (NAS), in contrast to SAN, uses file-based protocols such as NFS or SMB / CIFS where it is clear that the storage is remote, and computers request a portion of an abstract file rather than a disk block.

Network File System (NFS)

A file system originally created by Sun Microsystems as a way to allow multiple clients to access files on a central network storage device. When the Hypervisor accesses data on an NFS share, it accesses the files directly because the protocol itself provides the file system.

Network Load Balancing (NLB)

A Windows network component that uses a distributed algorithm to load-balance IP traffic across a number of hosts, helping to enhance the scalability and availability of mission-critical, IP-based services.

network virtualization

Network virtualization lets you combine multiple networks into one, divide one network into many and even create software-only networks between VMs. The basis of network virtualization is virtual network software, to which there are two approaches: internal and external. Internal network virtualization uses virtual network software to emulate network connectivity among VMs inside a host server. External network virtualization virtual network software to consolidate multiple physical networks or create several virtual networks out of one physical network.

NTFS

An advanced file system that provides performance, security, reliability, and advanced features that are not found in any version of the file allocation table (FAT).

parent partition

The parent partition can call hypervisor and request for new partitions to be created. There can only be one parent partition. In the first release of Hyper-V, the parent and root partitions are one and the same.

partition

A partition is the basic entity that is managed by the hypervisor. It is an abstract container that consists of isolated processor and memory resources with policies on device access. A partition is a lighter weight concept than a virtual machine and could be used outside the context of virtual machines to provide a highly isolated execution environment.

physical computer

The computer, or more specifically, the hardware that is running the Hyper-V role.

physical processor

It is the squarish chip that you put in your computer to make it run. This is sometimes also referred to as a "package" or a "socket".

private virtual network

A virtual network without a virtual network adapter in the management operating system. It allows communication only between virtual machines on the same physical server.

processor topology

This is the concept by which your logical processors correlate to your physical processors. For example, a two processor, quad-core system and a four-processor dual-core system both have eight logical processors but they have different processor topologies.

P2V

A physical-to-virtual server migration, also known as a P2V server migration, is the process of converting a physical workload into a VM. To perform a physical-to-virtual server migration, copy bits from the physical disk to the VM, inject drivers, then modify other bits to support the drivers. Some operating systems and virtual server migration tools let you perform a P2V server migration while the host is running, but others require a shutdown.

release key combination

The key combination (CTRL+ALT+LEFT ARROW by default) that must be pressed to move keyboard and mouse focus from a guest operating system back to the physical computer.

root partition

This is the first partition on the computer. This is the partition that is responsible for starting the hypervisor. It is also the only partition that has direct access to memory and devices.

saved state

A manner of storing a virtual machine so that it can be quickly resumed (similar to a hibernated laptop). When you place a running virtual machine in a saved state, Virtual Server and Hyper-V stop the virtual machine, write the data that exists in memory to temporary files, and stop the consumption of system resources. Restoring a virtual machine from a saved state returns it to the same condition it was in when its state was saved.

small computer system interface (SCSI)

A standard high-speed parallel interface used for connecting microcomputers to peripheral devices, such as hard disks and printers, and to other computers and local area networks (LANs).

snapshot

A VM snapshot backup is the most common way to protect a virtual machine. A VM snapshot is a copy of the state of a VM (and any virtual disks assigned to it) as it exists in server memory at a specific moment. The snapshot is usually saved to the SAN, where it can be recovered in case of a failure. Regular VM snapshot

backups can significantly reduce recovery point objectives.

storage area network (SAN)

A set of interconnected devices, such as disks and tapes, and servers that are connected to a common communication and data transfer infrastructure, such as Fibre Channel.

storage array

A disk storage system which contains multiple disk drives. It is differentiated from a disk enclosure in that an array has cache memory and advanced functionality, like RAID and virtualization.

storage virtualization

Storage virtualization separates the operating system from physical disks used for storage, making the storage location independent. The benefits of storage virtualization include more efficient storage use and better management. Dynamic provisioning is similar to storage virtualization, but it still requires more traditional storage management.

system center virtual machine manager (SCVMM)

A centralized management console that helps you manage and administer a virtual environment.

vfd or virtual floppy disk

The file format for a virtual floppy disk. See also virtual floppy disk.

vhd or virtual hard disk

The file format for a virtual hard disk, the storage medium for a virtual machine. It can reside on any storage topology that the management operating system can access, including external devices, storage area networks, and network-attached storage.

virtual hardware

The computing resources that the host server assigns to a guest VM make up the virtual hardware platform. The hypervisor controls the virtual hardware platform and allows the VM to run on any host server, regardless of the physical hardware. The virtual hardware platform includes memory, processor cores, optical drives, network adapters, I/O ports, a disk controller and virtual hard disks. Virtualization lets a user adjust the levels of these resources on each VM as needed.

virtual machine

A virtual machine (VM) is a file that includes an application and an underlying operating system combines with a physical host server and a hypervisor to make server virtualization possible. A virtual machine is a super-set of a child partition. A virtual machine is a child partition combined with virtualization stack components that provide functionality, such as access to emulated devices, and features like being able to save state a virtual machine. As a virtual machine is essentially a specialized partition, the terms "partition" and "virtual machine" is often used interchangeably. But, while a virtual machine will always have a partition associated with it, a partition may not always be a virtual machine.

virtual machine bus

A communications line used in Hyper-V by virtual machines and certain types of virtual devices. The virtual devices that use virtual machine bus have been optimized for use in virtual machines.

virtual machine configuration

The configuration of the resources assigned to a virtual machine. Examples include devices such as disks and network adapters, as well as memory and processors.

Virtual machine connection

A Hyper-V management tool that allows a running virtual machine to be managed through an interactive session.

virtual machine management service

The SCVMM service that provides management access to virtual machines.

virtual machine monitoring

Virtual machine monitoring actually means virtual machine performance monitoring. Virtual machine performance monitoring tools keep tabs on the state of VMs in an environment. Though it is possible to monitor the VM performance from within, but it's recommended to monitor it from outside the VM.

virtual machine snapshot

A virtual machine snapshot is a point in time image of a virtual machine that includes its disk, memory and device state at the time that the snapshot was taken. At any time can be used to return a virtual machine to a specific moment in time, at any time. Virtual machine snapshots can be taken irrespective of the state or type of child guest operating system being used.

virtual network

A virtual version of a physical network switch. A virtual network can be configured to provide access to local or external network resources for one or more virtual machines.

virtual network manager

The Hyper-V component used to create and manage virtual networks.

virtualization server

A physical computer with the Hyper-V role installed. This server contains the management operating system and it provides the environment for creating and running virtual machines. Sometimes referred to as a server running Hyper-V.

virtualization stack

The virtualization stack is everything else that makes up Hyper-V. This is the user interface, management services, virtual machine processes, emulated devices.

virtual processor

A virtual processor is a single logical processor that is exposed to a partition by the hypervisor. Virtual processors can be mapped to any of the available logical processors in the physical computer and are scheduled by the hypervisor to allow you to have more virtual processors than you have logical processors.

virtual switch

A virtual switch is the key to network virtualization. It connects physical switches to VMs through physical network interface cards and ports. A virtual switch is similar to a virtual bridge, which many virtualization platforms use, but it is more advanced. Virtual LANs, EtherChannel and additional virtual networking tools are only available in a virtual switch. Some virtual switches even offer their own security features.

virtualization WMI provider

The WMI provider for virtualization that can be used with the hypervisor API to enable developers and scripters to build custom tools, utilities, and enhancements for the virtualization platform.

VMDK

The Virtual Machine Disk (VMDK) file format is used to identify VMware virtual machines. (In virtualization, the hypervisor creates a VM file that consists of an operating system instance, an application and other associated components.) Other platforms that support the VMDK file format include Sun Microsystems xVM, Oracle VirtualBox, and QEMU. It competes with Microsoft's Virtual Hard Disk format, which is used in Virtual Server and Hyper-V.

21 CFR Part 11

This section describes how AVEVA System Platform and its software components adhere to the 21 CFR Part 11 requirements of the U.S. Food and Drug Administration (FDA).

About This Guide

This 21 CFR Part 11 Deployment Guide provides information about features relevant to the 21 CFR Part 11 requirements of the U.S. Food and Drug Administration (FDA) for the following products:

- AVEVA™ System Platform
- AVEVA™ Application Server
- AVEVA™ Operations Management Interface (OMI)
- AVEVA™ InTouch HMI
- AVEVA™ Historian

This document helps customers reduce the cost and time of application development. In going a step further, it provides customers from Food and Drug Administration (FDA) audited industries with a set of best practices in regards to the 21 CFR Part 11 requirements.

While not directly subject to regulation under 21 CFR Part 11, System Platform, InTouch HMI, AVEVA OMI, and Historian products incorporate features and functionality designed to facilitate the development of applications for use in FDA-regulated industries. Accordingly, this document to provides customers with a set of "best practices" in regards to certain products and the 21 CFR Part 11 requirements.

Note: The methods described in this document represent general guidance and may require adaptation or modification depending on the needs of your specific system implementation. For optimum results, before applying the advice contained in this guide, consult our systems integrator.

Auditing and security functions are tightly integrated with Microsoft products, and working knowledge of both Microsoft SQL Server and the Microsoft Windows operating system is required. It is assumed that you are familiar with administering a Microsoft SQL Server and using the administrative tools provided with the Microsoft Windows Server or Advanced Server operating system.

For more information on Microsoft SQL Server or the Microsoft Windows operating system, see your Microsoft documentation.

System Platform, InTouch HMI, AVEVA OMI, and Historian are AVEVA products for human-machine interface (HMI) software plus a plant historian. They are based on Microsoft Windows and can be used to control and monitor processes in FDA-audited industries. Historian is closely linked to Microsoft SQL Server. System Platform, InTouch, Historian, and procedural controls can be used to implement systems that comply with the FDA's 21 CFR Part 11 regulation.

This deployment guide is designed for closed systems. Closed systems are defined as systems where access is controlled by the people responsible for the content of the electronic records. Open systems are not addressed in this document.

References and Documentation

User guides, Readme files, and other publications and help systems are available on the System Platform installation media. The documentation is also available for download or as online help files through the respective application interface or from the AVEVA Knowledge and Support Center (a valid login is required):

https://softwaresupportsp.aveva.com/#/producthub?selectedTab=Documents

The following tables list the references and documentation available for System Platform products relevant to the 21 CFR Part 11 regulations.

Application Server and AVEVA OMI

The Application Server and AVEVA OMI documentation set includes the following guides:

Publication Name (file name)	Description
AVEVA System Platform Readme Readme.html	Includes descriptions of new product features introduced in System Platform, installation requirements, and any known issues.
Application Server User Guide (IDE.pdf)	Explains configuring and deploying Application Server and AVEVA OMI applications.
Industrial Graphic Editor User Guide (IndustrialGraphics.pdf)	Explains how you create and manage graphical symbols using the Industrial Graphic Editor within the Integrated Development Environment (IDE).
Application Server Scripting Guide (Scripting.pdf)	Reference for the Application Server scripting language.
AVEVA System Platform Help (NGX\index.htm)	Web-based help that provides information for Application Server and AVEVA OMI.
AVEVA OMI SDK Help (OMISDKHelp\index.html)	Web-based help that provides information for using the AVEVA OMI software developer kit to create AVEVA OMI apps.
Object Viewer User Guide (ObjectViewer.pdf)	Explains how to acquire run-time data using the Object Viewer.
AVEVA Alarm Client Control User Guide (AlarmClientControl.pdf)	Explains how to configure the alarm control (client) to show current and historical alarms and events in a grid.

Publication Name (file name)	Description
AVEVA Trend Client Control User Guide (TrendClient.pdf)	Explains how to configure a chart to trend real-time data values.
AVEVA Platform Manager User Guide (PlatformManager.pdf)	Explains how to start and stop system components.
AVEVA Galaxy Database Manager User Guide (galaxymanagement.pdf)	Explains how to backup and restore the Galaxy database.
Operations Control Logger (index.htm)	Web help containing the following components: The *Log Viewer User Guide e*xplains how to use the Log Viewer utility to determine system diagnostics. *Log Flag Editor Utility Guide e*xplains how to turn on and off certain diagnostics logging messages. *Log Monitor* explains how to use the monitor utility.
AVEVA Enterprise Licensing Help (index.htm)	Explains how to use the AVEVA Enterprise License Manager to manage the licenses required for System Platform.
AVEVA Protocols User Guide (Protocol.pdf)	Explains background information on the main protocols used between components of our products.
AVEVA GRAccess Toolkit API User's Guide (GRAccess.pdf)	Explains how to use the GRAccess programmable object model to configure an ArchestrA Galaxy in .NET and COM.

InTouch HMI

The InTouch documentation set includes the following guides:

Publication Name (file name)	Description
AVEVA™ InTouch HMI Creating Standards for InTouch HMI Components Guide ITStandards.pdf	Contains information on creating standards for various InTouch HMI components, and describes how to prepare the development environment, view the application in run time, and gain an understanding on how to work with tags, alarms, and data items in the InTouch HMI to connect your application to the physical devices in your plant environment.
AVEVA™ InTouch HMI Application Development Guide ITBuild.pdf	Contains information on creating and managing InTouch HMI applications locally and in a network environment, including: how to how to create visualization windows, how to draw and animate graphic elements, and how to use wizards and ActiveX controls in your application. This guide also includes a reference of the InTouch HMI scripting language and functions, along with details on working with Industrial Graphics in the Cloud.
AVEVA™ InTouch HMI Application Deployment Guide ITDeploy.pdf	Contains information on deploying InTouch HMI applications to work with terminal and remote desktop services, and describes how to configure InTouch HMI Network Application Development and use Managed applications at run time.
AVEVA™ InTouch HMI Application Run Time Guide ITOperate.pdf	Contains information on using WindowViewer and Web Client to view the InTouch HMI applications in run time, and also describes viewing application graphics in a web browser, with a focus on language switching, tag viewer, and various alarm components.
AVEVA™ InTouch HMI Application Maintenance Guide ITMaintain.pdf	Contains information on migrating and upgrading InTouch applications, components, and alarms, and also describes how to set up an InTouch HMI application on tablet PC or multi monitors.
AVEVA™ InTouch HMI Troubleshooting Guide ITDiagnose.pdf	Contains troubleshooting information to understand error messages and resolve issues with the InTouch HMI application and Web Client.
AVEVA™ InTouch HMI Management Guide ITManage.pdf	Contains licensing information for the InTouch HMI application and Web Client, as well as generic and InTouch-specific security configurations, and also describes using other supplementary components.

Historian

The Historian documentation set includes the following guides:

Publication Name (file name)	Description
Historian Administration Guide (HistorianAdmin.pdf)	This guide describes how to administer and maintain an installed Historian, such as configuring data acquisition and storage, managing security, and monitoring the system.
Historian Concepts Guide (HistorianConcepts.pdf)	This guide provides an overview of the entire Historian system and describes each of the subsystems in detail.
Historian Database Reference (HistorianDatabase.pdf)	This guide provides documentation for all of the Historian database entities, such as tables, views, and stored procedures.
Historian Glossary (HistorianGlossary.pdf)	This guide provides definitions for terms used throughout the documentation set.
Historian Retrieval Guide (HistorianRetrieval.pdf)	This guide describes how to retrieve data store in a Historian server using Transact-SQL queries, Historian Client tools, Historian Insight, and the Historian SDK.
Historian Scenarios Guide (HistorianScenarios.pdf)	This guide discusses how to use Historian to address some common customer scenarios.

Notes on System Architecture Options

Our software components can be used to design industrial IT systems using several architectures, according to the exact requirements of a customer. The choice of architecture can impact the best practices for designing an application to be validated.

The System Platform is a suite of products that provides a powerful and common framework for building industrial applications. System Platform integrates security, data quality, communications, and alarming within an infrastructure of common services.

System Platform includes the following products:

- InTouch HMI can be used as a visualization client of System Platform. InTouch HMI can be used to create managed applications that integrate with Application Server and stand-alone applications that support Industrial Graphics without requiring a Galaxy.
- AVEVA OMI is an advanced visualization client of System Platform. It is included with Application Server and leverages the power of Industrial Graphics, built-in and custom apps, and supports multiple windows.

- AVEVA Historian is a component of System Platform, and can also be used in combination with InTouch tag-based applications or in a stand-alone mode, for example when different HMI system(s) are already in place. Historian (formerly known as IndustrialSQL Server) for recording of historical data values
- Insight is an online information portal to collect, store, visualize, and analyze industrial data. Insight consolidates disparate data for complete visibility into how your business is performing and enables users to access data and information from anywhere.

Technological Control describes our product features, according to the different architectural options, that support 21 CFR Part 11 compliance.

Other References & Documentation

21 CFR Part 11

The following publications and web resources provide information about 21 CFR Part 11:

Electronic Records; Electronic Signatures Final Rule, 62 Federal Register 13430 (March 20, 1997)	This Code of Federal Regulation is the official rule on Electronic Records and Electronic Signatures management for FDA-audited industries.
Use of Electronic Records and Electronic Signatures in Clinical Investigations Under Part 11-Questions and Answers; Draft Guidance for Industry; Availability (*June 21, 2017*)	This draft document provides guidance to sponsors, clinical investigators, IRBs, CROs, and other interested parties on the use of electronic records and electronic signatures under part 11 in clinical investigations of medical products.
Risk-Based Approach to 21 CFR Part 11 (August 2003)	This ISPE White Paper describes how a risk-based approach to Part 11 could be used to benefit patient health while adversely impacting industry productivity.
Guidance for Industry, Part 11, Electronic Records; Electronic Signatures - Scope and Application (September 2003)	This guidance is intended to describe the Food and Drug Administration's (FDA's) current thinking regarding the scope and application of part 11 of Title 21 of the Code of Federal Regulations; Electronic Records; Electronic Signatures (21 CFR Part 11).

Validation

ISPE GAMP® 5: A Risk-Based Approach to Compliant GxP Computerized Systems (Second Edition)	This document is one guideline, used widely within FDA-regulated industries, for validation of computer systems. ISPE and the GAMP Forum produce the GAMP Guide. http://www.ispe.org/gamp/
General Principles of Software Validation; Final Guidance for Industry and FDA Staff (January 11, 2002)	This guidance presents principles of software validation considered to be applicable by the FDA. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation
Current Good Manufacturing Practice (CGMP) Regulations November 16, 2022	This summary describes regulations in 21 CFR related to Part 11. https://www.fda.gov/drugs/pharmaceutical-quality-resources/current-good-manufacturing-practice-cgmp-regulations

FDA publications can be downloaded from the United States Food and Drug Administration web site at http://www.fda.gov/.

Guides developed by the International Society for Pharmaceutical Engineering (ISPE) can be purchased through ISPE at http://www.ispe.org/.

The 21 CFR Part 11 Regulation

Developments in documentation technology, specifically electronic records and electronic signatures, offered companies advantages over paper-based documentation. Companies in regulated industries sought to use these electronic record and electronic signature capabilities to satisfy regulatory requirements.

The United States government responded by updating the Code of Federal Regulations (CFR) with guidance in the form of regulation 21 CFR Part 11, governing the use of electronic records and electronic signatures needed or used to satisfy FDA requirements.

Overview of Part 11

The Part 11 regulation contains three major divisions: Subpart A - General Provisions, Subpart B - Electronic Records, and Subpart C - Electronic Signatures. The outline of the regulation is as follows:

Subpart A - General Provisions

- 11.1 Scope
- 11.2 Implementation

- 11.3 Definitions

Subpart B - Electronic Records

- 11.10 Controls for closed systems
- 11.30 Controls for open systems

  Section 11.30 is outside the scope of this guide.
- 11.50 Signature manifestation
- 11.70 Signature/record linking

Subpart C - Electronic Signatures

- 11.100 General requirements
- 11.200 Electronic signature components and controls
- 11.300 Controls for identification codes/passwords

Subpart A, General Provisions, define what electronic records and electronic signatures must comply with this regulation. The records subject to Part 11 are those in electronic form "*created, modified, maintained, archived, retrieved, or transmitted, under any records requirements set forth in agency regulations.*" [1] The regulation also applies to any electronic records submitted to the FDA even if the record is not specifically identified in the FDA regulations. Any signatures applied electronically to such records must also comply with the Part 11 regulations.

The General Provisions also definitively state that electronic records and electronic signatures in compliance with this regulation will be considered the equivalent of paper records and handwritten signatures applied to paper.

Subpart B—Electronic Records

Controls for Closed Systems (11.10)

The regulation calls for a series of controls to ensure the authenticity, integrity, and confidentiality (when necessary) of electronic records in closed systems. The scope of this deployment guide is limited to closed systems. Controls for open systems (21 CFR 11.30) will not be addressed in this guide.

The controls for closed systems are summarized here but addressed in greater detail in chapters 3 and 4 of this deployment guide:

- 11.10 (a): Systems must be validated (tested to verify they operate as designed)
- 11.10 (b): Records must be available for inspection in both electronic and human readable form
- 11.10 (c): Records must be accessible for retrieval during the required retention period
- 11.10 (d): System access is limited to authorized individuals
- 11.10 (e): Operator entries and actions that create, modify, or delete electronic records must be tracked in a secure, computer-generated audit trail
- 11.10 (f): System checks will enforce sequencing of steps or events
- 11.10 (g): Authority checks will be used to ensure system use or electronic signatures only by authorized individuals

- 11.10 (h): Device checks will determine validity of inputs or operational instructions
- 11.10 (i): System users have the necessary education, training, and experience for their tasks
- 11.10 (j): Written policies that hold individuals accountable for actions initiated by their electronic signatures
- 11.10 (k): Controls over system documentation including access to and changes therein

Signature Manifestation (11.50)

Electronic records must include information associated with each electronic signature applied to the record. This information must be controlled to the same degree as the electronic records and all aspects of the signature will be included in the human readable form of the electronic record.

The required signature information includes:

- Printed name of the signer
- Date and time when the signature was applied
- Assigned role of the person applying the signature (e.g. author, reviewer, approver)

Signature/Record Linking (11.70)

Signatures, electronic or handwritten, applied to the electronic records must be linked to the records to prevent them from being removed or changed in any way that could be used to falsify records.

Note: This guide does not address the application of handwritten signatures to electronic records.

Subpart C—Electronic Signatures

General Requirements (11.100)

There are a number of electronic signature requirements a system must meet to be Part 11 compliant. These requirements are intended to provide evidence and confidence the electronic signatures in the system can be considered the equivalent of handwritten signatures. The general requirements are:

- 11.100 (a): Electronic signatures must be unique to an individual
- 11.100 (b): Organizations must verify an individual's identity before the individual can use electronic signatures
- 11.100 (c): Persons using electronic signatures must certify to the FDA their electronic signatures are intended to be the legal equivalent of their handwritten signature

Electronic Signature Components and Controls (11.200)

The implementation of electronic signatures can be accomplished through biometrics or other means. Specific controls are required on the signature mechanism depending on the method used. Those controls are:

11.200 (a): Non-biometric signatures.

1. Use at least two different identification components (e.g. user ID and password)

   (i): Multiple signatures applied by an individual in a continuous session require all electronic signature

components for the first signature and only one component for subsequent signatures

(ii): Multiple signatures applied by an individual but not in a continuous session require all signature components for each signature

2. Must be used only by their genuine users
3. User administration must be designed to require collaboration of two or more individuals to use another user's electronic signature

11.200 (b): Biometric signatures must be designed so they can only be performed by their genuine owner.

Controls for Identification Codes/Passwords (11.300)

Systems using a combination of identification code (e.g. user ID) and password as the electronic signature components must ensure the integrity of these signatures through a series of controls.

11.300 (a): Maintain user ID and password combinations so no two individuals can have the same combination.

11.300 (b): Codes and passwords are periodically checked or revised.

11.300 (c): Lost or potentially compromised identification devices (e.g. tokens, cards) or passwords are voided and replaced with a new equivalent.

11.300 (d): Transaction safeguards are used to prevent unauthorized use of IDs or passwords.

11.300 (e): ID or password generating devices (e.g. tokens) must be tested initially and periodically to ensure they are unaltered and function properly.

Revised Guidance

The release of Part 11 by the FDA was intended to permit the extensive use of electronic technology in a manner consistent with the FDA's need to protect public health. The result was a significant amount of discussion within the industry requiring the subsequent release of a compliance policy guide and draft guidance for the following: validation, glossary of terms, time stamps, maintenance of electronic records, and electronic copies of electronic records.

While the intent was to produce a wide use of technology, concerns about Part 11 developed within the industry to the point where the regulation was perceived as having the opposite effect (see ISPE White Paper for examples).

In February of 2003, the FDA issued draft Part 11 guidance with a final guidance following in August 2003. In this final guidance, the FDA presented the group's intention to limit the scope and application of Part 11. Specific industry concerns about Part 11 were noted in this final guidance:

- Unnecessarily restricts the use of electronic technology inconsistent with the regulations intent
- Can significantly increase the costs of compliance due to the Part 11 requirements
- Discourages innovation and technological advances without providing significant public health benefit

The revised guidance is intended by the FDA to address these industry concerns as the result of the original Part 11 regulation was the opposite of the intent. The emphasis within this revised guidance is clear:

- FDA is re-evaluating Part 11 as it applies to FDA regulated products
- Part 11 remains in effect

- The FDA will narrowly interpret the scope of Part 11
  - Fewer records will be subject to Part 11
  - Part 11 will apply to:
    - Records required to be maintained by predicate rules) that are in electronic format in place of paper
    - Records required to be maintained by predicate rule(s) that are in electronic format in addition to paper format and are relied upon to perform regulated activities
    - Records submitted to FDA under predicate rules
    - Electronic signatures that are the equivalent to handwritten signatures required by predicate rule (e.g. reviewed, approved, verified)
- Enforcement discretion will be used during the period of re-examination
  - FDA does not intend to take enforcement action to enforce compliance with the validation, audit trail, record retention, and record copying requirements of Part 11 as explained in the final guidance
  - Records must still be maintained or submitted consistent with the applicable predicate rules
- No requirements will be enforced for systems that were operational before August 20, 1997, the effective date of Part 11

The enforcement discretion is strictly limited to the Part 11 requirements but some of these aspects of a system may still apply to satisfy the predicate rules. Regarding the areas identified for enforcement discretion, Part 11 should be viewed as not adding to or increasing the regulatory requirements defined in other regulations. Part 11 does not remove or invalidate existing requirements defined in other regulations. For example, system validation is still required for some systems by 21 CFR 820.70(i).

As of March, 2023, the FDA added paragraph (p) to "Subpart A-General Provisions 11.1, Scope", to add a limitation on electronic records required relative to Subpart R Part 1.

Complying with Part 11

The first step to compliance with Part 11 is to determine if Part 11 applies to the system in question and if it does, which parts of Part 11 apply.

For example, an electronic records system that does not include electronic signatures does not need to comply with Subpart C. Each company needs to make a determination for each new system based on their understanding and application of Part 11. The ISPE and PDA guide also provides guidance for understanding and complying with Part 11. Whatever decision is made, this determination should be clearly documented and consistent with a company's standard procedures related to regulatory requirements.

Once a system is found to require Part 11 compliance, the company needs to determine how to comply with the applicable requirements. This requires a mixed solution of two types of controls: technological and procedural. Some specific requirements may even be addressed by both types of controls.

The presence of procedural requirements means that no technological solution, software, etc. can be compliant with the Part 11 regulation as it exists on its own. This guide will focus on the application of technological controls, but will also identify where procedural controls are required.

Compliance Matrix

The following table identifies which type of control is needed to comply with specific requirements defined in

the Part 11 regulation. An 'X' in the Procedural or Technological column indicates the control applies for the requirement listed in that row. Sections 3 and 4 of this guide are structured to mirror the regulation and present the reader with information and specific technical methods or options available to support Part 11 compliance. In each subsection within Sections 3 and 4, the specific regulation text being addressed is presented to aid in the interpretation and application of this guide (shown in bold italics).

21 CFR 11 Requirement	**Procedural**	**Technological**
Subpart B 11.10	X	X
Subpart B 11.10 (a)*	X	X
Subpart B 11.10 (b)*		X
Subpart B 11.10 (c)*	X	X
Subpart B 11.10 (d)	X	X
Subpart B 11.10 (e)*	X	X
Subpart B 11.10 (f)		X
Subpart B 11.10 (g)	X	X
Subpart B 11.10 (h)		X
Subpart B 11.10 (i)	X	
Subpart B 11.10 (j)	X	
Subpart B 11.10 (k)	X	X
Subpart B 11.10 (k) 1	X	X
Subpart B 11.10 (k) 2*	X	X
Subpart B 11.30*	N/A**	N/A**
Subpart B 11.50 (a)		X
Subpart B 11.50 (a) 1		X
Subpart B 11.50 (a) 2		X
Subpart B 11.50 (a) 3		X
Subpart B 11.50 (b)		X
Subpart B 11.70		X
Subpart C 11.100 (a)	X	X

21 CFR 11 Requirement	Procedural	Technological
Subpart C 11.100 (b)	X	
Subpart C 11.100 (c)	X	
Subpart C 11.100 (c) 1	X	
Subpart C 11.100 (c) 2	X	
Subpart C 11.200 (a)	X	X
Subpart C 11.200 (a) 1	X	X
Subpart C 11.200 (a) 1.i	X	X
Subpart C 11.200 (a) 1.ii	X	X
Subpart C 11.200 (a) 2	X	
Subpart C 11.200 (a) 3	X	X
Subpart C 11.200 (b)	N/A***	N/A***
Subpart C 11.300	X	X
Subpart C 11.300 (a)	X	X
Subpart C 11.300 (b)	X	X
Subpart C 11.300 (c)	X	
Subpart C 11.300 (d)	X	X
Subpart C 11.300 (e)	X	

* These sections are part of the enforcement discretion defined in the Part 11 Guidance for Industry

** Open systems are outside the scope of this guide

*** Biometric signatures are outside the scope of this guide

Procedural Controls

Procedural controls must be applied as part of any Part 11 solution. Specific procedural controls are outside the scope of our products and offerings to the industry. But, this guide addresses the procedural controls and provides some content to aid a company's efforts to understand and apply the necessary procedural controls.

Electronic Records—Subpart B

Controls for Closed Systems—11.10

"Persons who use closed systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, when appropriate, the confidentiality of electronic records, and to ensure that the signer cannot readily repudiate the signed record as not genuine. Such procedures and controls shall include the following:"

Companies need to define and execute a system with components (e.g. standard operating procedures, processes, tools) that will ensure the closed system controls are properly established and maintained. Periodic verification, or auditing, of the controls should be performed to maintain the integrity of the controls, once established.

Validation—11.10 (a)

"(a) Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records."

Validation is one of the requirements where enforcement discretion will be applied. In this area, that means validation of electronic record/electronic signature (ER/ES) systems will not face any new requirements due to the Part 11 record. System validation must still be performed in compliance with the predicate rule (21 CFR 820.70(i)).

This validation should follow a defined methodology. The GAMP5 guide recommends important validation principles for companies to consider using. Significant procedural activities to perform include defining requirements, documenting system design, and testing that the system performs as defined by the design - including documenting this testing or verification.

Record Protection—11.10 (c)

"(c) Protection of records to enable their accurate and ready retrieval throughout the records retention period."

The following procedural actions should be performed to protect and enable record retrieval:

- Define procedures for providing records to internal and external parties
- Specify retention requirements - specifically the retention period
- Define backup, recovery, archival, and retrieval processes for electronic records

This is another area where enforcement discretion will be applied but this discretion is specifically limited to generating copies of records. Electronic records should be available at a company's facility using that company's defined tools and methods or the records should be made available as copies in some common format (e.g. PDF, XML).

Access Limitations—11.10 (d)

"(d) Limiting system access to authorized individuals."

Procedures need to be defined that address system user administration, including who should be granted access and how that access is granted. Often systems require administrator level or otherwise high level users with great latitude in the actions they can perform within the system. These high level users should get special attention when considering rules, limitations, and other procedural safeguards that can be applied.

Audit Trail—11.10 (e)

"(e) Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying."

Reliability of records must still be maintained even though an audit trail may not be specifically required due to the FDA's enforcement discretion defined in the revised guidance. Procedures should be established that require documentation of the methods for ensuring reliable records, whether this includes an audit trail or not.

Authority Checks—11.10 (g)

"(g) Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand."

Procedures need to define how systems should perform authority checks. If any variation in authorization method is allowed, the specific scenarios and authorization methods for each must be specifically defined.

This requirement is related to the access limitation requirement (section 3.1.1.3). However, this requirement addresses allowing specific actions within the system whereas the access limitation requirement relates to general access to the system.

User Qualifications—11.10 (i)

"(i) Determination that persons who develop, maintain, or use electronic record/electronic signature systems have the education, training, and experience to perform their assigned tasks."

The method or methods used to determine persons who are qualified to interact with ER/ES systems must be defined.

Accountability—11.10 (j)

"(j) The establishment of, and adherence to, written policies that hold individuals accountable and responsible for actions initiated under their electronic signatures, in order to deter record and signature falsification."

Policies need to define the responsibilities of those users with electronic signature capabilities. This should include any consequences or other deterrents for misuse of the electronic signature function by those users. This requirement is intended to ensure electronic records can be trusted as signed.

Documentation Control—11.10 (k)

"(k) Use of appropriate controls over systems documentation including:

(1) Adequate controls over the distribution of, access to, and use of documentation for system operation and maintenance.

(2) Revision and change control procedures to maintain an audit trail that documents time-sequenced development and modification of systems documentation."

The controls placed on system documentation must be defined by procedures. Requirements for recording and tracking system documentation changes must also be defined.

Electronic Signatures—Subpart C

General Requirements—11.100

Signature Uniqueness—11.100 (a)

"(a) Each electronic signature shall be unique to one individual and shall not be reused by, or reassigned to, anyone else."

Procedures should be established to ensure an electronic signature is unique and can only be used by one individual Companies should also be prepared, with defined procedures, to handle situations where signature authorities are not available because no others can execute a signature on their behalf. A valid remedy is to establish rules for delegation of signature responsibility so work flow can progress even if primary signature authorities are not available to execute an electronic signature.

User Identity—11.100 (b)

"(b) Before an organization establishes, assigns, certifies, or otherwise sanctions an individual's electronic signature, or any element of such electronic signature, the organization shall verify the identity of the individual."

Verification of an individual's identity is a procedural activity. The process or methods of verification should be documented.

Certification—11.100 (c)

"(c) Persons using electronic signatures shall, prior to or at the time of such use, certify to the agency that the electronic signatures in their system, used on or after August 20, 1997, are intended to be the legally binding equivalent of traditional handwritten signatures.

(1) The certification shall be submitted in paper form and signed with a traditional handwritten signature, to the Office of Regional Operations (HFC-100), 5600 Fishers Lane, Rockville, MD 20857.

(2) Persons using electronic signatures shall, upon agency request, provide additional certification or testimony that a specific electronic signature is the legally binding equivalent of the signer's handwritten signature."

Certifying the persons using electronic signatures are intending to apply the legal equivalent of a handwritten signature is also only a procedural activity. This certification is applicable to all systems where a certified person is able to apply electronic signatures. Certifications are required for each person and not each system.

Components and Controls—11.200

Non-Biometric Signatures—11.200 (a)

"(a) Electronic signatures that are not based upon biometrics shall:

(1) Employ at least two distinct identification components such as an identification code and password.

(i) When an individual executes a series of signings during a single, continuous period of controlled system access, the first signing shall be executed using all electronic signature components; subsequent signings shall be executed using at least one electronic signature component that is only executable by, and designed to be used only by, the individual.

(ii) When an individual executes one or more signings not performed during a single, continuous period of controlled system access, each signing shall be executed using all of the electronic signature components.

(2) Be used only by their genuine owners; and

(3) Be administered and executed to ensure that attempted use of an individual's electronic signature by anyone other than its genuine owner requires collaboration of two or more individuals."

Procedures should be established to define what distinct identification components are considered valid for use in an electronic signature. These procedures should also define what is considered a continuous period of controlled access for the purpose of identifying when only one electronic signature component is required for subsequent signings.

These procedures also need to ensure persons only use or apply their own electronic signature and they do not share or distribute any components of their electronic signature such that others cannot falsely sign electronic records for them.

Finally, the procedures need to define and manage signature components such that a single person cannot attempt to use another's signature. For example, if the system administrator, who knows the user identification codes assigned, can also reset a person's password to a known value then that individual could falsify signatures for others without requiring any assistance form others.

Controls for Identification Codes & Passwords—11.300

"Persons who use electronic signatures based upon use of identification codes in combination with passwords shall employ controls to ensure their security and integrity. Such controls shall include:"

ID and passwords are given specific mention in the regulation as they are by far the most commonly used electronic signature components.

ID & Password Uniqueness—11.300 (a)

"(a) Maintaining the uniqueness of each combined identification code and password, such that no two individuals have the same combination of identification code and password."

This requirement is most likely addressed by the procedures required for electronic signature uniqueness in 21 CFR 11.100(a).

One possible additional consideration is that rules for password components can be used to make passwords more difficult to guess. For example, password length of 6 or more characters, at least one capital letter, at least one letter and one number, would all contribute to making it more difficult to guess another's password.

Password Changes—11.300 (b)

"(b) Ensuring that identification code and password issuances are periodically checked, recalled, or revised (e.g., to cover such events as password aging)."

Identification codes should be disabled for users that are no longer allowed access to a system. Users should be periodically reviewed to ensure they are currently assigned to the correct user groups as many systems grant access or permissions based on membership in defined user groups. These types of changes are often due to change in roles or separation from the company. Whatever method is applied, the procedures should not jeopardize the integrity of signatures already executed which means it may not be possible to completely remove a user from the system.

Passwords are commonly required to be periodically changed in an effort to minimize the likelihood an ID-

password combination can be compromised. This generally accepted practice is especially important in ER/ES systems. Additional rules, such as password cannot be changed to be the same as the user ID, passwords cannot be reused or reused within a specific time period, and others should also be considered to protect the integrity of passwords.

Compromised Devices—11.300 (c)

"(c) Following loss management procedures to electronically deauthorize lost, stolen, missing, or otherwise potentially compromised tokens, cards, and other devices that bear or generate identification code or password information, and to issue temporary or permanent replacements using suitable, rigorous controls."

Loss management procedures must be clearly defined and consistently applied to protect the integrity of electronic signatures when passwords or any other signature component or component generating device is lost or compromised.

Transaction Safeguards—11.300 (d)

"(d) Use of transaction safeguards to prevent unauthorized use of passwords and/or identification codes, and to detect and report in an immediate and urgent manner any attempts at their unauthorized use to the system security unit, and, as appropriate, to organizational management."

Procedures should define how to handle any unauthorized attempts to use a person's ID and/or password. This will depend on the technological controls to identify the unauthorized attempted use.

Device Testing—11.300 (e)

"(e) Initial and periodic testing of devices, such as tokens or cards, that bear or generate identification code or password information to ensure that they function properly and have not been altered in an unauthorized manner."

Procedures need to define the testing to perform and when it should be performed, including the frequency of the periodic tests.

Technological Control

Electronic Records—Subpart B

Controls for Closed Systems—11.10

"Persons who use closed systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, when appropriate, the confidentiality of electronic records, and to ensure that the signer cannot readily repudiate the signed record as not genuine. Such procedures and controls shall include the following:"

Validation—11.10 (a)

"(a) Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to

discern invalid or altered records."

Software Verification

FDA-audited industries are required to properly validate their applications. The software world uses the term validation and verification interchangeably. However, according to the document "General Principals of Software Validation; Final Guidance for Industry and FDA Staff", we provide software that is verified. The partial definition of software verification in the FDA document is:

Software verification provides objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase. Software verification looks for consistency, completeness, and correctness of the software and its supporting documentation, as it is being developed.[…]

All our products are verified and tested extensively prior to release. Furthermore, our commitment to quality ensures performance reliability.

Software validation is for a finished device or system hence it does not apply to off-the-shelf configurable software. Validation applies to systems created using the configurable software products.

System Validation

Validation of the applications created using the AVEVA products is entirely the responsibility of the FDA-audited industry. Creating and maintaining a validated state is simplified by features and capabilities within the AVEVA products.

For example, InTouch HMI and AVEVA OMI include standard user entry windows for both alphanumeric and numeric entries. These standard windows provide entry capabilities as part of the off-the-shelf products so the features do not need validation. It is only necessary to consider validating the connections to these windows to prove entered values end up in the right place within the system.

For example, graphic object templates can be used to create templates for user interface objects like control valves or process variable displays. The features of each template are defined once and then the template can be reused each time an object with the defined features is needed. Use of the template simplifies validation by reducing the scope of testing for this custom-configured item. All features of the object template should be tested fully once but each instance of the template does not require full testing of all the template features because each individual instance shares the features of the template. Validation of each instance of a template can focus on the custom configuration of the specific instance, for example, linking of the object to a specific system input or output.

Decisions to reduce testing scope are completely within the boundaries of currently accepted industry methodology. GAMP 5 specifically promotes the use of a risk-based approach to validation, including testing. Capabilities within the AVEVA products like those mentioned above provide a strong case of reduced testing when properly employed within a custom-configured system.

Record Availability—11.10 (b)

"(b) The ability to generate accurate and complete copies of records in both human readable and electronic form suitable for inspection, review, and copying by the agency. Persons should contact the agency if there are any questions regarding the ability of the agency to perform such review and copying of the electronic records."

Viewing Recorded Alarms and Events

Our products include the Alarm DB View ActiveX control, Alarm View Industrial Graphics, and Situational Awareness Library symbols to visualize data from the alarm database in InTouch HMI applications and AVEVA OMI ViewApps. AVEVA OMI ViewApps can show the state of real time and historical alarms with the Alarm Control.

For information on configuring the Alarm DB View ActiveX control, see "Viewing Recorded Alarms" in the *AVEVA InTouch HMI Application Development Guide*. For more information about viewing alarms occurring in ViewApps, see the AVEVA OMI web help.

Viewing Recorded Data

We offer products that can be linked to various report generators. Report generators get data through queries to databases. All our software is tested to ensure data stored in the databases are accurate and complete.

Provided the chosen report generator is properly configured, the reports generated should contain an accurate and complete set of data.

Historian Insight Reporting

AVEVA OMI provides the Insight App to run an active Insight browser session within a pane of a running ViewApp. The Insight app provides a visualization of operational data by retrieving information from the Historian and enabling web users to:

- Generate reports using data from Historian databases.
- Trend history data from Historian databases.
- Build and execute SQL queries against data from Historian and other databases.

The Historian Insight website can be accessed either directly or as a control placed within a pane of a ViewApp. When accessing the site directly, a custom starting page appears from which users can access the various Reporting Website features. When accessing through AVEVA OMI, the Insight app includes a set of properties that enable users to select the data that appears in a report. For information on using the Insight app, see the AVEVA OMI web help.

Historian Reporting

AVEVA Historian and AVEVA Historian Client provide many reporting tools. For information on using these tools, see the AVEVA Historian help and the AVEVA Historian Client help.

Record Protection—11.10 (c)

"(c) Protection of records to enable their accurate and ready retrieval throughout the records retention period."

Protection of records includes performing scheduled backups of data. Typically backups should made and maintained for InTouch applications, System Platform Galaxy, Historian history blocks and SQL Server tables, WWALMDB and A2ALMDB alarm and event databases and any other system databases.

InTouch Applications Backup

In architectures where InTouch applications are based on tags defined in the tag database (as opposed to using a plant model in System Platform), InTouch applications can be either Stand-Alone or Managed. Backups for Stand-Alone InTouch applications must be made and maintained manually by selecting and copying the application directory.

Managed InTouch applications are maintained using the System Platform Integrated Development Environment (IDE) if it is installed on the same computer as the InTouch HMI. Unlike stand-alone InTouch applications that are managed entirely by Application Manager, managed applications are more integrated into the System Platform environment. Managed InTouch applications appear in the Application Manager as "Managed" and can be edited only by starting WindowMaker from within the IDE. Backups for managed InTouch applications are maintained in the System Platform Galaxy and are backed up with all of the other Application Server objects.

Managed InTouch applications are preferred where they will be deployed into regulated environments. A Managed InTouch application uses the InTouchViewApp object to manage the synchronization and delivery of

files required by the associated InTouch application. The advantages of this managed application choice are all changes are recorded and comments are allowed when changes are made. For more information, see "Managed InTouch Applications" in the InTouch HMI online help.

An AVEVA OMI ViewApp uses a ViewApp object to manage the synchronization and delivery of files required to run a ViewApp. As with a managed application, all ViewApp events are records.

InTouch includes an application version feature - a system tag that increments each time something is changed in the InTouch application configuration. This tag is called **$ApplicationVersion**. **$ApplicationVersion** can be a valuable asset in the validation process because it can be recorded in the validation protocols to ensure validated application has not been changed or altered.

Fig01

Historian Backup

Historian consists of two entities requiring backup: the SQL Server runtime database, and history blocks. Each of these entities has a separate backup procedure. For runtime database and history block backups see, "Managing the AVEVA Historian Runtime Database" and "Managing Partitions and History Blocks" under "Managing Data Storage" in th*e AVEVA Historian Help, AVEVA Historian Administration Guide. .*

Redundant Historian

Historian may be configured to have a symmetrical "partner" Historian that can be used as a backup if the primary, or main, historian is not available. This is known as a "redundant historian" setup. No control configuration is required to take advantage of a redundant historian.

When the primary historian is unavailable, the Alarm Control automatically switches over to the configured partner historian. The control remains connected to the partner historian, even when the primary historian becomes available again. The Alarm Control switches back to an available primary historian if it fails to connect to the partner or during a new attempt to connect to the primary historian, such as when restarting Trend.

Alarm and Event Backup

Management of the alarm database is performed using two InTouch utilities. The Alarm DB Purge-Archive utility is used to remove records from the database permanently or archive them to files. If the database becomes corrupt, use the Alarm DB Restore utility to restore archived records. For instructions on using these utilities see, "Maintaining the Alarm Database" in the AVEVA InTouch HMI online help. .

System Platform Galaxy Backup

The configuration model of a System Platform application is stored in the Galaxy Repository, which is an MS SQL Server database. The backup function of the Galaxy Database Manager archives all files and configuration data required to recreate the selected Galaxy in an empty Galaxy Repository. For procedures on backing up a Galaxy see, "Using the Galaxy Database Manager", "Backing Up a Galaxy", in the the AVEVA System Platform IDE online help.

System Databases Backup

For backup of data stored in other system databases, in your case Microsoft SQL Server, see your Microsoft Documentation.

Access Limitations—11.10 (d)

"(d) Limiting system access to authorized individuals."

Application Server Security

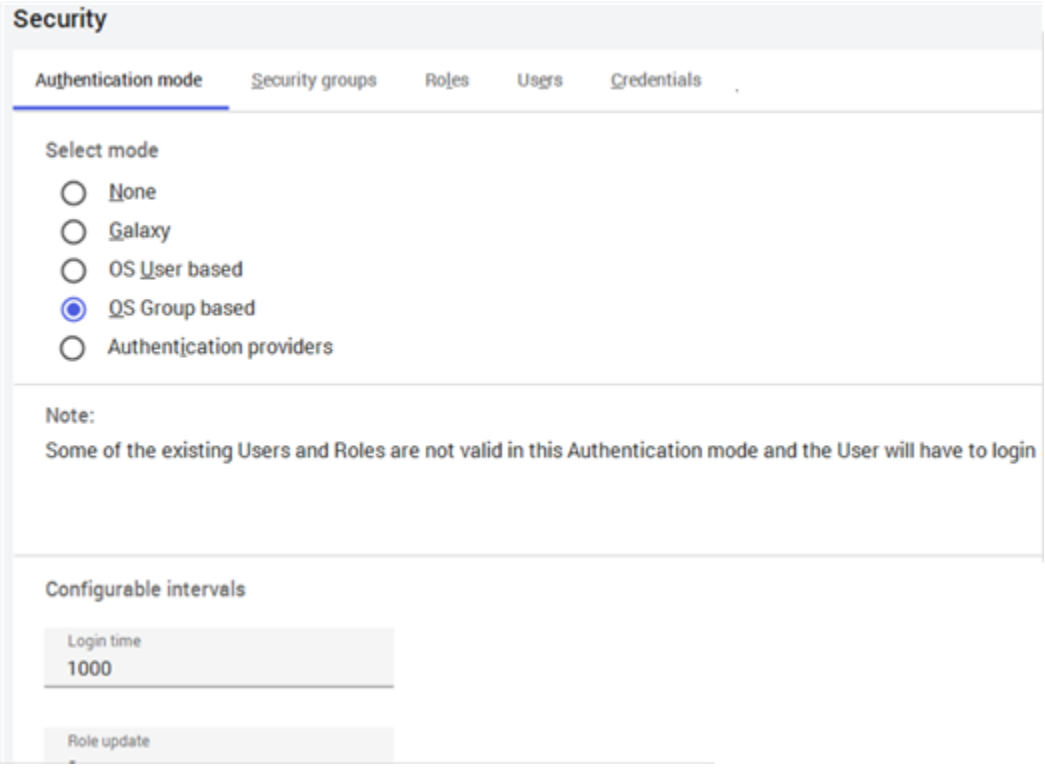
The Application Server IDE security system is a global function that applies to every object in the Galaxy database. It is a relationship-based security system between users and the objects and functions stored in a

Galaxy.

IDE security is designed to allow system administrators to easily define users and assign the operations they are allowed to perform. The security permissions are defined in terms of the operations the users can perform using automation objects.

FDA-audited industries should use the OS User Based or OS Group Based Security model for best results. Both OS Security models use Windows operating system authentication. This permits user name and password management, outside InTouch, directly in the Windows operating system environment. By using OS Security you benefit from the standard Windows functions for password aging, logon maximum trial, user name uniqueness and more.

If using OS Group Based Security Authentication Mode, make sure there is an understanding of the Windows operating system, particularly its user permissions, groups and security features. IDE OS Group-based security uses these Windows features. For more help, see the Microsoft online help or third-party documentation about Windows security.



When using local OS Groups as Roles, each node within a Galaxy must have the same OS Users, Groups, and user-group mappings to get the same level of access to the user at each node. In order to avoid this in regulated environments the use of a Windows Domain controller and Windows Active Directory is recommended in multiple node installations.

For information on OS Security configuration, see the section "About OS Group-Based Security," under "Configuring Security" in the AVEVA System Platform IDE online help.

IDE-based security includes advanced security mechanisms that also affect InTouch.

InTouch HMI Security

When using InTouch in architectures based on tag databases. InTouch offers multiple security configurations. FDA-audited industries should use the OS Security model for best results. OS Security model uses Windows operating system authentication. This permits user and password combination management, outside InTouch, directly in the Windows operating system. By using OS Security, you benefit from existing functions for password

aging, logon maximum trial, user name uniqueness and more.

For information on OS Security configuration, see the section "Using Operating System-Based Security," under "Securing InTouch" in the AVEVA InTouch HMI online help.

InTouch also offers an option for IDE-based security. When an InTouch node is configured to use ArchestrA security, the InTouch HMI uses methods and dialog boxes from Application Sever for logon and logoff operations. Users are configured in the Application Server IDE. An InTouch application configured to use IDE security provides the additional functions, which are useful in a regulated environment.

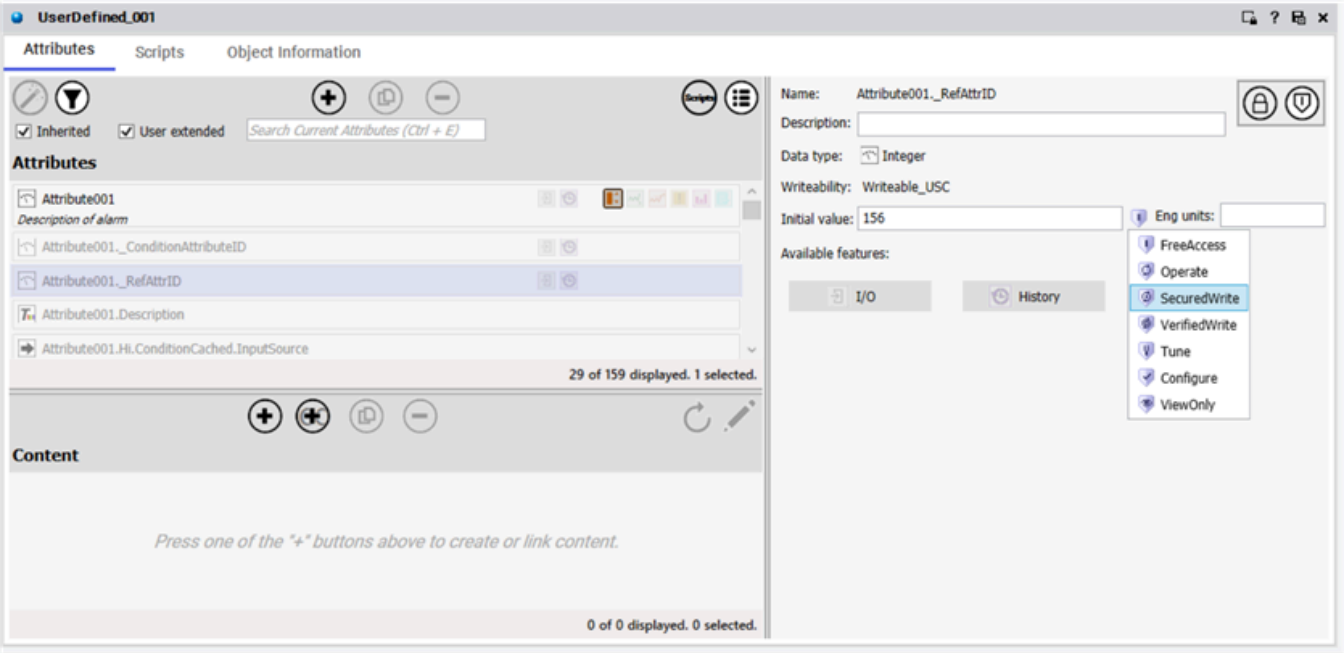
AVEVA OMI Security

AVEVA OMI extends security to the pane level of a ViewApp by restricting viewing of the pane's content to those user roles with sufficient access levels. Before you can configure security in a ViewApp, the following prerequisite tasks must be completed:

- Security roles must be assigned to those users who will interact with a running ViewApp. Each user role must be assigned an access level.
- Security must be configured to authenticate users by their user names and passwords as part of the ViewApp login process.
- Layout panes containing secure, restricted content must have an assigned access level.

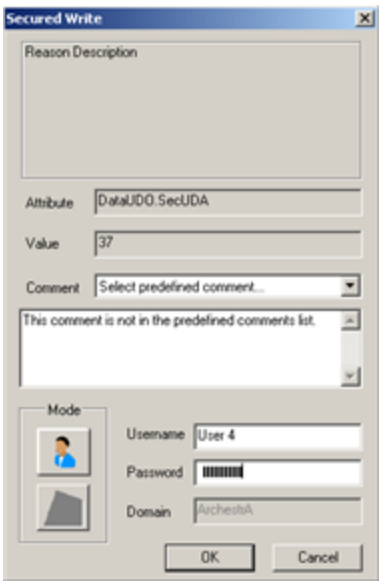
  Configuring Secured or Verified Writes with IDE Security

Attributes in a Galaxy can be configured to have an access control as Secured Write or Verified Write. Secured Write attributes require users to re-enter their passwords to complete the write to a Galaxy Attribute. Attributes configured with Verified Write require users to re-enter their passwords and also require authorization of a second operator to complete the write operation.

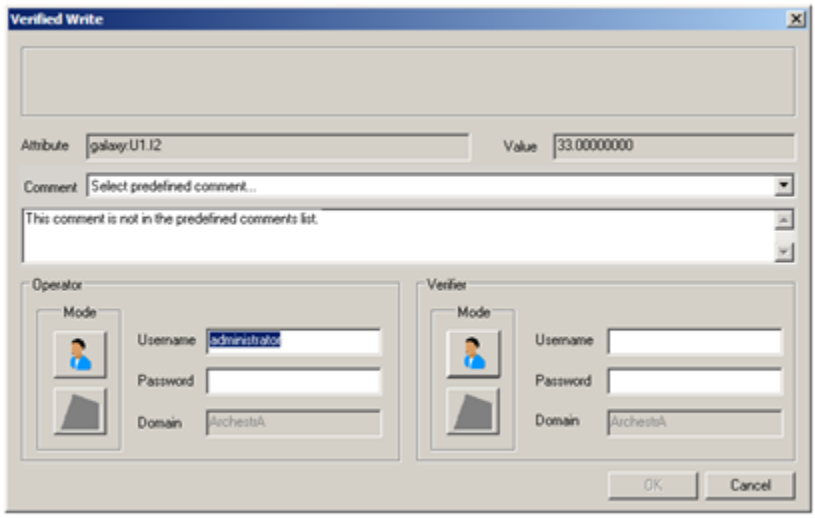


Using Secured Write and Verified Write with IDE-Based Security

Writing to an attribute that is configured for Secured Write in InTouch run time requires users to re-enter their passwords to complete the write operation. The following illustration shows the InTouch run-time dialog for Secured Write authentication.



Writing to an attribute that is configured for Verified Write, in InTouch run time requires users to re-enter their passwords and also requires authorization of a second operator to verify the write operation. The following illustration shows the InTouch dialog for Verified Write authentication.



The operator must have "Can Modify Operate Attributes" operational permission to perform a Secured or Verified Write. The verifier must have "Can Verify Writes" operational permission to confirm the Verified Write.

The operator can add a comment for the Secured or Verified Write operation by selecting from a predefined **Comment** list or by entering a comment in the **Comment** box.

Following a Secured or Verified Write a security Event is written to the event log, including the following information:

- The signer name
- Verifier name, if any
- Type of write: "Secured Write" or "Verified Write"
- Date timestamp
- Comment, if any entered by user

- Reason Description, if any provided
- Attribute description, if any, or the Short Description of the Application Object, if no Attribute description exists

  Historian Security

Historian uses two security mechanisms:

- Windows operating system security
- Microsoft SQL Server security

Historian uses Microsoft SQL Server security configured in SQL Server Authentication mode or Mixed mode. Mixed mode allows users to connect to an instance of SQL Server using either Windows authentication or SQL Server Authentication.

The security choice from either SQL Server Enterprise Manager or Historian Console is offered when registering a server. FDA-audited industries should use Mixed mode and Windows Authentication. Windows Authentication offers consistency with the OS Security models. SQL Server allows you to define Windows user groups as SQL Server users thus ensuring centralization of all user related information and facilitating management of all parameters.

When the Historian is installed, default SQL Server logins are created that can be used for logging on to the Historian from client applications. These default logins provide "out of the box" functionality so that logins do not have to be created to start using the system. The following table describes the preconfigured logins:

Login ID	Username in Database	Member of Role	Permissions
aaUser	aaUser	aaUsers	SELECT on all tables INSERT, UPDATE, DELETE on PrivateNameSpace and Annotation
aaPower	aaPower	aaPowerUsers	CREATE Table CREATE View CREATE Stored procedure CREATE Default CREATE Rule SELECT on all tables INSERT, UPDATE, DELETE on grouping tables

Login ID	Username in Database	Member of Role	Permissions
aaAdmin	aaAdmin	aaAdministrators	CREATE Table CREATE View CREATE Stored procedure CREATE Default CREATE Rule DUMP Database DUMP Transaction SELECT, INSERT, UPDATE, DELETE on all tables
aadbo	dbo	db_owner	Full database owner capabilities

Applications that are deployed in FDA-regulated industries should always change the default passwords for the SQL Server logins.

For information on Historian and related Microsoft SQL Server functions, see "Managing Security" in the AVEVA Historian online help.

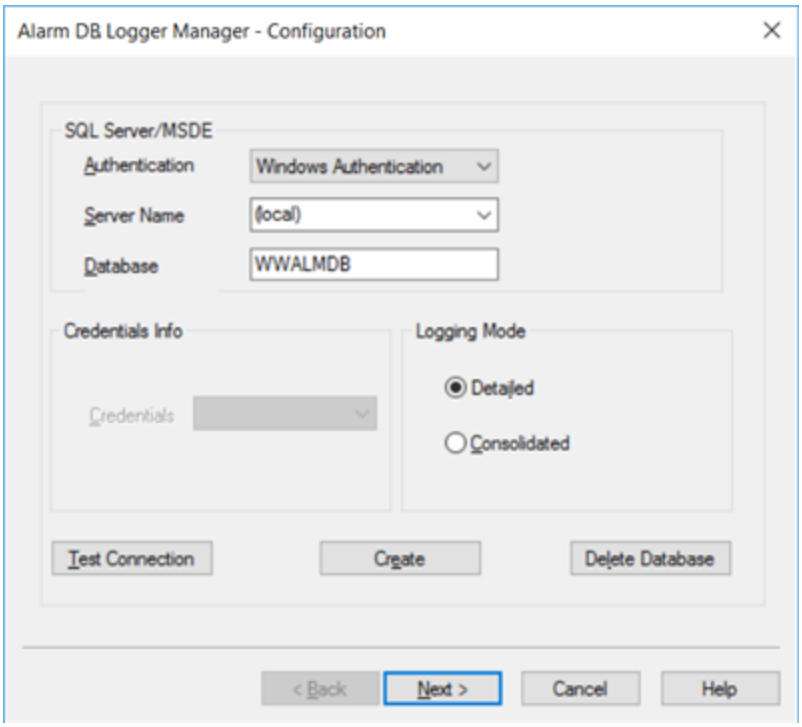
For more information on server registration, see "Registering AVEVA Historian Servers" in the AVEVA Historian online help. .

For a more secure installation, check the "Always prompt for login" information check box in the Registered Wonderware Properties window.

For information on SQL Server security and registration, see your Microsoft SQL Server documentation.

Alarm and Event Security

The InTouch Distributed Alarm system includes the Alarm DB Logger utility that logs alarms and events to an alarm database. The Alarm DB Logger Manager uses fixed accounts in the Microsoft SQL Server database to access the data. The DB Logger needs to have a write-access account which is specified using the Alarm DB Logger manager utility.



The fixed user accounts (names and passwords) present a possible compliance risk. Companies should consider the potential for un-audited changes to the alarm database and determine if any procedural controls should be employed to address the potential risks. These procedural controls could include limiting access to the database by isolating the system network and databases from the corporate network and physical limitations to HMIs that can access the alarm database.

Audit Trail—11.10 (e)

"(e) Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying."

Capturing System Information and Audit Trails

All InTouch and IDE tags defined as alarms are logged in the **WWALMDB** database. When an InTouch alarm provider is configured to use either operating system or ArchestrA authentication and an alarm occurs, the alarm record contains the full name of the operator, assuming the operator is logged on, along with the time and date and alarm details. If the alarm is subsequently acknowledged, and the node performing the acknowledgement is set to use operating system or ArchestrA security, the alarm record contains the full name of the acknowledgement operator. Otherwise, the alarm record contains a computer name concatenated with whatever is in the **$Operator** tag.

All tags that are configured as events have a record logged in the **WWALMDB** database each time an action happens with it or its value is changed. All event records, in the **WWALMDB** database, are logged with the full name of the logged on user and a time stamp in UTC. Logging the operator with an event can be forced by using secured and verified write attributes.

A comment field can also be configured in InTouch or the IDE and logged along with the alarm or event.

This logging of alarms and events that occurs while running a system can be used to create a report of system operation. In a production environment this information could be used to generate a batch report, which could

be an electronic record, that showed alarms and events (e.g. setpoint changes, user logon/off) during a batch or production run.

This information logged into the database could be part of an electronic record about system operation. While this is helpful information related to system operation, it does not constitute an audit trail.

An audit trail would be a record of any changes (additions, deletions, or modifications) to this data once it has been logged. For example, if another system operator changed an alarm limit value (a logged event) while someone else was logged in then the event recording that value change could be changed in the electronic record to indicate the actual operator making the change. That change to the electronic record would be subject to tracking in an audit trail.

SQL Server can be configured, by using triggers, to track and log changes made to any data, see the Microsoft SQL Server documentation.

Historian Modification Tracking

The Historian supports tracking of modifications (inserts and updates) to columns in the Runtime database. Modification tracking can be used to track changes to configuration data and changes to actual historian data. The Historian uses the same security defined for SQL Server for inserting and updating data. However, data values cannot be deleted from storage.

Modification tracking is system-wide; it is controlled via the use of the *ModLogTrackingStatus* system parameter. Modification tracking stores a record of modification events that include the old data, the new data and the user name of the user registered with Windows Authentication in the Historian Console. Information in the modification tracking tables is stored in the data files of the Microsoft SQL Server database.

Embedded Image (65% Scaling) (LIVE)

There are two types of modifications that can be tracked:

- Changes to configuration data. For example, additions or changes to tag, I/O Server, and storage location definitions. For more information, see "Modification Tracking for Configuration Changes" in the AVEVA Historian online help.
- Changes to history data. For example, data inserts and updates via Transact-SQL statements or CSV imports. For more information, see "Modification Tracking for Historical Data Changes" in the AVEVA Historian online help.

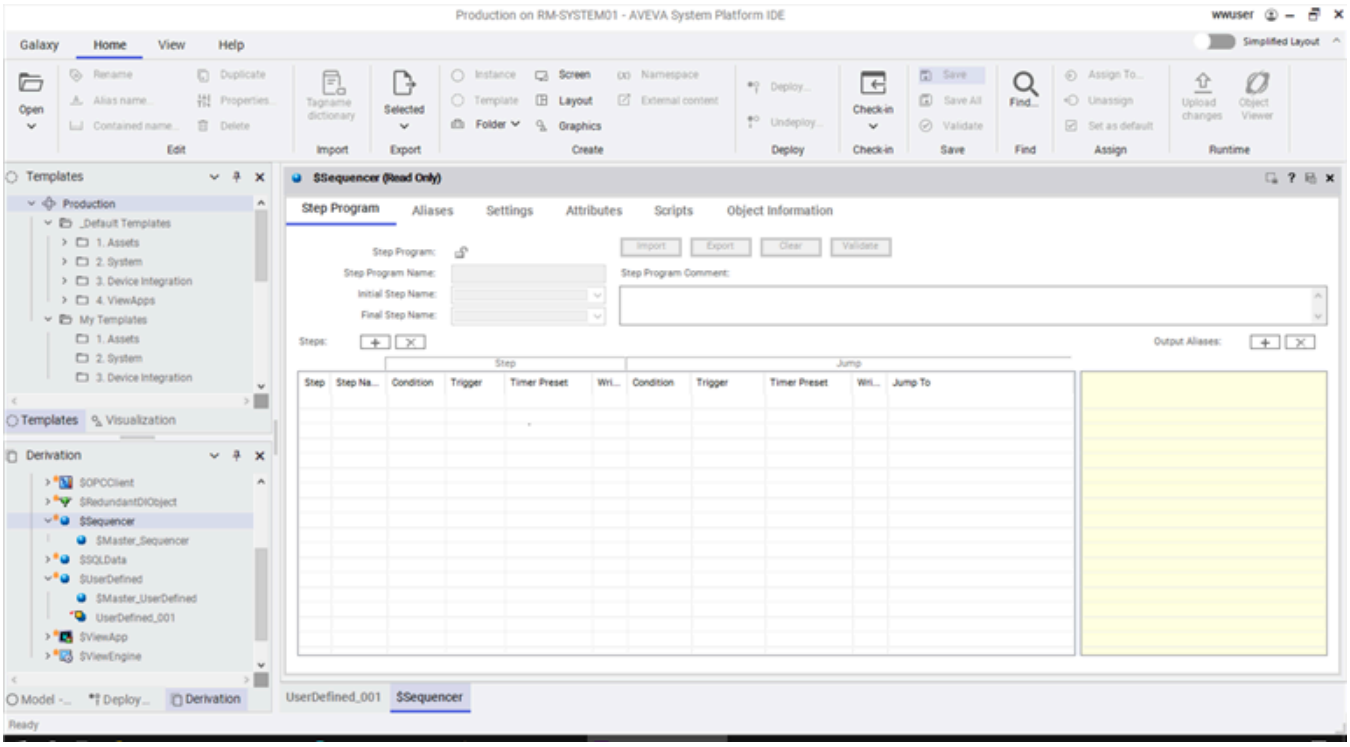
  Embedded Image (65% Scaling) (LIVE)

Sequencing—11.10 (f)

"(f) Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate."

Our products can be configured to perform and enforce operational checks and sequencing of steps and events with the use of scripting and IDE objects. A Sequencer object can be used to facilitate the configuration and verification of operational checks and sequencing of steps and events.

The compliance of these operations is up to the developer and should be verified during the testing and qualification phases of a project.



Authority Checks—11.10 (g)

"(g) Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand."

The features relevant to authority checks are the same as those discussed previously in 4.1.1.4 Access Limitations—11.10 (d).

Device Checks—11.10 (h)

"(h) Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction."

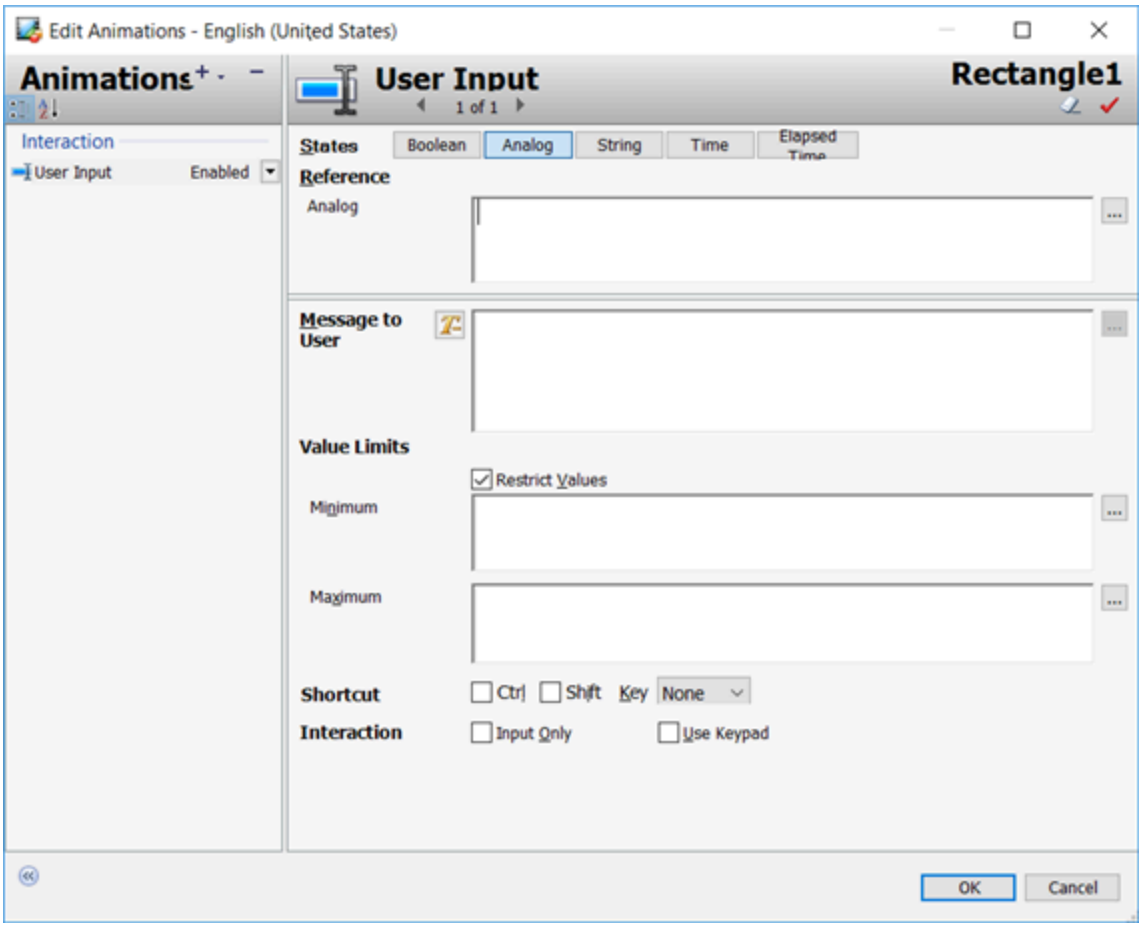
InTouch Numeric Input Validity Check

InTouch can be used to validate source data. The Analog User Input Touch Link can be used to verify that user input is within an allowable range. The figure below shows the Touch Link animation properties window. By entering a Min Value and Max Value user entries of analog data can be forced within a certain range.

fig11

Numeric Input Validity Check

AVEVA Industrial Graphics can also be used to validate source data. The User Input Animation can be used to verify that user input is within an allowable range. The figure below shows the User Input Link animation properties window. By checking the Restrict Values check box and entering Minimum and Maximum values user entries of analog data can be forced within a certain range.



Alphanumeric Input Validity Check

The examples above illustrate the validation of user entered analog data. String data can be checked and validated using scripting functions in both InTouch and the IDE. Below is script example that can be executed as an action animation, data change script, or any other method. This example checks a user string entry for length and any illegal special characters. It can easily be modified to perform other validation actions.

```
Error = 0;
ASCIICode = 0;
IF TestString01 <> "" THEN
   FOR Index = 1 TO StringLen(TestString01)
      ASCIICode = StringASCII(StringMid(TestString01,Index,1));
      IF (ASCIICode >= 48 AND ASCIICode <= 57) OR (ASCIICode >= 65 AND ASCIICode <= 90) OR
      (ASCIICode >= 97 AND ASCIICode <= 122)
THEN
      Error = 0;
   ELSE
      MessageBox("Entry contained illegal characters. Only alpha-numerics, underscores and
      dashes are allowed!","Illegal Characters Found",10);
      Error = ASCIICode;
      EXIT FOR;
   ENDIF;
NEXT;
ELSE
Error = 9999;
MessageBox("Entry cannot be null!","Invalid Entry",10); ENDIF;
TestString01 = "";
```

Data Quality

System Platform maintains a data quality attribute for all tags. Data quality is the degree of validity for a data value. Data quality can range from good, in which case the value is exactly what was originally acquired from the plant floor, to invalid, in which the value is either wrong or cannot be verified. As a data value is acquired, stored, retrieved, and then shown, its quality can degrade along the way, as external variables and events impact the system.

InTouch and the IDE can make use of the IsGood() and IsBad() functions to test the quality of data and make decisions based on the results. See the InTouch HMI online help and the AVEVA Industrial Graphics Editor online help for more information on using the data quality attribute.

Historian can use the data quality attribute in its data logging operations. This allows systems to identify any invalid data that has been logged or captured as part of an electronic record. See the AVEVA Historian online help for more information on how the Historian utilizes data quality information.

Documentation Control—11.10 (k)

"(k) Use of appropriate controls over systems documentation including:

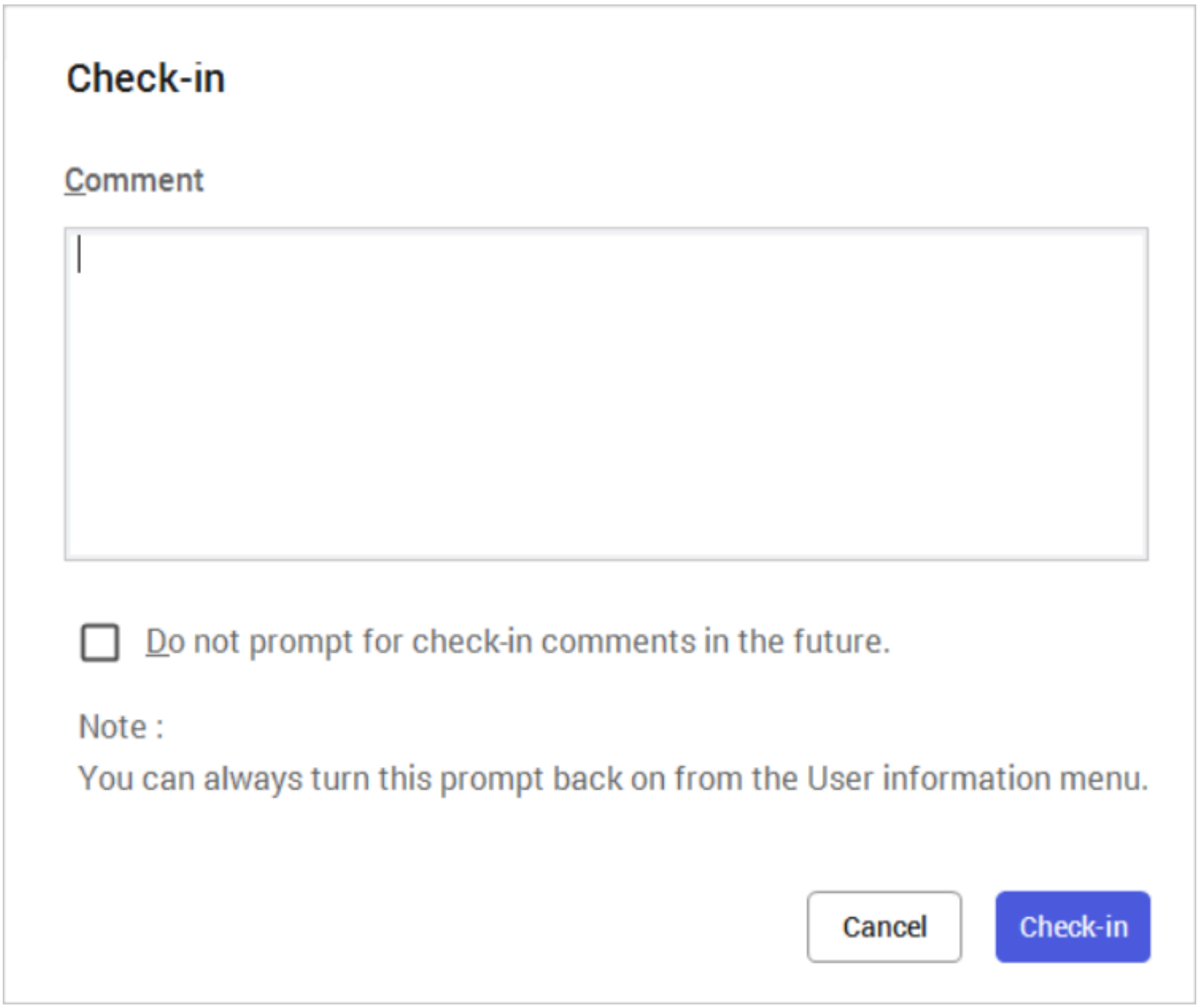
(1) Adequate controls over the distribution of, access to, and use of documentation for system operation and maintenance.

(2) Revision and change control procedures to maintain an audit trail that documents time-sequenced development and modification of systems documentation."

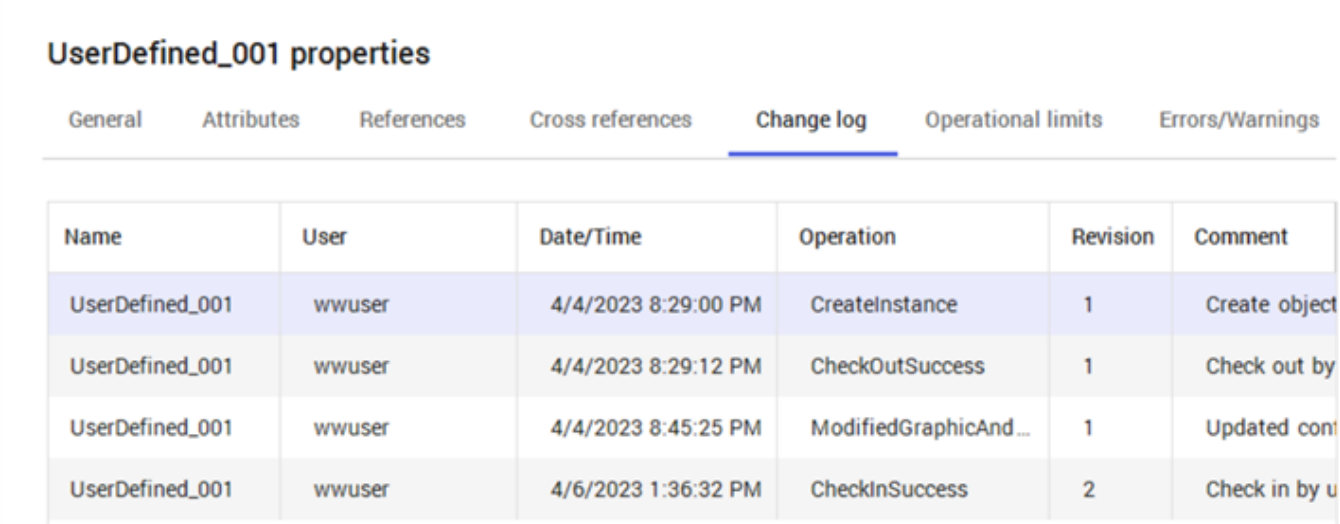
System Platform is configured with the Application Server IDE. This application provides a secure environment for developing and maintaining a system configuration. Galaxies deployed in regulated environments should employ OS User based or OS Group based security. All aspects of application development will then be controlled by a combination of Microsoft Windows and Galaxy security. See the AVEVA Historian online help for more information on how the Historian utilizes data quality information.

For more information on securing the IDE see "Working with Security" in the AVEVA Application Server online help.

The IDE also provides a revision history of Application Objects and symbols. A revision history includes records that track the life cycle of activities, such as object creation, check in/check out, deployment, and import/export. When objects or symbols are checked back in to the Galaxy after making changes, a dialog box prompts the user to enter comments about the changes. It is good policy to enforce mandatory check in comments in regulated environments.



A log of all changes and comments is available from the object properties menu. Following is an example log.



Signature Manifestation—11.50

"(a) Signed electronic records shall contain information associated with the signing that clearly indicates all of the following:

(1) The printed name of the signer;

(2) The date and time when the signature was executed; and

(3) The meaning (such as review, approval, responsibility, or authorship) associated with the signature.

(b) The items identified in paragraphs (a)(1), (a)(2), and (a)(3) of this section shall be subject to the same controls as for electronic records and shall be included as part of any human readable form of the electronic record (such as electronic display or printout)."

Signatures for Alarms and Events

The InTouch Distributed Alarm system includes the Alarm DB Logger utility that logs alarms and events to the alarm database. Alarm and event records are generated by the system and the signature is linked to a specific event. When an entire alarm or event row is retrieved, the signature is linked.

InTouch-generated event records include the operator and a comment which is the Alarm Comment field of the InTouch tag. InTouch verification events can be handled with scripts that utilize the *InvisibleVerifyCredentials()* function.

All IDE user-defined attributes generate events. The generated event records include the operator and a comment. The operator column can be forced by setting an Attribute to either Operate, Secured Write or Verified Write access. The comment column is the object description, all Attributes in an object will have the same comment therefore each event should be its own object. Boolean events can take advantage of the Boolean label extension for the Attribute, which will be logged in the Value String column of the event database. Verified Write events will show both the done by and checked by operator in the OperatorName column.

When an InTouch alarm provider, for example, InTouch or the IDE, is configured to use either operating system or IDE-managed authentication and an alarm occurs, the alarm record contains the full name of the operator in the Operator Full Name column, assuming the operator is logged on along with the time and date and event details. For example if a user is registered in the PLANT_FLOOR domain with a user ID of JohnS and a full name of John Smith, the Operator Full Name column contains John Smith. If the alarm is subsequently acknowledged, and the node performing the acknowledgement is set to use operating system or IDE security, the alarm record contains the full name of the acknowledgement operator. Otherwise, the alarm record contains a computer name concatenated with whatever is in the **$Operator** tag.

Applications deployed in an FDA-regulated environment should use OS security. InTouch Managed applications should use IDE security and the IDE should use OS group or OS user security.

For information about alarms and events and logging see the Alarms and Events topics in the InTouch HMI online help and "Working with Alarms and Events" in the AVEVA Application Server online help.

Other Signatures

AVEVA products can be configured to produce other electronic signatures with the use of scripting and object attributes configuration. The compliance of these records is up to the developer and should be verified during the testing and qualification phases of a project.

You can acknowledge alarms using the AVEVA Alarm Client Control by providing your signature. If the alarms are configured to require a signature for acknowledgement, the system checks whether any alarms are awaiting

acknowledgement and whether the alarms fall within the configured priority range. If so, you need to provide valid domain, user name, and password authentication to acknowledge the alarms.

You can also provide your signature to acknowledge the alarms by using Smart Cards. You need to have a Smart Card reader configured to your system. If an alarm requires a signature to be acknowledged, you must select the appropriate Smart Card inserted in the reader at run time. You need to provide the valid PIN to acknowledge the alarm. You can also choose to log on with your name, password, and domain instead of Smart Card.

Signature/Record Linking—11.70

21 CFR Part 11:

"*Electronic signatures and handwritten signatures executed to electronic records shall be linked to their respective electronic records to ensure that the signatures cannot be excised, copied, or otherwise transferred to falsify an electronic record by ordinary means.*"

Signatures for Alarms and Events

The InTouch Distributed Alarm system includes the Alarm DB Logger utility that logs alarms and events to the alarm database. Alarm and event records are generated by the system and include user name, time and date and all other event details.

You can acknowledge alarms using the AVEVA Alarm Client Control and in InTouch by providing your log-on credentials. In the Alarm Client Control, select the **Requires ACK Signature** check box to configure alarms to require a signature for acknowledging them.

For both the Alarm Client Control and the SignedAlarmAck() script function, when the signature requirement is enabled, the user credentials are required only if the alarms to be acknowledged fall within a specified priority range, or if no user is currently logged on to the InTouch application. This is because if a Galaxy is secured, then only an authenticated user can acknowledge alarms.

If the system is configured with a Smart Card reader, users can provide the Smart Card and the valid PIN to acknowledge the alarm.

In the IDE, the SignedAlarmAck() script function enables the user to configure AVEVA Industrial Graphics, set to act as alarms, to require a signature in order to be acknowledged.

Similar to the Alarm Client Control, you can use Smart Cards to provide user authentication.

Electronic Signatures—Subpart C

General Requirements—11.100

Signature Uniqueness—11.100 (a)

21 CFR Part 11:

"(a) Each electronic signature shall be unique to one individual and shall not be reused by, or reassigned to, anyone else."

FDA-audited industries should use the OS User Based or OS Group Based Security model for best results. Both OS Security models use Windows operating system authentication. This permits user name and password management, outside InTouch, directly in the Windows operating system environment. By using OS Security you

benefit from the standard Windows functions for password aging, logon maximum trial, user name uniqueness and more.

It is not recommended to use local OS Groups as Roles, as that requires each node within a Galaxy to have the same OS Users, Groups, and user-group mappings to get the same level of access to the user at each node. Defining users on individual nodes creates a possibility of the same user name being assigned to different users. For example, if user name jdoe was used on a node for John Doe and jdoe was used on a different node for Jane Doe, within the same system, alarm or event records would not be able to distinguish between the users. Managing users in a single location and authenticating by connecting to that location eliminates the potential for multiple users having the same user name, which in turn ensures signature uniqueness.

Components and Controls—11.200

Non-Biometric Signatures—11.200 (a)

"(a) Electronic signatures that are not based upon biometrics shall:

(1) Employ at least two distinct identification components such as an identification code and password.

(i) When an individual executes a series of signings during a single, continuous period of controlled system access, the first signing shall be executed using all electronic signature components; subsequent signings shall be executed using at least one electronic signature component that is only executable by, and designed to be used only by, the individual.

(ii) When an individual executes one or more signings not performed during a single, continuous period of controlled system access, each signing shall be executed using all of the electronic signature components.

(2) Be used only by their genuine owners; and

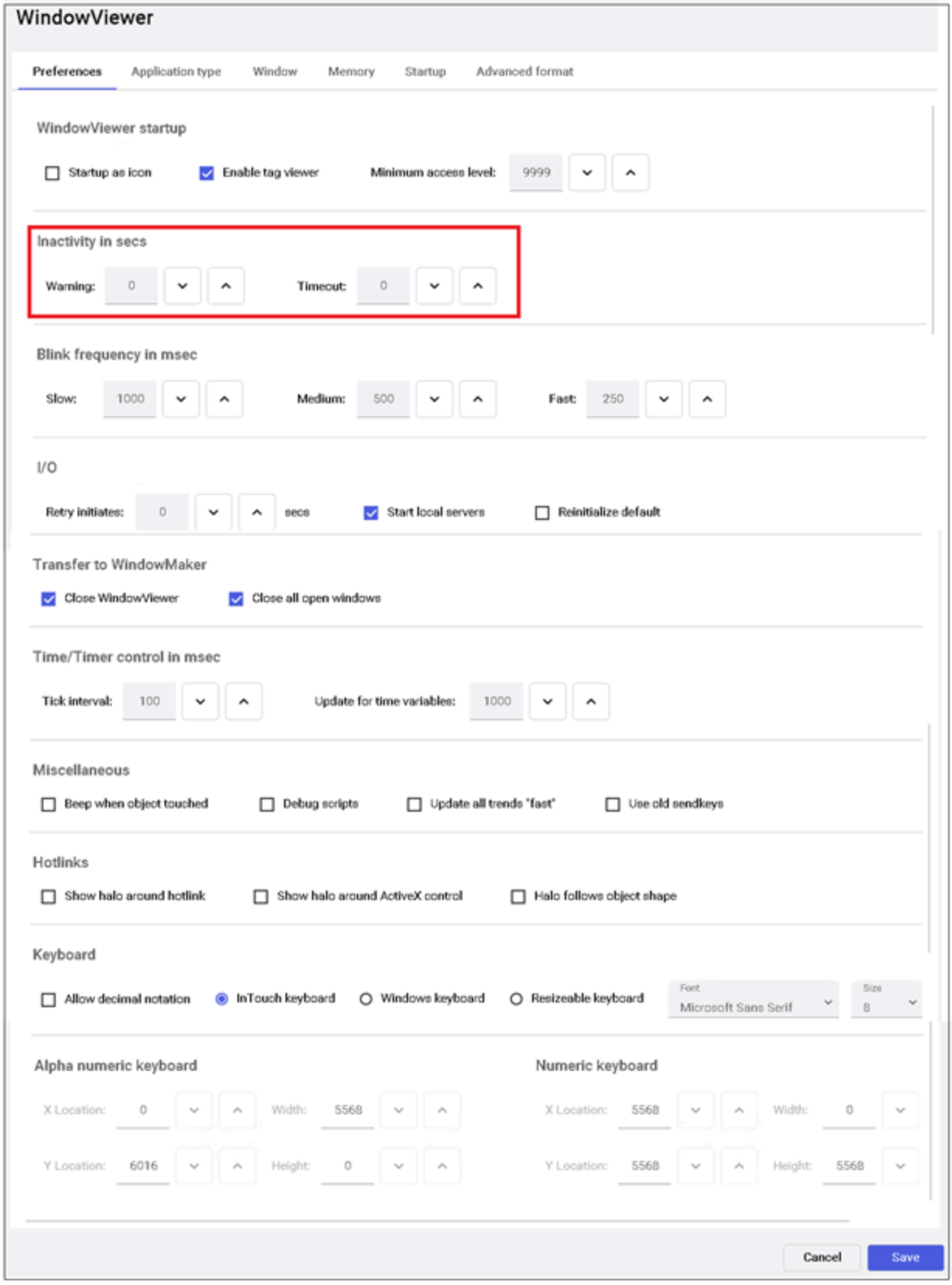
(3) Be administered and executed to ensure that attempted use of an individual's electronic signature by anyone other than its genuine owner requires collaboration of two or more individuals."

Systems should be set up to require user ID and password entry to authenticate users. Logging the authentication event or logon can then act as the first signing as all electronic signature components (ID, password) are required for logon and the user name of the logged in user can be recorded as part of the logon event.

Subsequent signature events, such as alarm limit change and alarm acknowledgement, can also include the user name to indicate the signing of the event. The user ID and password combination is not required to complete the signing while this same user is logged in as the duration of any logon for a user is a single, continuous period of controlled access.

If a user logs out manually, is logged out by another user logging in, for example to perform a checked-by function, or is inactive and logged out automatically by the system, the continuous period of controlled access ends. Any signatures would then require a new user to be logged in, which requires all electronic signature components including ID and password.

InTouch WindowViewer can be configured to automatically log off an inactive operator from an InTouch application. An operator must log on again after being logged off for inactivity. Setting an automatic inactivity log off period prevents unauthorized access to your InTouch application when operators leave their workstations unattended. Inactivity time periods should be evaluated for each system and vary according to the unique attributes and environment in which each system is operated.



For more information on using the WindowViewer inactivity features see "Security Configuration for InTouch HMI" in the AVEVA InTouch online help.

Controls for Identification Codes & Passwords—11.300

ID & Password Uniqueness—11.300 (a)

"Persons who use electronic signatures based upon use of identification codes in combination with passwords shall employ controls to ensure their security and integrity. Such controls shall include: (a) Maintaining the uniqueness of each combined identification code and password, such that no two individuals have the same combination of identification code and password."

This requirement is addressed by the content in Signature Uniqueness—11.100 (a).

Password Changes—11.300 (b)

"(b) Ensuring that identification code and password issuances are periodically checked, recalled, or revised (e.g., to cover such events as password aging)."

FDA-audited industries should use the OS User Based or OS Group Based Security model for best results. Both OS Security models use Windows operating system authentication. This permits user name and password management, outside InTouch, directly in the Windows operating system environment. By using OS Security you benefit from the standard Windows functions for password aging, logon maximum trial, user name uniqueness and more.

Transaction Safeguards—11.300 (d)

"(d) Use of transaction safeguards to prevent unauthorized use of passwords and/or identification codes, and to detect and report in an immediate and urgent manner any attempts at their unauthorized use to the system security unit, and, as appropriate, to organizational management."

System events can be created to document failed logon attempts in the event someone attempts to logon as a different user. There is no technological control to prevent unauthorized use of IDs or passwords if those ID and password combinations are compromised or known to more than the individual assigned a specific ID and password combination.

Other Technical Products

Additional AVEVA products incorporate features and functionality designed to facilitate the development of applications for use in FDA-regulated industries.

While it is outside the scope if this guide to document best practices for 21 CFR Part 11 for these additional products, following are summary descriptions of related AVEVA products and their capabilities relevant to regulated industries.

AVEVA Batch Management software effectively manages flexible batch operations found in the process industries, including life sciences, fine chemicals, and food and beverage/CPG. Adhering to the ISA-88 standards for batch control, it provides guidance and oversight to both recipe management and batch execution. Batch Management software coordinates everything with the plant control systems, interfaces with the operators, and directs batch activity, material flow, and production data to a historical database for a full electronic batch record (EBR). Customers can improve Compliance Governance through complete electronic system records for 'as planned' and 'as executed' information, including full Electronic Batch Records (EBR) in according with requirements found in FDA CFR 21, Part 11 regulations.

Operations & Performance Software— provides a configurable and highly scalable Manufacturing Execution Software System (MES) solution that is integrated with AVEVA System Platform IDE and InTouch HMI for unsurpassed connectivity and flexibility and can be applied to essentially any manufacturing or process industry. Customers gain accurate equipment setup according to site specific product specifications; central administration of process and product parameters; consistent interpretation of operating guidelines and procedures - all with complete electronic 'As-Built' historical records for documentation of products and processes.

Glossary

Biometrics

A method of verifying an individual's physical feature(s) or repeatable action(s) where those features and/or actions are both unique to that individual and measurable

cGMP

Current Good Manufacturing Practice

CFR

Code of Federal Regulations

Closed system

An environment in which system access is controlled by persons who are responsible for the content of electronic records that are on the system

DB

Database.

ER/ES

Electronic Record/Electronic Signature

FDA

Food and Drug Administration

GAMP

Good Automated Manufacturing Practice

Historian

The Historian component of the System Platform is a high-performance real-time database for historical information. It combines the power and flexibility of a relational database with the speed and compression of a true process historian, integrating the office with the factory floor or any industrial operation.

HMI

Human-Machine Interface

ID

Identification or an item used to verify one's identity (e.g. user name)

IDE

Integrated Development Environment

IT

Information Technology

InSQL

IndustrialSQL Server

IndustrialSQL Server

Now called "Historian"

ISPE

International Society for Pharmaceutical Engineering

OS

Operating System

Open system

An environment in which system access is not controlled by persons who are responsible for the content of electronic records that are on the system.

Part 11

21 CFR Part 11

SOP

Standard operating procedures

SQL

Structured Query Language

System Platform

System Platform provides a single platform for all the SCADA, Supervisory HMI, and MES and EMI Software Solutions needs of industrial automation and information personnel.

UDA

User Defined Attribute

UTC

Coordinated Universal Time

WinPlatform object

An object that represents a single computer in a Galaxy, consisting of a system-wide message exchange component, a set of basic services, the operating system, and the physical hardware. The WinPlatform object hosts the Application Engine (AppEngine).

System requirements and guidelines

Hardware requirements

Hardware Requirements Notes

Operating System, Firewall, .NET Framework, and Virtualization Requirements

Software requirements

Windows Operating System Notes

Supported Operating Systems at Time of Release

.NET Framework requirements and compatibility

.Net Notes

SQL Server Notes

Virtual Environment Notes

Firewall notes

Third-Party Application Prerequisites

Operating system notes

Operating System Notes: Common for AVEVA Products

Operating System Notes: InTouch HMI

Operating system notes: Application Server

Operating system notes: Historian Client

SQL Server considerations

Considerations for SQL Server

Considerations for SQL Server Express

Hardware requirements notes

Windows operating systems and SQL Server versions may impose hardware requirements that exceed the minimum requirements for System Platform 2023 R2. Refer to the following Microsoft Web sites for Windows and SQL Server hardware requirements.

Windows requirements

- Windows Server 2022 System Requirements

- Windows Server 2019 System Requirements
- Windows Server 2016 System Requirements
- Windows 11 System Requirements
- Windows 10 System Requirements

Note: System Platform 2023 R2 is not supported on any 32-bit Windows operating system. Windows 10, Version 1809 (64-bit), is the earliest version of Windows 10 that supports System Platform 2023 R2. For the complete list of supported Windows operating systems and SQL Server versions, see the AVEVA GCS Technology Matrix.

SQL Server requirements

- Hardware and Software Requirements for Installing SQL Server 2019
- Hardware and Software Requirements for Installing SQL Server 2017 and Prior

AVEVA Historian hardware guidelines

- AVEVA Historian is not supported on cluster hardware.
- Do not use the Historian computer as a domain controller.
- If you are running the Historian on a virtual server, the Historian must have adequate CPU, network, memory, and disk I/O resources at all times. Overloading the virtual server leads to unpredictable behavior.

For system sizing examples, see the *System Platform Installation Guide* (SP_Install_Guide.pdf).

Operating system, firewall, .NET Framework, and virtualization notes

- Firewall notes
- Operating System Notes: Common for AVEVA Products
- Operating System Notes: InTouch HMI
- Operating system notes: Application Server
- Operating system notes: Historian Client
- .NET Framework requirements and compatibility
- Virtual Environment Notes

Minimum Operating System and Browser Requirements for System Platform 2023 R2

The latest product information for each System Platform product is listed in the AVEVA Global Customer Support GCS Technology Matrix. Each link includes:

- General information about the selected product, such as the version number and release date
- Operating system requirements

- Microsoft SQL Server requirements
- Virtualization software compatibility
- Information about interoperability (which AVEVA products a specific product works with)
- Information about coexistence (which AVEVA products can be installed on the same node)

Minimum Required Operating System Version

The following table shows the **minimum** operating system for each product. See the GCS Technology Matrix for a list of all supported operating systems.

Product	Minimum Required Operating System Version	
	Client OS (x64 only)	**Server OS (x64 only)**
Application Server	Windows 10 LTSC Version 1809 or later Enterprise and IoT Enterprise SAC Version 21H2	Windows Server 2016 or later Standard, Data Center, and IoT Windows Server 2022 SAC 21H2
InTouch		
Historian		

Important: Regardless of which operating system you are using, we recommend that you download and install the latest Microsoft updates to enhance security and ensure product compatibility.

Note: Only Windows Server versions licensed under Microsoft's long term servicing channel (LTSC) are supported. Versions of Windows Server licensed under the Semi-Annual Channel (SAC) do not include Desktop Experience, and therefore, are not supported.

Supported Operating Systems at Time of Release

The latest product information for each System Platform product is listed in the AVEVA Global Customer Support GCS Technology Matrix. Each link includes:

- General information about the selected product, such as the version number and release date
- Operating system requirements
- Microsoft SQL Server requirements
- Virtualization software compatibility
- Information about interoperability (which AVEVA products a specific product works with)
- Information about coexistence (which AVEVA products can be installed on the same node)

  Note: Only Windows Server versions licensed under Microsoft's long term servicing channel (LTSC) are supported. Versions of Windows Server licensed under the Semi-Annual Channel (SAC) do not include Desktop Experience, and therefore, are not supported. For example SAC versions 1709, 1903, and 2003 are not supported.

64-bit only

System Platform Product / Component	SAC Windows 10/11 Pro/ Enterprise/IoT Enterprise	LTSC Windows 10/11 Enterprise/IoT Enterprise	LTSC Windows Server See Windows Server Notes
Application Server 2023 R2 • Galaxy Repository • AppEngine / Platform / Bootstrap • IDE (including remote) • AVEVA OMI ViewApp (run time)	Windows 10 21H2 Windows 10 22H2 Windows 11 21H2 Windows 11 22H2 Includes Professional, Enterprise and IoT Enterprise.	Windows 10 1809 Windows 10 21H2 Windows 11 Includes Enterprise and IoT Enterprise	2016 Standard and Data Center 2016 IoT 2019 Data Center, Essentials, and Standard 2019 IoT 2022 Standard and Data Center
InTouch HMI 2023 R2 • WindowMaker (no Modern apps) • WindowMaker (Modern apps) • WindowViewer (run time) / InTouchViewApp • InTouch Web Client (run time)			
InTouch for System Platform 2023 R2 • WindowMaker (Managed Apps) • WindowViewer (run time) / InTouchViewApp			
Historian Server 2023 R2			
InTouch Access Anywhere Server 2023 • ITAA Client (HTML5 browser) InTouch Access Anywhere Secure Gateway 2023			

Other System Platform 2023 R2 components / products • Historian Client 2023 R2 • OI Gateway • System Management Server • AVEVA Licensing • System Monitor Agent / Manager			

SAC: Semi-Annual Channel (subscription)

LTSC: Long Term Servicing Channel

LTSC is the newer name for LTSB (Long Term Servicing Branch)

Windows Server Notes:

- **SAC releases** of Windows Server do not include Desktop Experience and thus cannot be supported.
- **Windows Server 2016:** IoT, Standard, and Data Center are supported.
- **Windows Server 2019 (Desktop Experience):** IoT, Standard, and Data Center are supported
- **Windows Server 2019 (Core):** NOT supported

Important! Installing System Platform 2023 R2 on a computer used as a domain controller is not supported. For more information, see the Microsoft Security Best Practices Checklist.

System Platform 2023 R2 Web Clients

The following web clients are included with System Platform 2023 R2:

- InTouch Access Anywhere Client (HTML5 Browser)
- Historian Insight Client
- AVEVA Enterprise Licensing - Manager Client

The client programs listed above can be used with most common web browsers. Compatible browsers include:

- Microsoft Edge Non-Chromium and older
- Microsoft Edge Chromium 97.0.1072.76 and newer
- Firefox version 96.03 ESR and newer
- Safari version 15.2 and newer (Mac and iOS only) (Not Windows)
- Google Chrome version 98.0.4758.80 and newer
- Opera version 83.0.4254.16 and newer

Windows Operating System Notes

- Upgrading from one version of System Platform to this version is only allowed when system requirements for operating system version, SQL Server version, and .NET Framework version are met. Upgrading System Platform removes the prior version and installs the current version (System Platform 2023 R2).
- Support for 32-bit versions of Windows and SQL Server has been discontinued. Only 64-bit versions are supported.
- Semi-Annual channel (SAC) releases of Windows Server only include Core, and not Desktop Experience. Therefore, SAC releases of Windows Server are not supported.
- Newer operating system Service Packs (SPs) than those listed do not block the installation of AVEVA products. A warning message may appear during the installation process.
- Upgrading the operating system while System Platform is installed is not supported.

.NET notes

Versions of .NET Framework (other than 4.x versions) can coexist, but all .NET code, including QuickScript.NET scripts, run under .NET Framework 4.8 or higher. For more information about .NET Framework requirements and compatibility, see .NET Framework requirements and compatibility.

SQL Server notes

Upgrading SQL Server with AVEVA products installed is supported. See the AVEVA Global Customer Support GCS Technology Matrix for the complete list of supported SQL Server versions.

Only 64-bit versions of SQL Server are supported in this System Platform release. See Supported SQL Server versions at time of release for more information.

If an error message about an unsupported SQL Server version is displayed while installing or upgrading System Platform, check the following:

- Your installed version of SQL Server is no longer supported, for example, SQL Server 2012.
  - **How to fix:** Upgrade to a supported version. Refer to the following Microsoft resource for more information: Upgrade SQL Server
- Your installed version of SQL Server is supported but requires a service pack. For example, you have SQL Server 2016 SP2, which was supported on System Platform 2020 R2 SP1, but now upgrading to System Platform 2023 is blocked because this release requires SQL Server 2016 SP3 or newer.
  - **How to Fix**: Download and install the required service pack.
- You have a 32-bit version of SQL Server installed.
  - **How to Fix:** Refer to Issue 1249251 under Installation and Uninstallation Issues.

Supported SQL Server versions at time of release

64-bit only

System Platform Product / Component	SQL Server Express SSMS	SQL Server
Application Server 2023 R2 • Galaxy Repository • AppEngine / Platform / Bootstrap • IDE (including remote) • AVEVA OMI ViewApp (run time)	2016 SP3 v13.0.6300.2 or higher 2017 Core v14.0.1000.169 or higher 2019 Core v15.0.2000.5 or higher 2022 Core v16.0.1000.6 or higher	2016 Standard/Enterprise SP3 v13.0.6300.2 or higher 2017 Standard/Enterprise v14.0.1000.169 or higher 2019 Standard/Enterprise v15.0.2000.5 or higher 2022 Standard/Enterprise v16.0.100.4 or higher
InTouch HMI 2023 R2 • WindowMaker (no Modern apps) • WindowMaker (Modern apps) • WindowViewer (run time) / InTouchViewApp • InTouch Web Client (run time)		
InTouch for System Platform 2023 R2 • WindowMaker (Managed Apps) • WindowViewer (run time) / InTouchViewApp		
Historian Server 2023 R2		
System Monitor Manager		

Be sure to apply all cumulative updates before installing System Platform

The default database is SQL Server 2022 Express with Advanced Tools

For information about which version of SQL Server you are running, see Determine the version, edition, and update level of SQL Server and its components.

Virtual environment notes

The following virtualization software and cloud-based virtual environments are supported for System Platform 2023 R2:

- Hyper-V 5 and newer (version is based on the operating system utilized)
  - Both Gen 1 and Gen 2 VMs are supported

  - Windows 10 and Windows Server 2016 support Versions 7 and 8
- VMWare VSphere 6.5 and newer, including HA/DR
- VMWare Clients, including Horizon - Application Virtualization
- Cloud Virtualization - Azure
- VMWare Workstation, Version 12.5.9 (only) and Versions 14.1.3 and newer
- Stratus HA/DR and FT based solutions (zTC Edge 110i)
- Nutanix (AHV version 20170830.337, Nutanix AOS version - 5.10.8.1 LTS)

Firewall notes

In order to establish communication with other components and services on the network, System Platform and Platform Common Services (ASB services) require certain network ports to be opened in the Windows Firewall. Typically, these ports are automatically opened for the Windows Firewall during installation. However, if the Windows Firewall service is disabled or not running at the time of installation, or if an alternative firewall is in use, you will need to manually open the appropriate network ports in the firewall. For more information about port configuration, see "Configuring Service TCP Ports" in the *Application Server User Guide* (filename: IDE.pdf).

Note: The ArchestrA Service Bus (ASB) is superseded by Platform Common Services (PCS). However, some ASB references remain in System Platform.

Operating System Notes: Common for AVEVA Products

Before installing System Platform, download and install the latest Microsoft updates to enhance security and ensure product compatibility. Allow the Windows update process to finish before you start installing System Platform. This recommendation applies to all Windows versions.

ActiveX controls behavior on supported Windows operating systems

Due to the Data Execution Prevention (DEP) feature in Windows operating systems, any ActiveX control built with ATL version 7.1 or earlier will fail to host, or will behave unpredictably in InTouch, either in WindowMaker or WindowViewer.

The ActiveX controls and error message, along with solutions to resolve the behavior, are described in detail in TechNote 522, "Some ActiveX Controls NOT Supported in InTouch 2012 R2 (Version 10.6)". You can download this TechNote from the AVEVA Global Customer Support (GCS) website.

Configuring remote alarm retrieval queries

The process to configure remote alarm retrieval queries has changed for interactive applications such as InTouch HMI when running on currently-supported Windows and Windows Server operating systems.

When InTouch WindowViewer is started and generates alarms from an interactive Windows or Windows Server desktop session, an **AlarmViewer** control (running within InTouch HMI) on a remote node must be specially configured to query the alarms. The source alarms will not appear unless the **AlarmViewer** control's alarm query is configured.

This type of query only works for InTouch HMI as an alarm provider running in a Terminal Services session, not

for InTouch HMI running in a console session.

To configure the AlarmViewer's alarm query

1. After starting InTouch WindowViewer (alarm provider), open the **Operations Control Management Logger** and look for the most recent string generated by **AlarmMgr**. For example: "Registering AlarmMgr with SLSSVC as AlarmMgr 253.127.148.120". The indicated IP address will be unique to your alarm-providing node. Note the IP address for use in Step 2.
2. In the **Alarm Query** tab of the **AlarmViewer** control on the remote computer, configure the alarm query as follows, substituting your nodename of the alarm providing InTouch HMI for "nodename" below and substituting your IP address noted in the previous step:

   ```
   \\nodename:ip_address\intouch!$system
   ```

   where *nodename* is the name of the node that is providing the InTouch alarm and *ip_address* is the IP address that you determined in step 1.
3. Test to validate that the alarms generated from the alarm-providing node are shown accurately in the **AlarmViewer** control.

Terminal services behavior in Windows server operating systems

Windows Server no longer supports the /console switch as a means of starting the remote desktop (RDP) client, also known as Session 0 or Terminal Server Console session. Session 0 is no longer an interactive session, and is reserved only for Windows services. Windows Server treats all remote connections as remote RDP sessions regardless of /console, /admin, or any other switches used to make the connection.

This impacts InTouch HMI functionality such as Alarm Manager that depends on the Terminal Server Console session. The impact to Application Server is minimal as most Application Server processes run as services. One impact to Application Server is to carry forward the restriction introduced with the Windows Vista operating system which permits only one alarm provider. While both Application Server and InTouch HMI can be configured as alarm providers, only one alarm provider can be configured at any one time.

Refer to the InTouch HMI Readme for further information about InTouch HMI applications running in the Terminal Server Console.

If you are running WindowViewer within a Terminal Server session and want to access alarms from WindowViewer in a client session, you must use the syntax **\\terminalservernode:<IP address>\InTouch!$System** to access the alarms, with a colon (:) after the node name. The IP address is that of the client computer connected to the session.

Operating System Notes: InTouch HMI

InTouch HMI with supported Windows operating systems

- InTouch HMI 2023 R2 does not support the following legacy script functions: WWPoke(), WWExecute(), WWRequest(), ActivateApp() and SendKeys().
- If Recipe Manager is started using the path **Start\Program\AVEVA\InTouch\Recipe**, then select **Run as Administrator**.
- The InTouch Extensibility Toolkit might need to be started by right-clicking and selecting **Run As Administrator** to function properly.

- The onscreen keyboard options were changed as of Windows 7 and Windows Server 2008 R2. Although System Platform 2023 R2 does not support these operating systems, these changes apply to the currently-supported operating systems.
- Hovering to select from the Windows keyboard does not work in currently-supported operating systems.

InTouch HMI View applications and DDE support

NetDDE is not supported for InTouchView applications.

By design, an InTouchView application does not serve data to any other source, including InTouch HMI itself. When WindowViewer starts, it verifies if the application is an InTouchView application. When WindowViewer detects an InTouchView application, it does not register to become a DDE server. Industrial Graphics make use of the client layer when accessing InTouch tags, and appear as a third-party client trying to access WindowViewer as a data server. As a result, Industrial Graphics cannot communicate with InTouch tags when used with an InTouchView license.

In Industrial Graphics, **InTouch:<tagname>** is still a valid method of referring to an InTouch tag on a local node.

InTouch HMI support for Windows user account control

System Platform 2023 R2 with InTouch HMI 2023 R2 supports User Account Control-enabled operations on run-time nodes.

Operating system notes: Application Server

- The Bootstrap, IDE, and Galaxy Repository are supported by the following language versions of Microsoft operating systems: English, Japanese, Simplified Chinese, German, and French. The Galaxy Repository is also supported by the English, Japanese, Simplified Chinese, German, and French versions of Microsoft SQL Server.
- Upon installation, the selected language and regional settings must match those of the intended operating locale and must use the collation of SQL Server that matches the operating system locale.

Using Application Server with supported Windows operating systems

This section describes specific behaviors and restrictions when using the supported versions of Windows and Windows Server operating systems with Application Server.

- The DDESuiteLink Client connection to the local Communication Driver (also called the OI Server or DAServer) using Local DDE is supported ONLY when the Communication Driver is configured as "Desktop Mode (Must Start from Command line)" and activated from its executable file or launched from InTouch HMI. On Windows and Windows Server operating systems, Local DDE is NOT supported when the Communication Driver is activated in the Operations Control Management Console.
- For toolkits such as the Application Object Toolkit, GRAccess Toolkit, and MXAccess Toolkit to function properly, you may need to start the toolkit by right-clicking on the file and then clicking **Run As Administrator**.

Operating system notes: Historian Client

User Account Control can be enabled when the Historian Client application is running as non-administrator.

.NET Framework requirements and compatibility

IMPORTANT: System Platform 2023 R2 leverages Microsoft .NET Framework 4.8. Multiple versions of the .NET Framework can coexist, if other applications on the same machine have dependencies on other .NET versions.

All user-supplied .NET code that runs in the context of InTouch HMI and Application Server requires .NET Framework 4.8. Although .NET Framework 4.8 is highly compatible with applications that are built with earlier .NET Framework versions, you may have to update your scripts.

In scripts for Industrial Graphics and/or Application Server objects, some .NET codes could fail if proper text encoding is not used. This may cause a script to exit without completion. The UTF8Encoder is the default BinaryStream decoder in .NET Framework 4.5 and later. To enable a script to decode ASCII XML data, for example, insert the following snippet:

```
BinaryReader streamReader = new BinaryReader(ms, new ASCIIEncoding());
```

To learn more about changes introduced in different versions of the .NET Framework, refer to the following Microsoft resources:

- What's New in .NET Framework
  https://docs.microsoft.com/en-us/dotnet/framework/whats-new/
- What's obsolete in the .NET Framework class library
  https://docs.microsoft.com/en-us/dotnet/framework/whats-new/whats-obsolete
- Migration Guide to the .NET Framework 4.8, 4.7, 4.6, and 4.5
  https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/
- .NET Framework 4 migration issues
  https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/ee941656(v=vs.100)

Considerations for SQL Server

SQL Server is required for Application Server, InTouch, and Historian Server. We recommend that you install and configure the supported SQL Server version before you begin the System Platform installation program. If you select SQL Server Express during System Platform installation, it will be installed automatically (applicable to small installations only).

The System Platform installer will install all prerequisites. If SQL Server is not installed, SQL Server 2019 Express Core is installed automatically. If you wish to install a full version of SQL Server, exit the installation program, install a supported SQL Server version, then resume the installation.

Other considerations are:

- **Alarm DB Logger:** To use the Alarm DB Logger with SQL Server Express, you need to change the default authentication mode from Windows-based to Mixed Mode.
- **SQL Server Configuration Rights:** While installing InTouch HMI for System Platform, if the logged-on user performing the installation is not a SQL Server administrator, the Config SQL dialog box appears and requests SQL Server administrator credentials.

- **Maximum Server Memory:** The System Platform installation process will attempt to adjust Maximum Server Memory if it has the appropriate rights to configure SQL Server. After installing SQL Server, you can download SQL Server Management Studio and use it to confirm that the Maximum Server Memory is configured to approximately 65% of the total available RAM. By default SQL Server does not clamp this setting. If you do not have SQL Server Management Studio installed, you can download it from:

  https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15
- **The MSSQL Server user account** is not supported for the SQL Server Service. Instead, configure SQL Server to run as the local system or Network Service account. Named instances are not supported.
- **Migrating SQL Server Versions:** System Platform now requires a 64-bit version of SQL Server. Beginning with SQL Server 2016 (13.x), SQL Server is only available as a 64-bit application. Supported SQL Server versions at the time of release of System Platform 2023 R2 include:
  - SQL Server 2016 SP3 or SP3 with all cumulative updates
  - SQL Server 2017 with all cumulative updates
  - SQL Server 2019 with all cumulative updates
  - SQL Server 2022 with all cumulative updates

  Important! A 32-bit version of SQL Server cannot be upgraded to a 64-bit version through SQL Server Setup. However, you can back up or detach a database from a 32-bit instance of SQL Server, and then restore or attach the database to a 64-bit instance. For more information, see Supported Version and Edition Upgrades for SQL Server 2017.
- For more information and helpful procedures about upgrading/migrating SQL Server, see the following Microsoft references:
  - Upgrade SQL Server
  - ALTER DATABASE (Transact-SQL) Compatibility Level
- **SQL Server Rights Requirements:** SQL Server no longer automatically creates the BUILTIN\Administrators role delivered in earlier, unsupported versions SQL Server. The Application Server installation process will create the necessary operating system user group (aaAdministrators) and SQL Server role. This automated process will provide the rights required to allow operations within the Galaxy Repository without the need for blanket BUILTIN\Administrator rights. The aaAdministrators group must be present and enabled. If you accidentally delete the aaAdministrators group from the Windows operating system, you can run either of two options to restore it:
  - Run the Change Network Utility from the AVEVA folder in the Windows Start menu.
  - Run the SQL Access Configurator from the AVEVA folder in the Windows Start menu.

    If you accidentally delete the aaAdministrators group from the SQL Server security logins, you must run the SQL Access Configurator to restore it. Refer to the "About the System Platform Network Account" in the *Application Server User Guide* for additional information.

Considerations for SQL Server Express

- SQL Server Express is supported for use on an Application Server or Historian node, but is recommended for use only in small or development configurations. SQL Server 2019 Express Core is automatically installed when you select Application Server (with the Galaxy Repository selected) or the Historian, if, at time of installation, no other SQL Server elements are installed on the computer.

- The computing capacity of SQL Server Express is limited to the lesser of one CPU socket or four processor cores.
- For InTouch HMI-only installations, SQL Server is no longer a requirement since InTouch HMI has been reconfigured to work without a Galaxy Repository.

Additional SQL Server notes for Application Server

- Only 64-bit versions of SQL Server are supported.
- If multiple versions of SQL Server are installed, the one used as the Galaxy Repository must be the default instance. Named instances are not supported.
- The Galaxy Repository locks the SQL Server maximum memory usage to 65% of the computer's physical memory.
- TCP/IP must be enabled on the computer hosting a SQL Server database. The TCP/IP protocol setting can be verified from the SQL Server Network Configuration under SQL Server Configuration Manager.
- To use the Alarm DB Logger with SQL Server Express, you need to change the default authentication mode from Windows-based to Mixed Mode.

AVEVA Group plc
High Cross
Madingley Road
Cambridge
CB3 0HB
UK

Tel +44 (0)1223 556655

www.aveva.com
To find your local AVEVA office, visit **www.aveva.com/offices**