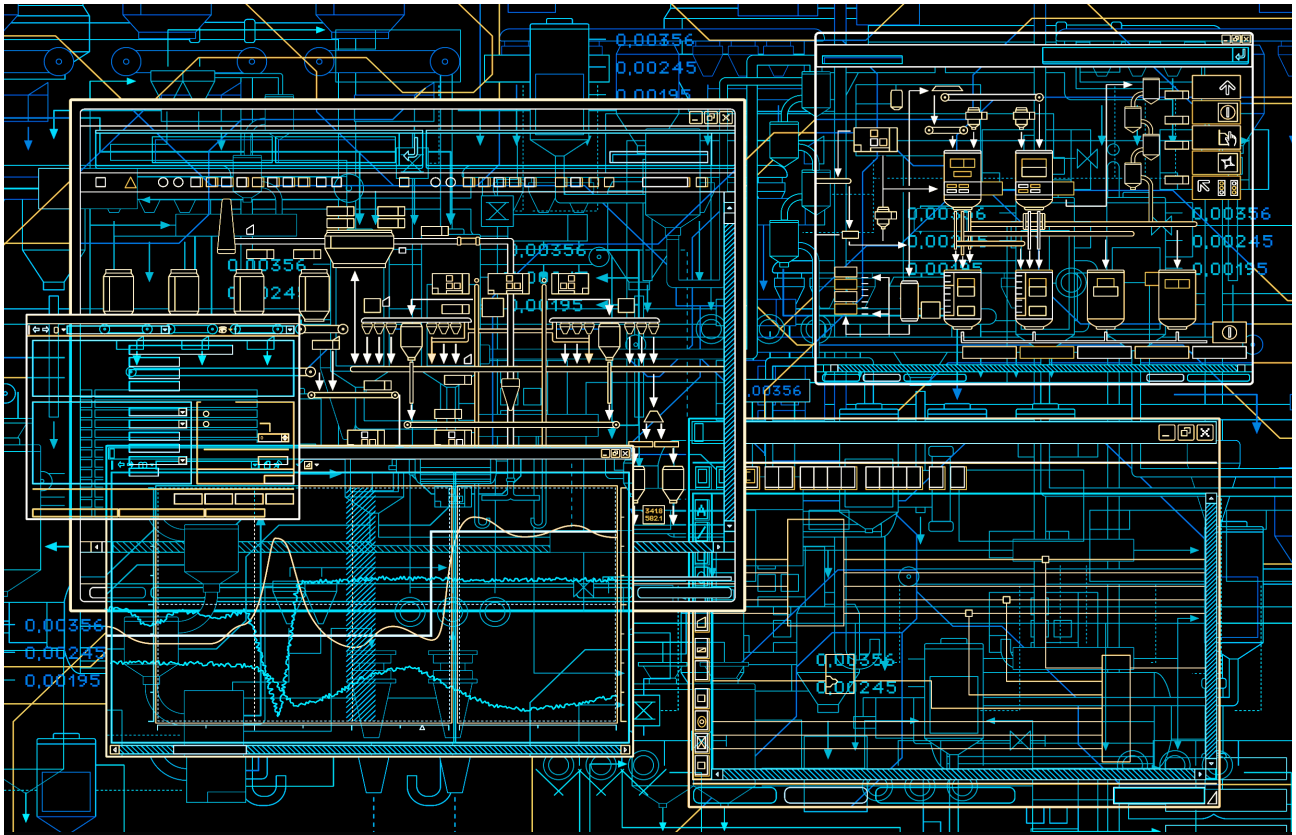


System 800xA Information Management

Configuration

System Version 5.1

Power and productivity
for a better world™

System 800xA Information Management

Configuration

System Version 5.1

NOTICE

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all of the ABB products referenced in this document support all of the features of the described or referenced standard. In order to determine the specific features supported by a particular ABB product, the reader should consult the product specifications for the particular ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license. This product meets the requirements specified in EMC Directive 2004/108/EC and in Low Voltage Directive 2006/95/EC.

TRADEMARKS

All rights to copyrights, registered trademarks, and trademarks reside with their respective owners.

Release: July 2015
Document number: 3BUF001092-510 D

Table of Contents

Section 4 - Configuring Calculations

Section 5 - Configuring History Database

Section 6 - Configuring Log Sets

Section 7 - Alarm/Event Message Logging

Section 8 - Historizing Reports

Section 9 - Historical Process Data Collection

Section 10 - Profile Data Collection

Section 11 - Configuring the Archive Function

Section 12 - Consolidating Historical Data

Section 13 - History Database Maintenance

Section 14 - Open Data Access

Section 15 - Configuring Data Providers

Section 16 - Authentication

Section 17 - System Administration

Appendix A - Extending OMF Domain to TCP/IP

About this User Manual

Any security measures described in this User Manual, for example, for user access, password security, network security, firewalls, virus protection, etc., represent possible steps that a user of an 800xA System may want to consider based on a risk assessment for a particular application and installation. This risk assessment, as well as the proper implementation, configuration, installation, operation, administration, and maintenance of all relevant security related equipment, software, and procedures, are the responsibility of the user of the 800xA System.

This User Manual provides instructions for configuring Information Management functionality for the 800xA System. This includes:

- Data access via data providers and Open Data Access.
- SoftPoint Services.
- Calculations.
- History Services for:
  - Logging numeric (process) data collected from object properties.
  - Logging alarms/events.
  - Storing completed reports executed via the application scheduler.
  - Archival of the aforementioned historical data.
  - Consolidation of message, PDL, and property logs.

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1Systems.

These Information Management functions comprise a repository of process, event and production information, and provide a single interface for all data in the system.

This book is intended for application engineers who are responsible for configuring and maintaining these applications. This book is not the sole source of instruction for this functionality. It is recommended that you attend the applicable training courses offered by ABB.

User Manual Conventions

Microsoft Windows conventions are normally used for the standard presentation of material when entering text, key sequences, prompts, messages, menu items, screen elements, etc.

Warning, Caution, Information, and Tip Icons

This User Manual includes Warning, Caution, and Information where appropriate to point out safety related or other important information. It also includes Tip to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

Electrical warning icon indicates the presence of a hazard that could result in *electrical shock.*

Warning icon indicates the presence of a hazard that could result in *personal injury.*

Caution icon indicates important information or warning related to the concept discussed in the text. It might indicate the presence of a hazard that could result in *corruption of software or damage to equipment/property.*

Information icon alerts the reader to pertinent facts and conditions.

Tip icon indicates advice on, for example, how to design your project or how to use a certain function

Although Warning hazards are related to personal injury, and Caution hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, fully comply with all Warning and Caution notices.

Terminology

A complete and comprehensive list of terms is included in *System 800xA System Guide Functional Description (3BSE038018*)*. The listing includes terms and definitions that apply to the 800xA System where the usage is different from commonly accepted industry standard definitions and definitions given in standard dictionaries such as Webster's Dictionary of Computer Terms. Terms that uniquely apply to this User Manual are listed in the following table.

Released User Manuals and Release Notes

A complete list of all User Manuals and Release Notes applicable to System 800xA is provided in *System 800xA Released User Manuals and Release Notes (3BUA000263*)*.

System 800xA Released User Manuals and Release Notes (3BUA000263)* is updated each time a document is updated or a new document is released. It is in pdf format and is provided in the following ways:

- Included on the documentation media provided with the system and published to ABB SolutionsBank when released as part of a major or minor release, Service Pack, Feature Pack, or System Revision.
- Published to ABB SolutionsBank when a User Manual or Release Note is updated in between any of the release cycles listed in the first bullet.

A product bulletin is published each time *System 800xA Released User Manuals and Release Notes (3BUA000263*)* is updated and published to ABB SolutionsBank.

Section 1 Configuration Overview

Information Management applications comprise a repository of process, event and production information. Information Management functionality includes:

- **History Services** for collection, online and offline storage, consolidation, and retrieval for process/lab data, alarms/events, and reports.

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

- **Real-time Database Services** for access to real-time process data from AC 800M controllers, and other ABB and third-party control systems when the applicable connectivity components are installed. Real-time database services also support the configuration of SoftPoints to hold application-generated data not directly connected to any process. SoftPoints are accessible by all other functions in the system. **Calculation Services** is used to apply calculations to other objects in the system, including both process and SoftPoint objects.
- **Display and Client Services** includes the following desktop client applications for data access: DataDirect (for Excel Data Access refer to Installing Add-ins in Microsoft Excel on page 295), Display Services, and Desktop Trends. These desktop tools may run directly on the Information Management server, or on remote computer clients. The Information Management server also supports data access by third-party applications such as Crystal Reports when Open Data Access is installed.
- **Application Scheduler**. This supports scheduling and execution of jobs implemented as system plug-ins. This includes:
  - Reports created with DataDirect, or a third-party report building package such as Crystal Reports.
  - Consolidation of Production Data and Message logs.

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

– Event-driven collection for process and lab data (property) logs.

– Calculations created via the Calculation Services.

Configuration Requirements

The following points briefly describe the configuration requirements and work flow for the Information Management functions.

- Service Providers
  The following Information Management functions are managed by service providers in the 800xA system: History, Archive, Calculations, SoftPoints, Scheduling, and Open Data Access Services. Verify all service providers are properly installed and enabled before using the Information Management functions. This is described in Section 2, Verifying Services.
- Configuring Data Access
  There are two methods by which client applications access data from the 800xA system. The Information Management Display and Client Services use the data providers supplied with ABB Data Services. Third-party tools such as Crystal Reports, and Microsoft Excel (without DataDirect plug-ins), use the ODA server. To configure data providers, refer to Section 15, Configuring Data Providers. To configure Open Data Access, refer to Section 14, Open Data Access.
- SoftPoint Configuration
  SoftPoint Services is used to configure and use internal process variables not connected to an external physical process signal. To configure SoftPoints, refer to Section 3, Configuring SoftPoints.
- Calculations Configuration
  Calculations Services is used to configure and schedule calculations that are applied to real-time database objects, including both SoftPoints and actual process points. To configure calculations, refer to Section 4, Configuring Calculations.
- History Configuration
  History Services database configuration requires a History database instance. The default instance is sized to meet the requirements of most History applications. It may be necessary to drop the default instance and create a new

larger instance depending on performance requirements. Refer to Section 5, Configuring History Database. Once the Oracle database instance is created, use the following procedures to configure historical data collection, storage, and archival for the system:

– **Configuring Property Logs**
Process data collection refers to the collection and storage of data from aspect object properties. This includes live process data, SoftPoint data, and lab data (programmatically generated, or manually entered). The objects that perform process data collection and storage are called *property logs*. Process data collection may be implemented on two levels in the 800xA system. The standard system offering supports short-term data storage with *trend logs*. When Information Management History Server is installed, long-term storage and permanent offline storage (archive) is supported with *history logs*. For instructions on how to integrate and configure trend and history logs, refer to Section 9, Historical Process Data Collection.

– **Alarm/Event Message Logging**
All alarm and event messages for the 800xA system, including Audit Trail messages, are collected and stored by the 800xA System Message Server. This provides a short-term storage facility with the capacity to store up to 50,000 messages. If the system has Information Management History Server installed, the messages stored by 800xA System Message Server may be forwarded to a message log for extended online storage. This message log can store up to 12 million messages. Refer to Section 7, Alarm/Event Message Logging.

– **Archive Configuration**
The archive function supports permanent offline storage for historical data collected in property, message, and report logs, as well as the 800xA System alarm/event message buffer. When a history log becomes full, the oldest entries are replaced by new entries, or deleted. Archiving copies the contents of selected logs to a designated archive media. For instructions on configuring archiving refer to Section 11, Configuring the Archive Function.

– **Configuring Log Sets**
Log sets allow one command to start or stop data collection for a number

of property, message, or report logs simultaneously. For instructions on configuring log sets, refer to Section 6, Configuring Log Sets.

– **Configuring Report Logs**
Report logs hold the finished output from reports scheduled and executed via the Application Scheduler and the Report action plug-in. Completed reports may also be stored as completed report objects in the Scheduling structure. In either case the Report action must be configured to send the report to a report log, or a completed report object. This is described in the section on creating reports in *System 800xA Information Management Data Access and Reports (3BUF001094*)*. For instructions on configuring the report log to store finished reports, refer to Section 8, Historizing Reports.

– **Consolidating Historical Data**
Historical process data, alarm/event data, and production data from remote history servers can be consolidated on a dedicated consolidation node. For instructions on consolidating historical data, refer to Section 12, Consolidating Historical Data.

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

– **History Database Management**
The History Services database may require some configuration and management, before, during, and after configuration of History Services. For example, it may be necessary to drop and then re-create the history database, allocate disk space for storage of historical data, or stagger data collection for property logs to optimize performance. These and other history administration procedures are described in Section 13, History Database Maintenance.

Section 2 Verifying Services

This section consists of the following topics:

Information Management Service Providers

The following Information Management functions are managed by service providers in the 800xA system:

- History.
- Archive.
- Calculations.
- SoftPoints.
- Scheduling.

The Service Provider represents the service running on a specific node. For example, if Calculation Services is running on more than one node, then a Calculation Service Provider is required for each server where it will run.

Service Providers are located under their respective *service* categories in the Service structure. Service Providers are contained within Services Groups. For some 800xA services, the Service Group structure supports redundancy by hosting redundant Service Providers. The only Information Management services that support redundancy are SoftPoints and Calculations.

The service group/service provider objects for History, Archive, Calculations Application Scheduler, and Open Data Access are automatically created for their respective Information Management servers when the Process Administration Service (PAS) is initialized during the post installation set-up for the Information Management server.

Calculations and Application Scheduler may also be installed and run on other types of server nodes in the 800xA system (not Information Management servers). In this case, the service group/service provider objects for those services must be created manually for each of those non-Information Management servers. This procedure is described in this section.

For SoftPoint Server, the service group/service provider object are created when a SoftPointSoftPoint Generic Control Network object is created in the Control structure. Other SoftPoint server set-up is also required. This is described in SoftPoint Set-up Requirements on page 40.

All procedures related to service group/service provider objects are done via the Plant Explorer. The remainder of this section describes how to open the Plant Explorer workplace, how to verify that these objects exist, create the objects if necessary, and enable/disable a service provider.

Opening Plant Explorer

Most configuration procedures described in this book are done using the Plant Explorer workplace. To start the Plant Explorer workplace:

1. From the Windows task bar, choose: **Start>Programs>ABB Industrial IT 800xA>System>Workplace**.
2. Select a System and then select the **Plant Explorer Workplace** from the list of available workplaces.

Checking Service Group/Service Provider Objects

Service Group/Service Provider objects are located under their respective Service containers in the Service structure. One Service Group/Service Provider set is required for each node where that service will run, Figure 1. Additional considerations for the History server are provided in History Server on page 36.

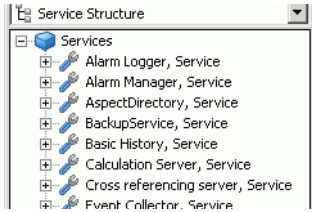


Figure 1. Example, Service Group/Service Provider Objects

After restoring a system containing IM logs a check should be done that the Service Group isn't missing for the IM log in Log Configuration. If the Service Group is missing a new group has to be configured.

Creating Service Group/Service Provider Objects

To create service group and service provider objects for a node:

1. Create the service group object under the applicable service container in the Service structure for example, Calculation Server.
   a. Select the service container (Calculation Server, Service) and choose **New Object** from the context menu.
   b. Assign a name to the new Service Group object. Typically, the name is based on the node name, for example IM_ROC78. Click **Create**.
2. Create the service provider object under the new service group object.
   a. Select the new Service Group object (for example, IM_ROC78, Service Group) and choose **New Object** from the context menu.
   b. Assign a name to the new Service Provider object. Typically, the name is based on the node name, for example IM_ROC78. Click **Create**.
3. Configure the service provider object to point to the node where the service (in this case Calculations) must run (reference Figure 2).
   a. Select the Service Provider object in the object browser.
   b. Select the Service Provider Definition aspect from the object's aspect list.
   c. Select the Configuration tab.

d. Use the Node pull-down list to select the node where the service will run.

e. Check the **Enabled** check box.

f. Click **Apply**.

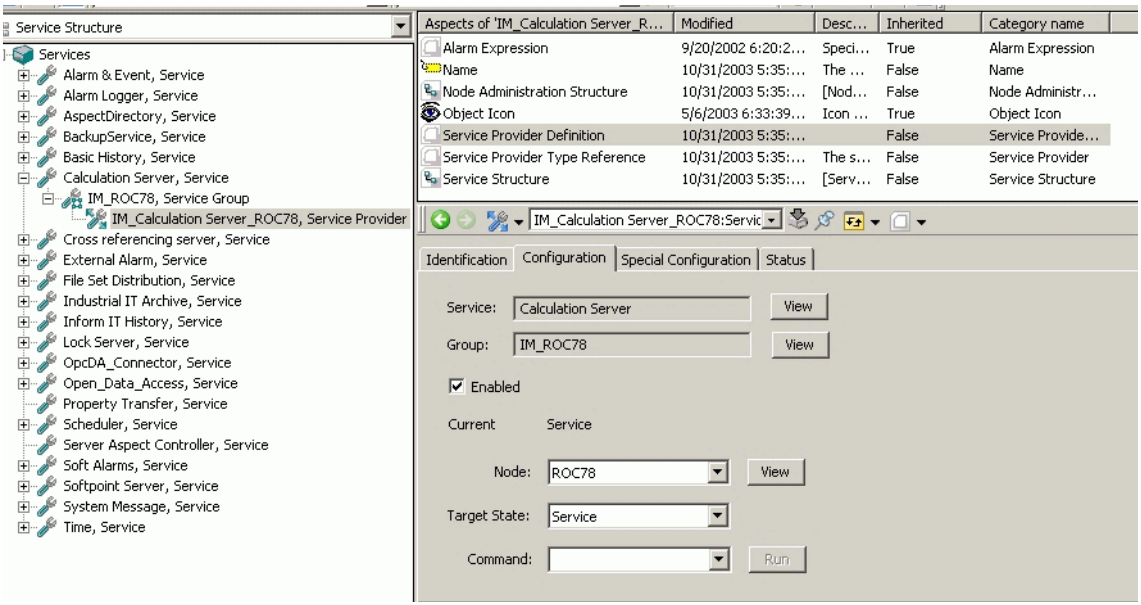


Figure 2. Configuring a Calculation Service Provider

History Server

History is installed in two parts: the operator trend functionality which is installed as standard and the Information Management History Server which is optional. Both functions are implemented as services in the Service Structure. The service categories are **Basic History, Service**, and **Inform IT History, Service** respectively.

Both of these service categories require a Service Group/Service Provider set for each node where the Information Management History Server runs. The Service Provider represents the service running on a specific node. An example is shown in Figure 3. The Service Group structure is intended to support redundancy for certain services (for instance Connectivity Servers). Redundancy is not supported for

Information Management History Server, so there is always a one-to-one correspondence between the History Service Group and its History Service Provider.

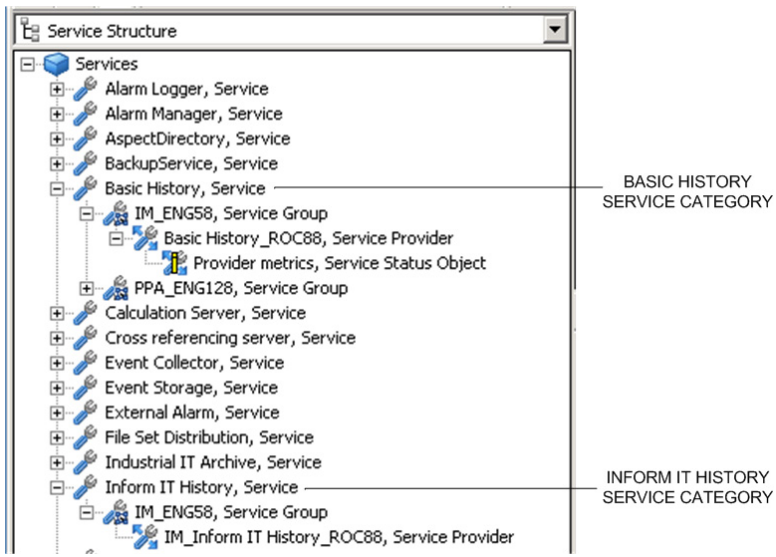


Figure 3. Basic and Information Management History Service Category

A Service Group/Service Provider set is automatically created in the Service Structure for the node when PAS is initialized during post installation.

If a service group/service provider object set needs to be created, use the prefix **Inform IT History** followed by a unique name, for example, **Inform IT History_Basic_eng62**, when specifying the name for the Information Management History Service Provider.

Inform IT History Service Providers are also represented in the Node Administration structure under their respective nodes. This is illustrated in Figure 4.

In the Node Administration structure, the History Service Provider marks the location of the aspect objects related to History configuration. The **InformIT History Object** under the History Service Provider is a container for history object groups - one group for each kind of History object: Log Sets, Message Logs, Report Logs, and so on. Archive objects are located under the

Archive Service. The History objects must be instantiated under the applicable group. For example, message log objects are instantiated under the **Message Logs** group.

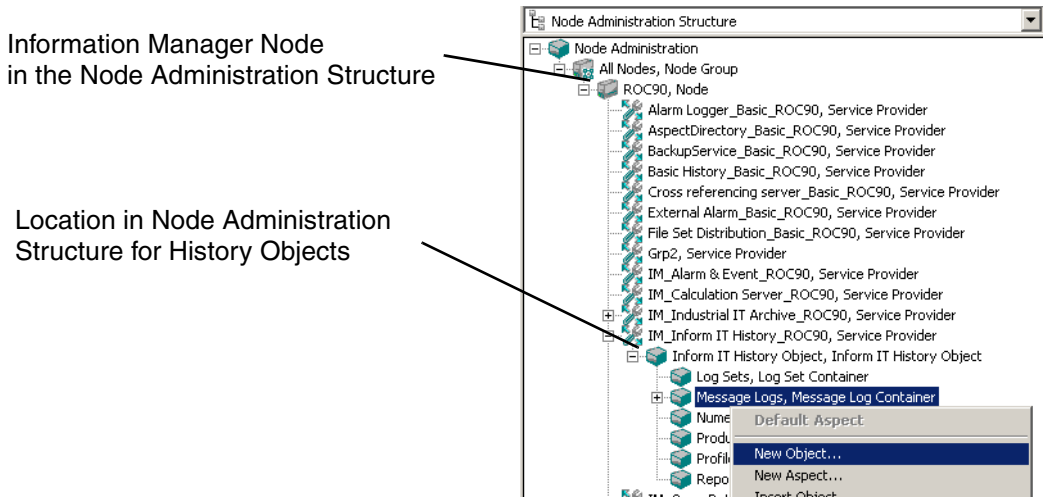


Figure 4. Adding a Message Log in the Node Administration Structure

Section 3 Configuring SoftPoints

SoftPoint Services is used to configure and use internal process variables not connected to an external physical process signal. Once configured, the SoftPoints may be accessed by other 800xA system applications as if they were actual process points. For example, SoftPoint values may be stored in property logs in History Services. Reporting packages such as Crystal Reports may access SoftPointSoftPoints for presentation in reports. Desktop tools such as Desktop Trends and DataDirect can read from and write to SoftPoints.

SoftPoint alarms can be configured and are directly integrated with system alarms and events. Engineering unit definitions for SoftPoints include minimum/maximum limits and a unit descriptor.

SoftPoint Services can be configured as redundant on a non-redundant Information Manager Server pair or on a redundant connectivity server. For greatest efficiency between Calculations and SoftPoints, put the primary Calculation Server on the same machine as the primary SoftPoint Server.

The SoftPoint Services software may run on multiple servers. Each SoftPoint server can have up to 2500 SoftPoint objects. Each SoftPoint object can have up to 100 signals; however, the total number of signals per server cannot exceed 25,000.

Data types supported are: Boolean, integer (32bit), single precision floating point (32 bit) and string. Also, double precision floating point (64 bit) is supported as an extended data type.

The SoftPoint object types specify the signal properties for the SoftPoint objects. This includes signal ranges for real and integer signals, alarm limits, event triggers, and whether or not the signals will be controllable. The process objects inherit the properties of the object types from which they are instantiated.

SoftPoint Set-up Requirements

SoftPoints are instantiated in the Control Structure under a SoftPoint Generic Control Network object. This object establishes the connection with a specific SoftPoint server. Configuration of the SoftPoint Generic Control Network and related objects is completed as a post installation procedure and should already be done. This will automatically configure the following:

- SoftPoint service group object and service provider object(s).
- virtual control network by which other system applications (History, desktop applications and so on) can access the SoftPoint data on that node.
- Source Definition aspect for each network. This is used to deploy the SoftPoints on their respective nodes.
- integrates the SoftPoint alarm and event messages into the 800xA system message service.

To support historical data collection for SoftPoints, create at least one History Source object under the SoftPoint Generic Control Network object in the Control structure. Instructions for adding History Source objects are provided in Configuring Node Assignments for Property Logs on page 189.

Configuring SoftPoints

Configure SoftPoints via the Plant Explorer. SoftPoint objects are instantiated as actual process objects in the Control Structure. They are instantiated from the SoftPoint object type templates configured in the Object Type Structure. SoftPoint objects must be created in the Control structure as sub-objects of the SoftPoints container for the applicable SoftPoint Generic Control Network. Once created, these objects can be inserted into other structures, for example the Functional structure.

- Creating a SoftPoint object adds a signal of each signal type in the SoftPoint object type.
- SoftPoint objects cannot be copied, cut, pasted, moved or renamed. Deleting an object deletes all included signals.
- When dragging an input/output from a SoftPoint property (in a SoftPoint object type) to a Calculation aspect, the selected information is not correct and needs to be edited. This is not a problem when dragging an input/output from an instantiated SoftPoint object. Use relative naming in this case.

Some SoftPoint properties are configured on an individual basis for instantiated SoftPoint objects in the Control structure. For example the event text associated with limiter trip points is configured for real and integer signal types.

Further, most of the default settings configured in the SoftPoint object types may be adjusted on an individual basis for SoftPoint objects in the Control structure.

When creating new SoftPoint objects, or making changes to existing SoftPoint objects, the new configuration will not go into effect until the changes are deployed.

When changes are applied to a SoftPoint object type, those changes are propagated to all SoftPoint objects that have been instantiated from that object type. A warning message is generated any time a change is applied to a SoftPoint object type, even if there are no instantiated objects for that type. Disable this warning message while configuring SoftPoint object types, and enable it only after the SoftPoint configuration has been deployed. To enable/disable this warning message, refer to Enabling/Disabling Warning for Changes to Object Types on page 72.

The basic procedures covered in this section are:

Adding a New SoftPoint Object Type

Create SoftPoint object types in the Object Type structure. These object types will be used as templates for instantiating actual SoftPoint objects in the Control structure. To add a new SoftPoint object type:

1. In the Plant Explorer, open the **Object Type Structure**

2. Select the **SoftPoint Object Types** group and choose **New Object** from the context menu.

3. In the **New Object** dialog click the **Common** tab and select **SoftPoint Process Object Type,** Figure 5.

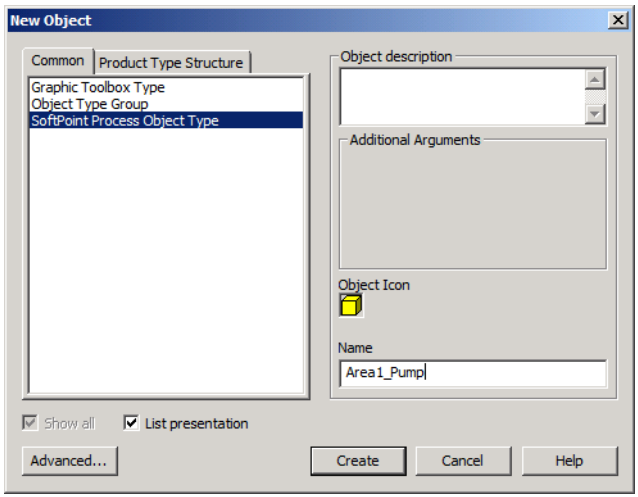


Figure 5. New Object Dialog

4. In the New Object dialog, assign a name to the new object, then click **Create**. This adds the new object type to the SoftPoint Object Type group, Figure 6.

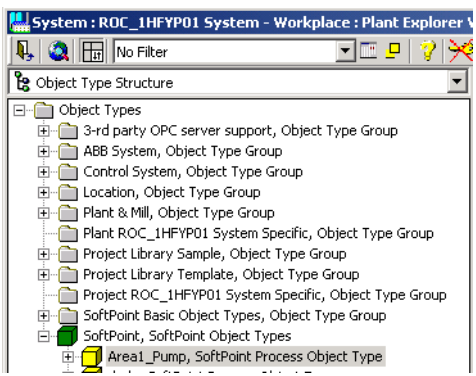


Figure 6. New Object Type Added

Configuring a SoftPoint Object Type

SoftPoint object types are configured via the Process Object Configuration aspect. This aspect is used to add signal types for the SoftPoint object type.

To display the configuration view for this aspect,

1. Select the new SoftPoint object type in the Object browser (left hand pane).
2. Then click on the **Process Object Configuration** aspect in the right hand pane, Figure 7.

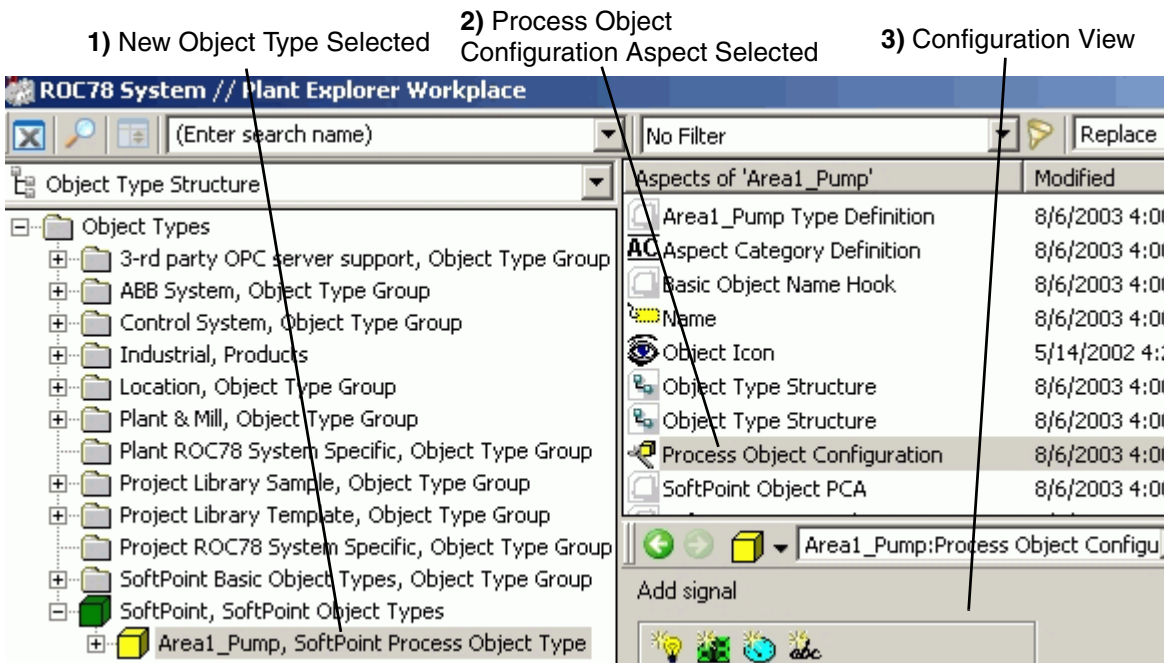


Figure 7. Configuring the New Object Type

Adding Signal Types

SoftPoint signals are specified through signal types. For example, a pump may require a binary signal to represent the pump's state and a real signal to represent the pump speed. A maximum of 128 signal types can be added to a SoftPoint object type. The available signal types are listed in Table 1.

Table 1. Signal Types

Signal Type	Object Type structure icon
Binary	
Integer	
Real	
String	

Signals must be added and deleted in the Object Type structure. Signals can not be added or deleted directly on SoftPoint objects in the Control structure. Do not copy, cut, paste, move or rename signal types. When a signal type is added to a SoftPoint object type, a signal of this signal type is added to all SoftPoint objects based on the SoftPoint object type. Deleting a signal type deletes all signals of this type.

The Add Signal buttons are located in the upper left part of the SoftPoint Process Object Configuration aspect, Figure 8.

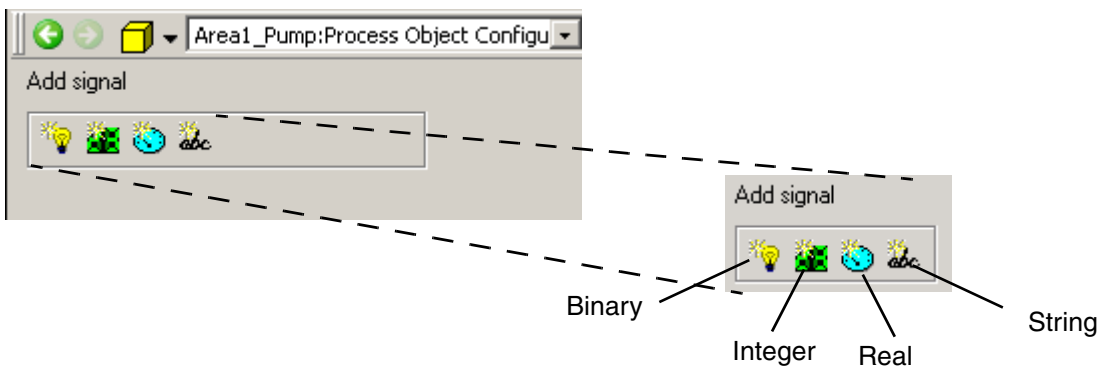


Figure 8. Add Signal Buttons

For example, to add a real signal type,

1. Click the Real Signal button. This displays the New Real Signal Type dialog.
2. Enter a logical name, for example **RPM**.

3. Click **OK**.
4. Repeat this procedure to add more signal types as required (up to 128 total).

Expand the object type's branch to refer to the added signal types, Figure 9.

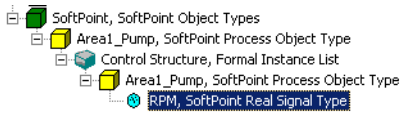


Figure 9. Signals Added

How to Delete a Signal Type

Deleting a signal type deletes all signals of this signal type from all SoftPoint objects instantiated from the object type. To delete a signal type:

1. Select the signal type from the Object Type structure.
2. Press the **Delete** key on the keyboard or right-click the signal type and select **Delete** from the context menu.

Deleting a SoftPoint Object Type

SoftPoint object types cannot be deleted once they have created objects based on that specific type. Deleting a SoftPoint object type deletes all included signal types.

To delete a SoftPoint object type:

1. Select the Object Type from the Object Type structure.
2. Press the **Delete** key on the keyboard or right-click the object type and select **Delete** from the context menu.

Configuring Signal Properties

Signal properties are configured via the Signal Configuration aspect, Figure 10.

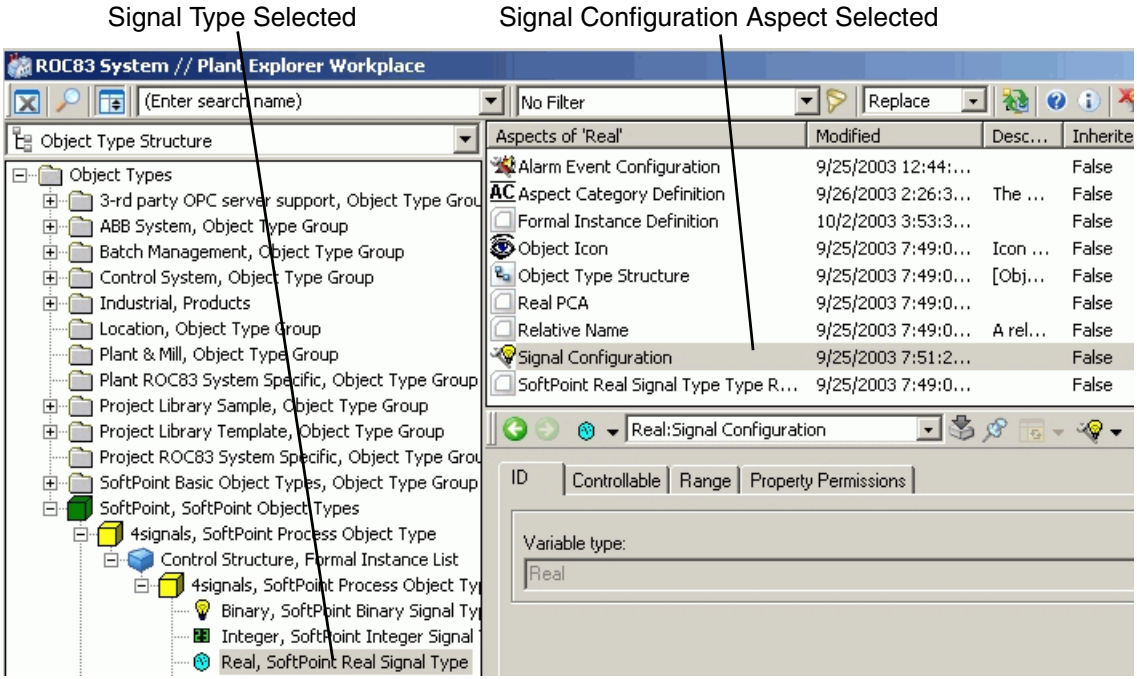


Figure 10. Displaying the Signal Configuration Aspect

Signal Properties

The following signal properties may be configured:

- Real and integer signal types can have an overall range. Refer to Configuring the Overall Range on page 46.
- Real, integer, and binary signal types can be made controllable. Refer to Making a Signal Controllable on page 48.
- Authentication can be configured for all signal types. Refer to Configuring Authentication for Signal Types on page 49.

Configuring the Overall Range

The overall range sets the absolute high and low limits for the signal. The default range is 0 to 100. To change these values:

1. Select the real or integer signal type from the Object Type structure.
2. Select the Signal Configuration aspect.
3. Select the **Range** tab, Figure 11.

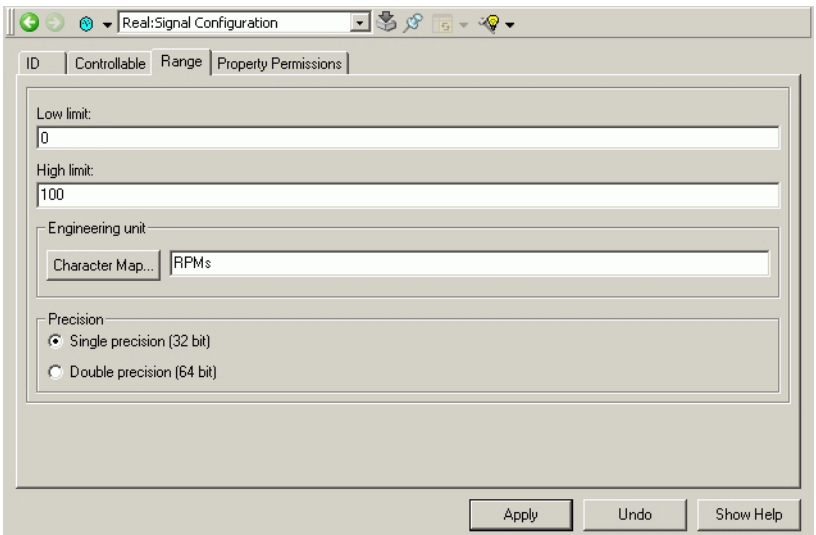


Figure 11. Real Signal - Range Tab

4. Enter low and high limit values in their respective fields.
5. Enter the engineering unit label. This can be up to 12 ASCII characters. Use the Character Map to include special characters. Refer to How to Use Special Characters in the Engineering Units Tag.
6. Specify precision for the signal as **Single** (32-bit) or **Double** (64-bit).
7. Click **Apply** when done.

How to Use Special Characters in the Engineering Units Tag

To select special characters for the engineering unit.

1. Click **Character map**. This displays the Character map dialog, Figure 12.

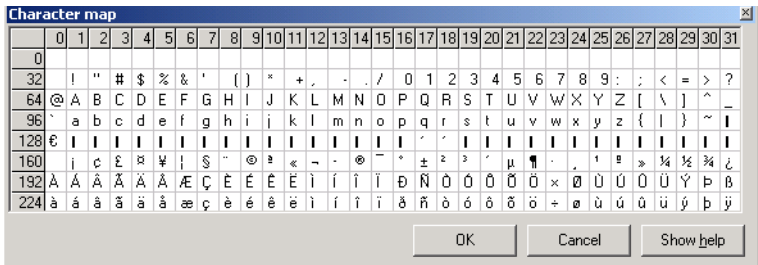


Figure 12. Character Map

2. Select a character.
3. Click **OK**. The Character map dialog box will be closed and the character added to the engineering unit.
4. Repeat step 1-3 to select additional characters.
5. Click **Apply**.

Making a Signal Controllable

Signals are NOT controllable by default. Signals must be made controllable to allow operators to manipulate the value, or to get external applications, including Calculations, to manipulate the value. For real and integer signal types, a control range can be specified. The control range sets boundaries within the overall range within which the signal value may be adjusted.

Refer to the applicable procedure depending on the signal type:

- Making a Binary Signal Controllable on page 48.
- Making Real and Integer Signals Controllable on page 49.

Making a Binary Signal Controllable

To make the binary signal controllable:

1. Select the binary signal type in the Object Type structure.
2. Select the Signal Configuration aspect.
3. Select the **Controllable** tab.

4. Check the **Is Controllable** check box.
5. Specify whether or not to **Log operator actions**.

Log operator actions applies to changes caused by external applications such as calculations as well as operator actions. Therefore, do not check this box unless log changes caused by these applications are required.

Making Real and Integer Signals Controllable

A real or integer signal that is controllable can also have a control range. The control range sets boundaries within the overall range within which the signal value may be adjusted. To make a real or integer signal controllable:

1. Select the signal type in the Object Type structure.
2. Click the **Controllable** tab Signal Configuration aspect.
3. Click the **Is Controllable** check box. This enables the **Advanced** button.
4. Click **Advanced** to display the dialog for setting the operator's control range. Check the **Control Limits** check box and enter Low and High limits, Figure 13. These limits must be within the overall signal range (refer to Configuring the Overall Range on page 46).

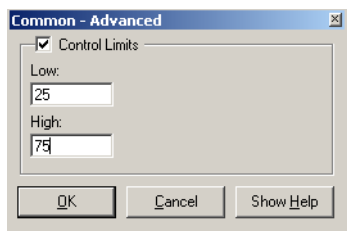


Figure 13. Setting the Control Range

Configuring Authentication for Signal Types

Configure *authentication* for SoftPoint signal types to help meet FDA 21CFR part 11 requirements. Authentication defines how many users must be approved (by entering their user name and password) before an operation can proceed, for example changing a signal value. Authentication may be set to one of three levels:

None (no users) **Reauthentication** (one user), and **Double authentication** (two users). The default is not to require authentication.

Authentication for SoftPoint signal types is configured on an individual basis for each signal property. Instantiated signals inherit authentication from their respective SoftPoint object types. Authentication may be changed for instantiated signals in the Control structure.

To configure authentication for a SoftPoint signal type, Figure 14:

1. Click the **Property Permissions** tab on the Signal Configuration aspect.
2. Select the Property from the property list.
3. Select the Authentication Level from the pull-down list and click **Apply**.

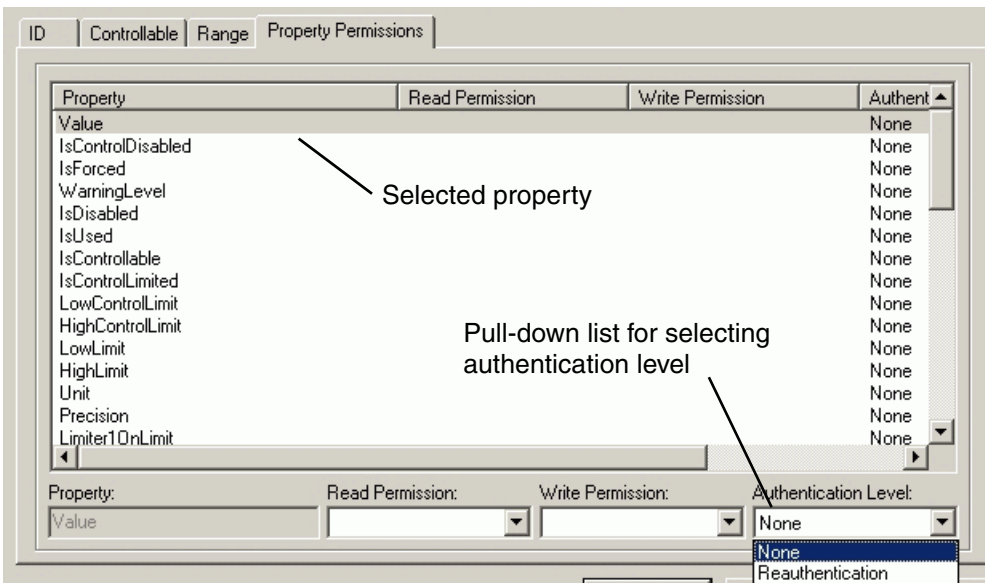


Figure 14. Configuring Authentication for a SoftPoint Signal

Configuring Alarms and Events for Signal Types

Real and integer signals can have up to four trip points (called *limiters*) within the signal's overall range. This is done via the real or integer signal type's Alarm Event

Configuration aspect. The procedure is described in Configuring Limiters for Real and Integer Signals on page 51.

Trip points must occur within the signal range and should be configured BEFORE adding the limiters. Refer to Configuring the Overall Range on page 46.

Binary signals, and limiters for real and integer signals, can be configured to generate an alarm and/or event when a state change occurs and how alarm acknowledgement will occur. This is done via the signal type's Alarm Event Configuration aspect. This procedure is described in Configuring Alarm and Event Handling on page 54.

Configuring Limiters for Real and Integer Signals

Real and integer signals can have up to four trip points within the signal's overall range that, when crossed, generate an alarm and/or event. These points are called *limiters*. To configure limiters:

1. With the signal type selected in the object browser, select the **Alarm Event Configuration** aspect from the object's aspect list, Figure 15.

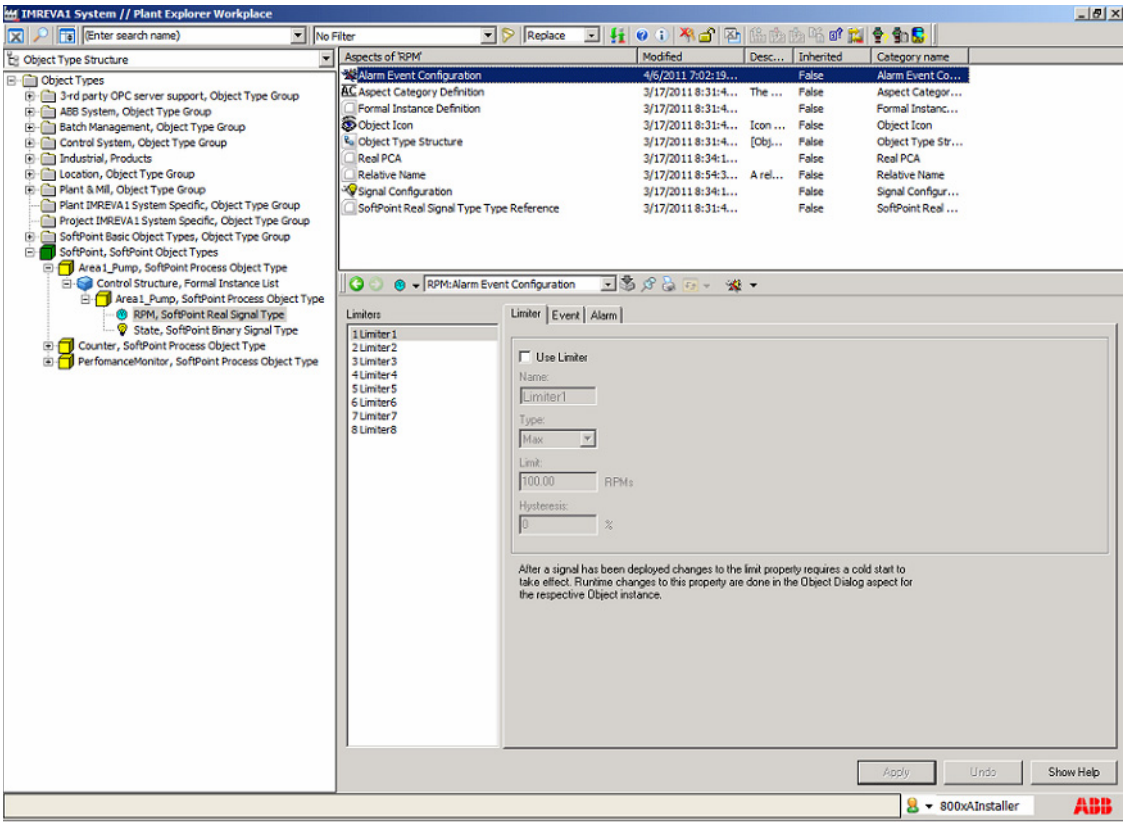


Figure 15. Alarm Event Configuration Aspect for Real and Integer Signal Types

Refer to Figure 16 for steps 2-6.

2. Select **1Limiter1** from the Limiters list under the **Limiter** tab and select the **Use Limiter** check box.

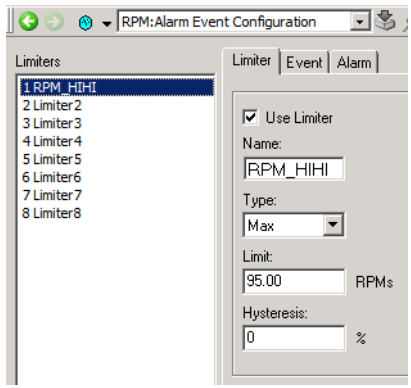


Figure 16. Adding a Limiter

3. Rename **Limiter1** to make it more easily identifiable, for example: **RPM_HIHI**.

4. Specify the **Type** to indicate whether the limiter will trigger an event when the signal rises above a maximum value (**Max**), or drops below a minimum value (**Min**).

5. Enter a **Limit**. This is the threshold that the signal must cross to trip an alarm (drop below if type is Min, or exceed if type is Max).

6. Set the **Hysteresis** to filter spurious alarms caused by signals that fluctuate around the limit. For a real signal type, the hysteresis range is **0.00** to **9.99** expressed as percent of signal range. For integer signal types, use a whole number. The default is 0.

   Events are triggered when the signal passes the limit value regardless of the Hysteresis. In order for a triggered alarm to reset so that subsequent alarms may be triggered, the signal must first re-cross the alarm limit in the opposite direction (decreasing for Max limit, or increasing for Min limit), and continue to decrease or increase until the point established by the Hysteresis is crossed. This requires the Hysteresis to be set to a value greater than zero (0).

   When the Hysteresis is set to zero, the alarm will reset as soon as the signal re-crosses the alarm limit.

7. Click **Apply**.

8. Repeat steps 2-7 to configure up to four limiters. An example of a completed limiter list is shown in Figure 17.

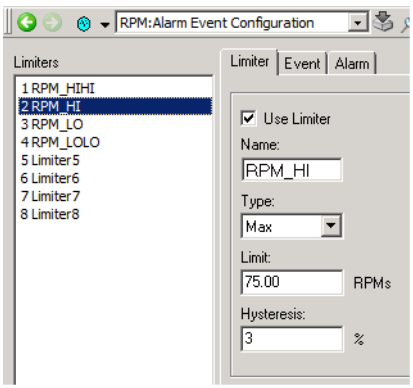


Figure 17. Example, Completed Limiter List

To remove a limiter, select the limiter and clear the **Use Limiter** check box under the **Limiter** tab and click **Apply**.

Alarm/event handling for limiters is configurable. Event generation is enabled for all limiters by default. Refer to Configuring Alarm and Event Handling on page 54.

The event text for each limiter must be configured in the instantiated object's Alarm/Event Configuration aspect in the Control structure. This is covered in Instantiating a SoftPoint Object in the Control Structure on page 58.

Configuring Alarm and Event Handling

Configuration options for binary signal types, and limiters are:

- Whether to generate an event and/or alarm when a state change occurs.
- How alarms will be acknowledged.
- The alarm text group.

Alarms cannot be generated for a signal unless Event handling is enabled for the signal. Configure Event handling and then configure alarm handling parameters.

Events

To specify that an event be generated when a signal state changes (reference Figure 18):

1. Select the signal type object.
2. Select the **Alarm Event Configuration** aspect.

Figure 18 demonstrates how to enable and configure events for a binary signal type. When configuring events for a real or integer signal limiter, the Alarm Event Configuration aspect will have a third tab for selecting the limiter whose event is being configured (Figure 15).

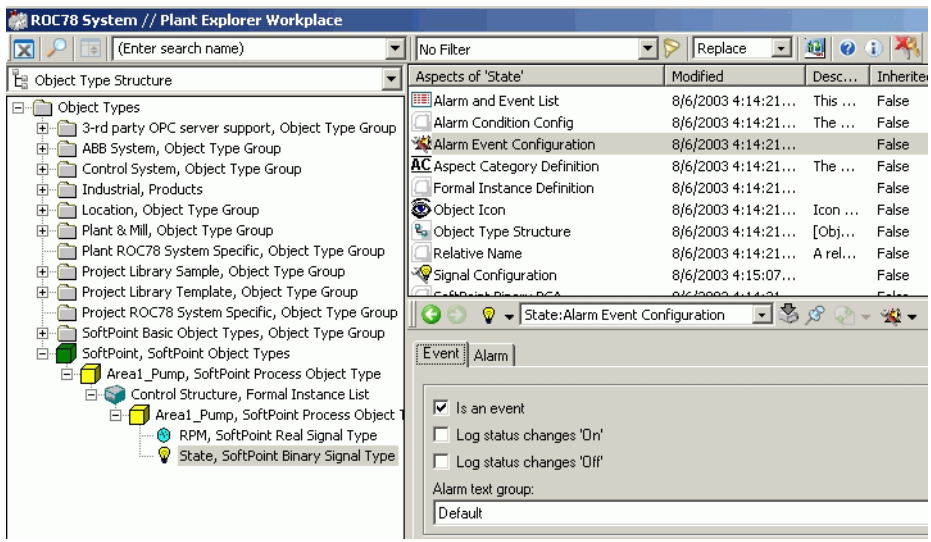


Figure 18. Alarm Event Configuration Aspect Selected

3. Click the **Is an event** check box on the **Event** tab. The default is to have events generated for both state changes On-to-Off, and Off-to-On). This is specified by having both **Log status changes** check boxes unchecked.

To generate events for both state changes, both the **Is an event** and **Is an alarm** check boxes must be checked, and both boxes for log status change unchecked.

4. As an option, use the check boxes to make the event occur exclusively for one state change or the other:

   – Checking **Log status changes 'On'** and NOT checking **Log status changes 'Off'** will result in events only when the state changes from Off to On (rising edge), Figure 19.

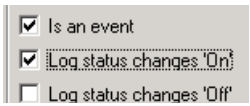


Figure 19. Event Generated Only When State Changes from Off to On

   – Checking **Log status changes 'Off'** and NOT checking **Log status changes 'On'** will result in events only when the state changes from On to Off (falling edge), Figure 20.

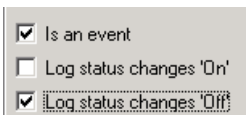


Figure 20. Event Generated Only When State Changes from On to Off

5. As a further option, change the Alarm text group. Alarm text groups specify the messages to be displayed for certain events. All events belonging to the same group will be displayed and/or printed with the same text for a particular change of state. Alarm text groups are configured as described in Defining Alarm Text Groups on page 67. To change the alarm text group for a signal, use the Alarm text group pull-down list. This list contains all Alarm text groups that have been configured in the system.

If alarm handling for the signal is needed, refer to Alarms on page 56.

Alarms

To show a signal in the alarm list, alarm line and printouts when the alarm status is activated, specify that the signal *Is an alarm*. This is done via the **Alarm** tab on the Alarm Event Configuration aspect. Also, the signal must be defined as an event by checking the **Is an event** check box under the **Event** tab of the Alarm Event Configuration aspect (Events on page 55).

The **Alarm** tab is also used to specify how alarm acknowledgement will occur, alarm priority, and a time delay to filter out spurious alarms.

To configure alarm handling for a signal:

1. Make sure the **Is an event** check box is selected on the Events tab.
2. Select the **Alarm** tab and check the **Is an alarm** check box, Figure 21.

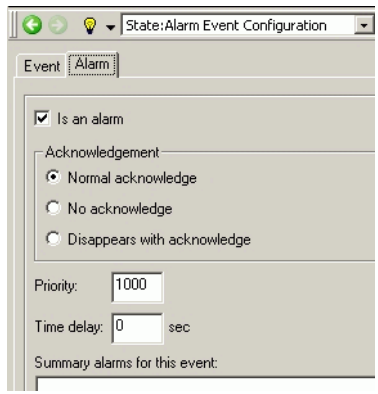


Figure 21. Alarm Tab

3. Select an alarm acknowledge option as described in Table 2.

Table 2. Alarm Acknowledge Options

Option	Description
Normal acknowledge	Show alarm as long as it remains active or unacknowledged.
No acknowledge	Show alarm as long as it remains active.
Disappears with acknowledge	Show alarm as long as it remains unacknowledged. When the alarm is acknowledged, it is removed from the alarm list regardless of the current alarm status. The system will not detect when the alarm becomes inactive. Thus printout is not available for when an alarm became inactive, only when it was acknowledged.

Alarm list colors are set in the Alarm and Event List Configuration aspect.

4. Enter a priority as a number from 1 (lowest) to 1000 (highest). The priority is used to sort the alarm list with the top priority alarm first. The default is 1.
5. Set a time delay. The time delay is the number of seconds the event has to remain active before any event actions take place. This filters out very short events. Entering zero (0) means there is no delay.
6. Click **Apply**.

Instantiating a SoftPoint Object in the Control Structure

SoftPoint objects are instantiated in the Control Structure. The SoftPoint objects inherit most of their operating parameters from the object type from which they are instantiated, and so require very little configuration. Typically all that is required is to specify event texts for limiter signals. Also, most operating parameters that were initially defined for the corresponding object type can be adjusted.

If a SoftPoint signal which has been instantiated is not required, it can be disabled by selecting the **Signal Not Used** check box under the **ID** tab of the Signal Configuration aspect.

Deploy the SoftPoint Generic Control Network to take effect of any changes made for SoftPoint objects and signals.

There are two methods for creating SoftPoint objects. Either create the objects one-object-at-a-time, or perform a bulk instantiation of a specified number of objects based on a selected object type. Refer to the applicable instructions below.

- Creating One Object at a Time on page 58.
- Bulk Instantiation of SoftPoint Objects in the Control Structure on page 60.

Creating One Object at a Time

To create a new SoftPoint object in the Control Structure:

1. In the **Control Structure**, select the SoftPoints container for the applicable SoftPoint Generic Control Network.
2. Right click and choose **New Object** from the context menu.

3. In the New Object dialog, select the object type whose properties will be used and enter a unique name for the object.
4. Click **Create**, Figure 22.

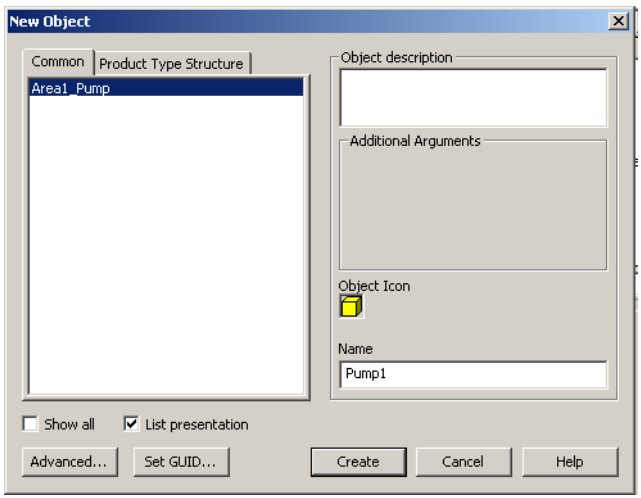


Figure 22. New Object Dialog

This adds a new SoftPoint object in the Control structure, Figure 23.

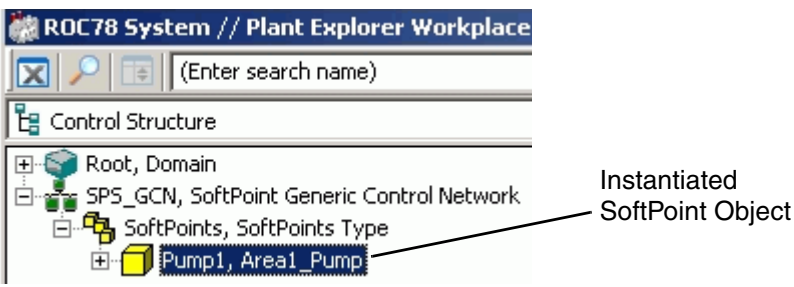


Figure 23. SoftPoint Object Added in the Control Structure

Bulk Instantiation of SoftPoint Objects in the Control Structure

When using this method to instantiate SoftPoint objects, for best performance limit the number of objects so as not to create more than 1000 signals at a time. For example, to create 100 objects with 25 signals each (2500 signals total), then instantiate 50 objects at a time.

To do this (reference Figure 24):

1. In the Control structure, select the **SoftPoint Generic Control Network** object for the node where the SoftPoints will be instantiated.
2. Click the **Generic Control Network Configuration** aspect.
3. Click the Add Objects button in the top left corner of this aspect.

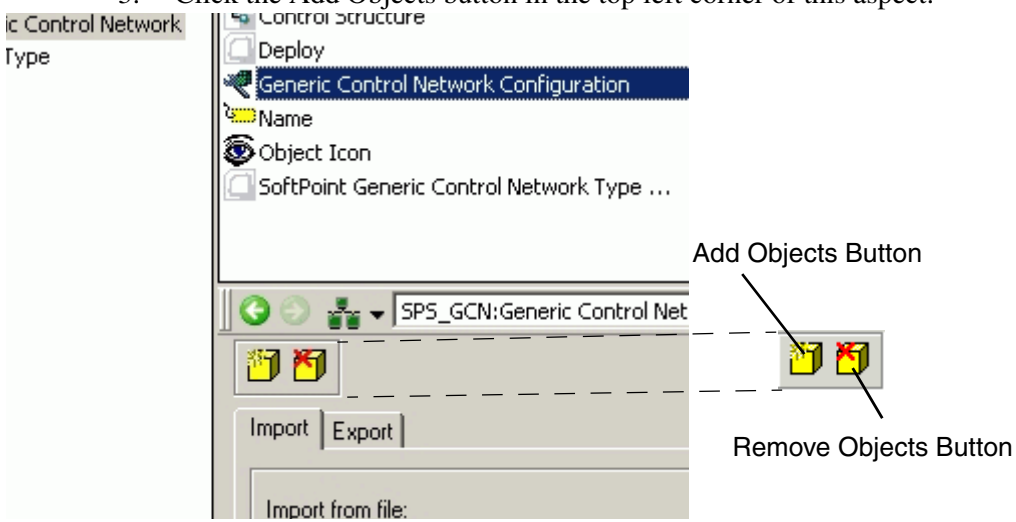


Figure 24. Selecting the Node Where SoftPoints Will Be Instantiated

This displays a dialog for selecting the object type on which to base the objects being instantiated, and for specifying the number of objects to be instantiated. An example is shown in Figure 25.

4. Use the browser on the left side to select the object type from which to instantiate the new objects (for example, *Area1_Pump* is selected in Figure 25).

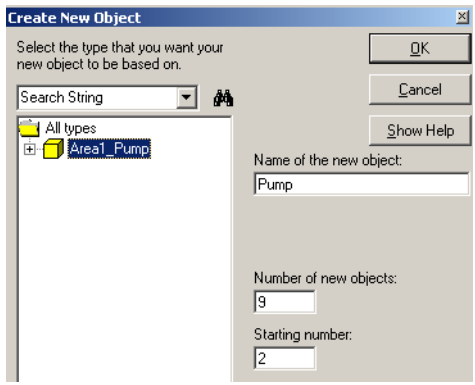


Figure 25. Create New Object Dialog

5. Specify the number of objects to create in the **Number of new objects** field. In this example, nine (**9**) new objects will be created.

6. Enter the **Name of the new object** that will be used as the base name for all new objects to be created. Each new object will be made unique by appending a sequential numeric suffix to the same base name starting with the specified **Starting number** (in this case: **2**).

7. Click **OK**. The result is shown in Figure 26.

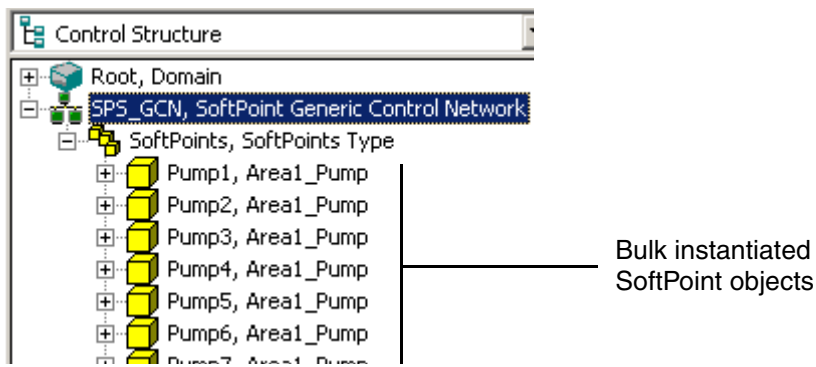


Figure 26. Example, Bulk Instantiation Result

Deleting Instantiated Objects

To delete instantiated objects:

1. Click the Delete objects button, Figure 27. This displays a dialog for selecting one or more objects to delete.

Figure 27. Delete Object Button

2. Select objects to be deleted in the **Delete Object** dialog.
3. Click **Delete**.

Adjusting Alarm/Event Properties

Alarm and event properties for binary signals in the Control structure are accessible via the Alarm Event Configuration aspect. Most of these properties are inherited from the respective signal types. One property that must be configured for signals in the Control structure is the Event Text. Other alarm/event settings can be adjusted as required. Refer to Adjusting Alarm/Event Properties for Binary Signals on page 62 and Adjusting Alarm/Event Properties for Limiters on page 65.

Adjusting Alarm/Event Properties for Binary Signals

The event text for binary signals is specified via the Alarm Event Configuration aspect. To do this (reference Figure 28):

1. Select the SoftPoint object in the Control structure, expand the object's branch to show the signals, and then select the signal.
2. Select the **Alarm Event Configuration** aspect.
3. The **Event** tab shows the default settings as configured for the corresponding SoftPoint object type.

   Optionally, to disable the default settings, uncheck **Use default event values** and then specify new settings in the left hand section of the tab, Figure 28. To restore the default settings, check **Use default event values**. The default settings can not be changed on this tab.

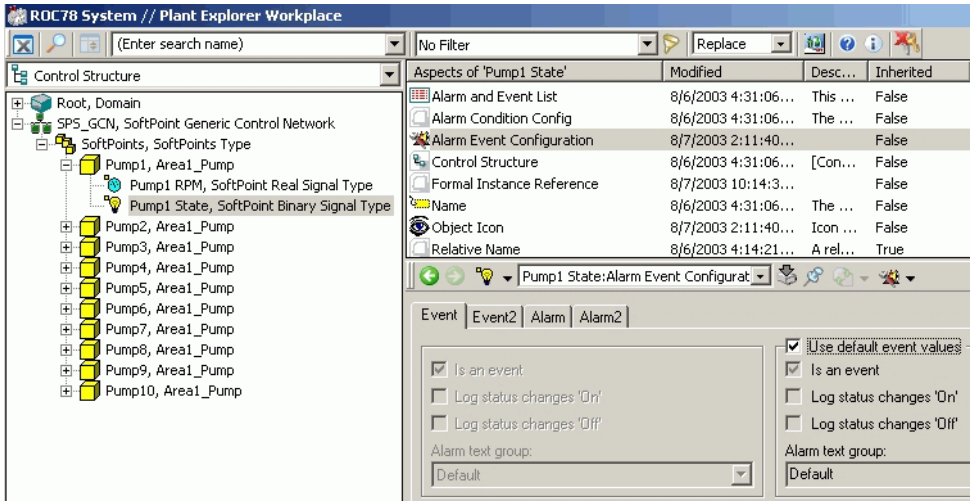


Figure 28. Binary Signal Alarm/Event Configuration

4. To enter the event text, click the **Event2** tab and then enter the text in the **Message**: field, Figure 29. The maximum number of characters in the **Message** field is 50. If additional event text is required, use the Extended event text: edit window.

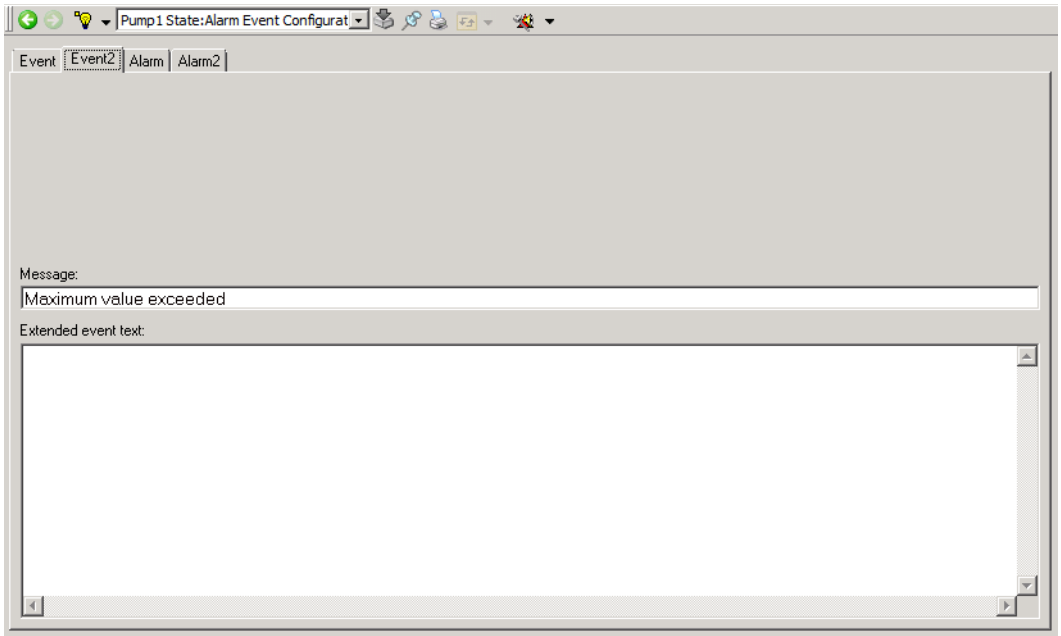


Figure 29. Entering the Event Text

5. The **Alarm** tab, Figure 30, is used to adjust the alarm handling parameters originally configured as described in Configuring Alarm and Event Handling on page 54.

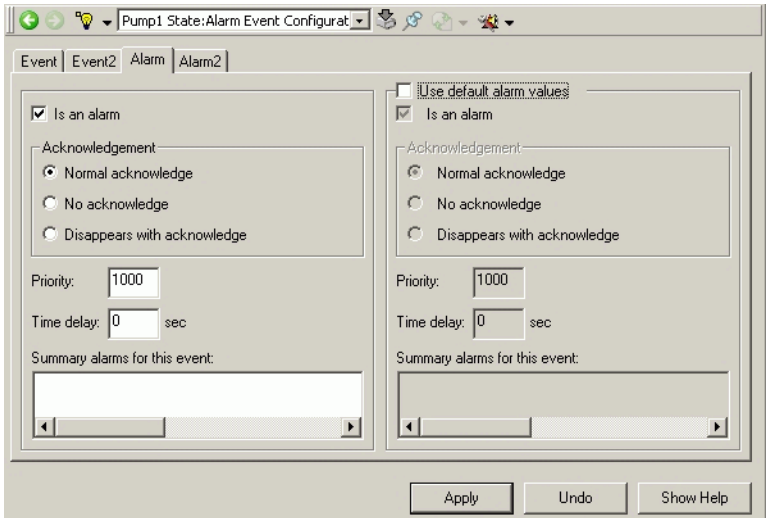


Figure 30. Adjusting Alarm Parameters

6. The Alarm2 tab, Figure 31, is used to configure class. Classes are groups of alarms without any ranking of the groups. The range is 1 to 9999.

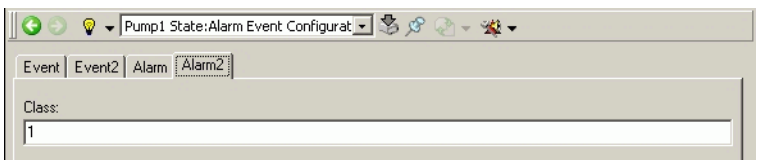


Figure 31. Alarm2 Tab

Adjusting Alarm/Event Properties for Limiters

The procedure for adjusting Alarm/Event properties for limiters is essentially the same as for binary signals, except that the Alarm Event Configuration aspect has an extra tab for selecting the limiter whose parameters need configuring, Figure 32. This **Limiter** tab also has a **Use Limiter** check box to disable/enable the limiter.

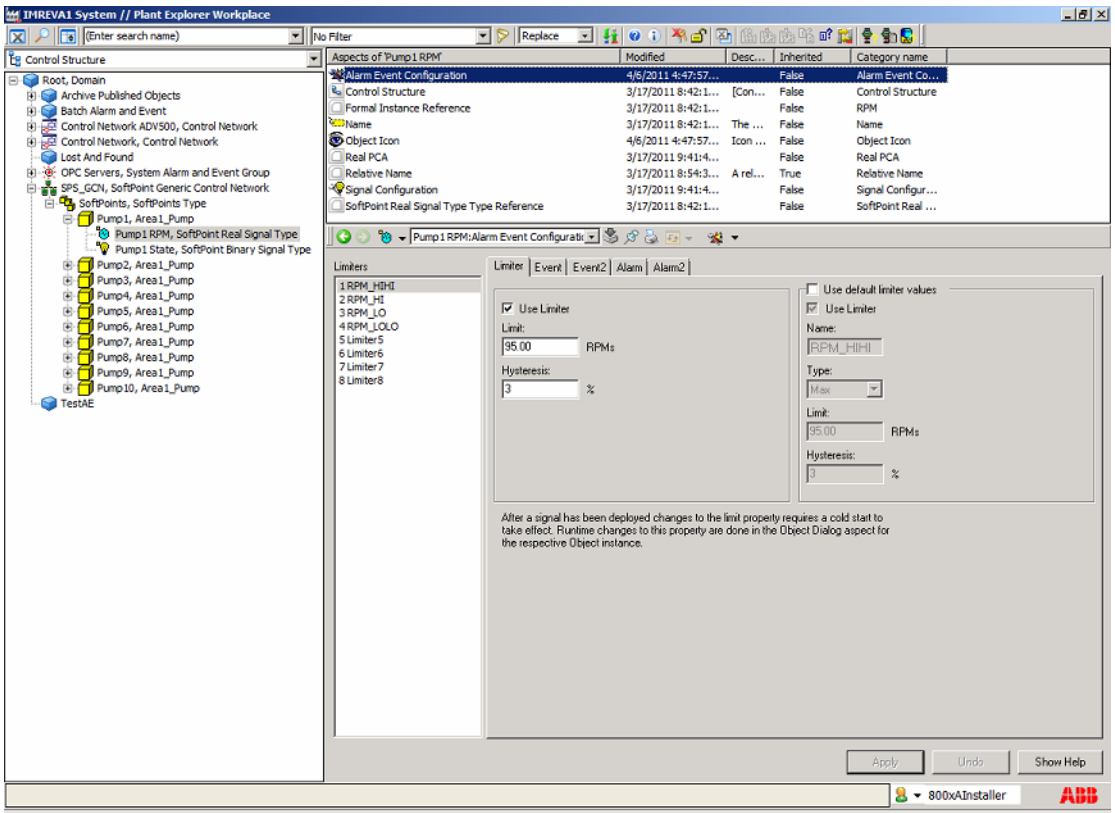


Figure 32. Adjusting Alarm/Event Parameters for Limiters

For details regarding all other tabs on this aspect, refer to Adjusting Alarm/Event Properties for Binary Signals on page 62.

Adjusting Properties for Signals

Other properties for signals in the Control structure are accessible through Signal Configuration Aspect, Figure 33.The **Signal Not Used** check box on the **ID** tab disables a signal.

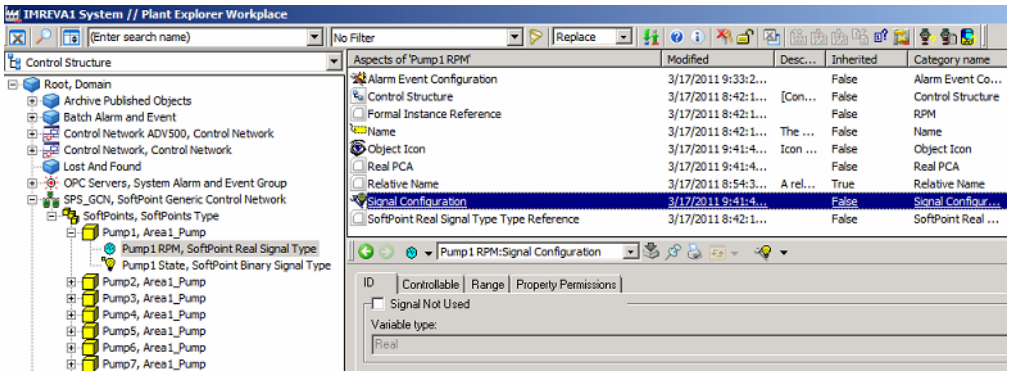


Figure 33. SoftPoint Signal Configuration Aspect

Most of the other signal properties are inherited from the respective signal types. As an option, disable the default settings from the Range tab, Figure 34.

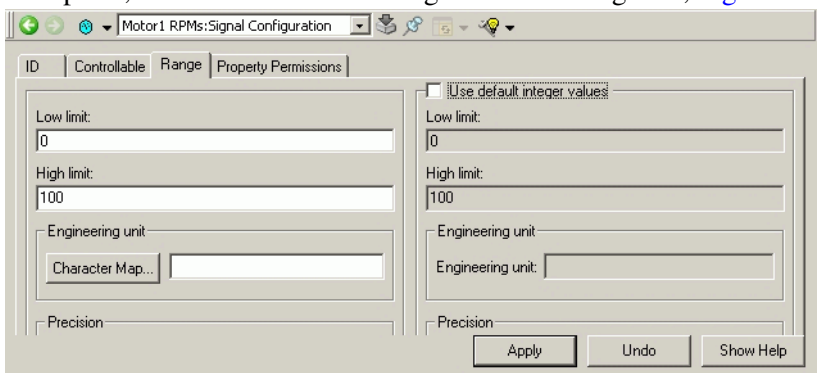


Figure 34. Unchecking the Use Defaults Check Box

Defining Alarm Text Groups

Alarm text groups can be used to specify messages to be displayed for certain events. All events belonging to the same group will be displayed and/or printed with the same text for a particular change of state. These texts can be a continuous string of up to 40 characters. Alarm Text groups are configured via the Alarm Event

Settings aspect of the SoftPoint Object Type group in the Object Type structure, Figure 35.

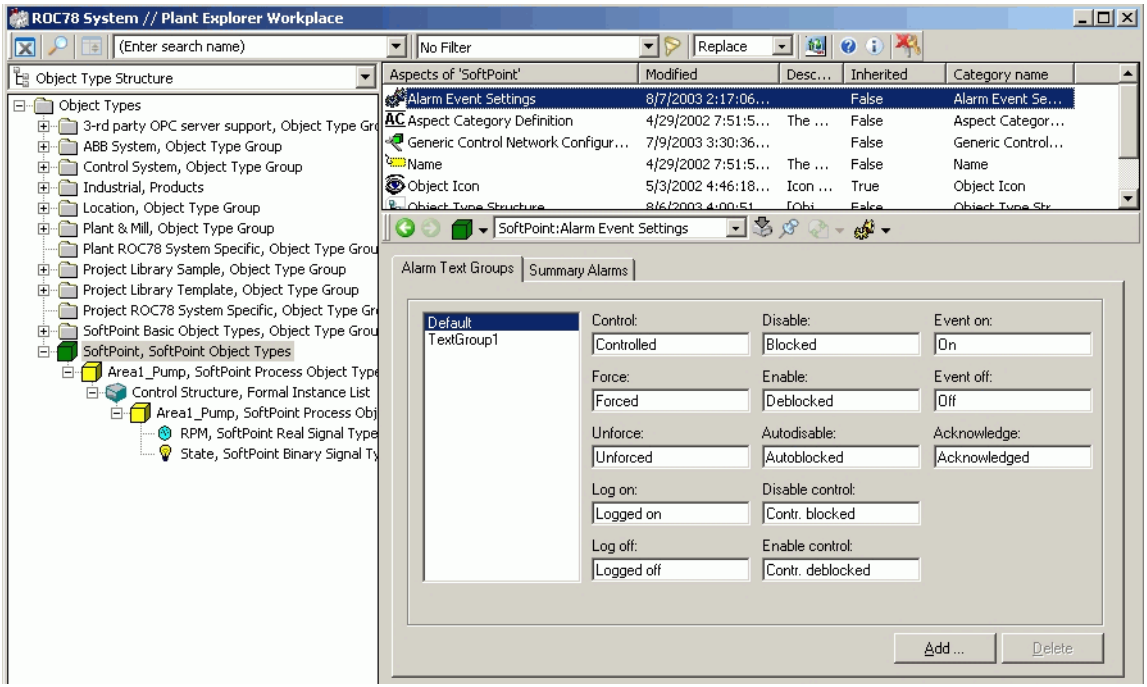


Figure 35. Alarm Text Group

Creating A New Alarm Text Group

To create a new alarm text group, or to edit an existing group:

1. Select the Alarm Event Settings aspect for the SoftPoint Object Types group in the Object Type structure.
2. Click the **Alarm Text Groups** tab.
3. To add a new group, click **Add**. This displays the New Alarm Text Group dialog, Figure 36. Enter a name for the new group in this dialog. To change an existing group, **Delete** the text group from the list and add a new text group.

Figure 36. Adding an Alarm Group

4. Enter the status texts in their respective boxes as described in Table 3. All texts can be up to 16 characters maximum. Refer to Name Rules on page 74. Properties with an asterisk (*) can only be change from the default group.

Table 3. Alarm Text Group Configuration

Property	Enter Text to be Displayed When:
Control*	a signal is under manual control.
Force*	a signal is forced.
Unforce*	a signal is unforced.
Disable control*	manual signal control is disabled.
Enable control*	manual signal control is enabled.
Disable	the event is disabled.
Enable	the event is enabled.
Autodisable	the event is autodisabled.
Event on	the event changes from off to on.
Event off	the event changes from on to off.
Acknowledge	the event is acknowledged.

5. Click **Apply**.
6. Restart the alarm and event server for the node where the SoftPoints are deployed. To do this:
   a. Go to the Service Structure.

b. Select the applicable Alarm and Event Service Provider under the Alarm and Event Services category.

c. Click on the **Service Provider Definition** aspect.

d. Click on the **Configuration** tab.

e. Uncheck the **Enabled** check box and click **Apply**.

f. Check the **Enabled** check box and click **Apply**.

Deleting An Alarm Text Group

An alarm text group cannot be deleted when it is in use (when the alarm text group is selected under the Event tab of the SoftPoint Alarm/Event Configuration aspect for at least one signal or signal type). In this case, select another alarm text group under the Event tab of the SoftPoint Alarm/Event Configuration aspect for all signals and signal types that have selected the group

To delete an alarm text group:

1. From the Text Group, select the text group to be deleted.
2. Click **Delete**.

Deploying a SoftPoint Configuration

When creating new SoftPoint objects, or making changes to existing SoftPoint objects, the new configuration will not go into effect until the changes are deployed. While SoftPoint configurations are being deployed, SoftPoint processing is suspended, current values and new inputs are held. This process is generally completed within five minutes, even for large configurations.

To do this (reference Figure 37):

1. In the Control structure, select the **SoftPoint Generic Control Network** object for the node where the SoftPoints will be deployed.
2. Click the **Deploy** aspect. This aspect provides a listing of configuration changes that have occurred since the last deployment, and indicates whether a full or incremental deployment is required.

Incremental deploys only those changes that have been made since the last deployment. A prompt is provided when a **Full** deploy is required.

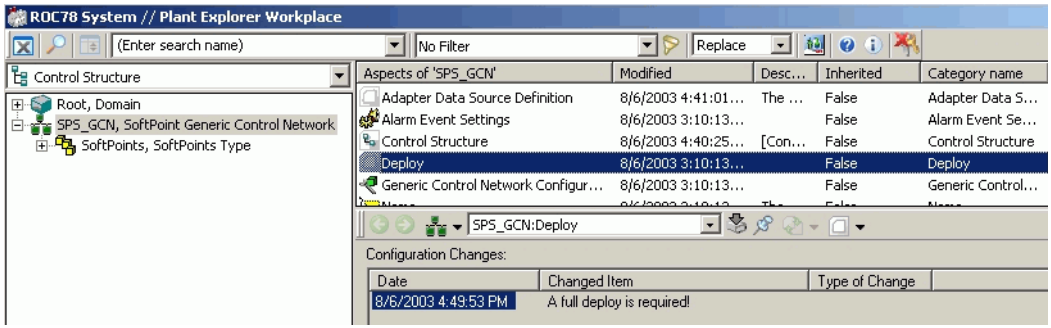


Figure 37. Deploying the SoftPoint Configuration

3. Click **Deploy** to start the deploy process. During this time, SoftPoint processing is suspended, current values and new inputs are held. This process is generally completed within five minutes, even for large configurations. Completion of the deploy process is indicated by the **Deploy ended** message, Figure 38.

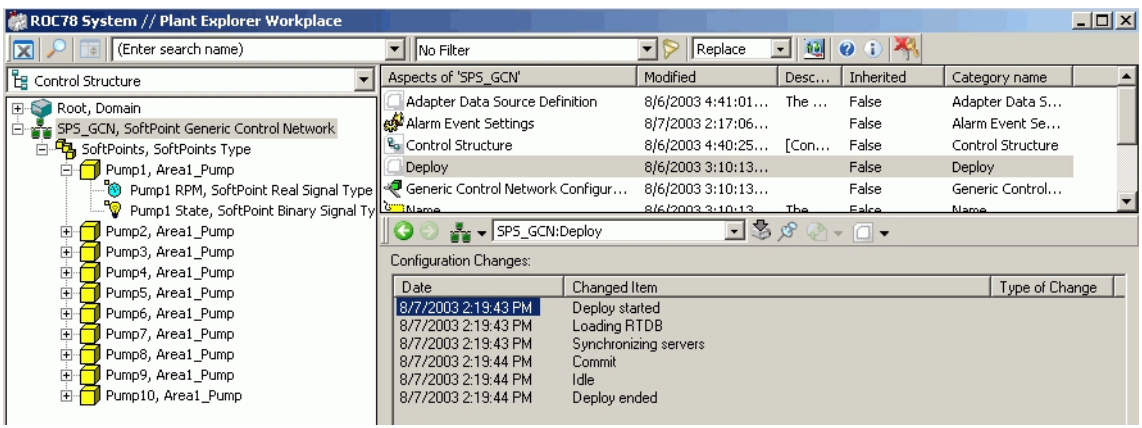


Figure 38. Deploy Ended Indication

Enabling/Disabling Warning for Changes to Object Types

Changes to a SoftPoint object type are propagated to all SoftPoint objects that have been instantiated from that object type. A warning message is generated any time a change to a SoftPoint object type is made, even if there are no instantiated objects for that type. Disable this warning message when configuring SoftPoint object types, and enable it only after the SoftPoint configuration has been deployed. To enable/disable this warning message, Figure 39:

1. Go to the Object Type structure.
2. Select the SoftPoint Object Type.
3. Select the Generic Control Network aspect.
4. Select the Edit Parameters tab.

5. To enable the message check the **Warn for time consuming operations** check box. Uncheck the check box to disable the message.

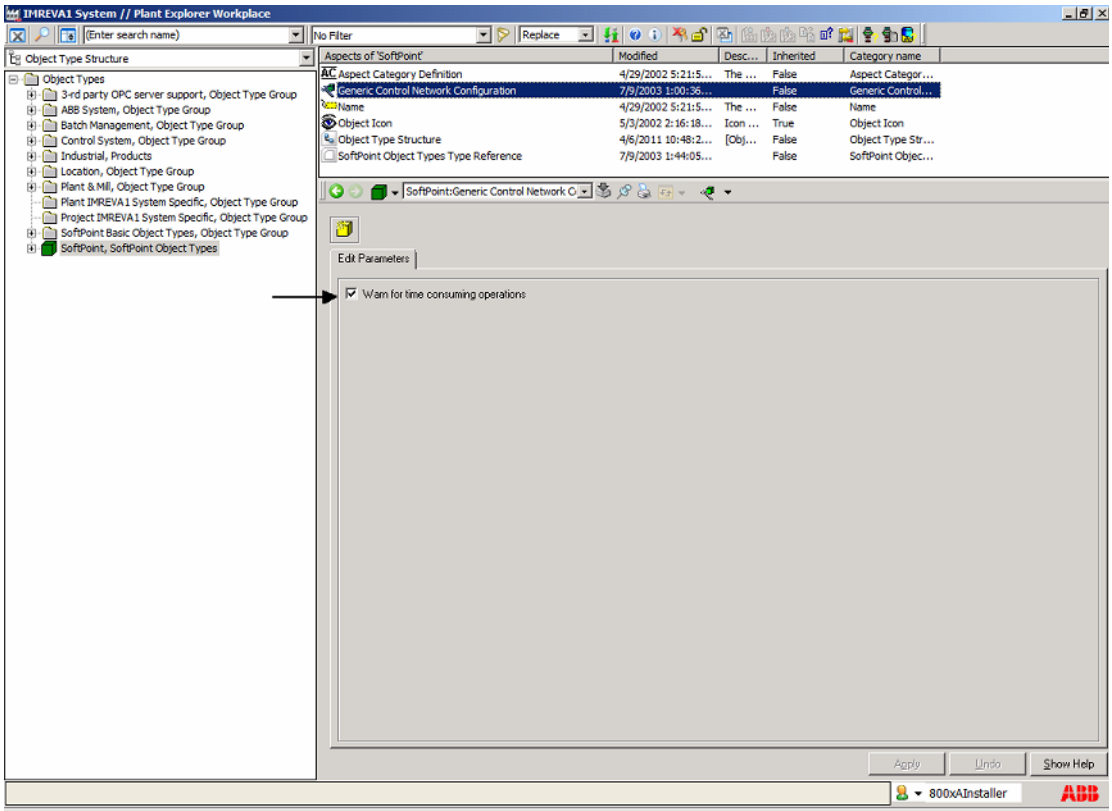


Figure 39. Enabling/Disabling the Warning Message

Name Rules

Use any combination of characters in Table 4 for SoftPoint object and signal names.

Table 4. Valid Characters for SoftPoint and Signal Names

Description	Characters
Lower case	a-z, å, ä, ö, æ, ø, ü
Upper case	A-Z, Å, Ä, Ö, Æ, Ø, Ü
Underscore	_
Numbers	0-9

Working with SoftPoints Online

The SoftPoint Object dialog is used to interact with SoftPoints online. This dialog is an aspect of the SoftPoint object in the Control structure, Figure 40.

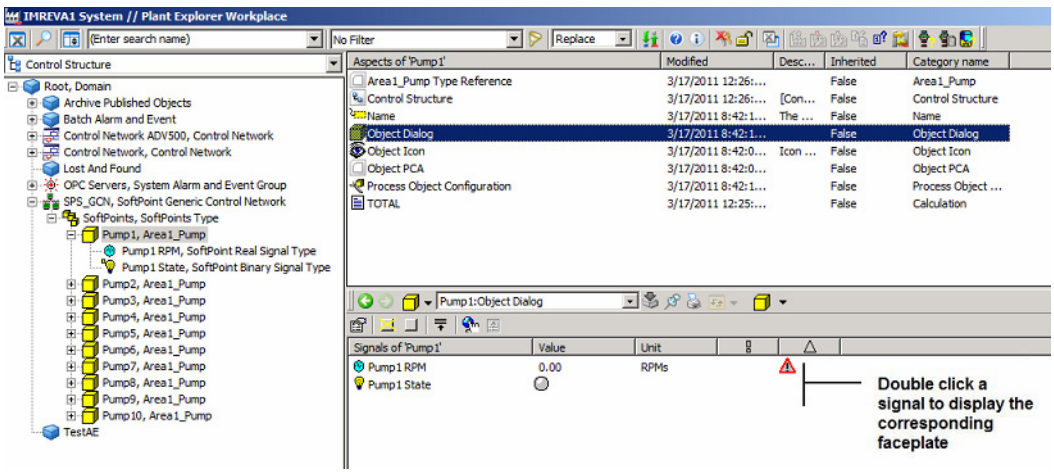


Figure 40. Accessing the SoftPoint Object Dialog

The SoftPoint Object dialog shows the signals configured for this SoftPoint object. Double-click on any signal to show the corresponding faceplate, Figure 41.

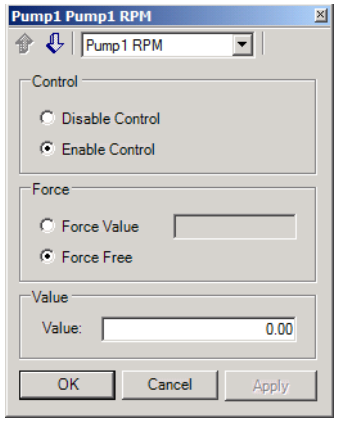


Figure 41. Example, Faceplate for Real Signal

Open the faceplate for another signal either by using the pull-down list, Figure 42, or by clicking the up/down arrows.

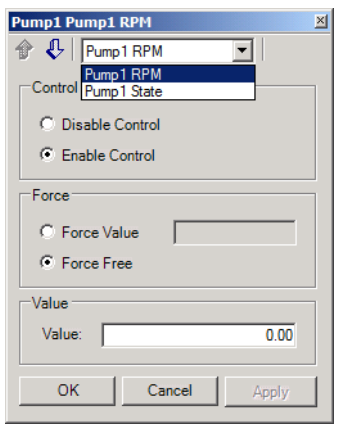


Figure 42. Changing faceplates

Attempting to access a SoftPoint object with a very long name, or performing a group OPC write to more than 64 SoftPoint signals at a time may result in the following error message:

The buffer size is insufficient

Error code: E_ADS_DS_INSUFFICIENT_BUFFER (0x8ABB04215)

Section 4 Configuring Calculations

When Calculations is installed on a node that is NOT an Information Management server, the Service Group and Service Provider objects must be manually created under the Calculations Service group in the Service structure.

Required objects are created during installation and post-installation when enabling Calculations to run on an Information Management server. The redundant Calculation Server option is set up during post installation.

Using Calculations

Refer to the Calculations Service Recommendations and Guidelines on page 133 for specific use cases related to improved stability and predictability of the Calculation Service.

Calculations Services is used to configure and schedule calculations that are applied to real-time database objects, including both SoftPoints and actual process points. Calculations may also be applied to object types allowing the calculation to be re-used each time a new object is instantiated from the object type. Calculations also have the ability to update/insert a historically stored value into a numeric log.

Calculations can be triggered by changes to the inputs, or be scheduled to execute cyclically or at a given date and time. A calculation aspect may be applied to any aspect object such as a unit, vessel, pump, or SoftPoint. Inputs can be any aspect object property, and outputs can be any writable point in the system. Input/output definitions can be made relative to the object for which the calculation is defined. Data quality and alarm generation are supported. Calculation logic is written in VBScript. The ability to write a timestamp to a SoftPoint or a log is provided to align calculations along with the inputs so they have the same timestamp for retrieval. Administrative rights are required to add, configure, or modify calculations.

The following calculation example finds the average value for up to four input variables, Figure 43. The variable mapping and VBScript are shown in Figure 44.

NOTE: This shows a calculation with one output. Other calculations may be configured with multiple inputs and outputs.

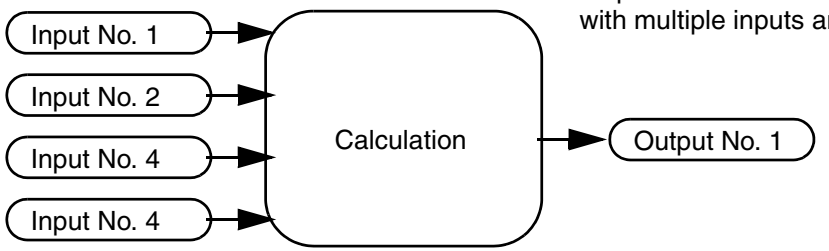


Figure 43. Example - Average

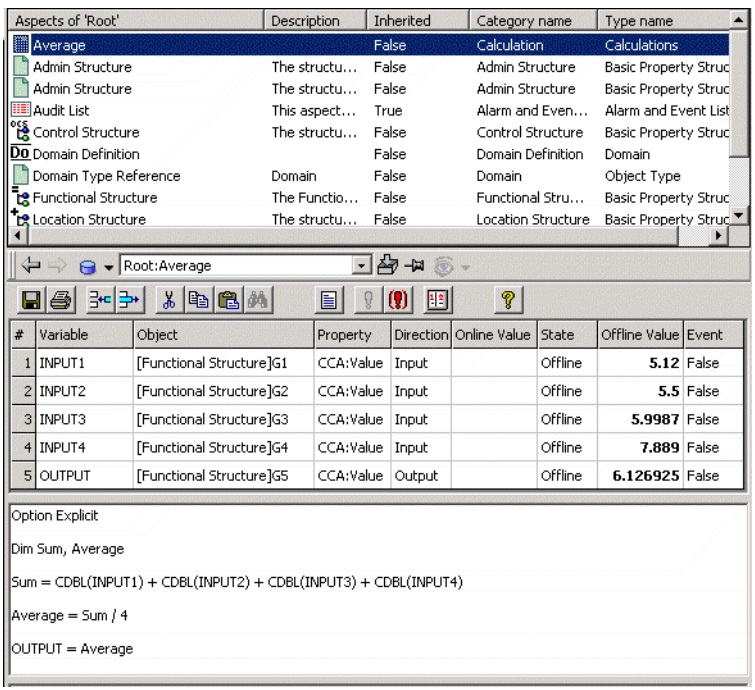


Figure 44. Example, Calculation for Average

Calculation Aspects

Calculations are created as aspects of objects or object types. An example is shown in Figure 45.

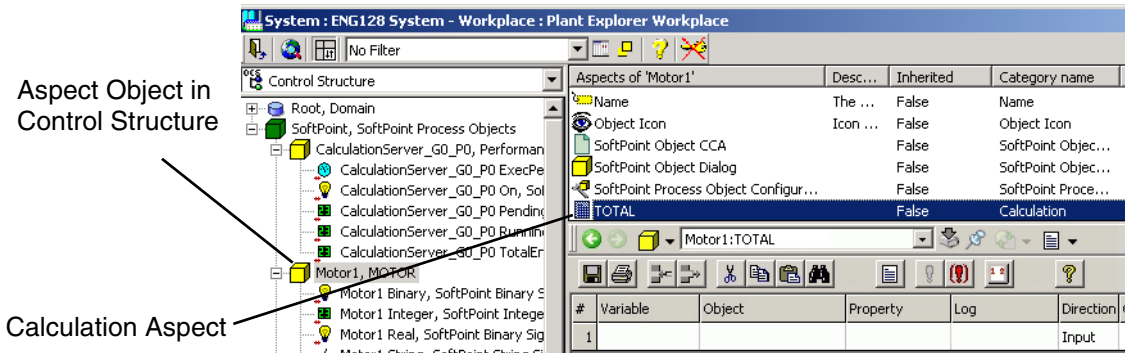


Figure 45. Calculation Aspect for SoftPoint Object in the Control Structure

User Interface

The user interface for Calculations Services is comprised of aspect views that are accessed via the Plant Explorer. The following aspect views are provided:

- **Special Configuration** for configuration of global parameters on the Configuration Server level. Refer to Configuring Global Calculations Parameters on page 80.
- **Calculation Editor** for configuring calculations. Refer to Editing the Calculation on page 84.
- **Calculation Scheduler** for scheduling calculations either cyclically or according to a schedule. Refer to Scheduling the Calculation on page 101.
- **Calculation Status Viewer** for monitoring and managing calculations. Refer to Managing Calculations on page 121.

Set-up and Administration

To verify the calculations service is properly installed and running, check the applicable Service Provider as described in Section 2, Verifying Services. To

configure global calculation parameters, refer to Configuring Global Calculations Parameters on page 80.

Configuring Global Calculations Parameters

The **Special Configuration** tab on the Service Provider Definition aspect, Figure 46, supports configuration of OPC update rate, scan rate for cyclic schedules, and cyclic offset to stagger the start of cyclically scheduled calculations. These are server-level parameters. They have valid default settings and do not need to be configured unless the defaults are not suitable for your specific application. These parameters are described in Table 5.

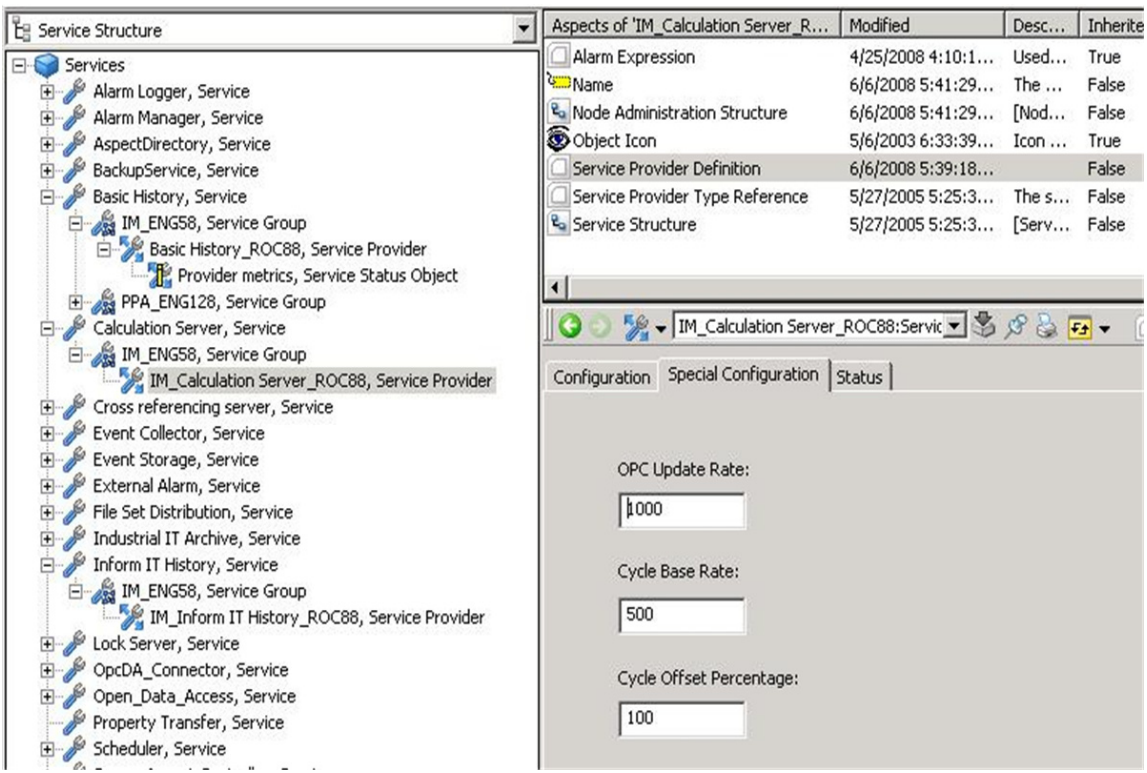


Figure 46. Calculations Server - Special Configuration

The other three tabs are common to all services.

Deviating from the defaults may cause system overload and errors. These special parameters should only be adjusted by qualified users.

Table 5. Calculations Server Special Configuration Parameters

Parameter	Description
OPC Update Rate	Rate at which input variables are updated by their respective OPC data sources. Range: 0 to 2,147,483,647 milliseconds Default: 1000 ms (1 second)
Cycle Base Rate	Rate at which the Calculations Scheduler scans the list of cyclically scheduled calculations. Typically, the default rate is used and is adjusted only if the Calculations Scheduler is consuming too much CPU time (as indicated by Windows Task Manager). Range: 100 to 3,600,000 milliseconds (3600000 = 1 hour) Default: 500 ms (1/2 second)
Cycle Offset Percentage	The first run for a cyclically scheduled calculation will occur when the specified cycle interval has elapsed after the calculation has been submitted. The Cycle Offset Percentage is used to stagger the execution of cyclic calculations to minimize *peaks* (where several cyclic calculations attempt to run at the same time). The cyclic offset reduces the cycle interval to advance the **first** execution of each cyclic calculation by a random amount between zero and the specified offset percentage. Subsequent executions of the calculation will occur according to the specified interval (without the offset). For example, if the cycle interval is 10 minutes and the offset percentage is 25, then the time when the calculation will FIRST be executed will be advanced by 0% to 25%. In other words, the time of first execution will be somewhere between 7.5 minutes and 10 minutes after the calculation is submitted for execution. Subsequent executions will occur every 10 minutes. Range: 0 to 100% Default: 100% No adjustment will be made to time-based calculations.

Configuring Calculations

This section describes how to add a calculation aspect to an aspect object, and how to use the aspect to create and schedule calculations.

Calculations are defined using VBScript. Any capabilities within VBScript or COM technologies can be used to manipulate data. External functions and libraries may also be used. Certain calculation properties are exposed for access by OPC client applications. These properties are described in OPC Access to Calculation Properties on page 108.

Once configured, calculations can be managed via the Calculations Status Viewer. This is described in Managing Calculations on page 121.

Adding a Calculation Aspect to an Object

Calculation aspects are typically added to a process or SoftPoint object in the real-time database (in the Control structure) or an object type in the Object Type structure. There are some special considerations when adding a calculation aspect to an object type. These are described in Enabling the Calculation to be Copied to Instantiated Objects.

Enabling the Calculation to be Copied to Instantiated Objects

A calculation added to an object type can be reused each time an object is created from the object type. To do this, specify that the calculation be copied when a new object is created. To get this functionality, complete the procedures for adding and configuring the calculation aspect and then refer to Instantiating Calculation Aspects On Object Types on page 107.

Also, the calculation aspect on the instantiated object is disabled by default. Enable it using either the Enable button on the calculation aspect's Tool Bar, or via Calculation Status Viewer. Refer to Managing Calculations on page 121.

Relative Object Referencing

Relative object referencing is used to run calculations independent of any specific object instances. This is described in Object Referencing Guidelines on page 113.

Adding a Calculation Aspect

To add a calculation aspect to an object:

1. Select the object that is getting a calculation. Right-click on the object and choose **New Aspect** from the context menu.
2. In the New Aspect Dialog, select the Calculation aspect and give the aspect a unique name. Figure 47 shows the new calculation aspect added to the object's aspect list.

The **Show All** check box may need to be enabled to find this aspect.

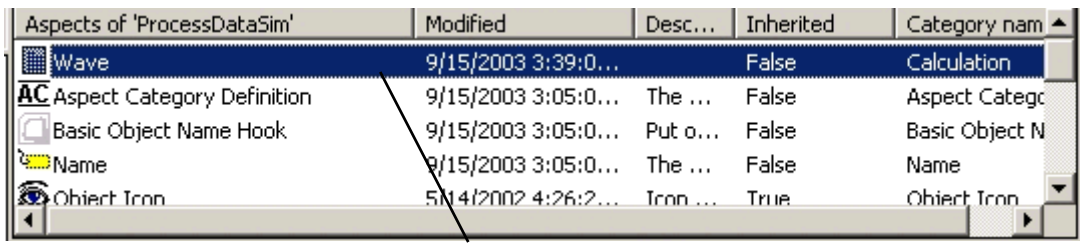


Figure 47. New Calculation Aspect Added

3. Click on the aspect to display its configuration view, Figure 48.

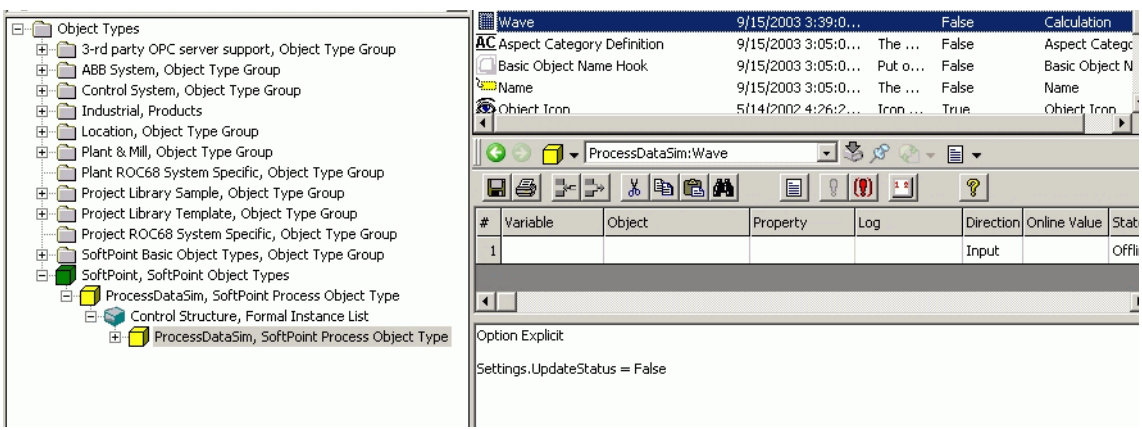


Figure 48. Displaying the Calculation Aspect View

Using the Calculation Aspect

Note the following BEFORE configuring calculations:

- Saving changes to a calculation replaces the existing version. One copy is maintained by versioning. However, to preserve multiple versions, copy and rename the calculation.
- More than one client can access a calculation for editing. The calculation editor will show changes made by all clients that are currently working on the calculation as the changes are saved.
- The Update status of the calculation is set to false by default. This is to optimize performance. Certain properties, mostly related to debugging, will not be updated when the calculation is run on line. The update status can be set to true for debug purposes; however, it is generally recommended not to run calculations in this mode.

The calculation aspect has two views - an editor and a scheduler. The editor is displayed by default. This is where calculation variables are defined, VBScript written and the trace feature enabled for debugging the calculation script offline.

Begin configuration by using the calculation editor to define variables and write the script as described in Editing the Calculation on page 84. When the calculation is ready, use the edit/schedule toggle button on the Tool Bar to switch to the scheduler view. Refer to Scheduling the Calculation on page 101.

Other procedures that can be performed via the editor include:

- Tracing the execution of a calculation - Refer to Tracing the Calculation Execution on page 100.
- Forcing the execution of a calculation - Refer to Running the Calculation Manually on page 104. The calculation can run on line or offline for simulations when run manually.
- Enabling/disabling the execution - Refer to Table 10.

Editing the Calculation

The calculation editor view is divided into three sections, Figure 49. The top part is a variable grid. This is used to map calculation input and output variables to their respective OPC tags. The middle part is an edit window for writing the VBScript. The bottom part is a trace window. When tracing is enabled, trace statements in the

VBScript are processed, and the results are displayed in the trace window. Enable or disable tracing via the editor's context menu, or programatically within the VBScript. The trace window has a limit of about 20Kbytes which is equivalent to about 10,000 characters.

Use the variable grid to map input and output variables to their respective OPC tags. Refer to Mapping Calculation Variables on page 85. Then use the edit window to write the VBScript. Refer to Writing VBScript on page 89.

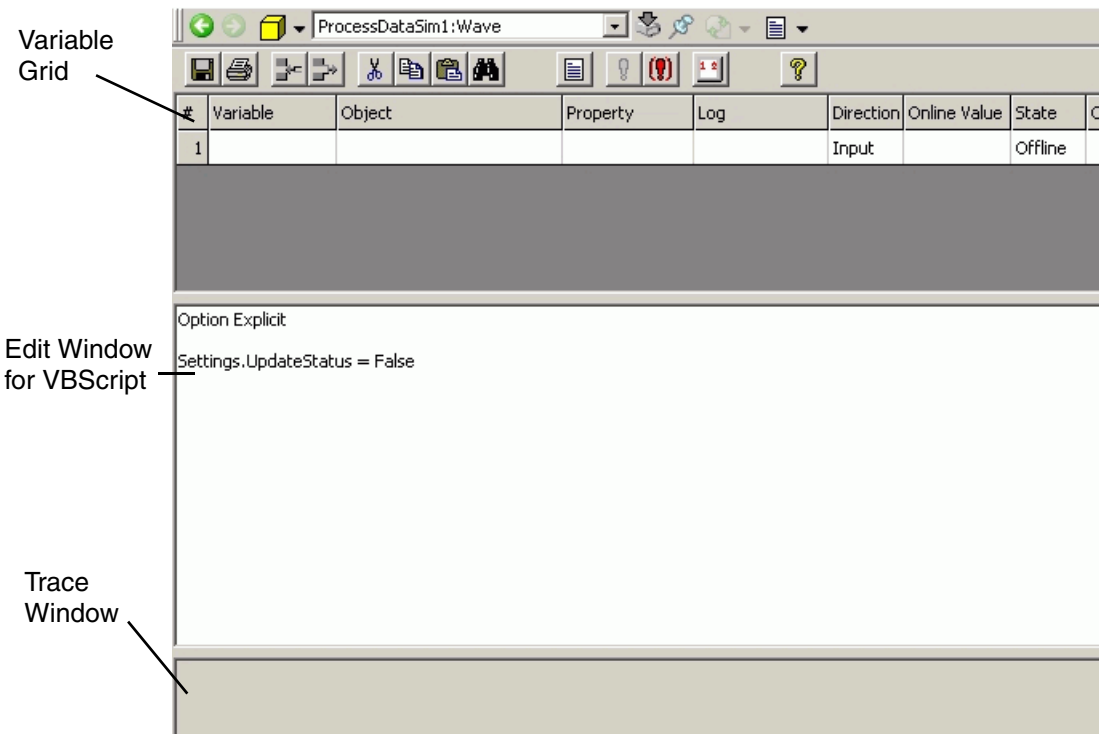


Figure 49. Editor View

Mapping Calculation Variables

The variable grid, Figure 50, is used to map calculation input and output variables to their respective OPC tags. In addition to specifying the OPC data points for the variables, use this map to specify:

- Whether to use the online data point value or a configured offline value.
- Whether to trigger the execution of the calculation when an input variable changes. Do not use analog variables to trigger the execution of a calculation.

#	Variable	Object	Property	Log	Direction	Online Value	State	Offline Value	Event
1	ppaLog	[Direct][Control Structure]Root/	INTEGER PCA:VALUE	PPA_Lab_data	Output Only	-----	Online	32	-----
2	IMLog	[Direct][Control Structure]Root/	INTEGER PCA:VALUE	IM_Lab_data	Output Only	-----	Online	0	-----
3	out2	[Direct][Control Structure]Root/	INTEGER PCA:VALUE		Output	42	Online	32	False

Figure 50. Variable Grid

Use the Insert Line and Delete Line buttons on the Tool Bar, Figure 51 to add and delete variables in this map as required. Up to 50 variables can be specified. The variables defined in this grid can be used without having to declare them in the VBScript. The parameters needed to specify each variable are described in Table 6.

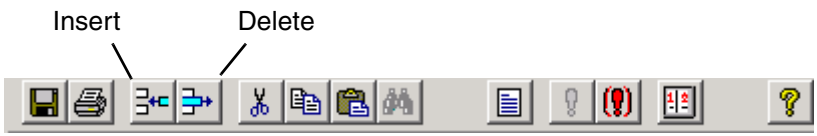


Figure 51. Insert/Delete Line Icons

Further guidelines for using variables are provided in:

- Variable Scope and Usage on page 91.
- Variable Properties and Methods on page 93.
- Variable Data Types and Conversion for Offline Variables on page 95.

When using relative object referencing to define calculations in the Object Type structure, select the property for the reference before adding the name of the specific structure to the path. After entering the structure into the path, the property list will no longer be populated because the list is built from the object reference which is no longer valid. For further information on relative object referencing, refer to Object Referencing Guidelines on page 113.

Table 6. OPC Mapping Parameters

Parameter	Description
Variable	This is the name of the variable as it is referenced in the VBScript. This does not need to be declared anywhere else. It is available for use immediately.
Object	This is an object reference. The object reference and the property parameter make up the data point specification. Enter the object reference by typing, or drag the corresponding object from the applicable structure. To drag the object, the Object field must first be selected. General object referencing syntax is described in Object Referencing Guidelines on page 113. As an alternative to using an absolute object reference, specify a relative object reference. This uses the same calculation for multiple object instances without having to change object references. A relative reference starts with the object to which the calculation is attached and must descend from that object. It is not possible to select a relative item outside the sub-tree of the selected object. Refer to Relative Object Referencing Guidelines on page 117.
Property	This pick list is populated with properties based on the object specified. For input variables, select the property to be read. For output variables, select the property to be written.
Direction	This specifies whether the variable is an input or output. • **Input** - the value is read into the variable before the calculation is executed. The OPC update rate is configurable. Refer to Configuring Global Calculations Parameters on page 80. • **Output** - the variable value is written to the property AFTER the calculation has executed. The output variable value is available for use in the script as it executes. **NOTE:** Object properties are NOT updated while the calculation is executing. For instance, a repeating loop that increments a variable value does not continually update the object. Instead, the value that occurs when the calculation is finished is written.

Table 6. OPC Mapping Parameters (Continued)

Parameter	Description
On Line Value	This is a read-only field that indicates the current value for the variable.
Offline Value	This value is used in place of the actual (Online) value when the Online flag in the State column is set to **False**. This is typically used for performing offline tests or declaring constants.
State	**Online**- Use the actual online value. For inputs, this is the online OPC property value. For outputs, this is the calculated variable value. **Offline** - Use the specified offline value
Event	This is only applicable for input variables. It is used to specify whether or not to execute the calculation in the event that the value for this input changes: • **True** - Execute the calculation when the input value changes. • **False** - Do not execute the calculation when the input value changes. **Note**: Analog signals should not be used as event triggers.
Log	The Log field is used to pick a log that will be used to store output data from the calculation. When the calculation is run, the data written to the output parameter will also be written to the numeric log, if present. If the timestamp already exists in the log for this data, then that data can also be updated. Also, the quality for the SoftPoint output object will be used as the data quality for the numeric log. The Log field is enabled when an object and a property for a variable is specified. This allows selection of a Lab Data log that has been created for that property. If no Lab Data logs have been created for the Object/Property combination, the pick list will be blank. The Lab Data log can be from either a trend log or a history log. Once specified and the aspect has been saved, the Direction column will be disabled with 'Output Only' displayed (Calculations can not read from a log). In addition, the Online Value and Event columns display '----' since they do not apply to an output only variable. The calculation variable can be treated like any other output variable, specifying the Value, Quality, and TimeStamp fields for the variable. When the calculation runs, the Value and Quality for the specified TimeStamp will be written to the specified 'Lab Data' log. If a value exists in the log for the specified timestamp, that entry will be overwritten with the new value/quality. If no value exists in the log for the specified timestamp, the value/quality will be inserted into the log with the specified timestamp (required).

Writing VBScript

When working with variables in VBScript, the assignment of any output variable must be explicitly cast to the output variable type using expressions such as:

```
OutputVar = CInt(variabledata)
```

Calculation logic is written in VBScript. Use the edit window in the middle part of the Calculation Editor to write the VBScript, Figure 52. Any capabilities within VBScript or COM technologies can be used to manipulate data. External functions and libraries may also be used.

```
Option Explicit

If OutputVar >=100 Then
  upDown = -1
Elseif OutputVar <=0 Then
  upDown = 1
End If

OutputVar = OutputVar +CInt(upDown)
OutputVar.quality =192

Settings.UpdateStatus = False
```

Figure 52. Example Script

For scripting guidelines related to calculations refer to:

- Settings.UpdateStatus on page 90.
- Variable Scope and Usage on page 91.
- Calculation Result on page 92.
- Variable Properties and Methods on page 93.
- Variable Data Types and Conversion for Offline Variables on page 95.
- Language Extensions on page 96.
- Language Restrictions on page 96.
- Writing to Quality of a Data Point from a Calculation on page 97.
- Writing Timestamps to SoftPoint Objects and Logs on page 98.
- SetLocale Function for Native Language Support on page 99.
- Tracing the Calculation Execution on page 100.

Settings.UpdateStatus

The Settings.UpdateStatus defaults to **False**. This is to optimize performance. Certain properties, mostly related to debugging, will not be updated when the calculation is run on line. These are described in Reading Calculation Status Information on page 122. Set the update status to true for debug purposes if necessary; however, it is generally recommended not to run calculations in this mode.

If a relatively few calculations are running at a modest rate of execution, run the calculations with Settings.UpdateStatus = True. This allows the Result property to link calculations. Refer to Calculation Result on page 92.

The calculation can also periodically set the update status true every n number of executions to periodically update status-related parameters. An example is provided in Improving Performance on page 131.

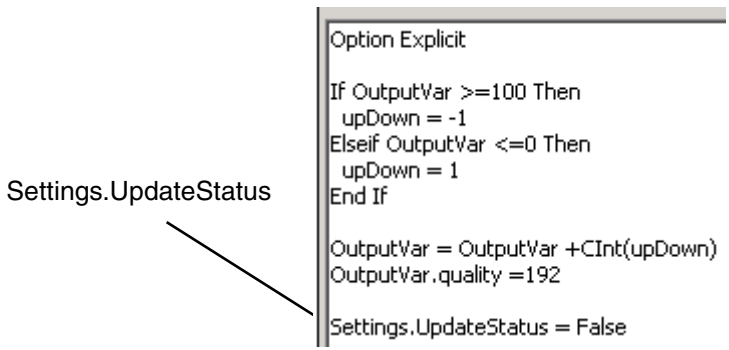


Figure 53. Example Script

To set the update status to true, either change the value of Settings.UpdateStatus to True, or by comment out the line.

DO NOT delete the line. This would require re-entering the line to reset the update status to False.

When True, versioning can not be used. This is because online values of the Calculation aspect differ between environments.

Variable Scope and Usage

This section describes the scope and usage of the online and offline variables declared in the variable grid. It also describes the *Result* variable which is a special internally managed variable. This variable is not declared in the grid or script. Its value may be persisted between executions and made available to other applications via OPC by setting the Update Status to **True**.

Online Variables

Online input variables declared in the grid contain read-only values that correspond to the value of the referenced object and property. The variable within the script can be modified, but these changes are not persisted between executions and the value after execution is not written to the actual process value.

Online output variables declared in the grid contain the actual process values of the referenced objects and properties before script execution. Also any modification of these variables from within the script will modify the actual process values after execution completes.

Offline Variables

Offline values are used in place of the actual (Online) values when the Online flag in the State column is set to **False**. These values are also used when executing the calculation in offline mode. Offline variables are not supported in an engineering environment.

Offline input variables are like constants. They contain the value as designated in the Offline Value column during execution. The variable may be modified within the script, but changes are not persisted between calculation executions. Further Offline values are not updated when the Update Status is set to False (default).

Offline output variables function like internal SoftPoints. Before calculation execution, the variable is set to the value contained in the Offline Value column. During execution, the variable can be modified via the script. After execution, the change (if any) is written back to the Offline Value. This offline value is not accessible to any client other than the calculation itself. This is useful for creating static variables (values are persisted between executions) where the values are only used internally within the calculation. For example, an internal counter could record the number of times the calculation has executed.

Calculation Result

The calculation result is a special calculation property that is treated as a variable within the VB script. It does not need to be declared in the grid or the script.

This property is only applicable when the Update Status is set to True.

Before execution, the previous result value is loaded into the variable and is available for use by the script. The result value can be set within the calculation's VB script, for example **result.value = 1**. After execution, the value is written back to the calculation's Result property. The result value is then accessible by OPC clients, and may also be referenced within other calculations as an input variable in the calculation's variable grid.

The result quality may also be set within the calculation script, for example: **result.quality = 0**. However, any change to this value is not saved once the calculation has completed. The result quality always reverts back to quality = good once the calculation has completed.

If history is to be collected for the result of a calculation, it is best not to collect directly from the Result property. For best performance, write the result to a SoftPoint from which the data may then be collected.

The calculation result can be used to link calculations without having to use an external SoftPoint. When Update Status is set to True, the result variable of the calculation aspect can be used to link one or more calculations. The result of one calculation can be used as the input and trigger for another calculation. This is illustrated in the following example.

Example:

M1 consists of two calculations: CalcA and CalcB. The result property of CalcA will be used to trigger CalcB.

The script for CalcA sets its result property to the some input, in this case INPUT1, Figure 54.

CalcB then subscribes to CalcA's result property as an input, Figure 55. In this case, the event flag is also set to **True** so that each time the result of CalcA changes, CalcB executes. CalcB could also be executed manually or scheduled instead.

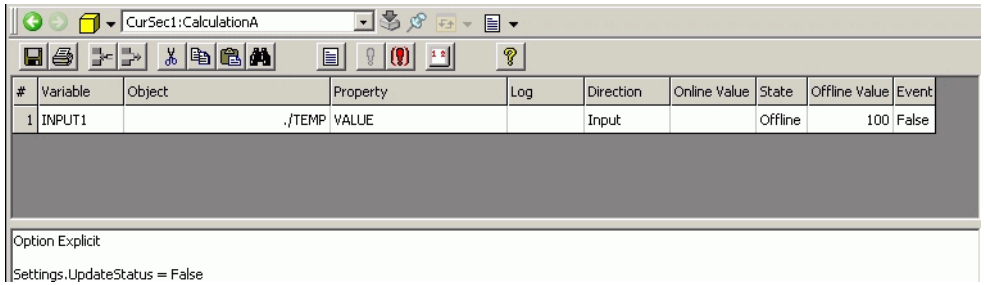


Figure 54. CalcA Aspect

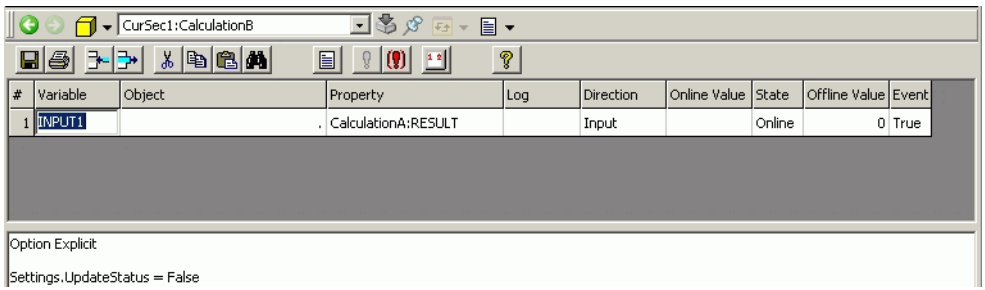


Figure 55. CalcB Aspect

Variable Properties and Methods

Variables added to the variable grid have properties and methods which can be referenced in the VBScript. These are described in Table 7 and Table 8. All properties are read/write. This is used to test error conditions by setting any of the properties from inside the calculation. Calculations only view OPC quality as either good or bad. The quality value for the calculation variable is read/write. For online variables, the quality value is set to the actual quality of the referenced OPC item. For offline variables, the quality is always set to **OPC_QUALITY_GOOD**, however this value may be modified by the calculation. The syntax for referencing a variable property or method is (refer to Figure 56 for an example):

variableName.propertyName or *variableName.methodName*

Table 7. Properties

Name	Data Type	Description
Value (Default property)	Variant	For online mode, this is the actual process value of the referenced OPCItem. For offline mode, this is the value defined in the Offline value field.
Quality	Long	For online mode, this is the quality value of the referenced OPCItem. For offline mode, the value is always **OPC_QUALITY_GOOD**. Also refer to Writing to Quality of a Data Point from a Calculation on page 97.
TimeStamp	Variant	For online mode, this is the Universal Time Coordinate (UTC) time stamp of the referenced OPC item. For offline mode, the time value is empty (although for testing purposes, the variable's time stamp can be set manually from within the script). For example: out.timestamp = Var1.timestamp out.timestamp = NOW() out.timestamp = CDATE(04/13/2006 11:30:25) Also refer to Writing Timestamps to SoftPoint Objects and Logs on page 98.

Table 8. Methods

Name	Returns	Description
IsQualityOk	Boolean	Returns **True** if the quality of the variable is equal to **OPC_QUALITY_GOOD**. Otherwise, returns **False**.
IsQualityBad	Boolean	Returns **True** if the quality of the variable is equal to **OPC_QUALITY_BAD**. Otherwise, returns **False**.

```
Option Explicit

' Clear out the trace window.
Debug.Clear

Volume.Quality = 0
Volume.Value = 10   ' Does the same thing as Volume = 10

' Check the quality using the IsQualityOk method...
If Volume.IsQualityOk Then
   VolumeCC = Volume / 1000
Else
  ' Put information to the trace window...
   Debug.Trace "Volume quality was invalid! : " & Volume.Quality
End If
```

Figure 56. Example Script

Variable Data Types and Conversion for Offline Variables

Offline variables can contain any of the data types supported in VBScript. However, to support this flexibility, the values are always stored in the aspect as a string. In most cases, this is not a problem because the functions and *most* operators in VBScript try to convert the variable to the appropriate data type before executing.

The addition (+) operator has a problem when it is used to concatenate text strings. Other operators that cause problems for strings are greater than (>) and less than (<). Therefore, when working with variables (and with VBScript in general), be sure to explicitly convert the variables to the desired data type before performing any operations. A conversion example is shown in Figure 57.

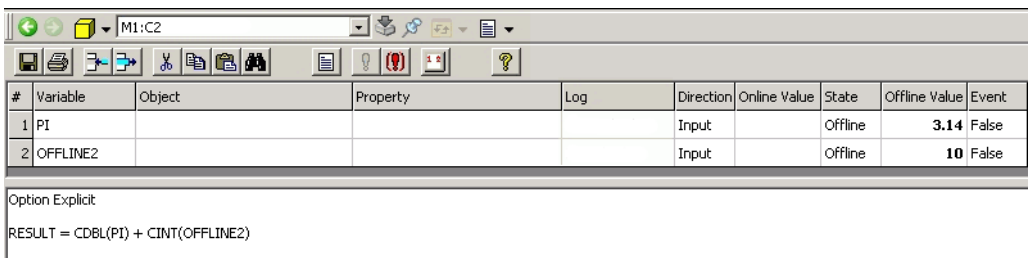


Figure 57. Conversion Example

In addition, some OPC properties will not convert to the appropriate data type before allowing the data to be committed. For example, attempting to write the value 255 to the NAME:DESCRIPTION property of an object will not work. However writing the value "255" or using the CSTR conversion function will work. This is illustrated in Figure 58.

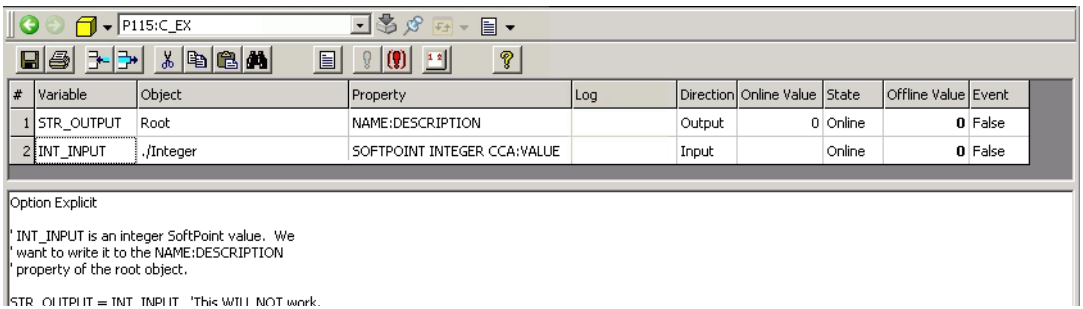


Figure 58. Example, CSTR Conversion

Language Extensions

Language extensions provide additional functionality as described in Table 9.

Table 9. Language Extensions

Keyword	Data type	Description
This	ABBAspect	Exposes aspect properties and methods for the object that *owns* this calculation. Any property or method of type ABBAspect may be referenced. For example: **This.object.name** references the name aspect of the object that owns this calculation.
Debug	User-defined type	Used to reference the **Trace** and **Clear** debug methods. Refer to Tracing the Calculation Execution on page 100.

Language Restrictions

Due to the nature of calculations, the following restrictions are placed on the language:

- **User Interface** - The VBScript language supports functions that produce a user-interface. For example, the MsgBox function displays a window that

remains visible until the user manually closes it. All user-interface functions are turned off for calculations.

- **Timeout** - To prevent the user from performing operations that could potentially hang the calculation service, a configurable timeout period is employed. If the calculation does not complete in the time specified, it is forced to exit with an error status indicating a timeout has occurred. This prevents endless loops and other coding errors from adversely affecting the system.
- **Calculations variables can not be used as a control variable in a *For* loop.**

  Using a variable declared in the calculation grid as a control variable will cause problems in a "For" loop. For example, INPUTA is a variable declared in the Calculations grid. This code will result in an illegal assignment error for the calculation. The problem area is shown in **bold** type.

```
For INPUTA = 1 To 10
     Do Something
Next
```

  The variable CAN be used as an upper limit for the loop.

```
Dim i
For i = 1 To INPUTA
    Do Something
Next
```

Writing to Quality of a Data Point from a Calculation

Writing to the quality of a point using OPC is not permitted according to the current OPC standard; however, calculations are permitted to write to the quality of a SoftPoint. This requires the Calculation server and SoftPoint server to be on the same node.

To write to quality, assign the Quality property of the variable (declared in the grid) to either good (192) or bad (0). Attempting to write to the quality of a non-SoftPoint OR a SoftPoint on a remote server which is not properly connected via DCOM, the write is ignored.

For example, INPUTA is a variable declared in the Calculations grid referencing a SoftPoint signal.

- To write bad quality to INPUTA, enter: **INPUTA.Quality = 0**
- To write good quality to INPUTA, enter: **INPUTA.Quality = 192**

It may be useful to propagate the data quality from one SoftPoint signal to another, typically from an input to output. Given a calculation with an input variable IN_1 and an output variable OUT_1, the following line of code will propagate quality from the input to the output: **OUT_1.Quality = IN_1.Quality**

Writing Timestamps to SoftPoint Objects and Logs

The script in a calculation object can write the value, quality and a specific timestamp associated with the value/quality to a SoftPoint object or a log. For example, in the following script, the output variable will take on the value, quality, and timestamp of the input with most recent timestamp:

```
if in1.TimeStamp >= in2.TimeStamp then
  out3.Value = in1.Value
  out3.TimeStamp = in1.TimeStamp
else
  out3.Value = in2.Value
  out3.TimeStamp = in2.TimeStamp
end if
```

When the calculation runs, the timestamp assigned to the out3 variable will get written to the SoftPoint object or log, rather than the current time. If current time is desired, then the out3.TimeStamp field should not be updated in the calculation. If the TimeStamp field of any output variable is assigned in the calculation, that timestamp will be written to the SoftPoint object or log. If the TimeStamp field is not updated, the current time will be assigned in the SoftPoint server.

This feature applies to SoftPoint objects serviced by the SoftPoint server on the local node and when writing the log parameter. OPC tags do not support this feature.

Timestamps that are read in from OPC servers and the SoftPoint server come into the script as UTC timestamps and are also written to those servers as UTC timestamps. Most VBScript time functions (such as Now(), Second(), etc.) are local timestamps and may need to be manipulated accordingly.

SetLocale Function for Native Language Support

If the Windows Regional Settings are set to a locale other than English (United States), and the decimal symbol in that locale is specified as a comma (,), customize the regional options to use the dot (.) as the decimal symbol. Refer to the table of Windows Installation Requirements in Section 2 of *System Installation (3BSE034678*)* for more information.

Calculation Services supports Windows Regional settings for French, German, and Swedish languages. If floating point values are to be used in calculations, be sure to include the *SetLocale* function in the calculation script in order for the calculation to process the floating point values correctly. An example is illustrated in Figure 59. The syntax is:

SetLocale("*lang*")

where lang is the code for the native language:

- fr = French.
- de = German.
- sv = Swedish.

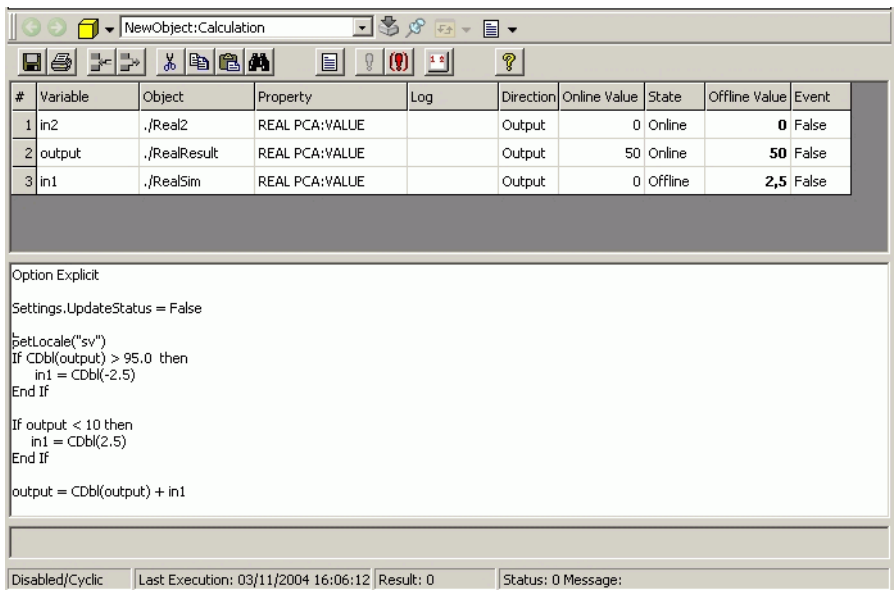


Figure 59. SetLocale Function for Floating Point Values in Native Language

Tracing the Calculation Execution

Tracing may be enabled when the Update Status is set to True. When tracing is enabled, trace statements in the VBScript are processed, and the results are displayed in the trace window. Enable or disable tracing via the editor's context menu, or programatically within the VBScript.

Tracing Calculation Variables

The trace can be fine tuned by keying on one or more specific variables. The syntax is: **debug.trace** <*argument*>. Use any argument that VBScript can convert to a string, for example: **debug.trace** ***input1.quality***. An error occurs when using an argument which cannot be converted to a string.

The example script in Figure 60 shows how to enter a text string in the trace window, and then append a variable property. The text string must be entered in double quotation marks, and the variable.property specification must use the & prefix to append it to the text string.

```
Option Explicit

' Clear out the trace window.
Debug.Clear

Volume.Quality = 0
Volume.Value = 10   ' Does the same thing as Volume = 10

' Check the quality using the IsQualityOk method...
If Volume.IsQualityOk Then
   VolumeCC = Volume / 1000
Else
   ' Put information to the trace window...
   Debug.Trace "Volume quality was invalid! : " & Volume.Quality
End If
```

Figure 60. Example Script

Clearing the Trace Window

To clear the trace window from the VB script enter: **debug.clear**.

Tool Bar

The tool bar, Figure 61, provides access to Calculation Editor functions. Table 10 provides additional notes.

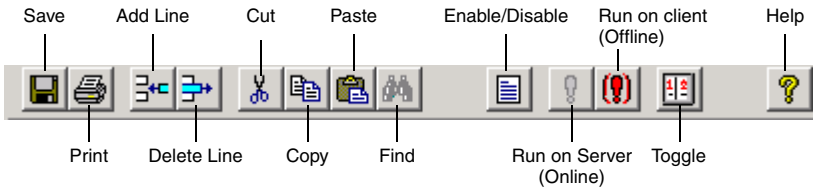


Figure 61. Tool Bar

The Calculation Editor View does not prompt the user when exiting and changes are pending. Use Save button before leaving the display or pin the aspect.

Table 10. Tool Bar Notes

Icon	Notes
	Saving changes completely replaces the original calculation. To preserve the original, make a second copy to edit.
	Enable/Disable. Each Calculation must be assigned to a specific Service Provider BEFORE enabling the calculation. This procedure is described in Distributing Calculations on Different Servers on page 124.
	Run Calculation on the Server (Online). The calculation must be saved and enabled before it is run.
	Run Calculation on the client (offline). The calculation must be saved before it is run.
	Toggle between calculation editor and scheduler views.

Scheduling the Calculation

Assign each calculation to a specific service provider prior to enabling and running the calculation. Refer to Managing Calculations on page 121.

The following scheduling options are supported for Calculation aspects:

- Calculations may be triggered by changes to input variables. Configuring an event trigger for the calculation is done via the Event column in the Variable Grid. Refer to Mapping Calculation Variables on page 85.
- Calculations may be scheduled via the Scheduler view in the Calculation Editor. This supports cyclical or periodic scheduling (date and time).

  To display the calculation Scheduler view click the Calculation Editor/Scheduler toggle button on the Tool Bar. Then refer to:
  – Cyclic Schedule on page 102.
  – Time-based Schedule on page 103.
- Calculations may be scheduled via the Application Scheduler. This supports:
  – A specified weekday during the month (for example first Monday, third Wednesday, last Sunday, or every Tuesday).
  – A specific day of the month (for example 1st, 12th, 31st, or every day).
  – A list of scheduled dates and times.
  – The evaluation of an expression.

  Refer to Scheduling Calculations via the Application Scheduler on page 104.
- Calculations may be run manually (one-time execution). To run a calculation manually refer to Running the Calculation Manually on page 104.

The complexity and scheduling frequency of a calculation may put an excessive load on the system. For example, the time required to open the Plant Explorer may take longer. This type of performance problems can happen when the UpdateStatus is enabled. For help, refer to Improving Performance on page 131.

Cyclic Schedule

To specify a cyclic schedule, make sure the **Schedule** check box is unchecked, and then check the **Cycle** check box. In the corresponding fields specify the interval unit (hours, minutes, seconds), and the number of intervals. The example in Figure 62 shows a one-second cycle.

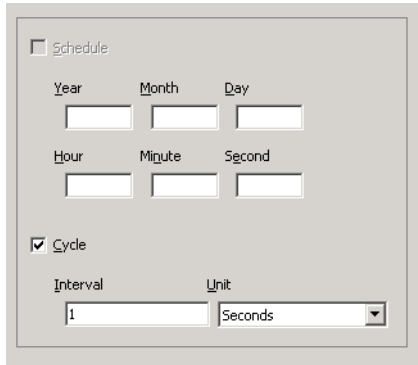


Figure 62. Cyclic Schedule

Time-based Schedule

To specify a time-based schedule, make sure the **Cycle** check box is unchecked, and then check the **Schedule** check box. In the corresponding fields specify the time components (year, month, day, hour, minute, second). The * character is a wildcard entry which means **every**. Figure 63 shows a time-based schedule that will execute at 12:01:01 (one minute and one second after noon) on every day of every month, for every year.

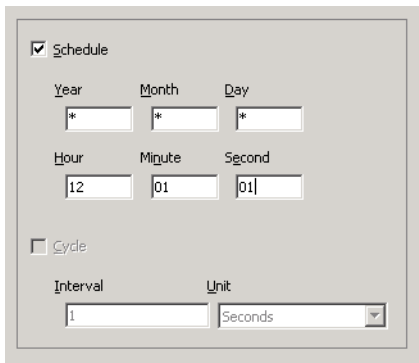


Figure 63. Example - Time-based Schedule

Running the Calculation Manually

When a calculation is run manually, it can also be run offline (simulation) or online (on the client).

Offline Execution

In the offline mode, the calculation uses the offline values specified on the variable grid. This way the calculation execution does not affect actual process values. The calculation must be saved before it can run.
To run a calculation offline, click the offline button  .

Online Execution

In the online mode the calculation uses the online values specified on the variable grid. Save and enable the calculation before running the calculation in this mode.

Assign each Calculation to a specific Service Provider BEFORE enabling the calculation. This is described in Distributing Calculations on Different Servers on page 124.

To run the calculation online, click the online button  .

Scheduling Calculations via the Application Scheduler

Calculations may be scheduled via the Application Scheduler. This supports scheduling based on:

- A specified weekday during the month (for example first Monday, third Wednesday, last Sunday, or every Tuesday).
- A specific day of the month (for example 1st, 12th, 31st, or every day).
- A list of scheduled dates and times.
- The evaluation of an expression.

To do this, create a job with a Calculation action in the Scheduling structure. This section demonstrates how to add a job and use the scheduling definition aspect to set up a periodic schedule. It also describes how to add and use the Calculation Action aspect. For further information on jobs and scheduling options, refer to the section

on scheduling in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

Adding a Job and Specifying the Schedule

Jobs are created in the Scheduling structure. To create a job:

1. In the Plant Explorer, select the **Scheduling Structure**.
2. Right-click on **Job Descriptions** and choose **New Object** from the context menu.
3. Add a **Job Description** object (under Scheduling Options) and assign the object a logical name (Calc1Sched for example).
4. Click **Create**. This creates the new job under the Job Descriptions branch, and adds the Schedule Definition aspect to the object's aspect list.
5. Click the **Scheduling Definition** aspect to display the configuration view, Figure 64. This figure shows the scheduling definition aspect configured as a periodic schedule. The calculation will be executed once every hour, starting July 2th at17:00 (5:00 PM), and continuing until July 9th at17:00.

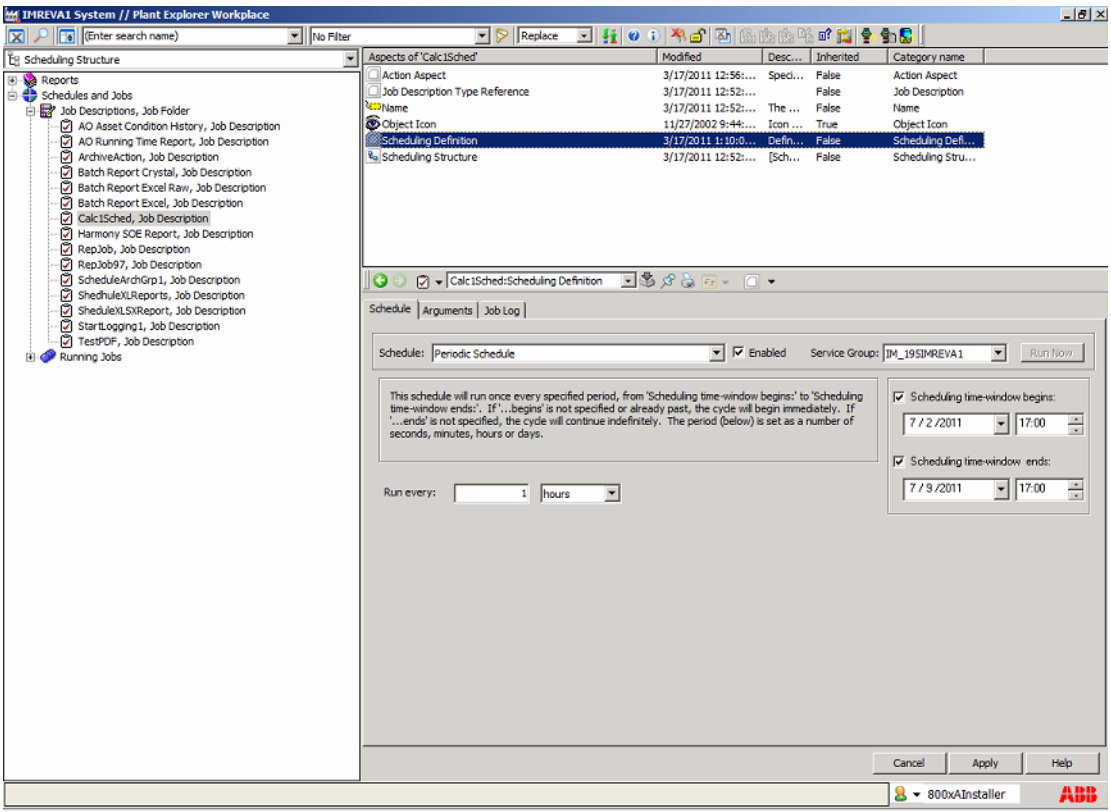


Figure 64. Scheduling Definition Configuration View

Adding and Configuring the Calculation Action

Actions are implemented as aspects on an object which is on or under Job Descriptions in the Scheduling Structure. To add an action:

1. Right-click on the Job object (for example Calc1Sched, Job Description) and choose **New Aspect** from the context menu.
2. In the New Aspect dialog, browse to the Scheduler category and select the Action aspect (path is: **Scheduler>Action Aspect>Action Aspect**). Use the default aspect name, or specify a new name.

3. Click **Create** to add the Action aspect to the job.
4. Click on the Action aspect to display the configuration view.
5. Select **Inform IT Calculation Action** from the Action pull-down list. This displays the Calculation plug-in, Figure 65.

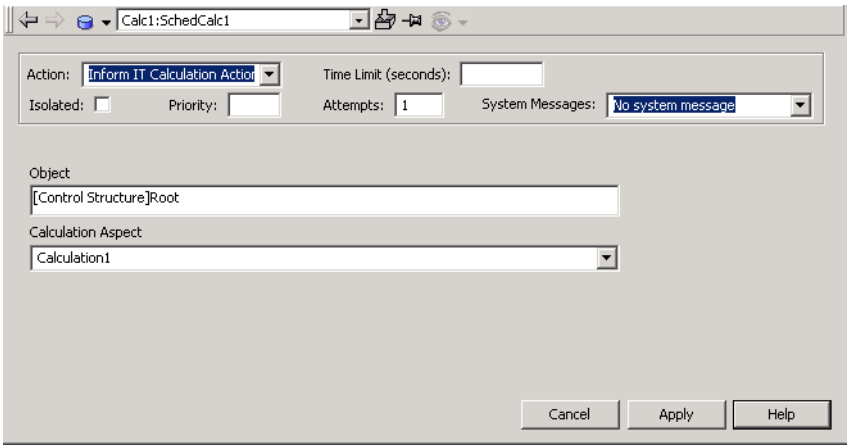


Figure 65. Plug-in for Calculations

6. Specify the calculation to be scheduled by entering the path to the object containing the calculation in the Object field. Either type the full pathname to the object in the field, or drag the object from the object browser in the Plant Explorer.
7. Next select the calculation aspect using the Calculation Aspect combo box. As an option, type the name of the calculation aspect.
8. Click **Apply** to save the action. The referenced calculation will execute according to the schedule defined in the associated job description object.

Instantiating Calculation Aspects On Object Types

When adding a calculation to an object type, the calculation will not be copied into the objects created from the object type unless it is specified that the calculation aspect be copied. To do this:

1. Go to the Type Definition aspect for the object type. Figure 66 shows an example of an object type named ProcessDataSim.

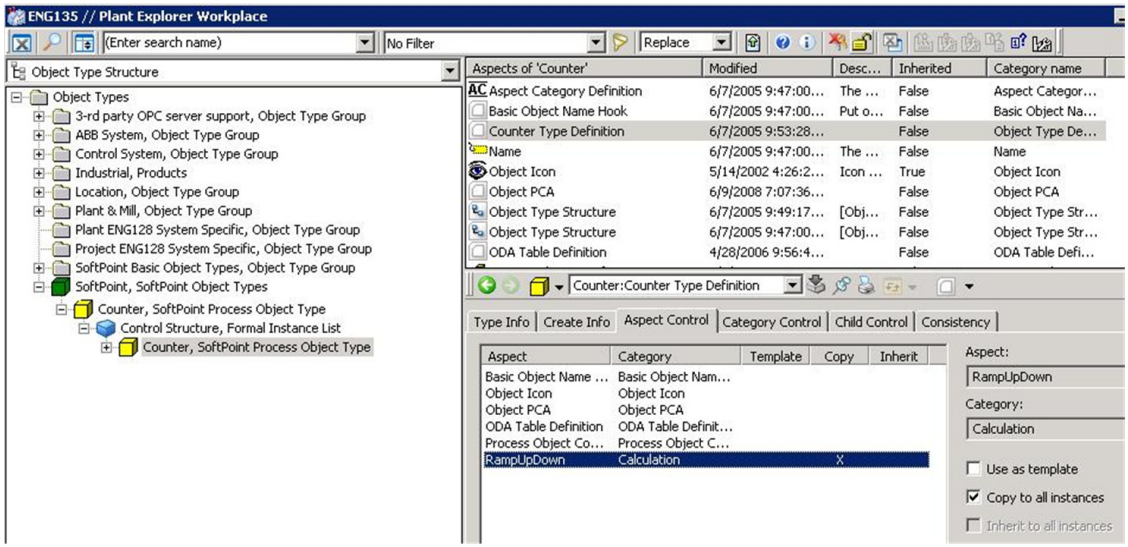


Figure 66. Type Definition Aspect

2. Click the **Aspect Control** tab.
3. Select the aspect, in this case: **Calculation**.
4. Check the box labeled **Copy to all instances**.
5. Click **Apply**.

Remember to enable the calculation aspect for the instantiated object. Use the Enable button on the calculation aspect Tool Bar, or the Calculation Status Viewer (Managing Calculations on page 121).

OPC Access to Calculation Properties

Table 11 describes calculation properties which are accessible by OPC clients. Typically, Result and Timeout are the only properties that may need updating. The other properties are generally intended for read-access only.

Table 11. Calculation Properties

OPC Item Name	Data type	Description
Result (Result.Value)	Variant	Use this to set the value of the calculation result from within the calculation's VB script. This result may be accessed by OPC clients, or by other calculations. Default Value: **0**. For further information refer to Calculation Result on page 92.
Timeout	Long	Maximum number of seconds the calculation should be allowed to run. Default: **30**
ExecutionTime	Long	The amount of time in milliseconds that the calculation took to run. This time does not include OPC reads or writes.
LastExecution	Time	Time of the calculation's last execution.
Status	Long	Status of the calculation. Contains the error code returned by the calculation engine. Default: **0x00000000**
StatusMessage	String	Text describing the status code.Default: ""
TriggerText	String	Text identifying the time or event that triggers the execution of a calculation. Default: ""
DisableOnError	Boolean	Boolean value indicates whether the calculation should become disabled when execution produces an error. Default: **TRUE**
TracingEnabled	Boolean	Boolean value indicating if tracing is enabled for the calculation. Default: **FALSE**
TraceMessage	String	String containing any debug messages from the last execution. Default: ""
SourceCode	String	Executable code of calculation
CalculationLanguage	String	Programming language of executable code. Default: **VBScript**

Making Bulk Changes to Instantiated Calculations

After creating several instances of an object type containing a calculation, it may become necessary at some point to modify all calculations belonging to the objects

of that type. For example, a defect may be found in the source code of the calculation. Opening and editing each calculation by hand to make the same fix could be time consuming and introduce additional defects. The *Clone* function of the Calculation Status Viewer is used to make such changes quickly and consistently.

To do this:

1. Using the Calculation Editor aspect, modify a single calculation with the necessary change. This aspect will become the source for the clone. For example, when disabling tracing for the calculation.
2. Open the Calculation Status Viewer. To do this, go to the Service structure, select the **Calculation Server, Service**, and then select the **Calculation Status Viewer** Aspect, Figure 74.

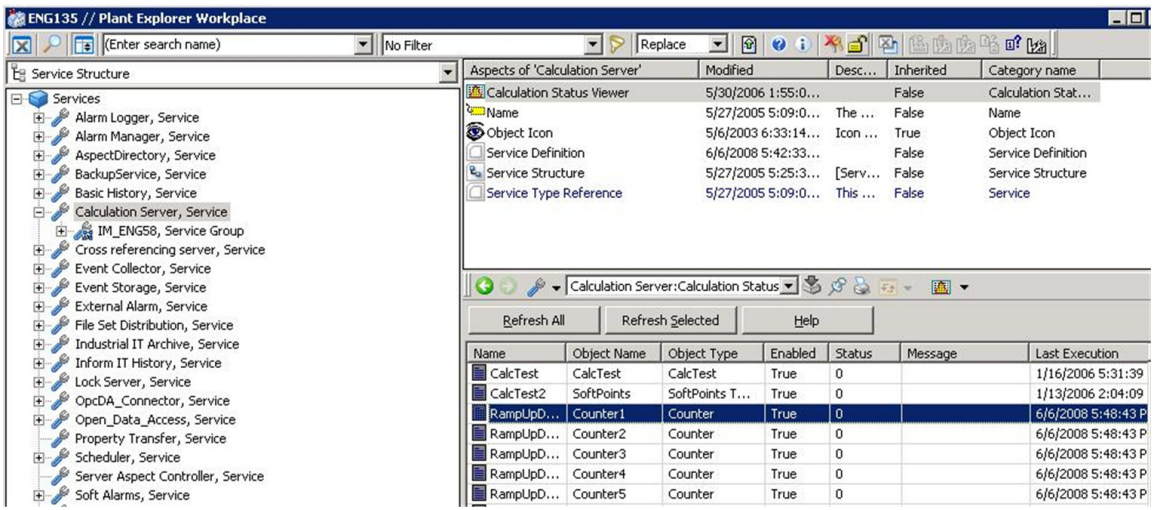


Figure 67. Displaying the Calculations Status Viewer

3. Using the Calculation Status Viewer, select the group of calculations to be modified with the change. Be sure to include the source calculation in the selected set. Then right click on the Calculation Status Viewer list and choose **Clone** from the context menu, Figure 68.

   This menu item will be available only if the following conditions are met:

– The current user has aspect modification privileges.

– All calculations in the selected set are disabled.

– The set of selected calculations contains more than one element.

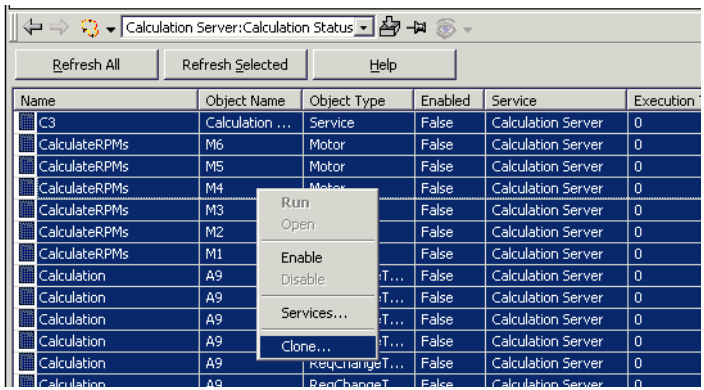


Figure 68. Calculation Status View

This displays the Clone dialog.

4. Choose the source for the clone by selecting from the list in the Clone dialog, the aspect that was modified in step 1. For example, in Figure 69 **C3** is selected as the source.

   All other calculations in the Clone list will acquire the selected property values of the source calculation.

5. Next, select the properties to be cloned. Checking a box indicates that this property should be cloned from the source calculation to all other calculations in the Clone list. In this example, the **TRACEENABLED** property is selected, Figure 69. Note that TRIGGERTEXT is for the Scheduler.

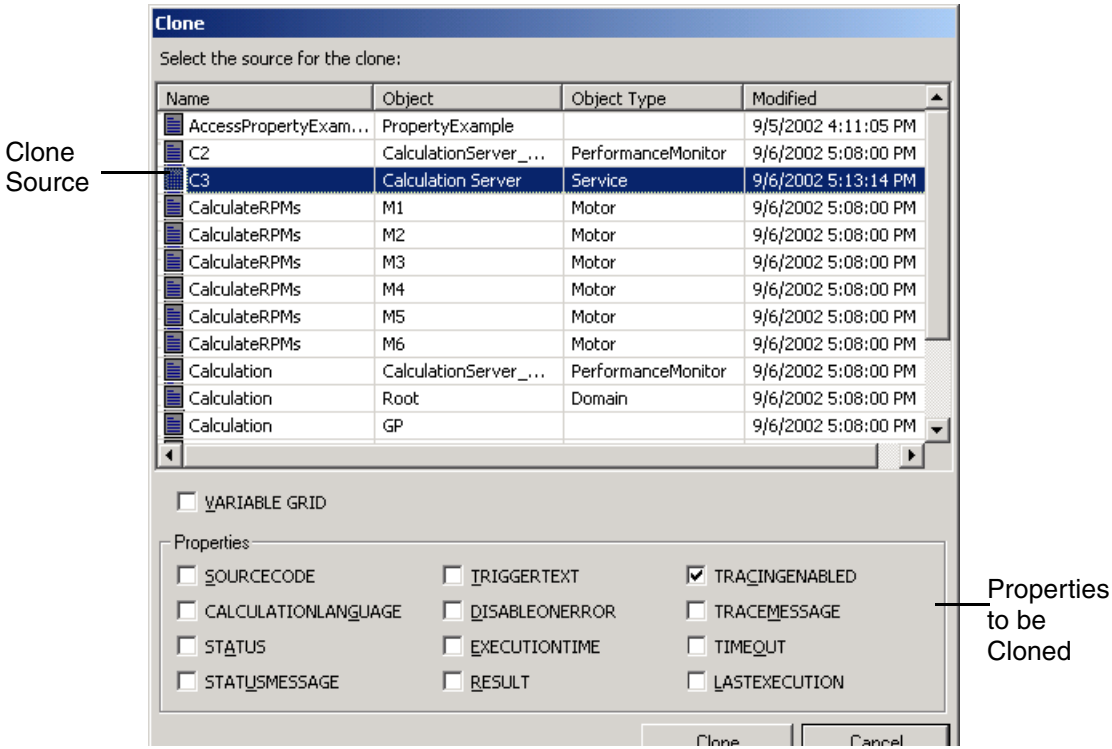
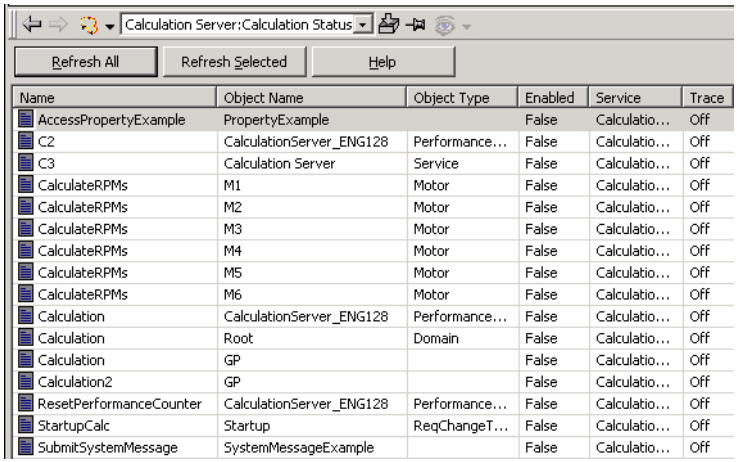


Figure 69. Selecting the Clone Source and Properties to be Cloned

6. Click **Clone**. This displays a confirmation dialog.
7. Click **Yes** to update all calculations with the selected property values of the source calculation.

Figure 70 shows the tracing turned off for all calculation aspects cloned from the selected source (C3).



Name	Object Name	Object Type	Enabled	Service	Trace
AccessPropertyExample	PropertyExample		False	Calculatio...	Off
C2	CalculationServer_ENG128	Performance...	False	Calculatio...	Off
C3	Calculation Server	Service	False	Calculatio...	Off
CalculateRPMs	M1	Motor	False	Calculatio...	Off
CalculateRPMs	M2	Motor	False	Calculatio...	Off
CalculateRPMs	M3	Motor	False	Calculatio...	Off
CalculateRPMs	M4	Motor	False	Calculatio...	Off
CalculateRPMs	M5	Motor	False	Calculatio...	Off
CalculateRPMs	M6	Motor	False	Calculatio...	Off
Calculation	CalculationServer_ENG128	Performance...	False	Calculatio...	Off
Calculation	Root	Domain	False	Calculatio...	Off
Calculation	GP		False	Calculatio...	Off
Calculation2	GP		False	Calculatio...	Off
ResetPerformanceCounter	CalculationServer_ENG128	Performance...	False	Calculatio...	Off
StartupCalc	Startup	ReqChangeT...	False	Calculatio...	Off
SubmitSystemMessage	SystemMessageExample		False	Calculatio...	Off

Figure 70. Result, Tracing Off for All Calculations

Object Referencing Guidelines

Two methods for referencing an object are supported:

- Absolute Referencing Guidelines uses the absolute (literal) path to the object.
- Relative Object Referencing Guidelines uses a relative path rather than the actual (absolute) path. Use this to create reusable applications that are independent of any specific object instances.

  For example, if the process has multiple units performing the same function and having the same basic design (same valves, pumps, I/O signals, etc.), with relative object referencing, one calculation is created to reference the same signals in all units throughout the process.

Before using the object referencing, also refer to Performance Considerations for Object Referencing on page 120. This describes how to improve performance during the enable process, and reducing the chance for errors.

Absolute Referencing Guidelines

The following is an example of an absolute object reference:

plant2.section1.drive3.motor4

Each component (plant2, section1, and so on) references a specific object in the structure hierarchy. Each component is delimited by a *separator* character. The dot (**.**) and slash (**/**) are default separators. Specify different characters to use as separators using the aspect [Service Structure]Services/Aspect Directory-> SNS Separators. The Name Server must be restarted after any changes.

If a separator character is actually part of an object name, it must be delimited with a "\" backslash in the object reference. For example, the string **AI1.1** will search for an object named 1 below an object named AI1. To search instead for an object named AI1.1, reference the object as: **AI1\.1**.

Absolute object references can be refined using the following syntax rules:

- **.** (dot) - search for the following object (right of dot) below the previous object left of dot). This includes direct children as well as their offspring.
- **..** (double dot) - go one level up to the parent.
- **[down]** - (default) searches from parent to children.
- **[up]** - searches from child to parent.
- **[direct]** - Limits search to just direct children. Refer to Direct on page 114.
- **[***structure***]** - Limits search to a specified structure, for example **Functional Structure** or **Control Structure**. Refer to Structure Category on page 115.
- **[name]** - Limits search to the *main* object name (of category *name*). Refer to Name on page 116.

This syntax can be used for powerful queries. Also, restricting a search using these keywords can significantly improve the search speed. For example, the following query path will start from a Signal object, and will retrieve the board for the signal and the rack where this board is located:

.[up][Control Structure]{Type Name}board[Location Structure]{Type Name}rack

Direct

By default, queries search for the referenced object at all levels below the parent object, not just for direct children. To limit the search to direct children, use the **[Direct]** keyword.

For example, the query path plant2.motor3 will find motor3 in all of these structures: heyden.plant2.section3.**motor3**, plant2.section3.drive4.**motor3**, and plant2.**motor3**. The query [Direct]plant2.motor3 will find motor3 only at plant2.**motor3**.

For further considerations regarding the Direct keyword, refer to Using the [Direct] Keyword on page 120.

Structure Category

All structures are searched by default. Restrict the search to a specific structure by specifying the structure name, Table 12. For example: [Functional Structure]drive3.motor2 searches only in the Functional Structure. The query [Functional Structure]drive3.motor2[]signal4 starts in the Functional Structure until it finds an object named motor2 below an object named drive3, and then searches all structures for an object named signal4 below motor2.

Use the [Direct] keyword in combination with a structure reference, for example: [Direct][Functional Structure]plant2.section1. Do not specify several structures. For example, the following syntax is not allowed: [Functional Structure][Control Structure]motor2. If this functionality is needed, use the IAfwTranslate::PushEnvironment method of the Name Server.

To negate a query path, add the not symbol. For example, [!Functional Structure]motor2 searches for a motor2 *not* occurring in the Functional Structure.

Table 12. Structure Names

Structure Name	Description
Location Structure	Location orient view
Functional Structure	Function oriented view
Control Structure	Contains controllers, networks, IOs etc.
Product Structure	Product oriented view
Object Type Structure	Defines and structures Object Types
User Structure	Contains users and user groups
Workplace Structure	Contains workplaces

Table 12. Structure Names (Continued)

Structure Name	Description
Aspect System Structure	
…….	

For further considerations regarding Structure, refer to Using the Structure Name on page 120.

Name

An object can have a number of different names. Each name is assigned as an aspect. The name categories are described in Table 13. The main name is the name that associated with the object in the Plant Explorer. This is generally the name used for absolute object references. By default a query will search for all names of a given object. This may provide unexpected results when objects, other than the one that was intended to be referenced, have names that fit the specified object reference. For example the query A* may return objects whose main name does not start with A, but whose OPC Source Name does start with A. This can be avoided by using the [name] keyword. This restricts the search to main names (category name) only. For example: {Name}A*.

Table 13. Name Categories

Name Category	Description
Name	Main name used to identify objects in the Plant Explorer.
Relative Name	Name to be used for relative references
Xxx Designation (e.g. Functional Designation)	Used as relative reference designation. “Xxx Designation” is used for the “Xxx Structure”
Xxx Id (e.g. Product Id)	Legacy names from AEW 2.0. Not to be used.
OPC Source Name	Read-only name filled by the Uploader. Used to identify alarms.
Aspect Name	The name of the aspect.

Examples

Table 14. Example Queries

Query	Description
“{Functional Designation}M1.R2”	Look only at name aspects of category ‘Functional Designation’. First search an object M1 and from there an object R2.
“[Control Structure]board11[Functional Structure]signal3”	Search for board11 in the Control Structure, then switch to the Functional Structure and search signal3.
“[Control Structure]board11[]signal3”	Search “board11” in the Control Structure and from there search all structures to find signal3.
“[Direct][Functional Structure]plant.a1”	Start with the root object plant in the Functional Structure and search for a direct child named a1.

Relative Object Referencing Guidelines

Relative object referencing creates reusable solutions. This functionality is supported by attaching a Relative Name aspect, Table 13, to an object.

It is recommended that relative naming be done at the object type-level. This way all instances of the object type will automatically have the proper relative name.

This is illustrated in the following example where a motor object type requires a calculation that references two signals to start and stop the motor. The motor object type is instantiated in two units - Unit 1 and Unit 2, Figure 71. The main names (Name aspect) for the signals are instance-specific and thus change from one unit to the next.

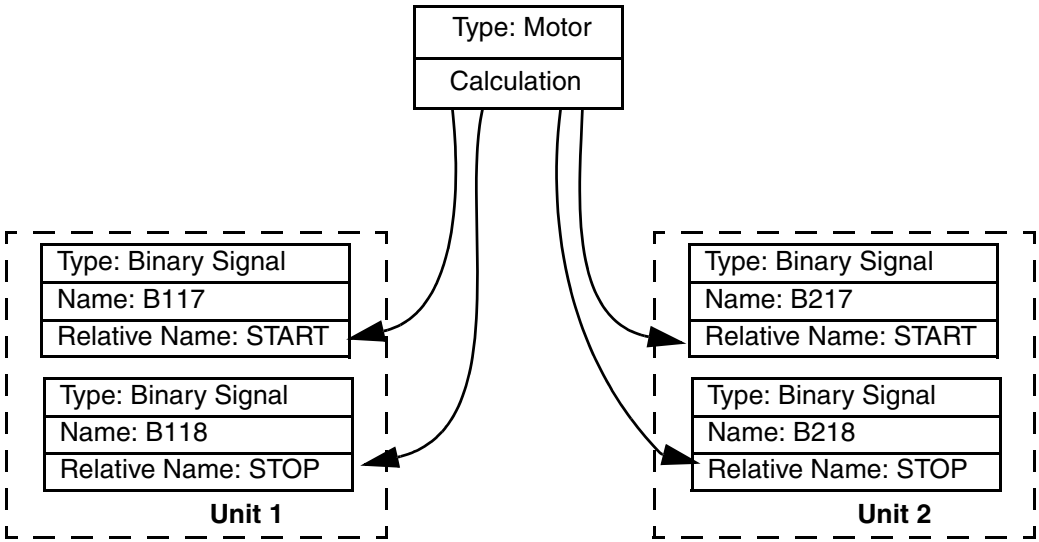


Figure 71. Relative Referencing to Child Objects

Follow these guidelines to create a reusable solution (based on example in Figure 71):

1. Create a motor object type with two binary signals.
2. Use the Relative Name aspect for each of the two binary signals to specify a relative name: for example **START** and **STOP**, Figure 72.

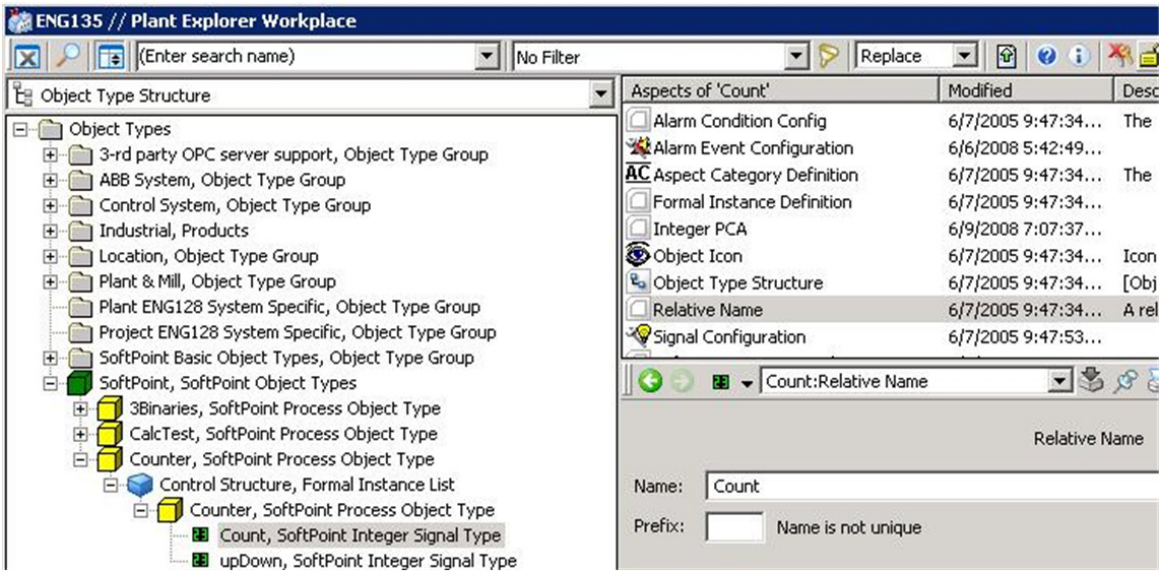


Figure 72. Example, Relative Name Aspect

3. Add the Calculation aspect to the motor object type. When adding the variables for the two binary signals, specify the object reference, Figure 73.

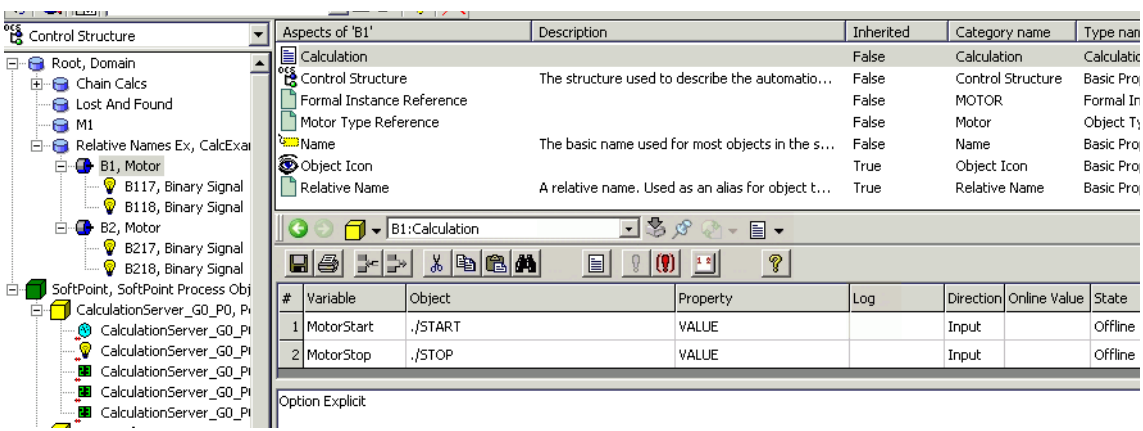


Figure 73. Example, Relative Object Referencing

The dot (.) at the beginning of the object reference specifies that the reference will start at the object that owns the calculation aspect, in this case the motor object. Only motor objects in the Control Structure will be referenced.

4. Instantiate the motor object type in the Control Structure under the objects representing Units 1 and 2 respectively. When the calculation runs for the motor on Unit 1, the references for START and STOP will point to B117 and B118 respectively. When the calculation runs for the motor on Unit 2, the references for START and STOP will point to B217 and B218 respectively.

Performance Considerations for Object Referencing

When using relative or absolute object references for calculation inputs and outputs, there are techniques that can be used to improve performance during the enable process and reduce the chance for errors.

Using the [Direct] Keyword

When using relative or absolute references, the path to the object should include the [Direct] keyword. By using this keyword, the platform's name server will resolve the object reference more efficiently.

Examples:

[Direct][Control Structure]Area_51/Motor_1/On

[Direct][Functional Structure]Sector_7G/Pump_1/On

.[Direct][Control Structure]Signal5

Using the Structure Name

Second, when creating relative references to objects it is important to include the name of the structure where the object will be found. This is absolutely necessary when the referenced object appears in more than one structure.

Examples:

.[Direct][Control Structure]Signal5

.[Direct][Functional Structure]P1/A

Creating a Calculation as an Object Type

When defining calculations in the Object Type structure, there is an additional consideration. After entering the structure into the path, the property list will no longer be populated as the list is built from the object reference, which in the context of the Object Type Structure is no longer valid. Therefore it is best to select the property for the reference before adding the name of the specific structure to the path.

Managing Calculations

Calculations are managed via the Calculations Status Viewer. This viewer supports the following functionality:

- Reading Calculation Status Information.
- Enabling/Disabling Calculations.
- Distributing Calculations on Multiple Servers.

In addition, the ability to collect and display performance statistics generated by the Calculation server can be configured.

Some tasks performed in the Status Viewer may be lengthy, for example enabling or disabling a large number of calculations. DO NOT close or change the Status Viewer aspect while any Status Viewer task is in progress. The view MUST remain open until the task is finished; otherwise, the client which is running the task will be disconnected from the Calculation service.

Take the following measures to ensure that the view remains unchanged and the client remains connected:

- If the Calculation Status Viewer aspect is hosted within the Plant Explorer workplace, use the *Pin / Unpin* feature to ensure that no other aspect replaces it during the task. Also, do not close the hosting workplace until the operation is complete.
- If the Calculation Status Viewer aspect is being displayed in a child window, do not close the window until the operation has completed. Also do not close the workplace from which the child was launched, as this will close all child windows.

To access this viewer, go to the Service structure, select the **Calculation Server, Service**, and then select the **Calculation Status Viewer** Aspect, Figure 74.

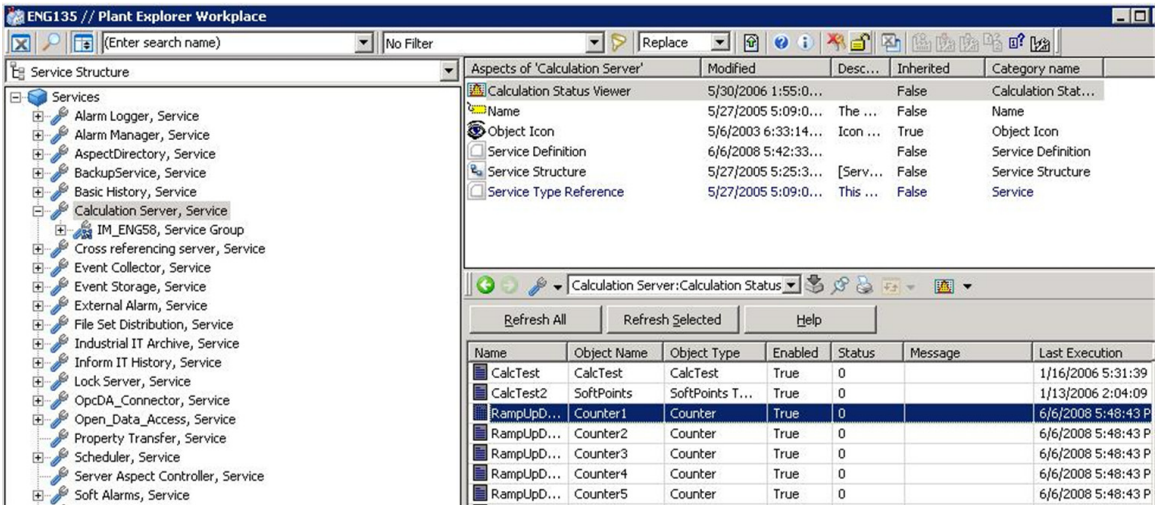


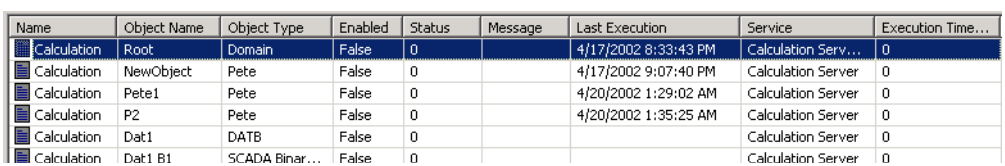
Figure 74. Launching the Calculations Status Viewer

Then refer to the applicable procedure as described below:

- Reading Calculation Status Information on page 122.
- Enabling/Disabling Calculations on page 123.
- Distributing Calculations on Different Servers on page 124.
- Performance Tracking on page 125.

Reading Calculation Status Information

The status viewer has eight columns of information for each calculation, Figure 75. These are described in Table 15.



Name	Object Name	Object Type	Enabled	Status	Message	Last Execution	Service	Execution Time...
Calculation	Root	Domain	False	0		4/17/2002 8:33:43 PM	Calculation Serv...	0
Calculation	NewObject	Pete	False	0		4/17/2002 9:07:40 PM	Calculation Server	0
Calculation	Pete1	Pete	False	0		4/20/2002 1:29:02 AM	Calculation Server	0
Calculation	P2	Pete	False	0		4/20/2002 1:35:25 AM	Calculation Server	0
Calculation	Dat1	DATB	False	0			Calculation Server	0
Calculation	Dat1 B1	SCADA Binar...	False	0			Calculation Server	0

Figure 75. Calculations Status Viewer

Table 15. Calculation Status Parameters

Parameter	Description
Name	Calculation Name specified when the aspect was created.
Object Name	Name of the object for which the calculation aspect was created.
Object Type	Type of object for which the calculation aspect was created.
Enabled	Indicates whether a calculation is enabled (True) or disabled (False). Toggle states via the right-click menu. Refer to Enabling/Disabling Calculations on page 123.
Status	Status codes for errors generated by related applications including OPC, Microsoft, and SoftPoint Server. These codes are intended for ABB Technical Support personnel.
Message	Textual message related to status.
Last Execution	Date and time that the calculation was last executed.
Service	Calculation Server where the calculation runs. This assignment can be changed. Refer to Distributing Calculations on Different Servers on page 124.
Execution Time	Time it took for the last execution of the calculation.

Sorting of columns with numbers or dates is done alphabetically which can yield unusual sort for dates and numbers.

The following properties will not be updated when the Settings.UpdateStatus line in the script is set to False (default condition): Result, Execution Time, last Execution, Status, Status Message.

Enabling/Disabling Calculations

To enable or disable one or more calculations, select the calculations in the viewer, then right click and choose **Enable** (or **Disable**) from the context menu, Figure 76.

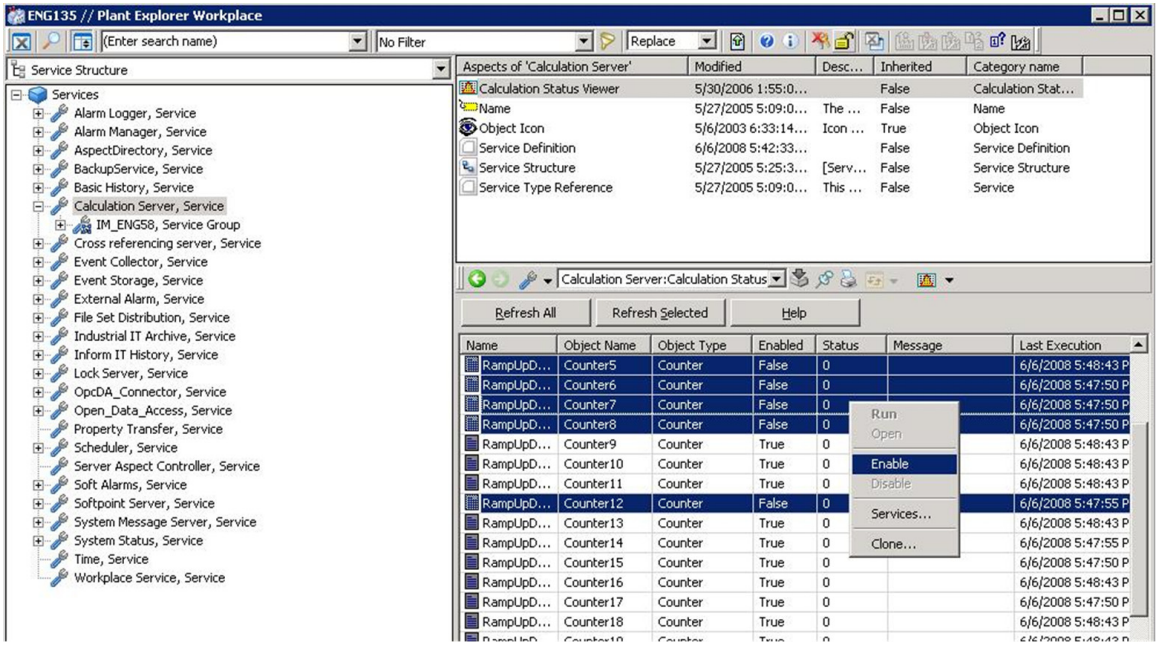


Figure 76. Enabling Calculations

The Enable button in calculation window does not compile the VBScript before actually enabling the calculation. Test script in offline mode.

Assign each Calculation to a specific Service Provider BEFORE enabling the calculation. This procedure is described in Distributing Calculations on Different Servers on page 124.

Distributing Calculations on Different Servers

Assign each calculation to a specific Calculation Service Provider BEFORE enabling the calculation. If the system has multiple Calculations Servers, assign calculations to different servers to distribute the processing load.

To do this, select the calculations to be assigned to a particular server, then right-click and choose **Services** from the context menu. This displays the Calculation Services dialog, Figure 77.

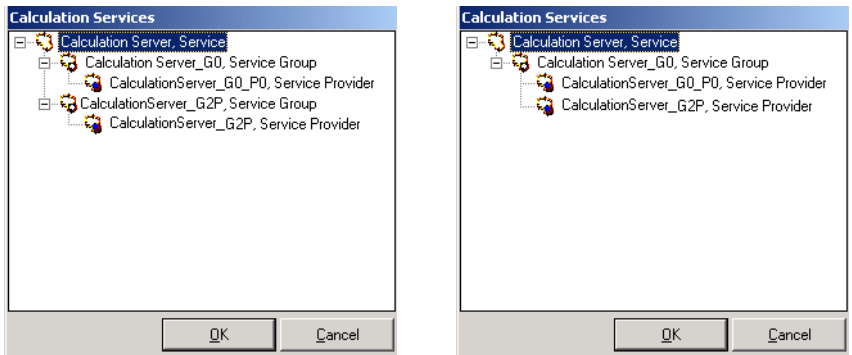


Figure 77. Calculation Services Dialogs (Dual on left and Redundant on right)

This dialog lists all Calculation Service Groups and Service Providers that exist in the current system. Assign the selected calculation to a Calculation Service Group. In a non-redundant configuration, only one Calculation Service Provider exists in a Calculation Service Group. In a redundant configuration, two service providers exist in one group. Their can be multiple Calculation Service Groups.

Performance Tracking

The Calculation Server generates performance statistics that used to monitor and adjust the load on the server. These statistics may be collected by a SoftPoint object created in the Control structure. Employ a trend aspect to monitor the performance over time, identify peaks during execution, and take corrective action if necessary. Guidelines for implementing this functionality are provided in:

- Collecting Performance Data for a Calculation Service on page 125.
- Accessing the Performance Data on page 130.
- Improving Performance on page 131.

Collecting Performance Data for a Calculation Service

To collect the performance statistics for a Calculations Server:

1. Create a PerformanceMonitor object type under the SoftPoint Objects category in the Object Type structure. This is facilitated by the PerformanceMonitor.afw import file provided with the Calculations Services software.

2. Instantiate the PerformanceMonitor object type in the Control structure. The PerformanceMonitor object must have the same name as the Calculations Service Provider whose performance data it is logging. This may require renaming the Calculations Service Provider.
3. Deploy the SoftPoint configuration.

The SoftPoint Server must be running on the same machine as the Calculation Server for data to be collected.

Detailed instructions for configuring SoftPoint objects are provided in Section 3, Configuring SoftPoints. The following sections provide guidelines related to the PerformanceMonitor object for collecting performance statistics.

Creating the PerformanceMonitor SoftPoint Object Type

Use the PerformanceMonitor.afw file to create the new SoftPoint object type and associated signal types. This file is located in the help/util directory where the Calculations Services software is installed (Default C:\Program Files\ABB Industrial IT\Inform IT\Calculations\help\util).

Use the Afw Import/Export tool to load PerformanceMonitor.afw. Browse to the file in Windows Explorer, and then double-click the file, or open the file from the Afw Import/Export tool.

With the file open in the Afw Import/Export tool, choose **Action>import All**. This creates the PerformanceMonitor object type under the SoftPoint Process Object Type category in the Object Type Structure, Figure 78.

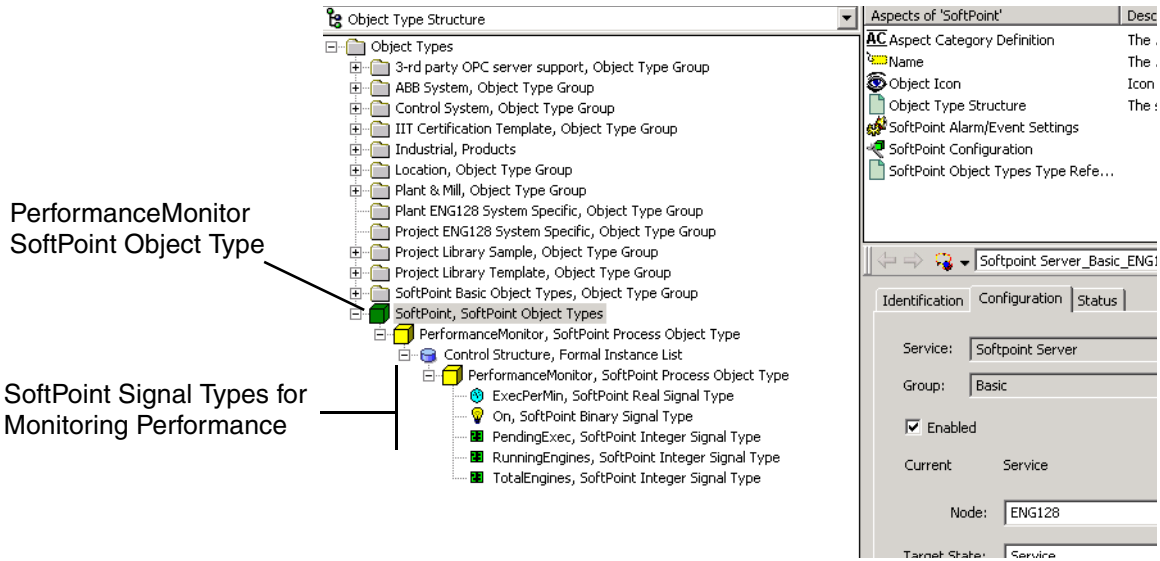


Figure 78. Performance Monitor Object Type

Instantiating the PerformanceMonitor in the Control Structure

To start collecting performance data for a particular service, create a new instance of the PerformanceMonitor object type in the Control Structure. The name given to the performance monitor object instance must match the name of the Calculation Service Provider to be monitored.

The SoftPoint object name cannot contain spaces. If the Calculations Service Provider name contains spaces (default name contains spaces), rename the Service Provider before instantiating the PerformanceMonitor SoftPoint object.

To change the name of a Calculation Service Provider, go to the Services structure and select the name aspect of the provider as shown in Figure 79.

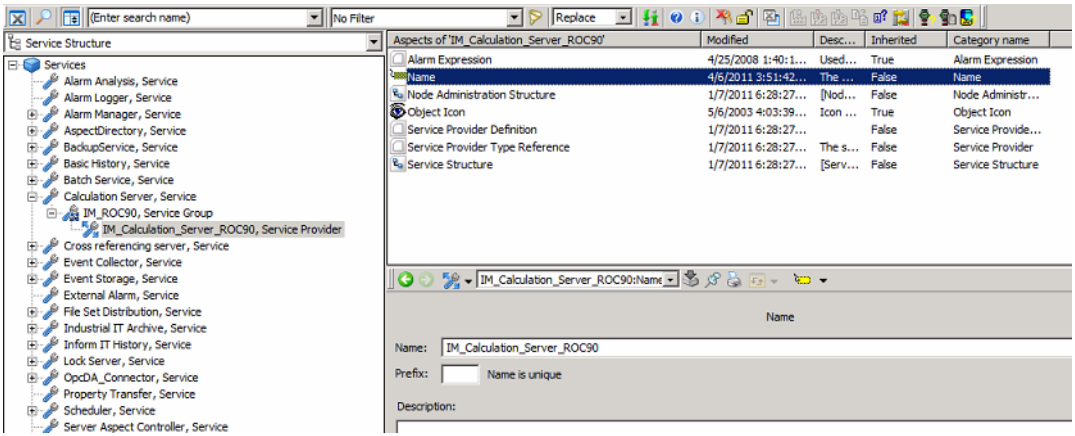


Figure 79. Changing the Service Provider Name

Create the PerformanceMonitor instance under the SoftPoint Object category in the Control Structure. Remember to use the name of the Calculation Service Provider when naming the SoftPoint object, Figure 80.

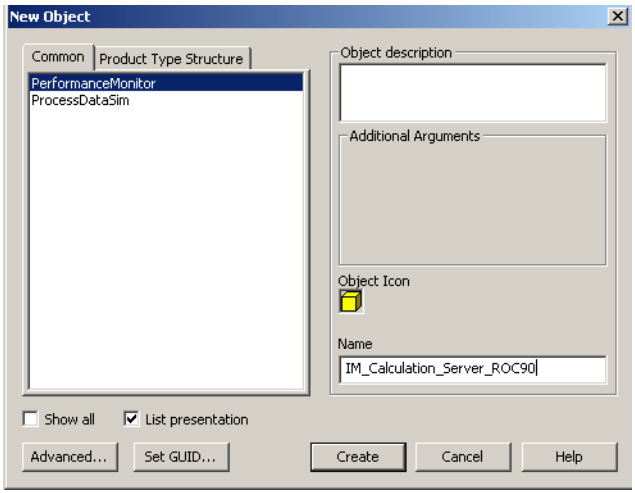


Figure 80. Instantiating the PerformanceMonitor Object

The instantiated PerformanceMonitor object is illustrated in Figure 81.

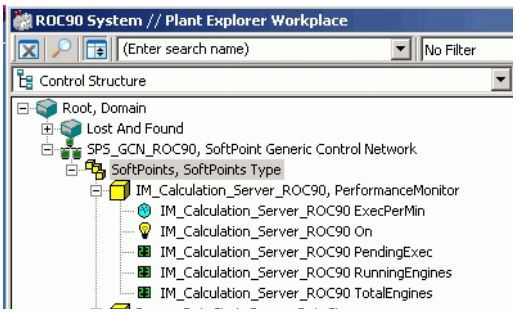


Figure 81. Instantiated Performance Monitor

After creating the Performance Monitor object, use the SoftPoint configuration aspect to deploy the new SoftPoint configuration. Refer to Deploying a SoftPoint Configuration on page 70Deploying a SoftPoint Configuration for details.

Accessing the Performance Data

The SoftPoint signals for monitoring performance are described in Table 16. Monitor these signals via the Object dialog as shown in Figure 82.

Table 16. SoftPoint Signals for Performance Monitoring

Signal	Data Type	Description
ExecPerMin	Real	The number of calculations per minute the calculation service is currently executing. This value is updated every five seconds.
PendingExec	Integer	The number of calculations waiting to be executed by the calculations service (Maximum: 10,000). This value is updated every second. If this number grows at a steady rate, the calculation service may be overloaded or another process may be using all the available processor time.
RunningEngines	Integer	The number of calculation engines that are currently executing in parallel. Each engine executes one calculation. A maximum of 100 engines may be executing concurrently. This value is updated every second.
TotalEngines	Integer	The total number of engines currently allocated to run in parallel (Maximum: 100). This value is updated every second.
On	Binary	If the state of this signal is TRUE, the signals available for monitoring will be updated. By default, this signal is turned off (VALUE = FALSE).

It may be more useful to attach a trend aspect to the PerformanceMonitor object to monitor trends in the performance statistics over time. For instructions on how to do this, refer to the section on using operator trend displays in *System 800xA Operation (3BSE036904*)*.

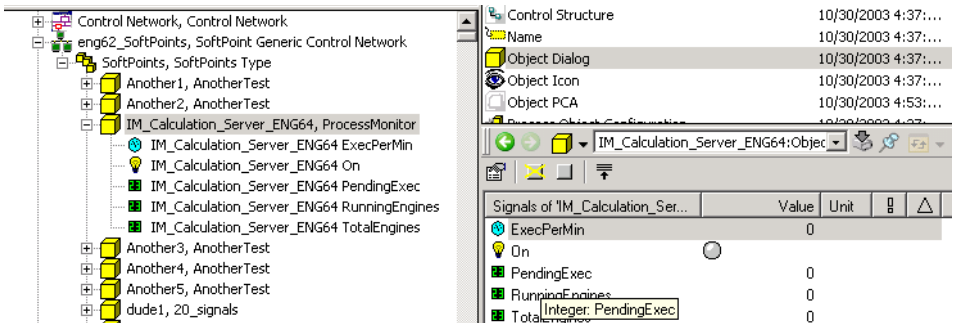


Figure 82. Viewing Performance Data Via the SoftPoint Object Dialog

Improving Performance

Refer to the Calculations Service Recommendations and Guidelines on page 133 for specific use cases related to improved stability and predictability of the Calculation Service.

Several factors affect the execution time for a calculation. These include:

- Number of inputs and outputs.
- Size and nature of the calculation source code.
- Persistence of the data in the aspect directory after calculation execution.

Data persistence can account for up to one-third of the total execution time of a calculation. This is controlled on an individual calculation basis by the **Settings.UpdateStatus** command in the calculation script. By default this command is set to **False**. This means STATUS, STATUSMESSAGE, EXECUTIONTIME, LASTEXECUTION, TRACEMESSAGE, RESULT properties, and offline output values are NOT updated in the aspect directory upon completion of the calculation execution. It is generally recommended that the calculation update status be set to False.

Set the Update Status in the calculation script to true to have these properties updated. For example, update the calculation properties every *n* number of executions by using the Settings.UpdateStatus in a loop that sets the value to TRUE after *n* iterations. An example is shown in Figure 83.

Execution times for calculations may be reduced up to 33%. For calculations that do not execute often or for calculations with logic that require a significant amount of

time to execute, this savings will be negligible. The savings will be most valuable for smaller calculations that execute often.

Refer to the considerations below to better understand the consequences of using the UpdateStatus feature to disable persistence of the calculation properties.

#	Variable	Object	Property	Direction	Online Value	State	Offline Value	Event
1	ExecCount	Root	PTEST:INTEGERVALUE	Output	0	Offline	**0**	False

```
'-------------------------------------------------------------
' Example illustrating the use of Settings.UpdateStatus
'-------------------------------------------------------------

If ExecCount >= 10 Then              ' If the execution count is greater than 10,
   ExecCount = 0                     ' reset the counter.  By default Settings.UpdateStatus
                                     ' is True so that the data will be persisted in this case.
Else
   Settings.UpdateStatus = False     ' Otherwise, set the UpdateStatus flag to False.
                                     ' The status data will not be persisted to the Aspect Directory.
   ExecCount = ExecCount + 1         ' Also increment the execution count.
End If
```

Figure 83. Example, Using UpdateStatus

Considerations when the UpdateStatus is set to False are:

- The STATUS and STATUSMESSAGE properties for the calculation are NOT updated after execution. Any errors that may have occurred during execution will not be visible via the Status Viewer aspect or the calculation editor.
- The LASTEXECUTION property for the calculation is NOT updated after execution. The last execution time of the calculation will not be visible via the Status Viewer aspect or the calculation editor making it more difficult to verify the calculation is running. However, use the Now function within the source code of the calculation to write the current time to an external SoftPoint.
- The EXECUTIONTIME property for the calculation is NOT updated after execution. The amount of execution time for the calculation is not shown.
- The RESULT property for the calculation is NOT updated. Any changes to this from within the source code of the calculation will not be persisted. Chaining of calculations will require an external data point.
- The TRACEMESSAGE property for the calculation is NOT updated after execution. Calculation tracing is not available.
- Offline output variables will NOT be updated. Any changes to them from within the source code of the calculation will not be persisted. External data-points must be created and used to accomplish the same goal.

Calculations Service Recommendations and Guidelines

The following recommendations for improved stability and predictability of the Calculation Service have been identified.

Use Case 1 - Settings.UpdateStatus

The script command ***Settings.UpdateStatus*** is used to assist with the debugging of a calculation. The value is set to ***FALSE*** by default if it is not present. This setting should be set to ***TRUE*** only while debugging a calculation. If the value is left as ***TRUE*** for long periods of time, there will be additional loading on the Aspect Server and high levels of memory consumption, potentially leading to system failure. The system loading is related to the frequency and quantity of calculations executed.

It is recommended to review all calculations to insure that the ***Settings.UpdateStatus*** variable is set to ***FALSE*** or that it is not part of the calculation at all.

Use Case 2 – Assignment of Output Variables

The script command language is based on Visual BASIC scripting. It has been observed that in some cases the Visual BASIC scripting language can not determine the data type of the output variables and can produce unpredictable results.

It is recommended to **CAST** all output variables to the data type of the actual output property.

Use Case 3 - SoftPoint Signals Show Bad Data Quality when Calculations are not Updating Data Quality

The quality of calculations inputs are used as the output if the calculation body does not expressly set the quality attribute of the output variable. A race condition can occur where bad data quality can be sent to SoftPoints before the input received indicating the true data quality from the input signal. This can occur when the SoftPoint Server is restarted either manually or as part of the reboot of the subsystem that executed the SoftPoint.

It is recommended when using a SoftPoint as an output destination that as part of the calculation the data quality should always be set.

Use Case 4 – Usage of External Libraries and COM/DCOM from a Calculation Aspect

The Calculation Service is designed as an execution platform for performing calculations. The Calculation Service is not a programming platform to integrate any and all external functions. This is especially important when considering the use of function libraries and functions which may use COM/DCOM or external communication. It is not recommended to use functions that may produce unpredictable results or may require external network based communication as part of the processing.

Use Case 5 – Usage of Calculations in a Redundant Configuration

The Calculation Service is typically configured nonredundant by default. The Calculation Service can be configured as a redundant configuration. The configuration changes must be performed manually for a redundant Calculation Service, The Service Group must be extended with a second Service Provider. Only one Calculation Service Provider can be defined on a single node.

Calculations defined to execute on the Service Group where the Service Group is define with a primary and secondary Service Provider will execute on the primary Service Provider and if a failure occurs of the primary Service Group the calculations will be moved to the secondary Service Provider as defined in the Service Group. The execution on the secondary service is independent and no status or state information is exchanged between the primary and secondary servers other than what is available in the aspect directory. When the primary server has returned to service, the calculations will be moved back to the primary Service Provider and the secondary Service Provider will return to standby.

Use Case 6 - Calculations Service Diagnostic Messages

The Calculations Service may send out at least one diagnostic each time the Calculation executes and can not write to one of the output variables.

After Calculations are deployed, review the Calculation Service event list for diagnostics indicating that the outputs are failing. Correct any outputs that are pointing to incorrect or nonexistent object properties.

Section 5 Configuring History Database

This section provides details on configuring History databases.

Changing the Default History Database Instance

History Services database configuration is not possible until a History database instance is created. A History database instance is created as a post installation step. If the database requirements were known, then the instance may already be properly configured. If those requirements were not known, a default database instance was probably created. The default database meets the requirements of many History applications. When necessary, drop the default instance and create a new larger instance depending on performance requirements. This should be done BEFORE configuring the database and collecting historical data.

This procedure requires logging in as a HistoryAdmin user. Refer to Managing Users on page 589.

Dropping the Default Database Instance

To drop the default database instance:

1. Launch the History Database Maintenance wizard, Figure 84.
   - from the configuration assistant, select **Create History Database** then click **Run Selected Configuration Tool**.
   - from the Start menu, choose **Start>Programs> ABB Industrial IT 800xA>Information Mgmt>History> Oracle Instance Wizard**.

The Maintenance option is for viewing and adjusting table sizing. Do NOT use this option unless the requirements for managing Oracle databases is known.

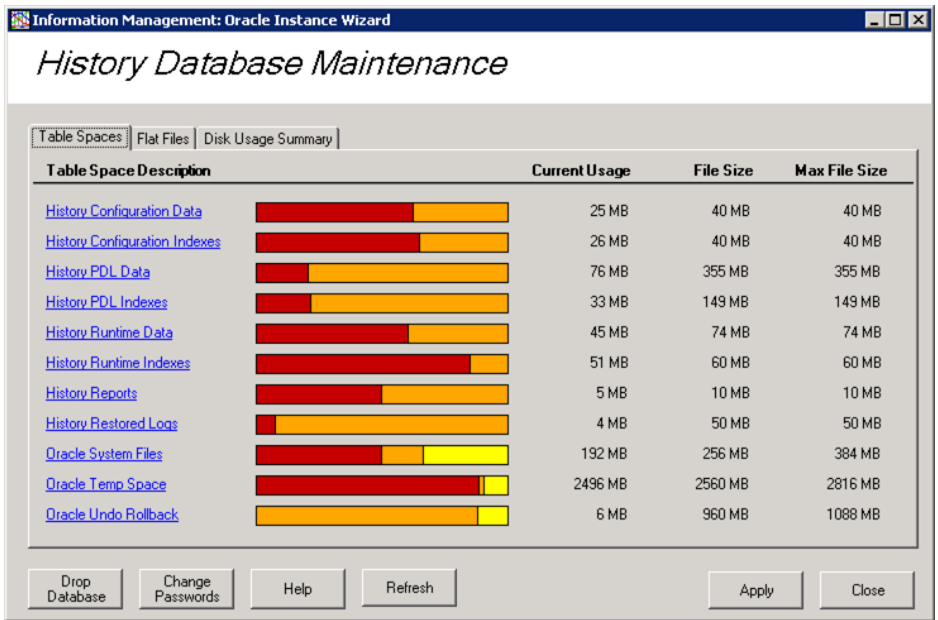


Figure 84. Instance Maintenance Wizard

2. Click **Drop Database**.

   A warning message is displayed to confirm whether or not to drop the database instance: “Are you sure you want to drop the Oracle Instance? All configurations and data that have not been backed up will be lost.”.

The History database configuration and all Oracle-based Historical data will be lost when the database instance is dropped.

3. Click **Yes** to continue dropping the database instance. The time to complete this process may take up to several minutes. Progress is indicated by the History Database Instance Delete message.

Once the default database instance has been dropped, create a new properly sized instance. Refer to the Post Installation book Information Management section for information on creating a new database instance.

Duplicating an Existing Configuration

Once a properly sized instance is created, the configuration can be duplicated on several machines. A configuration file exists in the following location:

C:\ProgramData\ABB\IM\History\instance_config.txt

Use this configuration file with the **Existing Configuration** option.

Maintaining the Oracle Instance

Use the instance wizard to maintain data files after the database instance is created.

The general procedure for modifying a data file is:

1. Choose **Start>Programs> ABB Industrial IT 800xA>Information Mgmt>History> Oracle Instance Wizard**. The wizard, Figure 84, supports the activities described in Table 17.
2. Click the appropriate Table Space Description on the History Database Maintenance display.
3. Make the changes to the data file and then click **OK** to the Data Files changes.
4. On the History Database Maintenance display click **Apply** to update Oracle and the configuration file. Clicking **Cancel** after changes are made displays a message to verify the cancel.

Table 17. Utilities Supported by the Instance Maintenance Wizard

Tab	Refer to	When to Use
Tablespaces	Extending Oracle Tablespace Via Instance Maintenance Wizard on page 138	Greater storage capacity than provided by default set up is required.
Flat Files	Flat File Maintenance via the Instance Maintenance Wizard on page 139	Pick a drive and size limit.

Tablespace Maintenance Considerations

Oracle tablespaces will be sized correctly if the correct choices are made for each log type in Oracle Instance wizard. Also putting Oracle and file-based data on dedicated data drives rather than the system drive, allows tablespaces to autoextend up to 32 Gigabytes, if available, on the drive where they reside. Additional data files must be added manually. Not making the correct selections based on your storage requirements can cause tablespace to run out.

Be sure to configure the Oracle History table space for Index and Runtime large enough to support your storage requirements. If these are not properly sized (when the Oracle database instance is created), messages will be lost when these tables reach their maximum size.

Extending Oracle Tablespace Via Instance Maintenance Wizard

This report provides a summary of the currently used table space, the current allocation and the maximum file size available in both a bar graph and table format.

When the indicated Used Space for a tablespace reaches its Total Space, the corresponding Datafile may autoextend if it is configured to do so. This is specified in the Autoextend column. If **Auto Extend** is checked, the Datafile capacity will be extended by the amount indicated in the **Extend By (MB)** column. The Datafile may be extended in this manner up to the specified **Max Size (MB)**.

Add Datafiles for a Tablespace by clicking the **Add New Data File** button. This adds a new row to the table. Modify any row as required directly in the table as described below. Click **OK** to save the changes or **Cancel** to exit without saving.

Disk Drive	Use the pull down to select a drive letter.
File Name	Enter a name for the file. Include the file extension .DBF
File Size	Specify the starting capacity for this file in megabytes. This will be indicated in the Cur Size column.
Auto Extend	Check to set Auto Extend. If left unchecked, the Datafile will not autoextend.

Extend By	Specify the size of additional increments to extend by when the used space reaches the file size space. This is ignored if Autoextend is disabled.
Max Size	Enter the limit for auto extending the Datafile. This is ignored if Autoextend is disabled.

Flat File Maintenance via the Instance Maintenance Wizard

Use the Instance Maintenance Wizard **Flat File** tab, Figure 85, to select a location for flat file space from a list of drives available for History data storage. Select the Flat File Quota (MB) column for a drive and enter the size of the desired flat file space.

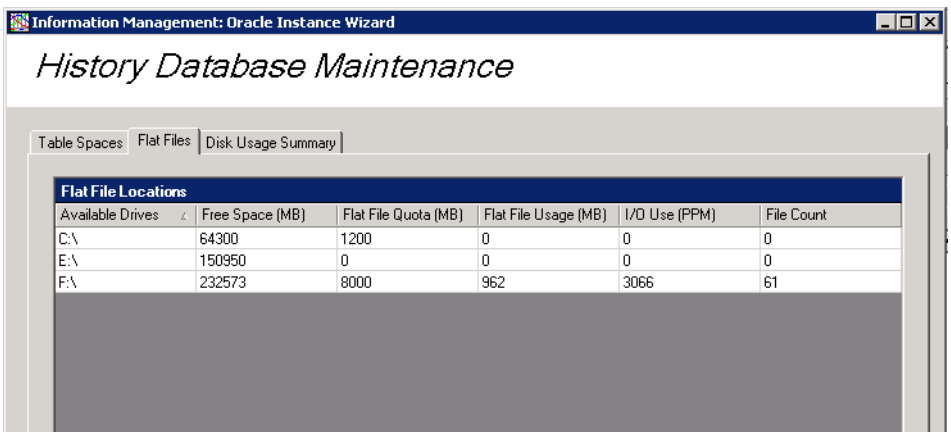


Figure 85. Flat Files Maintenance Tab

If there is 1 GB of Flat File space on a drive and this is reduced to 0 MB on that drive, then the wizard automatically makes the Flat file space 1 GB. This keeps the Flat File space and prevents it from growing beyond its current size.

Disk Usage Summary via the Instance Maintenance Wizard

Use the Instance Maintenance Wizard **Disk Usage Summary** tab, Figure 85, to refer to disk space usage from a list of drives available for History data storage. Select the a drive to refer to the usage.

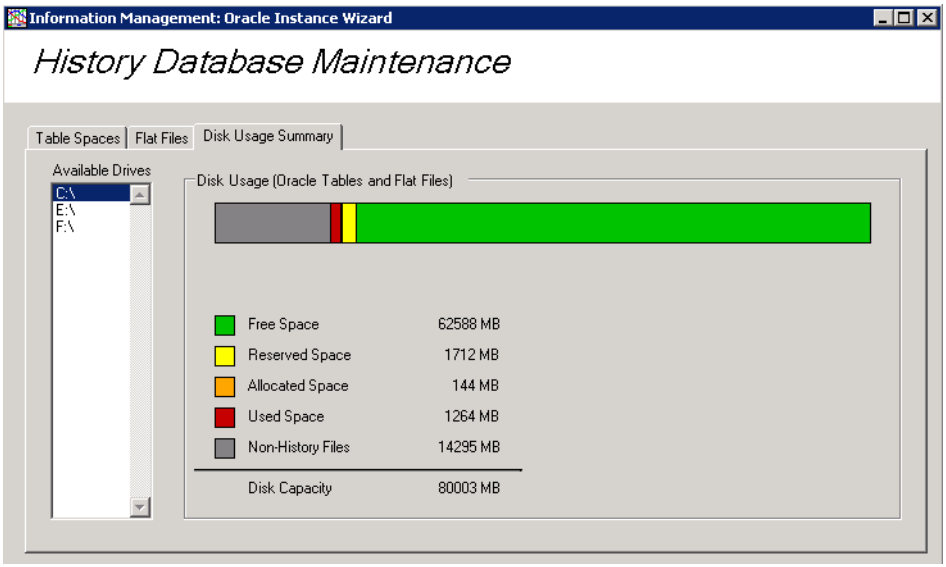


Figure 86. Disk Usage Summary Tab

Oracle Passwords

The default Oracle users are described in Oracle Users on page 592. When creating the Oracle database the user has to enter a password.

Oracle has defined a set of guidelines to enter the password. Refer to Oracle User Password Guidelines on page 142.

Optionally, they can specify a separate password for the *history* read only account. If a separate password is not specified for the *history* account, the same password for the other named accounts is used for the *history* account. It is recommended that the administrator Oracle accounts have the same password as the System 800xA Service account. These Oracle accounts are never used by customers and a common knowledge of the password is not required for day-to-day operations. If operators and other users logon to Oracle, they should utilize the *history* read only account. When utilized, the *history* account should have a different password than the administrative Oracle user account. Oracle passwords can be changed from the System 800xA Install account at any time without affecting the operation of the

Information Management server. This can be done by using the **Change Password** button as shown in Figure 87.

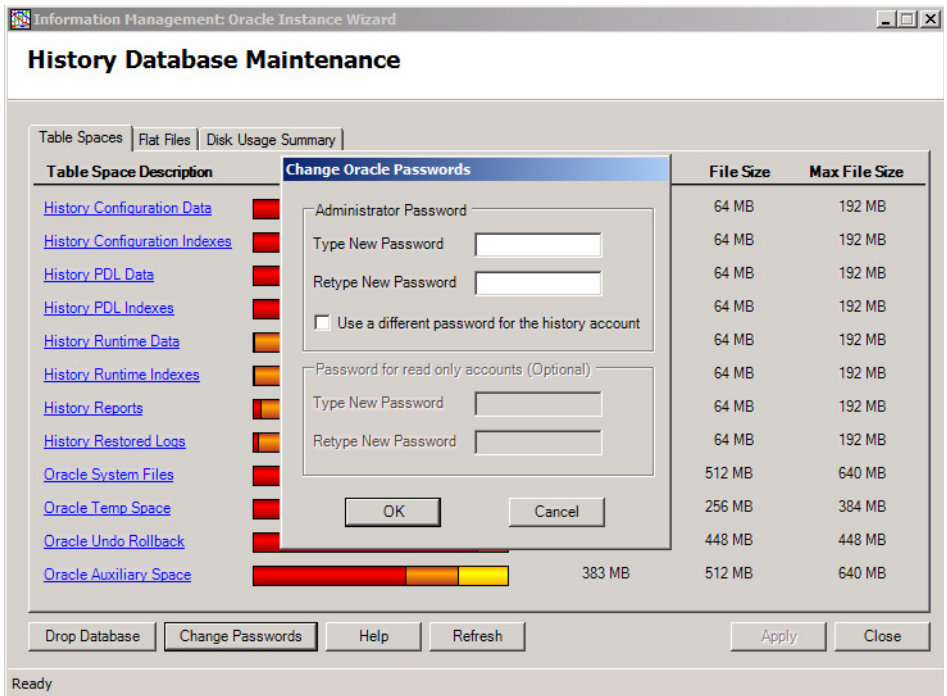


Figure 87. Change Oracle User Account Passwords

If the *history* account password is modified, any reports or other methods that may have hard coded the password would have to be updated to run with the new password.

In Oracle 11, passwords are case sensitive.

Oracle User Password Guidelines

The Oracle user password guidelines are as follows:

- Passwords must not exceed 30 characters.
- It is possible to use mixed case letters and special characters in the password to make it more secure.
- It is possible to include multibyte characters in the password.
- Use the database character set for the password characters, which includes underscore (_), dollar ($), and number sign (#) characters.
- Enclose the following passwords in quotation marks:
  - Passwords containing multibyte characters.
  - Passwords beginning with numbers or special characters and containing alphabetical characters. For example: **"123abc","#abc",** and **"123dc$".**
  - Passwords containing any character other than alphabetical characters, numbers, and special characters. For example: **"abc>","abc@"** and **" ".**
- No need to specify the following passwords in quotation marks:
  - Passwords beginning with an alphabet (a-z, A-Z) and containing numbers (0-9) or special characters ($,#,_). For example: **abc123, a23a** and **ab$#**.
  - Passwords containing only numbers.
  - Passwords containing only alphabetical characters.

Section 6 Configuring Log Sets

Log sets is used to start or stop data collection for a number of property or message logs simultaneously with one command. A log can be assigned to one or two log sets. The only restriction is that all logs in the log set must reside in the same node. For convenience, build log sets prior to building the logs so the log can be assigned to a log set when it is configured. Logs can still be added to log sets later.

Log set should be avoided if possible. They exist primarily for migrations from Enterprise Historian systems to 800xA. There are configurations where log sets are required when profile logs are used. However, for non-profile applications they should be avoided if possible.

Trend logs cannot be started and stopped with log sets. This functionality is only applicable for history logs.

The operating parameters for a log set are specified in a Log Set aspect. A dedicated aspect is required for each log set that needs to be configured. To create a Log Set aspect, create a Log Set object under a specific server node in the Node Administration structure refer to Adding a Log Set on page 144. The Log Set aspect is automatically created for the object. To configure the Log Set aspect, refer to Log Set Aspect on page 145.

Once the log sets are built, assign the logs to their respective log sets. This is done via each log's configuration aspect:

- For property logs this is done via the IM Definition tab of the Property Log Configuration aspect and is described in Assigning a Log to a Log Set on page 267.
- For message logs, refer to Message Log Aspect on page 154.

Adding a Log Set

This procedure describes how to add a Log Set object to a specific Information Management server node in the Node Administration Structure. To do this:

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node where the log set is to be added (for example, ROC90in Figure 88).
3. In the object tree for the selected node, navigate to **InformIT History**_*nodeName* **Service Provider > InformIT History Object**.

   Under the InformIT History Object find containers for each of the Inform IT History object types. The History objects (in this case, a log set) must be instantiated under the corresponding container.
4. From the **Log Sets** group and choose **New Object** from the context menu.

   This displays the New Object dialog with the **Inform IT Log Set** object type selected, Figure 89.

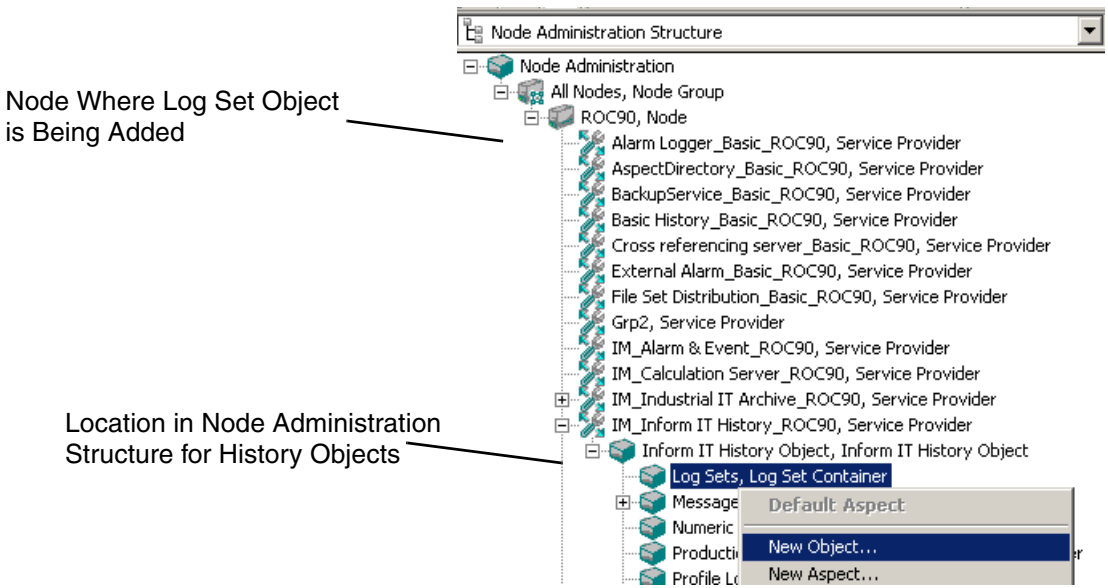


Figure 88. Adding a Log Set in the Node Administration Structure

5. Enter a name for the object in the Name field, for example: LogSet1.
6. Click **Create**. This adds the object under the Log Sets group, and creates a corresponding Log Set Aspect.

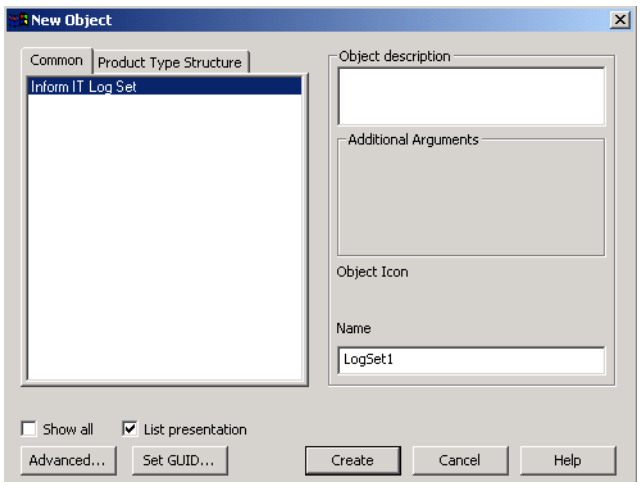


Figure 89. Selecting the Inform IT Log Set in the New Object Dialog

Log Set Aspect

Use the Log Set aspect, Figure 90, to:

- Specify Log Set Operating Parameters.
- Activate/Deactivate Logs in a Log Set.
- Delete a Log Set.

For Profile Historian applications, all Profile Logs associated with a specific machine must be grouped in the same log set, and that log set must have the same name as the machine.

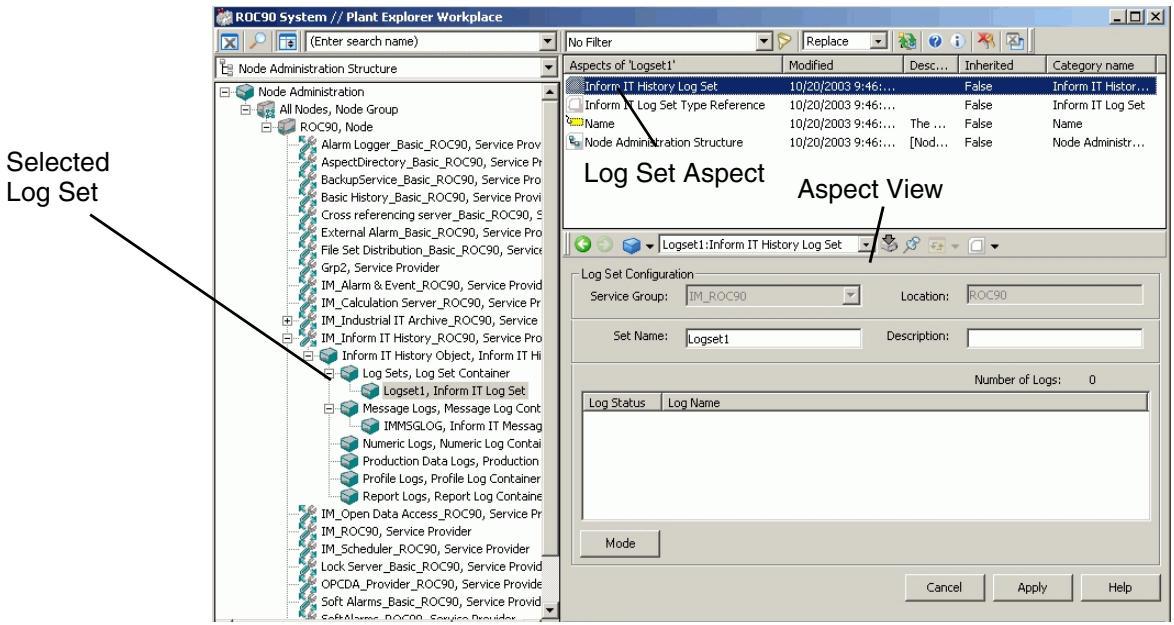


Figure 90. Accessing the Configuration View for the Log Set Aspect

Specify Log Set Operating Parameters

Select the Log Set aspect, and then configure the log set operating parameters as described in Table 18. Click **Apply** when finished.

Table 18. Log Set Attributes

Attribute	Description
Service Group	When adding the Log Set object in the Node Administration structure, the Service Group defaults to the Basic History Service Group for the selected node. This specification cannot be changed.
Location	This is a read-only field which indicates the node where the log set will run based on the selected Service Group.

Table 18. Log Set Attributes (Continued)

Set Name	The log set name can be up to 32 characters. If the length limitation is exceeded it will be truncated to size. This name will replace the object name given when this object was created. DO NOT use spaces in the log set name or the object name will be truncated at the space when the log set name replaces the object name. The log set name can not be changed in the Configuration View of the Log Set aspect. Once it is entered in the Set Name field and **Apply** is clicked, DO NOT change the name of the aspect. This will cause any new log configurations pointing to the history template that references the renamed log set to fail when created.
Description	The description is optional. The length limitation is 40 characters.
Log Status	This field indicates whether the logs in this log set are active or inactive. Refer to Activate/Deactivate Logs in a Log Set on page 147.
Logs in Set	This is a read-only field that lists the logs in this log set. No entries can be made. To assign logs to a log set, refer to Assigning a Log to a Log Set on page 267.
Number of Logs	This field indicates the total number of logs in the log set.

Activate/Deactivate Logs in a Log Set

Activating or deactivating a log set affects all of the logs belonging to that log set. To activate or deactivate a log set, go to Log Set Aspect, click **Mode** and choose **Activate** or **Deactivate** from the context menu.

Delete a Log Set

Before a log set can be deleted, it must not contain any logs. To delete a log set:

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node with the log set to be deleted.
3. In the object tree for the selected node, navigate to **InformIT History_***nodeName* **Service Provider > InformIT History Object>Log Sets**.
4. Right-click on the Log Set object that to be deleted and choose **Delete** from the context menu.

Section 7 Alarm/Event Message Logging

This section describes how to integrate the 800xA System Message Server with Information Management History Server, and how to configure message logs. All alarm and event messages for the 800xA system, including Process, Operator, and Audit Trail messages, are collected and stored by the 800xA System Message Server. This provides a short-term storage facility with the capacity to store up to 50,000 messages. The messages can be organized into filtered lists for viewing. Refer to *System 800xA Operation (3BSE036904*)*.

If Information Management History Server is installed, the messages stored by 800xA System Message Server may be forwarded to a Information Management message log for extended online storage. This message log can filter and store up to 12 million messages. In addition, an Information Management History Server allows messages from multiple servers to be consolidated onto a dedicated consolidation node, and the messages can be saved on an archive media for permanent offline storage.

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

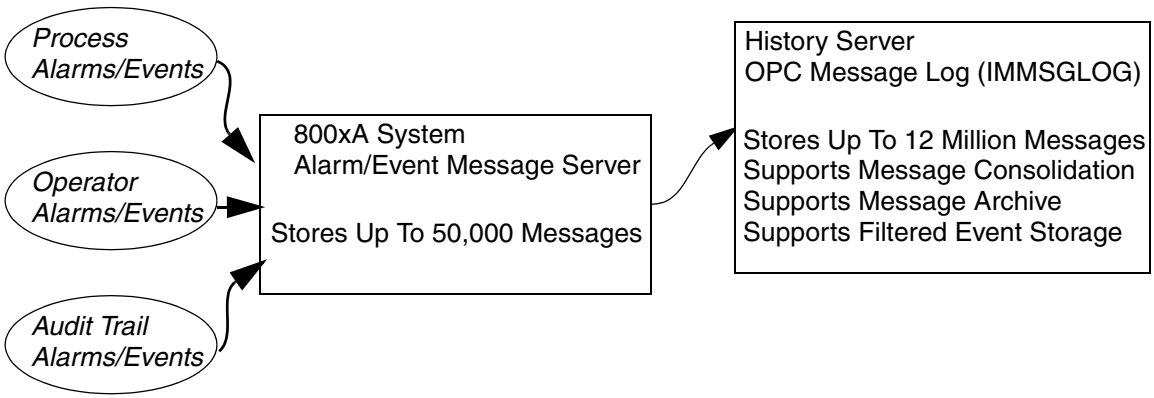


Figure 91. Alarm/Event Message Logging Overview

Begin with any one of the following topics to suite your depth of knowledge of message logs:

- If unfamiliar with message logs, refer to Message Log Overview on page 150 for a quick introduction.
- To configure message logs refer to Configuring Message Logs on page 152.

Message Log Overview

Message logs store events and system messages that are generated by control and operator applications. Three types of message logs are available, depending on where the events occur and what functionality is needed:

- Alarms and events which are buffered in the 800xA System Message Service may be forwarded to an **OPC_MESSAGE** log type. This type of storage is modeled after the data definitions in the OPC/Alarm and Event specification. This includes Audit Trail messages. The Audit Trail function tracks operator changes such as activating/deactivating historical logs, changes to production data, log configuration changes, and so on. The log stores the identification of the person that made the change, time that the change was made, previous and new value for the data being changed, as well as other information.

- The **PDLMSGLOG** log type is a special implementation of the OPC_MESSAGE log type for storing batch events related to Batch Management.

Configuring Oracle Access for Message Logs

Message log data is stored in an Oracle database. Client applications such as Display Services and DataDirect access Oracle-based History data, including message logs, via an ADO data provider named DBA. The data provider must be configured and must reside on the same node where the message logs reside. The ADO data provider configuration is performed as a post-installation procedure and can be verified via the ADSS Config tool in the Windows Control Panel.

Accessing Message Log Data

Messages can be read via interactive dialogs in DataDirect and Desktop Trends. DataDirect is also used to specify re-executable functions for implementing reports. Data can also be accessed directly using SQL queries. Refer to *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

Using an Alarm and Event List to Read Messages

Alarm and Event List aspects are used to view up to 50,000 messages stored in the 800xA system message service. This aspect can be configured to read messages from the Information Management message log rather than the 800xA system message service. The message log has the capacity to store up to 12 million messages for browsing farther back in time. Use the message log to maximize the scope of the Alarm and Event List aspect when it is used as the basis for a report. Refer to Accessing a Message Log with an Alarm and Event List on page 162.

The Alarm and Event List aspect may also be configured to retrieve archived messages on a specified archive volume, or archived messages which have been published or restored.

Message Log Consolidation

Alarm/event messages from multiple servers may be consolidated onto a dedicated consolidation node using the Application Scheduler and IM Consolidation action plug-in as described in Section 12, Consolidating Historical Data.

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

Offline Storage

Events and messages stored in a message log can be copied to an offline storage media. Refer to Section 11, Configuring the Archive Function.

Checking the Connection to 800xA System Message Services

If messages related to history are not being collected by the 800xA message services, check for the presence of the history OPC alarm server. Refer to Integrating System Message and History Servers on page 495.

The Event Log Collector collects all the events from 800xA system message queues. When a Message queue gets full it will overwrite the oldest events as newer events come in. If a burst of many events is expected for a particular Event Category, then consider filtering those events or increasing the storage sizes in the 800xA system message configuration. For example, a burst for a particular Event Category of 100 events per second for a duration of 5 minutes means that the storage size must be configured to handle 30,000 events. This can be filtered. The typical system message configuration default storage is sized for 10,000 events.

Configuring Message Logs

The operating parameters for a message log are specified via a Message Log object. A dedicated object is required for each message log. Typically, only one message log is required for each message log type being used (OPC, PDL, or DCS). These Message Log objects must be added under a specific node in the Node Administration structure.

To proceed, first create the Message Log object in the Node Administration structure, as described in Adding a Message Log on page 153. Then configure the Message Log, as described in Message Log Aspect on page 154.

Adding a Message Log

To add a Message Log object (reference Figure 92):

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node where the message log is to be added (for example ROC90 in Figure 92).
3. In the object tree for the selected node, navigate to **InformIT History**_*nodeName* **Service Provider > InformIT History Object**.

   Under the InformIT History Object find containers for each of the Inform IT History object types. The History objects (in this case, a message log) must be instantiated under the corresponding container.
4. Right-click on the **Message Logs** group and choose **New Object** from the context menu. This displays the New Object dialog with the **Inform IT Message Log** object type selected.
5. Enter IMMSGLOG for the object in the Name field then click **Create**. This adds the object under the Message Logs group. Using IMMSGLOG causes the log name to default to IMMSGLOG and the IM server IP address will be automatically appended (refer to Message Logging for 800xA System Alarm and Event Server on page 156).

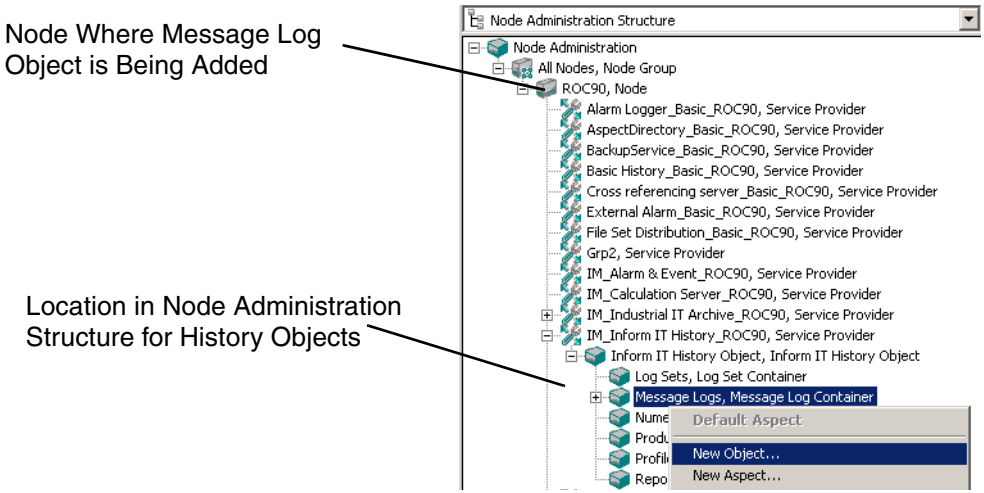


Figure 92. Adding a Message Log in the Node Administration Structure

Message Log Aspect

Use the Message Log aspect, Figure 93, to:

- Specify Message Log Operating Parameters.
- Delete a Message Log.

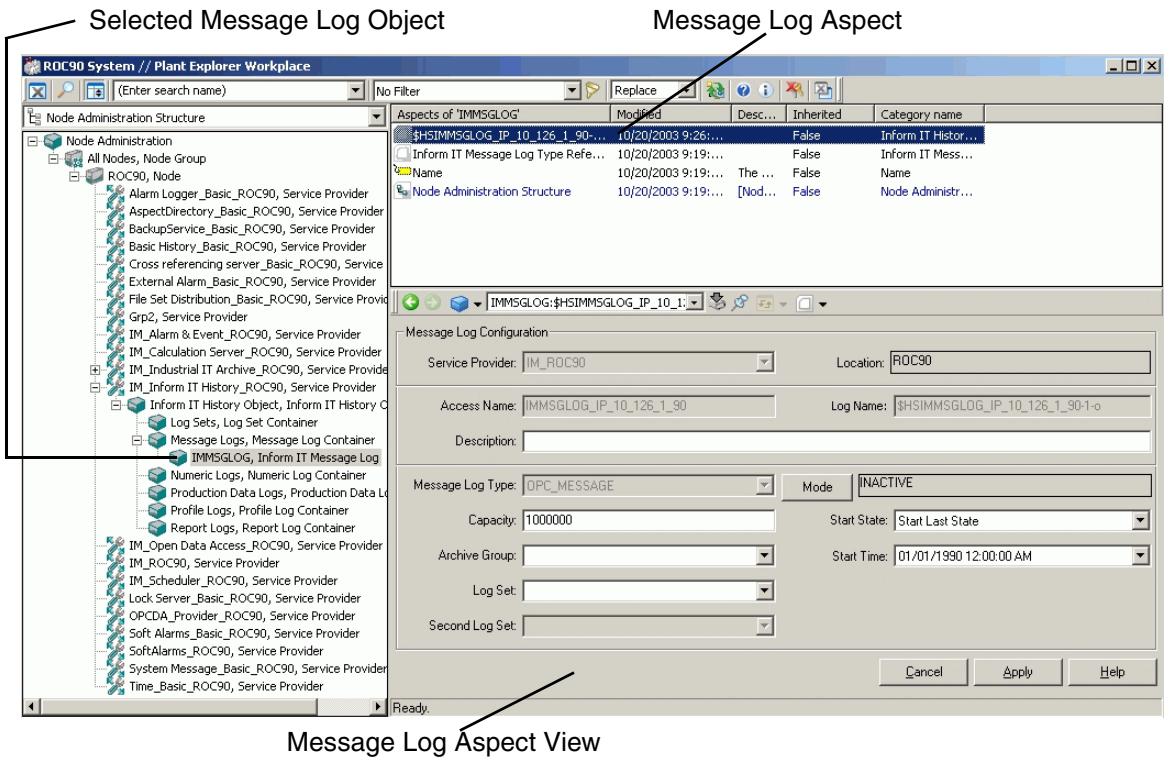


Figure 93. Accessing the Configuration View for the Message Log Aspect

Specify Message Log Operating Parameters

To configure a message log:

1. Select Message Log aspect.
2. Select the log type, and then follow the guidelines for the message logging application as listed below. Details regarding Message Log Attributes are provided in Table 19.
   - Message Logging for 800xA System Alarm and Event Server on page 156.
   - Message Logging for Batch Management on page 157.

3. When finished, click **Apply**.

- **Access Names** - Access names for message logs can not exceed 64 characters. The access name will replace the object name given when this object was created. DO NOT use spaces in the log set name or the object name will be truncated at the space when the log set name replaces the object name.
- **Activation** - Activation is optional for OPC message logs. Data will be stored in either state. If the log is active, none of the configuration parameters can be modified.
- **Capacity** - Specify a log capacity for the message log. The log capacity for a message log defines the number of messages it can hold. Details are provided in Table 19.

Message Logging for 800xA System Alarm and Event Server

The 800xA System Alarm/Event Server requires a message log of the type **OPC_MESSAGE**. The log name defaults to **IMMSGLOG_IP**_*ipaddress* where *ipaddress* is the IP address of the node where the log resides, Figure 94. This log will collect from the default system.

If an IMMSGLOG is being configured to consolidate messages from other OPC message logs, and the IMMSGLOG is to be dedicated to consolidation and will not be used to collect messages for the local node, use a prefix to avoid appending the local node IP address, for example: **CIMMSGLOGforEng444**. For further information on setting up message log consolidation, refer to Consolidating Message Logs and PDLs on page 401.

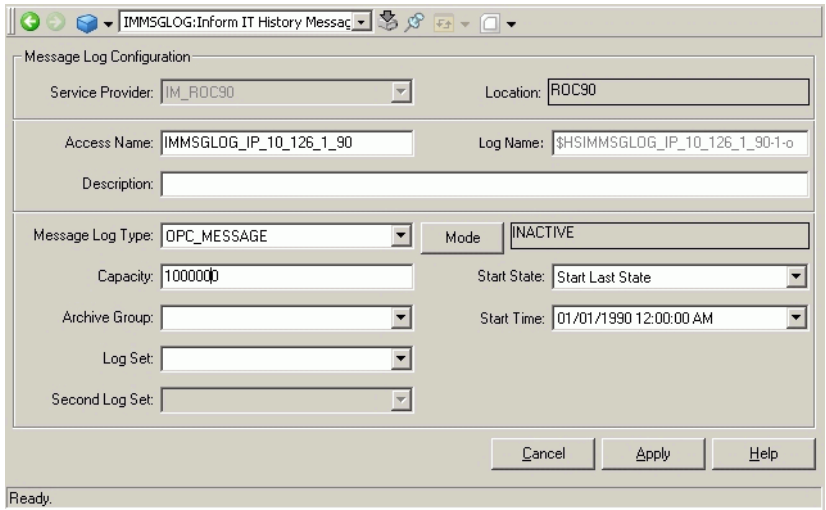


Figure 94. OPC_MESSAGE Log for 800xA System Alarm/Event Server

Message Logging for Batch Management

Batch Management requires a message log of the type **OPC_MESSAGE**. This log must be given the unique access name: **PDLMSGLOG**. When this access name is entered, the IP address for the local History server is automatically appended to the access name, Figure 95.

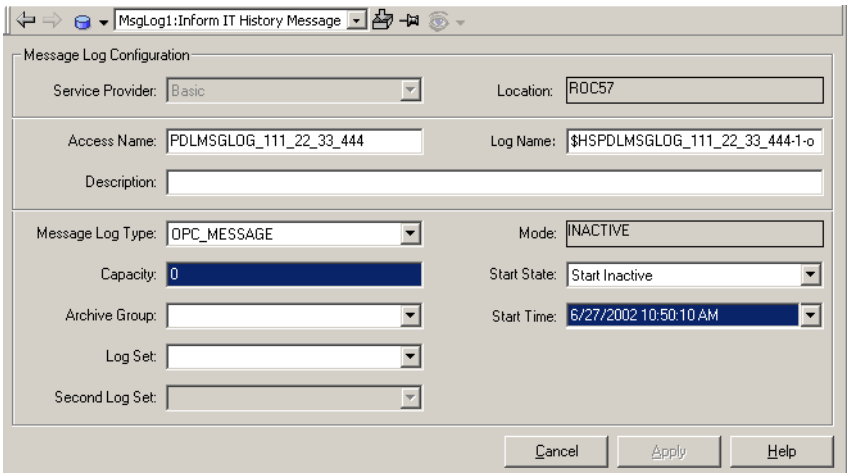


Figure 95. Access Name for Batch Management Message Log

When configuring a PDL_MESSAGE log to consolidate messages from other PDL_MESSAGE logs, configure ONE consolidation message log to collect from ALL other PDL_MESSAGE logs. This log must be named using the conventions previously described. DO NOT use a prefix when specifying the access name.

For further information on setting up message log consolidation, refer to Consolidating Message Logs and PDLs on page 401.

Message Log Attributes

Message Log Attributes are described in Table 19.

Table 19. Message Log Attributes

Attribute	Description
Service Provider	When adding the Message Log object in the Node Administration structure, the Service Provider defaults to the Basic History Service Group for the selected node. This specification cannot be changed.
Location	This is a read-only field which indicates the node where the message log will run based on the selected Service Group.

Table 19. Message Log Attributes (Continued)

Attribute	Description
Message Log Type	Select the message log type before configuring any other parameters. • For 800xA System Alarm/Event messages select **OPC_MESSAGE**. • Batch Management messages select **OPC_MESSAGE**. • For Advant OCS messages (for consolidating message logs from earlier Enterprise Historian systems) select **DCS_MESSAGE**.
Access Name	Assign the access name according to the selected message log type. Refer to: • Message Logging for 800xA System Alarm and Event Server on page 156. • Message Logging for Batch Management on page 157. The access name text string can be 64 characters maximum. If the length limitation is exceeded it will be truncated to size. **NOTE:** The access name will replace the object name given when this object was created. DO NOT use spaces in the log set name or the object name will be truncated at the space when the log set name replaces the object name.
Description	This field may be used to enter an optional text string. It does not have to be unique system wide. The length limitation is 40 characters. All characters are allowed. Description has a special application for DCS MESSAGE logs in systems with Master software.
Log Name	History assigns a default log name by adding a prefix `$HS` and a suffix `-1-o` to the access name. The last character in the log name indicates whether the log is original (from disk) or restored from tape: o = original, r = restored The default log name can be edited but can not be more than 64 characters. It must be unique system wide.
Capacity	The log capacity for the message log must be configured. This is the number of messages the log can hold. If a log is full, new entries replace the oldest entries. Specify the capacity as the number of messages needed to maintain online. The maximum capacity is 12 million (**12000000**) entries; however, the field will accept numerical values up to 9999999. Note that configuring the log capacity larger than required will needlessly consume more disk space.

Table 19. Message Log Attributes (Continued)

Attribute	Description
Archive Group	This is the archive group to which this log is assigned. This assignment can also be made via the Archive Group aspect as described in Configuring Archive Groups on page 341.
Start State	Log state upon starting or restarting the node. Start Inactive Start Active Start Last State - restart in state log had when node shut down (default)
Start Time	The earliest allowable system time that the log can become active. If the specified start time has already past, the log will begin to store data at the next storage interval. If the start time is in the future, the log will go to PENDING state until the start time arrives. The default is 01-01-1990 00:00:00.

Table 19. Message Log Attributes (Continued)

Attribute	Description
Log Set and Second Log Set	Log set to which this log belongs. The log can be reassigned to a new log set. **Note**: The user of log sets should be avoided.
Log State	This is the current state of the log. • Inactive - stores data, attributes can be modified. • Active - stores data, attributes cannot be modified. • Pending - log has been activated and is waiting for the Start Time in order to change to active. Individual logs can be activated or de-activated using the **Mode** button in the Message Log aspect and choosing **Activate** or **Deactivate**. Multiple logs can be activated and de-activated on a log set basis. When activating a log set, the Log State attributes of all logs in the set go to Active. The time when a log is scheduled to start collecting data is the log's *alignment time*. Alignment time is the intersection of the log's Start Time and its storage interval. For instance if the start time is 06-30-1999 10:00:00, and the storage interval is15s, the log will store data at 00, 15, 30, and 45 seconds of each minute starting at 10:00:00 on June 30, 1999. This is the alignment time. If the start time for a log has already passed, it will be aligned based on the storage interval. For example, a log with a storage interval of 1m will store data at the beginning of each minute. So its alignment time is the beginning of the next minute. The log stays in PENDING state until its alignment time. For the log whose storage interval is 1 minute, it will be PENDING until the start of the next minute.

Delete a Message Log

A message log must be inactive to be deleted. To delete a message log:

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node where the message log is to be deleted.

3. In the object tree for the selected node, navigate to **InformIT History**_*nodeName* **Service Provider > InformIT History Object>Message Logs**.
4. Right-click on the Message Log object that to be deleted and choose **Delete** from the context menu.

Accessing a Message Log with an Alarm and Event List

The configuration that supports this functionality is illustrated in Figure 96. To do this, the following three types of aspects on an object must be created and configured:

- Alarm and Event List.
- Alarm and Event List Configuration.
- A/E Linked Server Configuration.

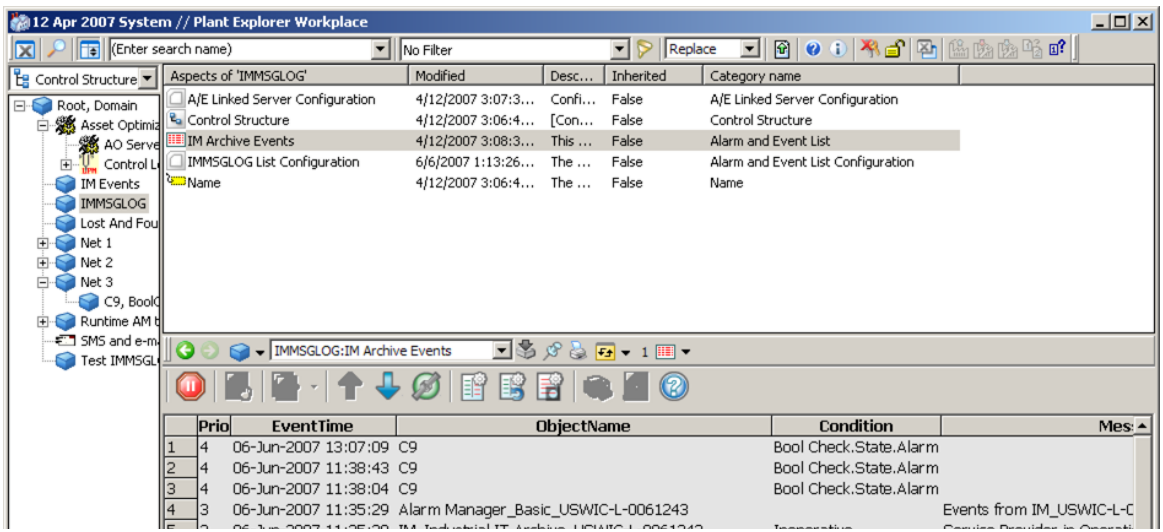


Figure 96. Configuration for Reading Message Log from Alarm and Event List

These aspects may be added to any object in any structure. However, it is generally recommended that a generic object be created and then all three aspects added to that object. This allows use of the same Alarm and Event List view in multiple locations in the plant (in more than one structure). To do this, create one object with these three aspects and use the Insert Object function to represent it in multiple locations. Otherwise, the three aspects would need to be added to multiple objects.

The A/E Linked Server Configuration aspect must be configured to specify the Information Management server where the message log resides. It must also be configured to specify one of four classes of alarm/event data to read:

- Messages currently stored in the message log.
- Published messages.
- Restored messages.
- Messages residing on a specified archive volume.

To read more than one class of message, configure a separate object with all three aspects for each class.

The Alarm and Event Configuration aspect must be configured to direct the Alarm Event List aspect to access messages from the Information Management message log rather than the 800xA system message service. This is done via the Source field on this aspect.

Finally, the Alarm and Event List aspect must be configured to use to the Alarm and Event List Configuration aspect described above.

To create this configuration:

1. Create a generic type object in the structure where the Alarm and Event list will reside, for example, the Functional structure as shown in Figure 97. Give the object a meaningful name. For example, use the Information Management server node name in combination with the type of message log to be handled by the Alarm and Event List aspect (current message log, restored, published, or specified archive volume).

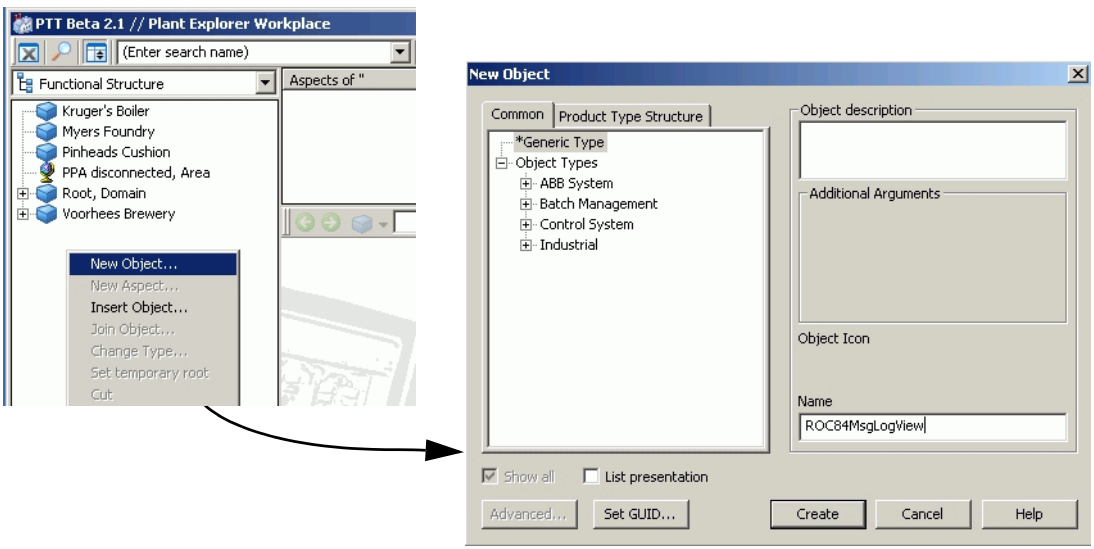


Figure 97. Creating a Generic Type Object

2. Add the three aforementioned aspects to the new Generic type object.

a. The **Alarm and Event List** aspect is located under the Alarm and Events category in the New Aspect dialog, Figure 98.

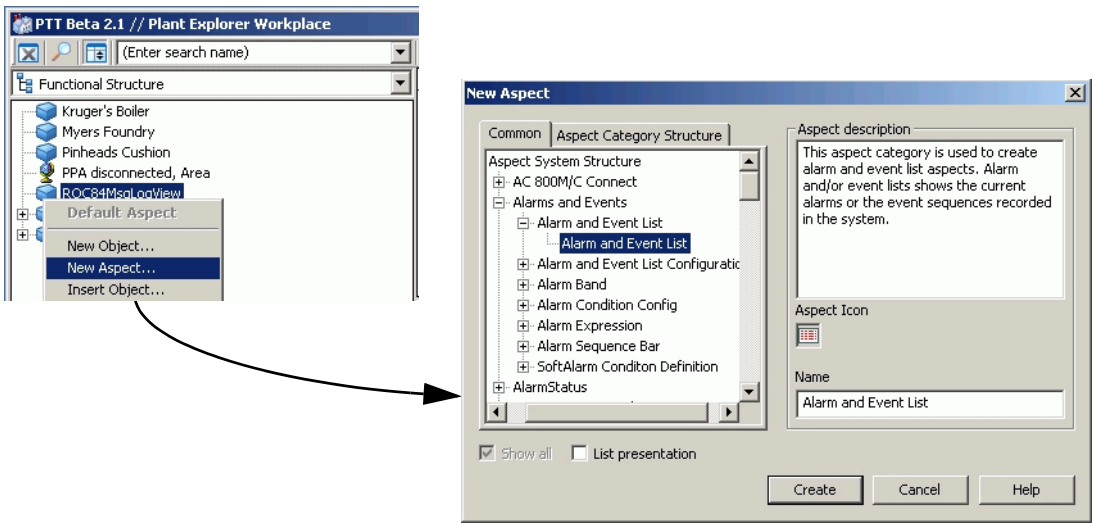


Figure 98. Adding the Alarm and Event List Aspect

b. The **Alarm and Event List Configuration** aspect is added differently.

– Open the Config View of the Alarm and Event List aspect.

– Use the Library Structure to select the Common System Event List Configuration aspect and click **Apply**. Refer to Figure 99.

– Click Copy Template and then Apply. An Alarm and Event List Configuration aspect is now added to the object. Refer to Figure 100.

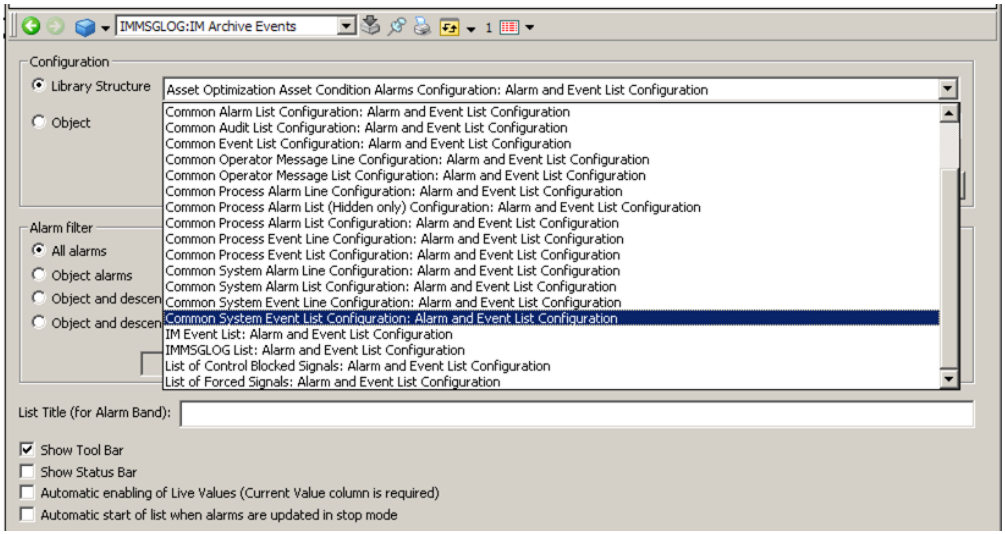


Figure 99. Common System Event List Configuration Aspect

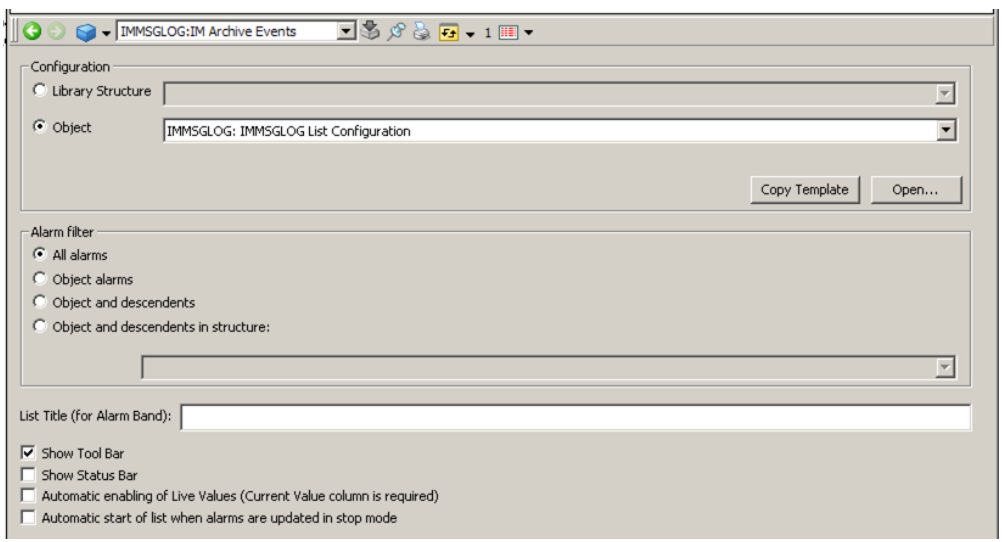


Figure 100. Creating Alarm and Event List Configuration Aspect

c. The A/E Linked Server Configuration aspect is located under the Industrial IT Archive Group, Figure 101.

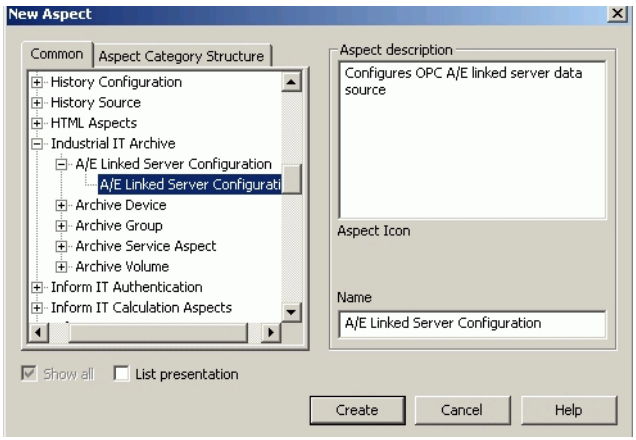


Figure 101. Creating the A/E Linked Server Configuration Aspect

3. Configure the A/E Linked Server Configuration aspect to specify the Information Management server where the message log resides, and to select the type of message to read. To do this:

   a. Select the A/E Linked Server Configuration aspect.

   b. Use the pull-down list to select the service group for Information Management server, Figure 102.

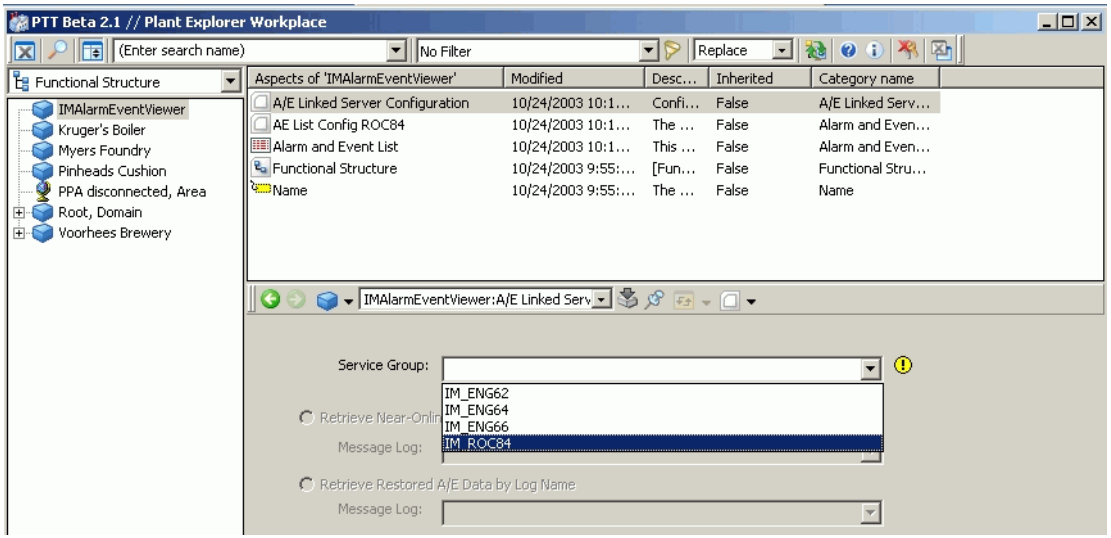


Figure 102. Selecting the Information Management Server

c. Select one of the four mutually exclusive radio buttons to select the type of messages to read, Figure 103. The categories are described in Table 20.

To read more than one type of message, configure a separate object with all three aspects for each type.

d. Click **Apply** when done.

Table 20. Retrieved Data Categories

Category	Description
Retrieve Near-Online A/E Data by Log Name	Retrieves messages currently stored in the message log.
Retrieve Restored A/E Data	Retrieves messages that have been restored from an archive media.
Retrieve Published A/E Data by Log name	Retrieves messages from all published volumes.
Retrieve Archived A/E Data by Location	Retrieves messages from a selected archive volume.

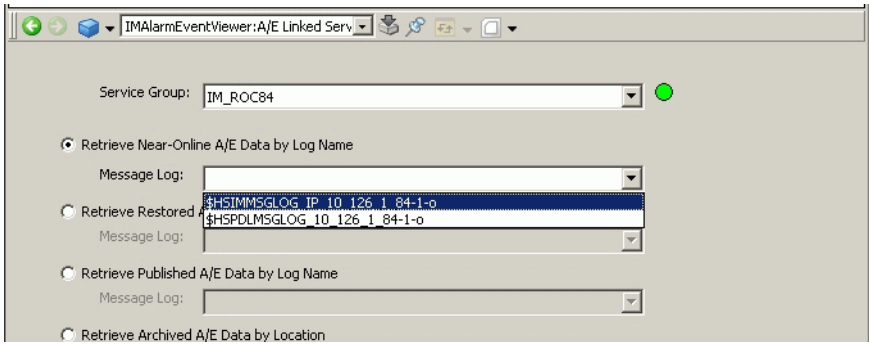


Figure 103. Selecting the Class of Message to Retrieve

4. Configure the Source on the Filter tab of the Alarm and Event List Configuration aspect to direct the Alarm and Event List aspect to read from the Information Management message log. To do this (reference Figure 104):

   a. Select the Alarm and Event List Configuration aspect.

   b. Select the **Filter** tab.

   c. Select **IM Archive Events** from the Source pull-down list.

   d. Click **Apply**.

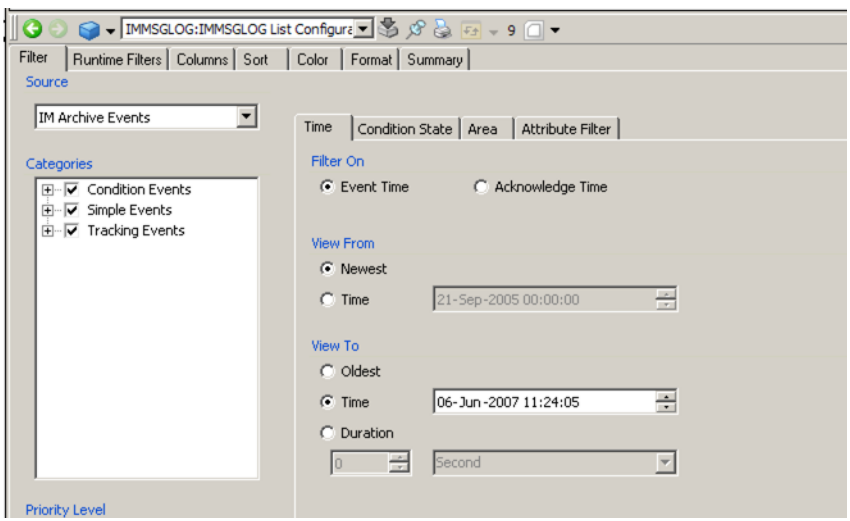


Figure 104. Configuring the Event Source

e. Enable the Condition Events, Simple Events, and Tracking Events options in the Categories area.

f. Click **Apply**.

Creating an Inform IT Event Filter

The events from the PPA Event Storage Service can be filtered from being collected, by the history event log collector through Inform IT Event Filter aspect configuration.

To create the Inform IT Event Filter aspect, do the following steps:

1. Select the Inform IT History object in the Node Administration structure.
2. Add an Alarm and Event List Configuration aspect.
3. Name the aspect as **Inform IT Event Filter**.

The name is case-sensitive.

4. Configure the aspect. Filter Configuration can be made in the following three ways:
   - Filtering based on Event Categories on page 171.
   - Filtering based on Event Attributes on page 171.
   - Filtering based on Combination of Categories and Attributes on page 173.

Filtering based on Event Categories

- In the **Filter** Tab, under **Categories** section, exclude one or more Category by selecting all and then clear the Category to be filtered as shown in Figure 105. In this case, **SoftAlarm** is being excluded from the events.

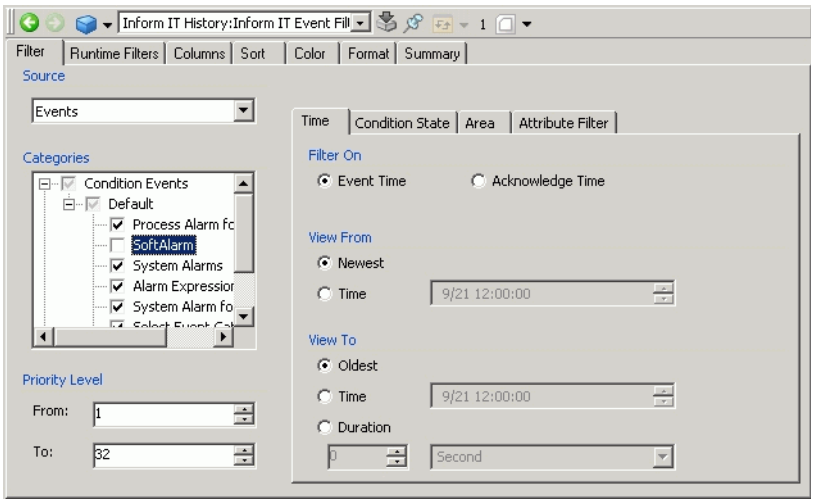


Figure 105. Inform IT Event Filter Aspect

- If no Category is selected or if all Categories are selected, then no filter is applied.
- Tracking Category and IM History Category are never filtered and any setting is ignored.

Filtering based on Event Attributes

Event collection is based on the specified Event Attributes. Unlike the **Filtering based on Event Categories,** Tracking Category and IM History Category can also be filtered if configured to.

In the **Filter** Tab, under the **Categories** section, all the categories must be selected.

The Filtering text used is not case-sensitive.

Refer to the following examples on using Attribute filters:

Example 1: Stores all the messages except those with sub string *Test message CCC,* as shown in Figure 106.

Example 2: Collects only those messages that satisfy the specified criteria, as shown in Figure 107. All other messages are excluded from collection.

Example 3: Filters all the messages that consist either of the following words: *Test*, *Message* or *CCC*.

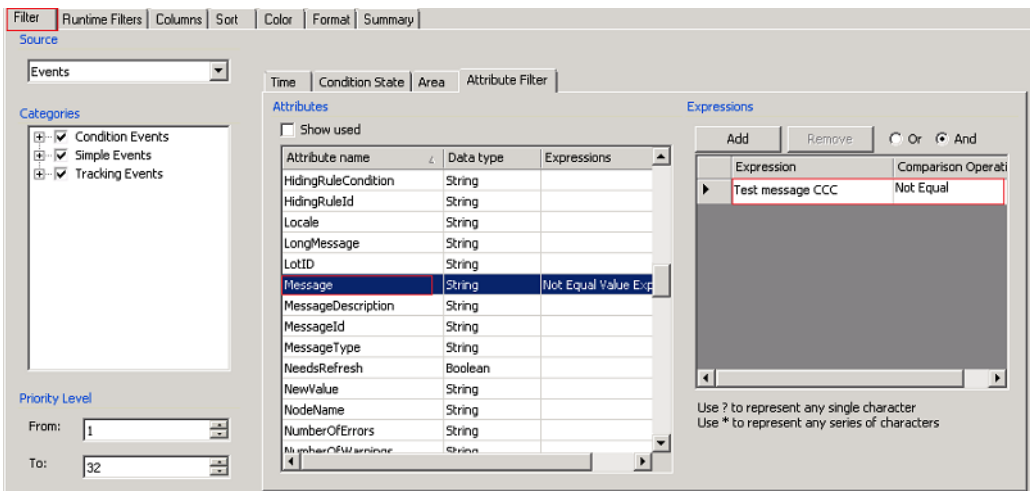


Figure 106. Example 1

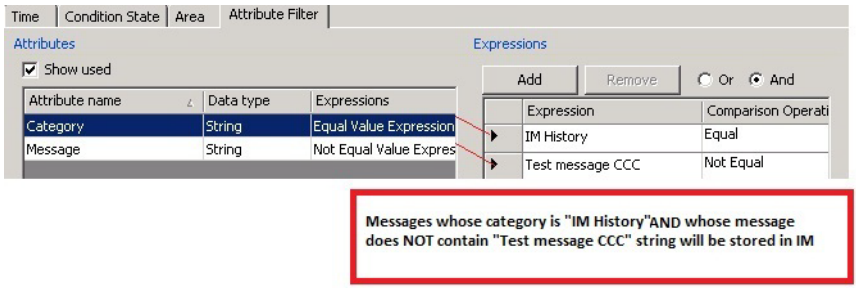


Figure 107. Example 2

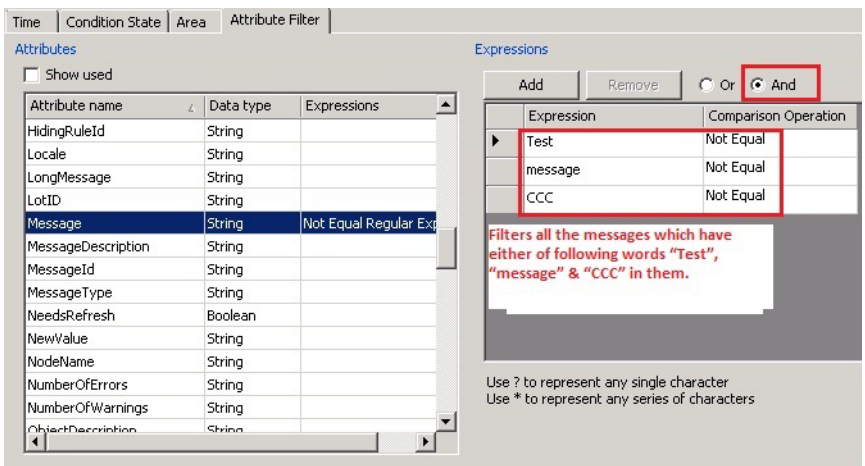


Figure 108. Example 3

Filtering based on Combination of Categories and Attributes

The messages satisfying conditions under the Categories section and the Attribute filter tab are stored in IM Message log. An example shown in Figure 109 indicates

that only the messages satisfying all the three conditions are stored in IM message log.

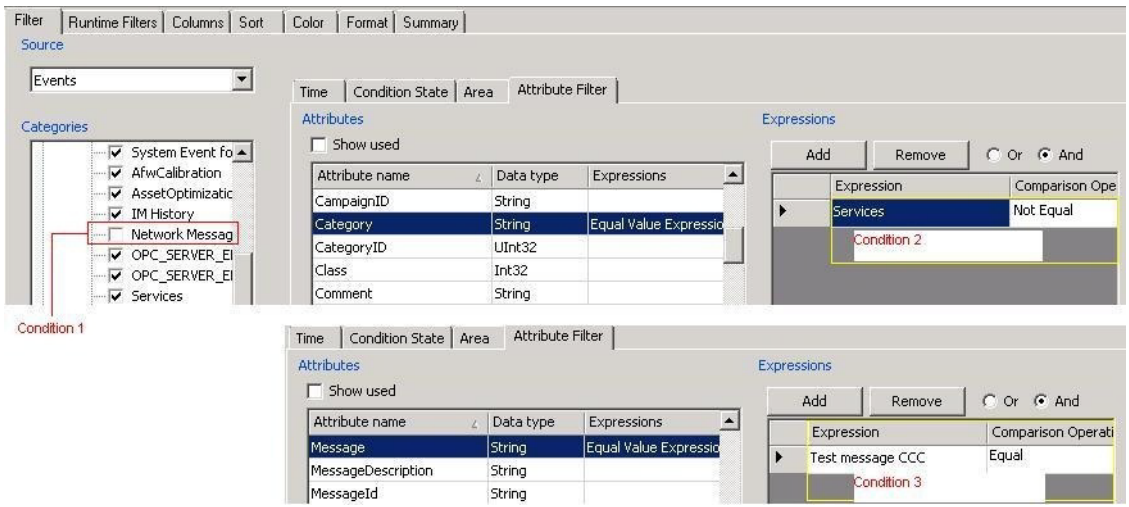


Figure 109. An example of combination of the two options

Categories to be filtered should not be used in mutually exclusive manner. In this case no messages are collected by IM. An example is shown in Figure 110.

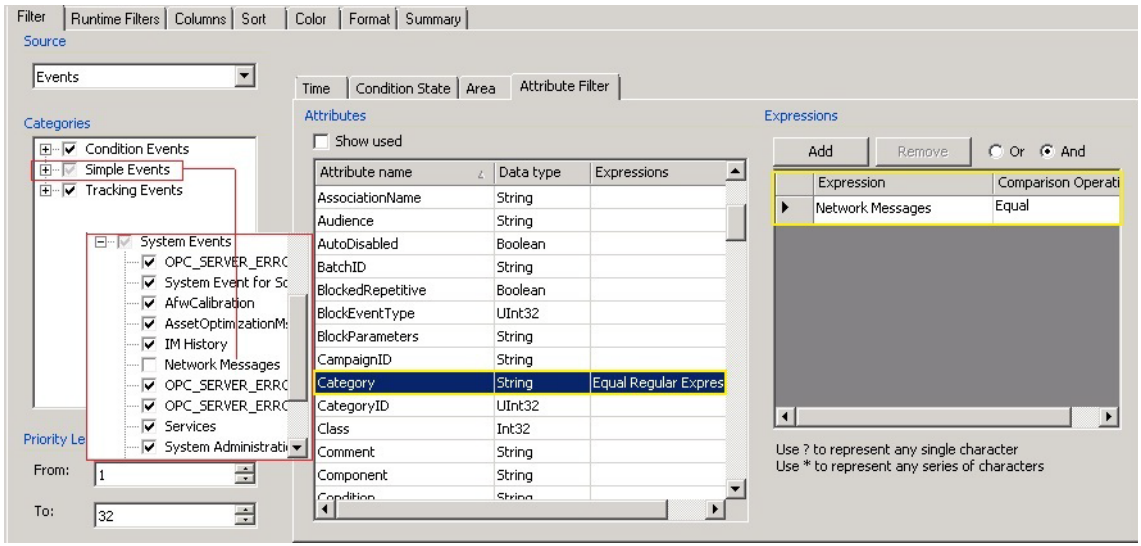


Figure 110. Combined Attributes being mutually exclusive

Filtering produces an event message. While setting a filter permanently saves storage space, the filter can also be used to manage bursts of activity or other special conditions.

Section 8 Historizing Reports

This section describes how to configure report logs to store finished reports scheduled and executed via the Application Scheduler and Report action plug-in. Completed reports may also be stored by embedding them in a completed report object in the Scheduling structure. This must be specified within the Report Action.

Reports stored either as objects, or in a report log can be archived on either a cyclic or manual basis. This is described in Section 11, Configuring the Archive Function. Access to archived reports logs is via the View Reports aspect as described in the section on viewing archive data in *System 800xA Information Management Data Access and Reports (3BUF001094*)*. When reviewing a restored report log, it will be presented in the tool appropriate for the type of file the report was saved as (i.e. Internet Explorer will be launched if the report was saved in .html format).

Generally, one report log is dedicated to one report. For instance, a WKLY_SUM report will have a log especially for it. Each week's version of the WKLY_SUM report is put into the report log. Different report logs are then used for other reports. More than one type of report can be sent to one report log if desired. Each report log can be specified as to how many report occurrences to save.

Report Log Configuration Guidelines

The operating parameters for a report log are specified in a Report Log aspect. A dedicated aspect is required for each report log to be configured.

To create a Report Log aspect, create a Report Log object under a specific node in the Node Administration structure. The Report Log aspect is automatically created for the object. To create the Report Log object in the Node Administration structure, refer to Adding a Report Log on page 178.

To configure the Report Log aspect, refer to Report Log Aspect on page 179.

Adding a Report Log

This section describes how to add a Report Log object to the applicable node in the Node Administration Structure. To do this (reference Figure 111):

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node where the report log is to be added (for example, ROC90 in Figure 111).

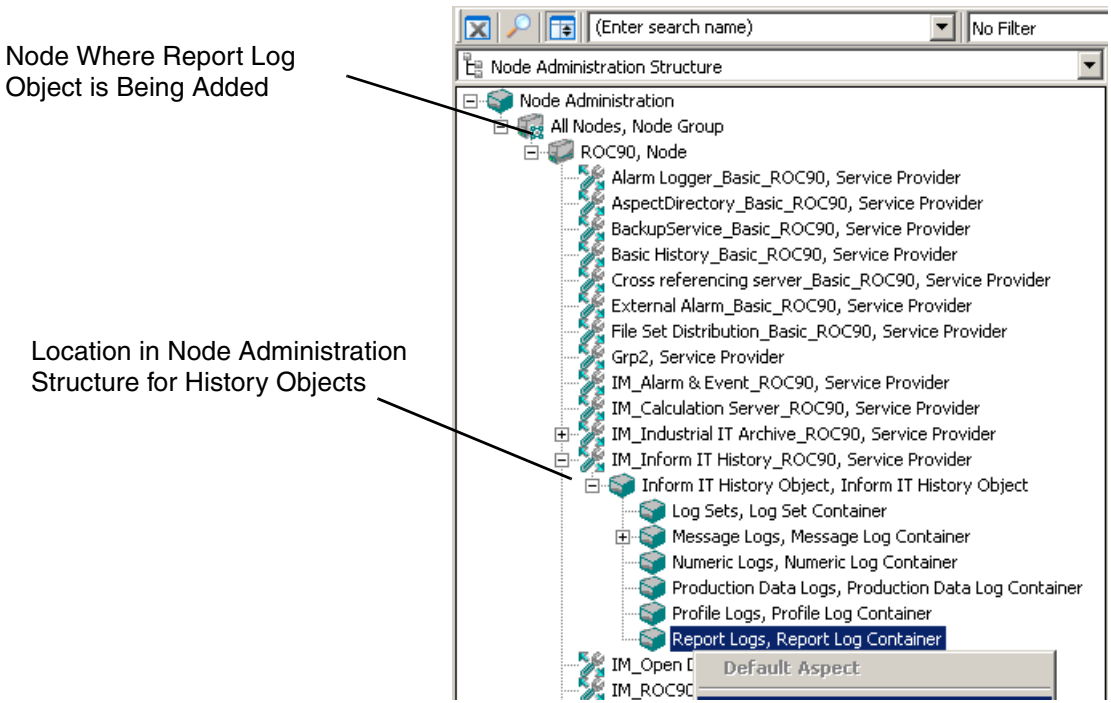


Figure 111. Adding a Report Log in the Node Administration Structure

3. In the object tree for the selected node, navigate to **InformIT History**_*nodeName* **Service Provider > InformIT History Object**.

   Under the InformIT History Object find containers for each of the Inform IT History object types. The History objects (in this case, a report log) must be instantiated under the corresponding container.

4. Right-click on the **Report Logs** group and choose **New Object** from the context menu. This displays the New Object dialog with the **Inform IT Report Log** object type selected.

5. Enter a name for the object in the Name field, for example: ReportLog1, then click **Create**. This adds the object under the Report Logs group, and creates a corresponding Report Log Aspect.

Report Log Aspect

Use the Inform IT History Report Log aspect, Figure 112, to:

- Configure Report Log Operating Parameters.
- Delete a Report Log.
- View a Report Log.

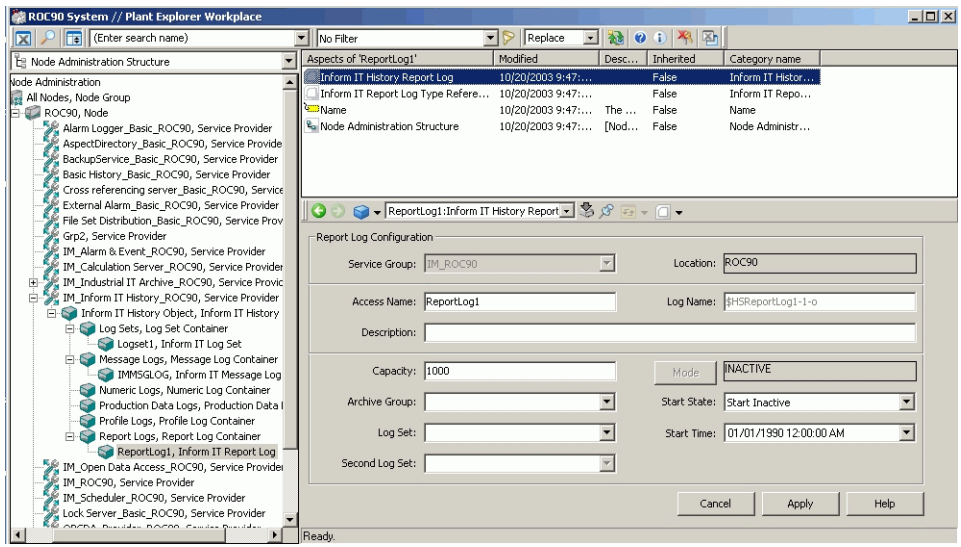


Figure 112. Report Log Aspect

Configure Report Log Operating Parameters

When building a report log, keep the following in mind:

- Report logs do not have to be activated in order to store reports. Therefore, log sets, log state, start state, and start time are not applicable for report logs.
- Be sure to specify the log capacity for a report log. Refer to Table 21.
- The report log's access name is the name used in the Report Scheduler to identify the log when reports are sent to it.

Build the report log via the Report Log Aspect. Make the appropriate entries as described in Table 21. Then click **Apply**.

Table 21. Report Log Attributes

Attribute	Description
Service Group	When adding the Report Log object in the Node Administration structure, the Service Group defaults to the Basic History Service Group for the selected node. This specification cannot be changed.
Location	This is a read-only field which indicates the node where the report log will run based on the selected Service Group.
Access Name	This is the name assigned the log for retrieval. Generally, use the name of the report as specified in the report building package. Use either the access name or log name to retrieve the data in a log. The access name text string can be 64 characters maximum. If the length limitation is exceeded it will be truncated to size. **NOTE:** This name will replace the object name given when this object was created. DO NOT use spaces in the access name or the object name will be truncated at the space when the log set name replaces the object name.
Description	An optional text string for documentation purposes only. It does not have to be unique system wide. The length limitation is 40 characters maximum. All characters allowed.

Table 21. Report Log Attributes

Log Name	The log name must be unique system wide. It is derived from the access name. History assigns a default log name by adding a prefix `$HS` and a suffix `-n-o` to the access name. *n* is an integer that uniquely identifies different logs in a composite log. This value will always be 1 since no other logs can exist in a composite log containing a report log. The last character in the log name indicates whether the log is original (from disk) or restored from tape: o = original, r = restored The default log name can be edited but can not be more than 64 characters. It must be unique system wide.
Capacity	This is the number of reports the log can hold. Be sure to specify a log capacity for report logs. The field will accept numeric values up to 9999. If a log is full, new entries replace the oldest entries.
Archive Group	This is the archive group to which this log is assigned. This assignment can also be made via the Archive Group aspect as described in Configuring Archive Groups on page 341.

Delete a Report Log

To delete a report log:

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node where the report log is to be deleted.
3. In the object tree for the selected node, navigate to **InformIT History_Basic Service Provider > InformIT History Object>Report Logs**.
4. Right-click on the Report Log object that to be deleted and choose **Delete** from the context menu.

View a Report Log

View report logs via the **Inform IT History View Report Logs** aspect for the Report Logs container in the Node Administration structure. For details, refer to the section on viewing archive data in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

Section 9 Historical Process Data Collection

Process data collection refers to the collection and storage of numeric values from aspect object properties. This includes properties for live process data, SoftPoint data, and lab data (programmatically generated, or manually entered). This functionality is supported by *property log* objects.

Process data collection may be implemented on two levels in the 800xA system. The standard system offering supports storage of operator trend data via *trend logs*. When the Information Management History Server function is installed, extended and permanent offline storage (archive) is supported via *history logs*. This section describes how to integrate and configure operator trend and history logs.

Begin with any one of the following topics depending on your current depth of knowledge on property logs:

- To learn the concept of property logs, refer to Property Log Overview on page 184.
- If History Source aspects have not yet been configured in the system, refer to Configuring Node Assignments for Property Logs on page 189. This must be done before the logs are activated and historical data collection begins.
- To learn about important considerations and configuration tips before actually beginning to configure property logs, refer to History Configuration Guidelines on page 194.
- For a quick demonstration on how to configure a property log, refer to Building a Simple Property Log on page 196.
- For details on specific property log applications, refer to:
  - Lab Data Logs for Asynchronous User Input on page 211.
  - Event-driven Data Collection on page 212.
  - History Logs with Calculations on page 219.

Property Log Overview

Property logs typically collect synchronous (periodic) data from properties that represent real-time process measurements. Property logs may also collect asynchronous data that is input manually (lab data), or synchronous data generated by a user program.

To begin implementing process data collection, one or more History Log Templates is created in the Library Structure. Each template serves as a model for a specific data collection scheme such as the one shown in Figure 113. The template defines a log hierarchy where component logs provide different views of the historical data.

Actual (operational) property logs are instantiated within a Log Configuration aspect which is based on a specified History Log template. The Log Configuration aspect is added to the object whose property values will be collected, typically in the Control structure. One property log is required for each property whose values will be collected. The Log Configuration aspect supports as many properties and as many property logs as required for the object. Therefore, it is recommended that just one Log Configuration aspect for an object be created. When more than one Log Configuration aspect for an object is created, it is recommended that all property logs for a given property should be included in the same Log Configuration.

The first component log inserted within a property log hierarchy is typically a trend log[1] which is inserted as a *direct* log type. This log collects *directly* from an OPC data source and resides on the connectivity server local to the data source. It supports viewing and storage of operator trend data. This log should be configured to store data for a time period that is slightly greater than the time the History Server may go off line. This will allow the history (extended storage) log to back fill data that was collected by the trend log while the History Server was offline. The trend log also supports redundancy when parallel connectivity servers are configured.

The Information Management History Server function can connect history logs in *hierarchical* fashion to the direct trend log. These history logs support long term storage, historical data consolidation, and offline storage. The first hierarchical history log that collects directly from the direct trend log is referred to as the *primary* history log. When additional history logs are connected to the primary history log, those logs are referred to as *secondary* history logs.

1. One exception to this scenario is when a property log is created to collect (consolidate) historical data from another Information Management server in another system.

For example, in Figure 113, the trend log stores high resolution data for a short time span, while the history log stores the same high resolution data for a much longer time span.

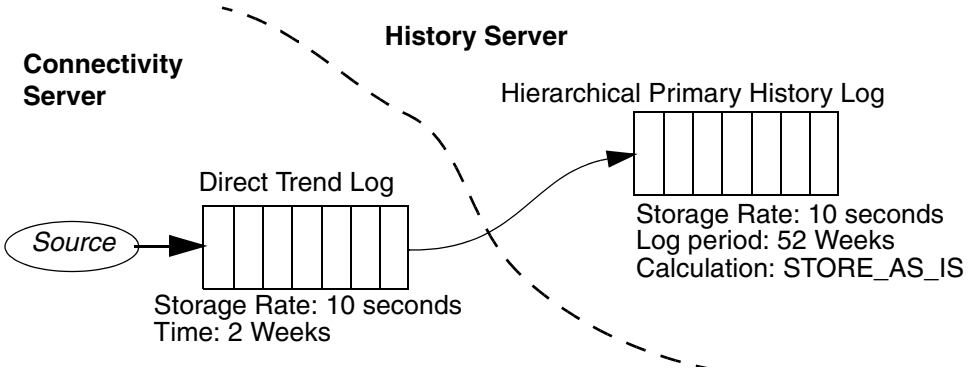


Figure 113. Example, Property Log Hierarchy

Dual Logs

As an option, configure a *dual log* where the same trend log feeds two history logs on two different history server nodes, Figure 114. This may be required when the application cannot wait for history data to be back-filled in the event that a history server goes off line. For example, shift reports may be required to execute at the end of each 8-hour shift and cannot wait days or weeks for the data to be back-filled. Configuration guidelines are provided in Dual Logs on page 211.

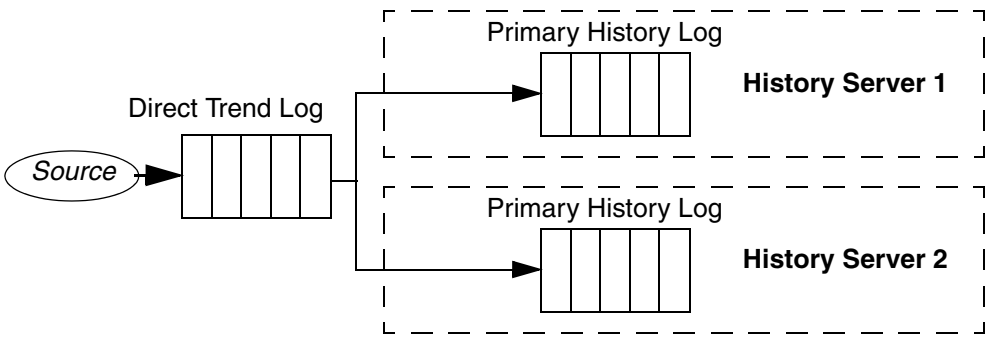


Figure 114. Example, Dual Log Configuration

Data Presentation in trend format is supported by 800xA System trend displays, Desktop Trends, and trend displays built with Display Services. These displays are simple graphs of process variables versus time, as shown in Figure 115. All analog and digital attributes recorded in History can be trended. Trend sources can be changed and multiple trends can be displayed on one chart to compare other tags and attributes. Stored process data may also be viewed in tabular format in Microsoft Excel using DataDirect add-ins (refer to Installing Add-ins in Microsoft Excel on page 295).

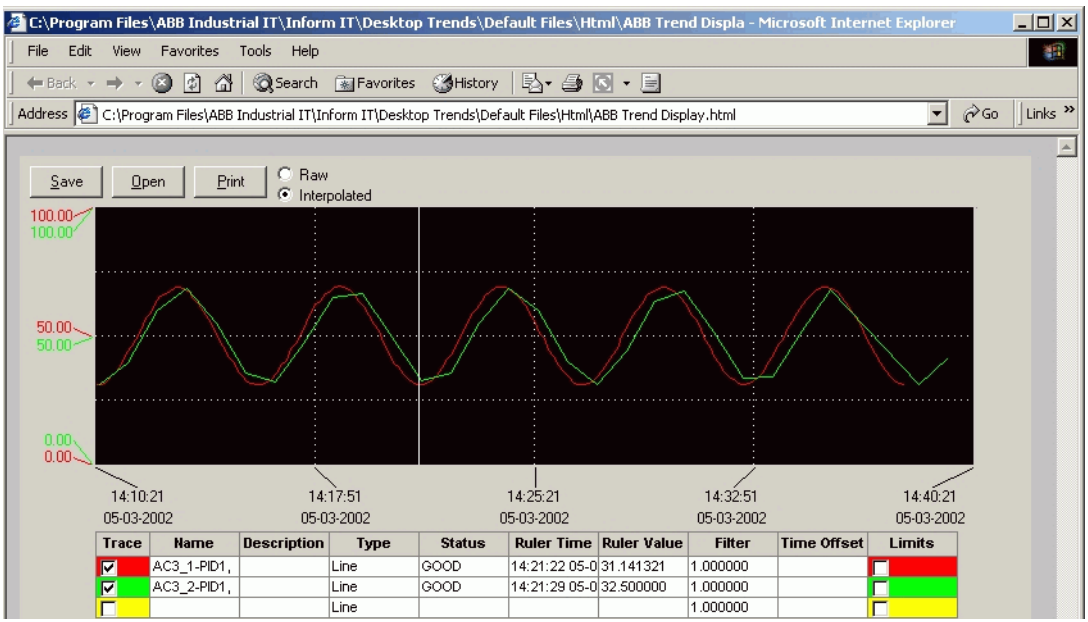


Figure 115. Historical Data on a Trend Display

The following sections provide a brief overview of the historical collection and storage functionality supported by history logs:

- Blocking and Alignment on page 187.
- Calculations on page 187.
- Data Compaction on page 187.
- Event Driven Data Collection on page 187.
- Offline Storage on page 188.
- Considerations for Oracle or File-based Storage on page 188.

- Seamless Retrieval on page 188.
- Consolidation on page 188.

Blocking and Alignment

A *Sample Blocking Rate* can be configured for the *primary* history log. This controls the frequency at which data is collected from the source trend log. The blocking rate can be used to optimize performance by phasing the rate at which numeric data is collected from the trend log, and phasing the writing of historical data to disk. For further information refer to Sample Blocking Rate on page 235.

Calculations

Although both trend and history logs can have calculations performed on collected data prior to storage, it is generally recommended that raw data be collected and stored, and then the required calculations can be performed using the data retrieval tool, for example, DataDirect. To do this for history logs, the STORE_AS_IS calculation algorithm is generally used.

This calculation algorithm causes data to be stored only when it is received from the data source. For OPC type logs, this calculation lets the OPC server's exception-based reporting be used as a deadband filter. This effectively increases the log period so the same number of samples stored (as determined by the log capacity) cover a longer period of time. Refer to STORE AS IS Calculation on page 238 for further details.

Data Compaction

The recommended method for data compaction is to use the STORE_AS_IS calculation algorithm. (STORE AS IS Calculation on page 238). The attributes on the Deadband tab may be used in certain circumstances, for example to support upgrades from earlier installations that used deadband, or when configuring calculation algorithms such as minimum, maximum, and average for history logs.

Event Driven Data Collection

Data collection for property logs may be event-driven. The event which triggers data collection is specified as a job via Application Scheduler. For further information refer to Event-driven Data Collection on page 212.

Offline Storage

All process data stored in a property log can be copied to an offline storage media. This can be done on a cyclic or manual basis. Refer to Section 11, Configuring the Archive Function.

Considerations for Oracle or File-based Storage

Log entries can be stored in Oracle tables or in files maintained by History. File storage is faster and uses less disk space than Oracle storage. File-based storage is only applicable for synchronous property logs. The default storage type is file-based TYPE5 which supports variable size based on the type of data being collected. ORACLE is mandatory for Asynchronous logs. This type supports collection for floating point data only. Other storage type are available. For further information refer to File Storage vs. Oracle Tables on page 255.

Seamless Retrieval

Seamless retrieval makes it easier to access historical data. For trends, when scrolling back in time beyond the capacity of the specified log, the seamless retrieval function will go to the next (secondary) log in the log hierarchy. Applications do not need to know the name of the log in order to retrieve data. Attributes such as data source, log period, calculation algorithm, and retrieval type can be specified and History will search for the log which most closely fits the profile.

Consolidation

Data from property logs on history server nodes in different systems (as configured via the 800xA system configuration wizard) may be consolidated onto one central consolidation node as illustrated in Figure 116.

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

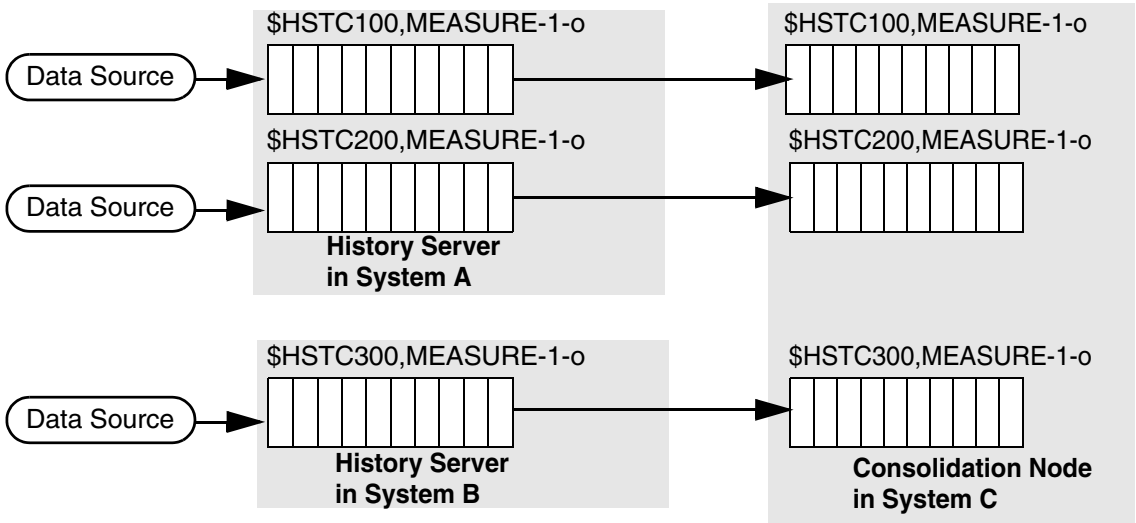


Figure 116. Example - Consolidating Property Log Data

Configuring History Source Aspects

If History Source aspects have not been configured in the system, refer to Configuring Node Assignments for Property Logs on page 189. This must be done before the logs are activated and historical data collection begins. If this is already done, then refer next to History Configuration Guidelines on page 194.

Configuring Node Assignments for Property Logs

Trend logs and history logs use different methods to establish their respective node assignments (on which node each log will reside).

For history logs, the node assignment is established by selecting the node's History Service Group when the log is added to a property log template. This is described as part of the procedure for configuring a log template in Creating a Log Template on page 197.

For trend logs the node assignment is established when the Log Configuration aspect is instantiated on an object, for example a Control Application object in the Control structure. When this occurs, the Log Configuration aspect looks upward in

the object hierarchy to find and associate itself with the first History Source aspect it encounters. The History Source aspect points to a History Service group on a specified Connectivity Server. To ensure that logs are assigned to the correct nodes, add one or more History Source aspects in the applicable structure where the log configuration aspects will be instantiated. Care must be taken to situate the History Source aspects such that each log will find the History Source that points to its respective Connectivity Server. Two examples are illustrated in Figure 117 and Figure 118.

- Property logs will not collect data until they are associated with a History Source aspect.
- History Source aspects must be placed in the same structure where log configuration aspects are to be instantiated.
- Add History Source aspects as early as possible in the engineering process. The change to update the History Server for a Service Group can take a long time for large configurations. All logged data will be lost during change of Service Group for affected log configurations.

Using One History Source

Figure 117 shows an example where all logs are to be associated with the same Connectivity Server. In this case, only one History Source aspect is required, and it should be placed near or at the top of the applicable structure, for example, at the root.

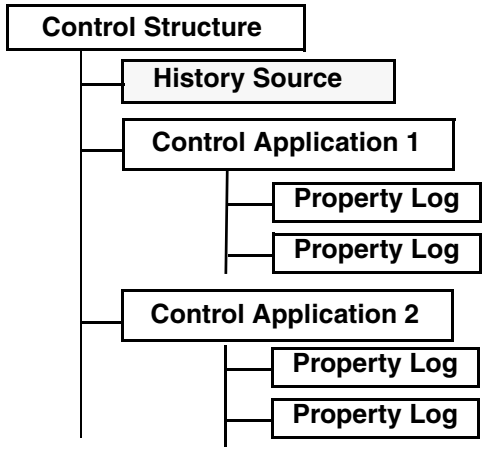


Figure 117. Using One History Source

Using Multiple History Sources

In Figure 118, the property logs for Control Applications 1 and 2 must reside on different Connectivity Servers. This is accomplished by adding a History Source aspect to each of the Control Application objects.

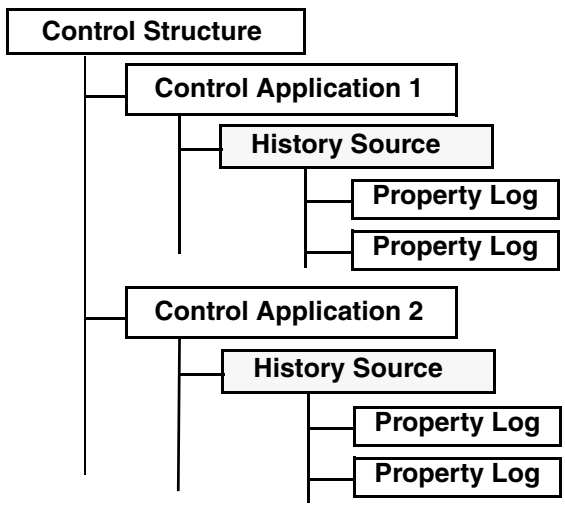


Figure 118. History Source Associations

Pointing History Source to History Service Group

The History Source aspect must be configured to point to the History Service Group for the same node where the OPC Data Source aspect points. This is illustrated in Figure 119.

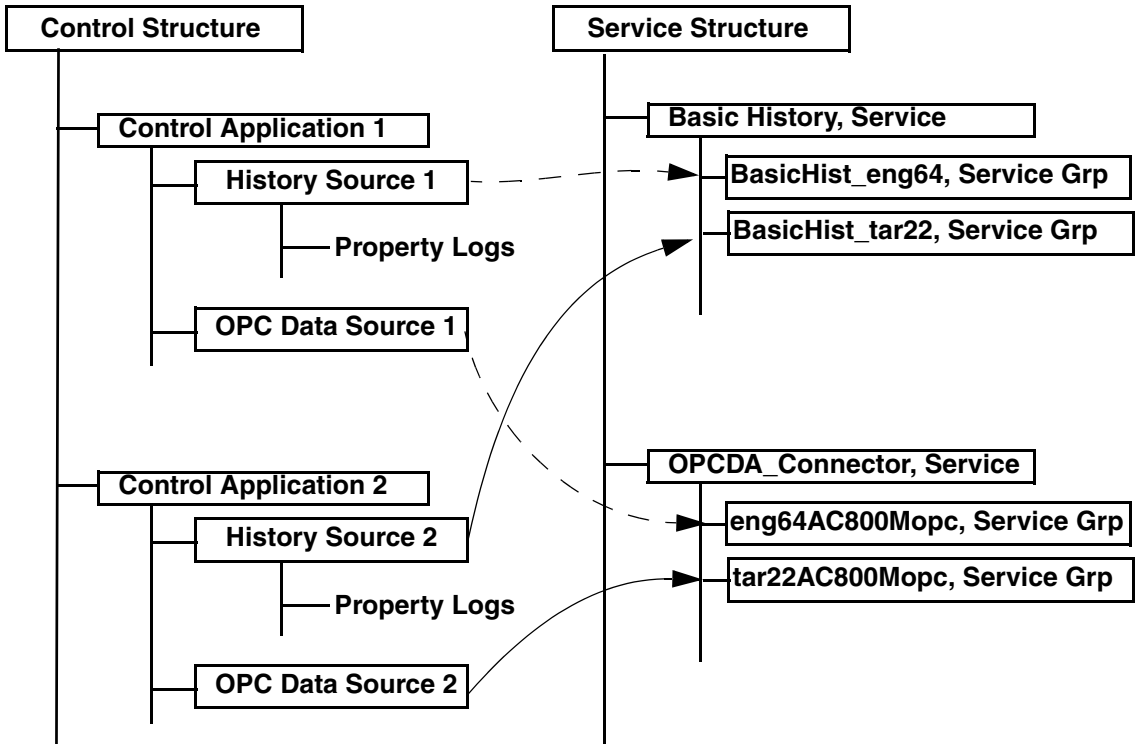


Figure 119. Example, History Source Configuration

Adding A History Source Aspect

To add a History Source aspect:

1. Select the object where the aspect is to be added and choose **New Aspect** from the context menu.
2. Select **History Source** in the New Aspect dialog and click **Create**, Figure 120.

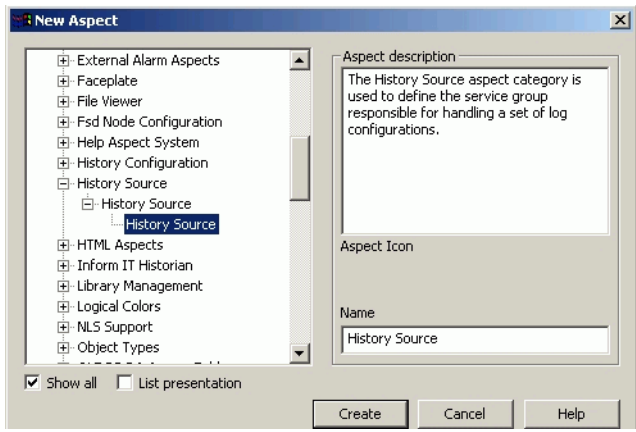


Figure 120. Selecting the History Source Aspect

3. Click on the History Source aspect and use the Service Group pull-down list to select the Service Group for a specific node, Figure 121 (use the same node specified in the OPC Data Source aspect (Figure 119).

4. Click **Apply**. This completes the History set-up requirements.

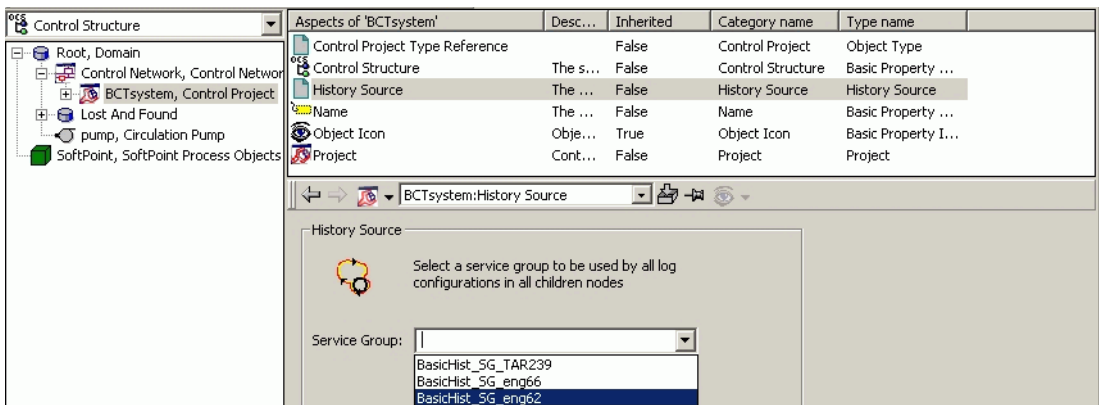


Figure 121. Configuring the History Source

Before configuring property logs, refer to History Configuration Guidelines on page 194 for important considerations and tips. For a quick demonstration on how to configure a property log, refer to Building a Simple Property Log on page 196.

History Configuration Guidelines

This section provides a quick reference for various topics that can be reviewed before actually starting configuring the history application. This includes:

- Data Collection and Storage Features for Optimizing Usage on page 194.
- Allocating Disk Space for File-based Logs on page 194.
- Configuring History Objects - Procedural Overview on page 195.
- Post-configuration Requirements on page 196.

Data Collection and Storage Features for Optimizing Usage

How certain data collection and storage parameters are configured directly affects CPU load and disk usage. To optimize CPU load and disk usage, consult the following sections before configuring logs:

- Data Collection Attributes on page 227.
- Deadband Attributes on page 242.
- File Storage vs. Oracle Tables on page 255.

Also refer to Property Log Configuration Reference on page 266. These examples provide guidelines and recommendations for common data collection/storage applications:

- Example 1 - Storing Instantaneous Values, No Compaction on page 248.
- Example 2 - Storing Instantaneous Values Using Compaction on page 248.
- Example 3 - Storing Calculated Values on page 250.
- Example 4 - Storing Calculated Values in Logs on page 252.
- Example 5- Storing Calculated Values with Compaction on page 253.

Allocating Disk Space for File-based Logs

To expand storage capacity for file-based logs (Storage Type = TYPE1 - TYPE5), specify the directory (or disk), and how much of the disk is to be allocated to History.

Be sure to allocate disk space BEFORE creating any file-based logs. Do this online via the Directory Maintenance History Utility as described in Directory Maintenance for File-based Logs on page 467.

Configuring History Objects - Procedural Overview

Configure property logs online using the Plant Explorer Workplace, or offline via the Bulk Log Configuration Import/Export utility. The Plant Explorer Workplace provides a graphical user interface for history configuration. The Bulk Log Configuration Import/Export utility uses Microsoft Excel to create a list of object properties, and then match object properties with their respective log templates. This method is much quicker than using the Plant Explorer when there is a large number of object properties that will use the same log template.

The Plant Explorer must be used to configure all other History objects. This includes archive devices and archive groups (if using archive functionality), message logs, report logs, and log sets. The Plant Explorer must also be used to create log templates. The log templates can then be used with the Bulk Log Configuration Import/Export utility to instantiate log configuration aspects on their respective objects.

To archive History data, build archive devices and archive groups. First build the archive devices, and then build the archive groups. Refer to Section 11, Configuring the Archive Function.

Log sets are used to start and stop data collection for multiple logs as a single unit. Build logs sets before the property logs if this functionality is used. Refer to Section 6, Configuring Log Sets.

Next configure message logs (Section 7, Alarm/Event Message Logging), and Report Logs (Section 8, Historizing Reports).

After creating the supporting History objects required by the application, build the basic property log structures to be used as templates for the property logs. A quick demonstration is provided in Building a Simple Property Log on page 196.

Finish the remainder of the History database configuration using the Bulk Log Configuration Import/Export utility. Refer to Bulk Configuration of Property Logs on page 270.

Post-configuration Requirements

- Make sure History Source objects have been added as required. Refer to Configuring Node Assignments for Property Logs on page 189.
- History logs will not collect data until they are activated. Refer to Starting and Stopping Data Collection on page 439.
- Use the stagger function as described in Stagger Collection and Storage on page 472 to stagger collection and storage times to avoid spikes in CPU usage which may occur when sample and storage rates cause a large number of messages to be written to disk at one time. Do this BEFORE activating logs. After stagger is complete, restart the History software and then activate the logs.
- Test the operation of one or more property logs. Use the Status tab on the Log Configuration aspect (Status on page 262), or use one of the installed desktop tools such as DataDirect or Desktop Trends.
- Sometimes changes to Log Configurations Aspects may not take effect until PPA History Services is restarted in the Connectivity Server where the Trend Log resides. To prevent this problem from occurring, restart the History Services after any History Log configuration activity. When redundant Connectivity servers are used, the restart should be staggered.

Building a Simple Property Log

This tutorial shows how to configure a property log for the data collection scheme illustrated in Figure 122.

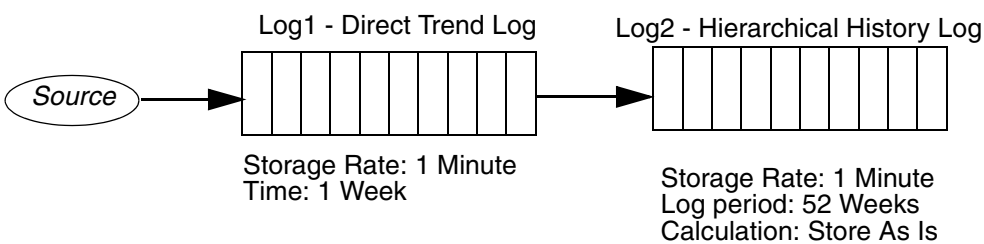


Figure 122. Example, Property Log Template

There are two basic parts to this tutorial: creating a log template, and configuring the log configuration aspect. Start with Creating a Log Template below.

Creating a Log Template

Property logs are instantiated from log templates in the Library structure. Therefore, the first step in implementing property data collection is to create the log template. This template establishes the log hierarchy and data collection scheme for the property logs. The instructions for doing this are organized in six parts:

- Adding a Log Template Object in the Library Structure on page 197.
- Adding a Trend Log on page 199.
- Configuring a Trend Log on page 199.
- Adding a History Log on page 201.
- Configuring a History Log on page 201.

Start with Adding a Log Template Object in the Library Structure on page 197.

Adding a Log Template Object in the Library Structure

To add a log template object in the Library structure:

1. In the Plant Explorer, select the Library Structure.
2. Add a History Log Template Object under the History Log Templates library. To do this:
   a. Select **History Log Templates > Default Templates** and choose **New Object** from the context menu.
   b. Select **History Log Template** in the New Object list, enter a name for the log template object, then click **OK**.

This name identifies the log template when selecting a template for a property log. A meaningful name will help with recognizing the template. One way is to describe the data collection scheme, for example: **Log_1M_1Wk_52Wk** (direct log for instantaneous data @ 1-minute sample rate storing for 1-week period, and one hierarchical log storing the data at same rate for a 52-week period). If there are multiple history servers, consider including some means to indicate the node on which the template will be applied.

The log template name is included in the Data Source name when a log configuration aspect is attached to an object (object name:property name,log template name). While a long name is supported, keep it reasonable.

c. Click **Create** when finished. This adds the History Log Template object to the History Log Template Library.

Opening the Log Template Configuration View

Select the new log template object, then select the object's **Log Template** aspect. This displays the Log Template view where the log hierarchy is built and data collection parameters specified, Figure 123.

The hierarchy starts with the *Property Log* placeholder which represents the data source (the object property for which data will be collected). All component logs in the hierarchy will be added under the Property Log placeholder.

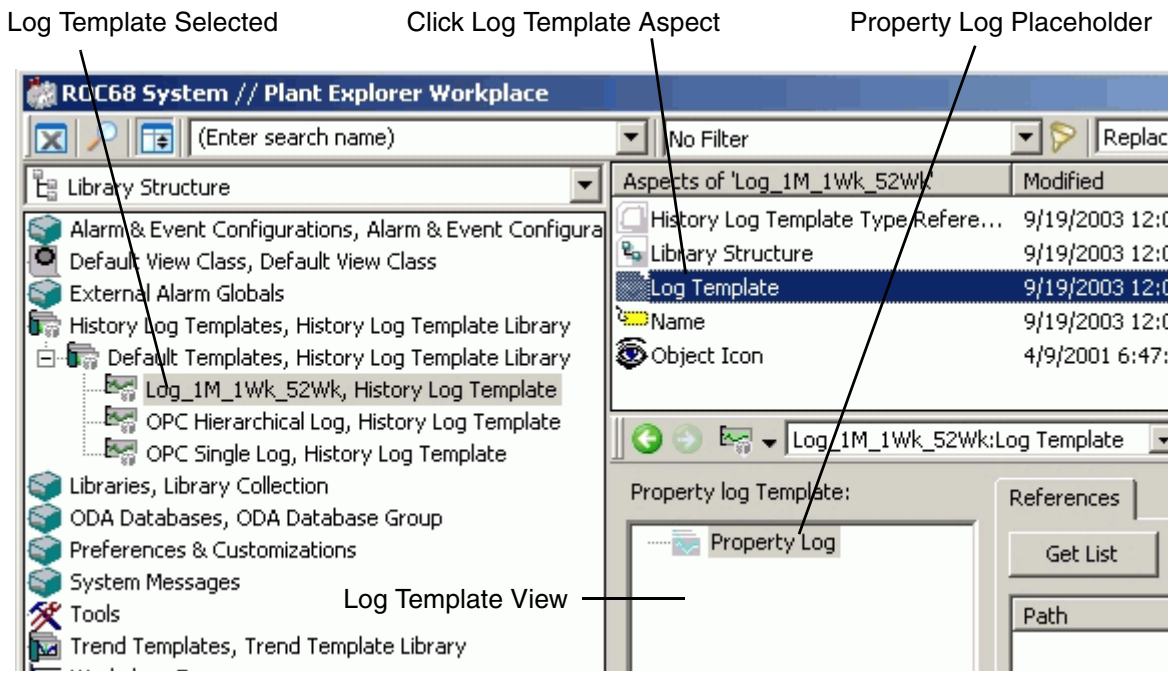


Figure 123. Display the History Log Template Configuration View

Continue with Adding a Trend Log on page 199.

Adding a Trend Log

The first log in the hierarchy is typically a trend log which is added as a direct log type[1]. This log collects directly from an OPC source. To add this log:

1. Select the Property Log to highlight it and choose **Add Log > OPC** from the context menu. This creates a Direct log with an OPC source.

The **Add** button at the bottom of the dialog is a less direct way of doing the same thing. Clicking the **Add** button opens the New Log Template dialog used to specify the following.

- **Source**. Select **OPC** from the Source pull-down list.
- **Log Type**. This defaults to **Direct**. This means the log collects directly from the data source. The other option is to select a **Lab Data** log. Lab data logs collect asynchronous data entered manually or through user programs. For collecting real-time synchronous data, leave the Log Type as **Direct**.
- The Collector Link option is for direct trend logs that collect from a remote Enterprise Historian.

2. Click **OK** when finished. This displays the configuration view for the direct type trend log.

Configuring a Trend Log

This configuration view has two tabs, Figure 124.

1. The one exception to this rule is when a property log is created to collect historical data from an earlier platform such as an Enterprise Historian with MOD 300 or master software.

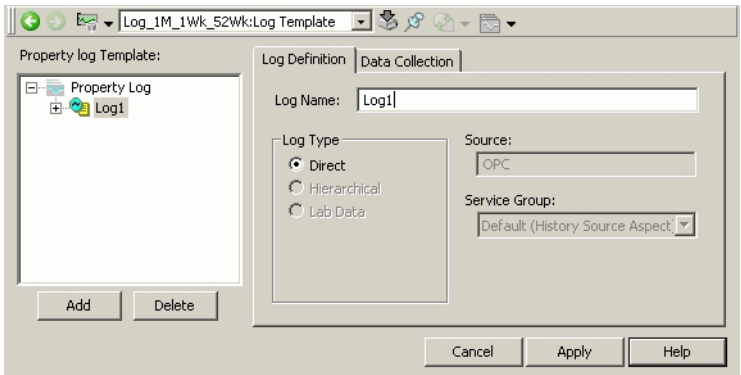


Figure 124. Basic History Trend Log Configuration View - Log Definition Tab

1. Use the **Log Definition** tab to specify a log name, Figure 125. Again, use a name that identifies the function this log will perform. For example, the name in Figure 125 identifies the log as a trend log with a 1-week storage size.

Figure 125. Specifying a Name for the Trend Log

   The Log Type, Source, and Service Group are fixed as specified in the New Log Template dialog and cannot be modified.

2. Click the **Data Collection** tab and configure how this log will collect from the OPC data source. For this tutorial, set the Storage Interval Max Time to **1 Minutes**, and the Storage Size Time to **1 Weeks**, Figure 126. For further information regarding trend log data collection attributes, refer to Collection Attributes for Trend Logs on page 227.

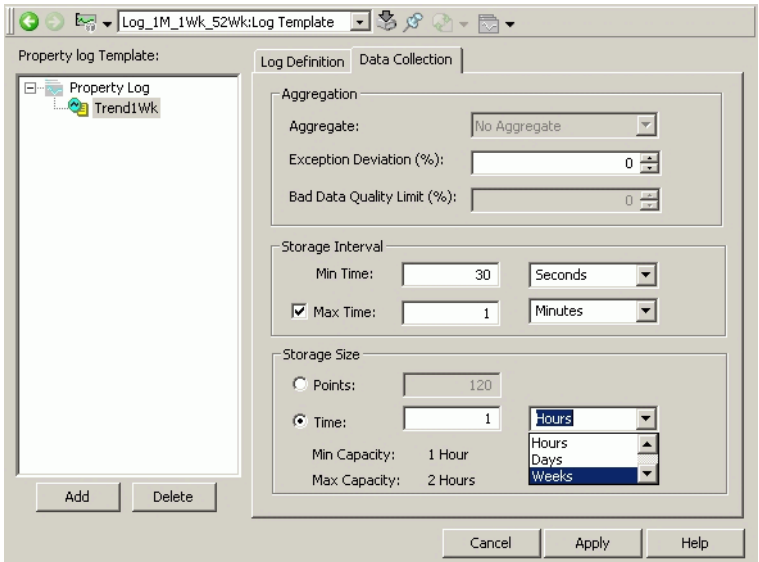


Figure 126. Data Collection Configuration for the Trend Log

Adding a History Log

Add a history log as a hierarchical log type under the direct trend log. This log will collect from the trend log. To add this log:

1. Select the trend log and choose **Add Log > IM History Log** from the context menu. This defines a **Hierarchical** log type with the Collector Link checked **Linked** checked, and the **IM History Log** selected in the Server Type. This establishes the log as a history log with all the associated Information Management functionality.
2. Click **OK** when finished. This displays the configuration view for the history log.

Configuring a History Log

The log configuration view for history logs has four tabs, Figure 127. Most of the log attributes on these tabs have valid default values, or are not mandatory to complete the configuration. Attributes that MUST be configured are covered in the

following steps. To configure any other attributes, refer to Property Log Attributes on page 220.

1. Use the **Log Definition** tab to specify a Log Name and Service Group.

   The Service Group pull-down list contains all History servers defined in the system. Use this list to specify the server where this log will reside. The list defaults to the local node.

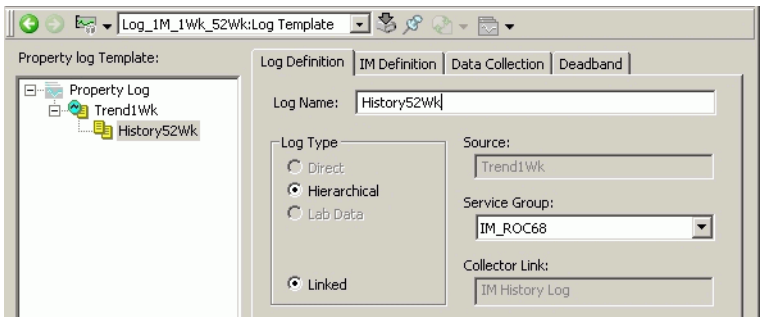


Figure 127. History Log - Log Definition Tab

2. Click the **IM Definition** tab, Figure 128.

   This tab is used to configure the Information Management functionality for this log. This includes assigning the log to its Log Set(s) and Archive Group. It is also used to configure a Start State and Collection Type.

When selecting archive groups from the Numeric Log IM configuration page, it is necessary to select the IM node on the first tab, apply the change, leave the aspect, and return. When the aspect reloads, the pick list will be correct.

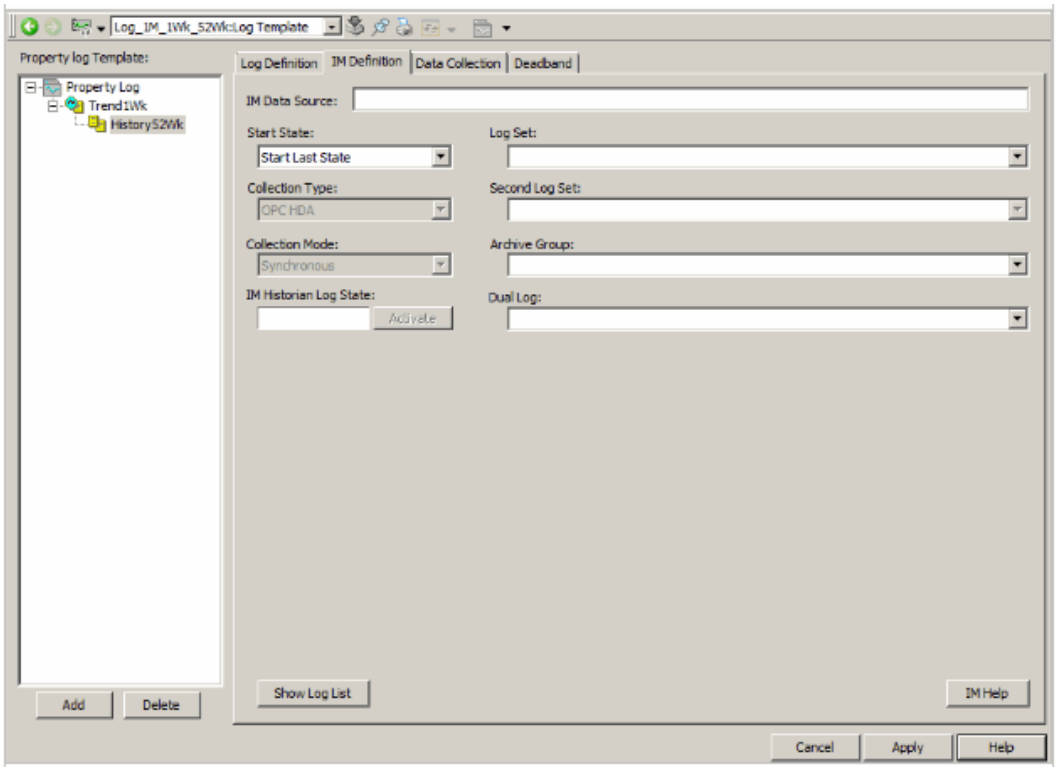


Figure 128. IM Definition Tab

Collection Mode defaults to Synchronous and cannot be modified. For the purpose of this tutorial, all attributes on this tab may be left at their default values.

3. Click on the **Data Collection** tab, Figure 129. For this example, this log needs to collect samples at a 1-minute rate and store 52 weeks worth of data. Certain data collection attributes must be configured to implement this functionality. These attributes are described in the following steps.

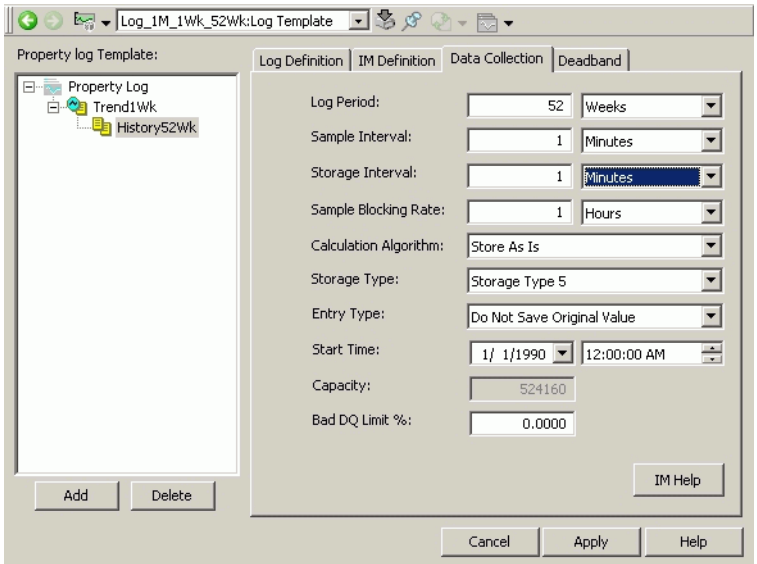


Figure 129. Data Collection Tab

a. Configure the Sample Interval to establish the base sample rate, for example: **1 Minutes**. The Storage Interval automatically defaults to the same value. Also, the Sample Blocking Rate defaults to a multiple of the Sample Interval, in this case, one hour.

b. Then enter the Log Period, for example: **52 Weeks**. This determines the Log Capacity (*log capacity = log period * storage interval*).

It is strongly recommended that the defaults be used for calculation algorithm (Store As Is) and storage type (type 5). Typically, calculations are not used for data collection. Desktop tools can perform calculations during data retrieval.

Store As Is causes data to be stored only when it is received from the data source. In this case the sample and storage intervals are essentially ignored for data collection, and are used only to properly size the log.

4. Click **Apply** when finished with both the trend and history log configurations. This completes the template configuration. Deadband is not here.

Configuring a Log Configuration Aspect

Add a Log Configuration aspect to the object from which data will be collected. The Log Configuration aspect is a container for one or more property logs. A dedicated property log is required for each object property whose values will be logged. The log hierarchy and data collection scheme for each property log will be based on a specified History Log Template.

The Log Configuration aspect will support as many properties and as many property logs as required for the object. For simplicity, it is recommended that just one Log Configuration aspect be created for an object. When creating more than one Log Configuration aspect for an object, there is a restriction that all property logs for a given property must be in the same Log Configuration aspect.

The instructions for this procedure are organized in six parts:

- Adding a Log Configuration Aspect to an Object on page 205.
- Opening the Log Configuration Aspect on page 206.
- Adding a Property Log for an Object Property on page 207.
- Reviewing the Log Configuration on page 209.
- Post-configuration Requirements on page 210.
- What to Do Next on page 210.

Start with Adding a Log Configuration Aspect to an Object on page 205.

Adding a Log Configuration Aspect to an Object

The Log Configuration aspect is typically added to an object in the Control Structure.

1. In the Plant Explorer, select the Control Structure (or other structure where the object resides).
2. Navigate to the object being configured for data collection, right-click, and choose **New Aspect** from the context menu. This displays the New Aspect dialog.
3. Find the History Configuration aspect category in the New Aspect dialog, and select the **Log Configuration** aspect, Figure 130. Use the default name, or specify a more meaningful name. Click **Create** when finished.

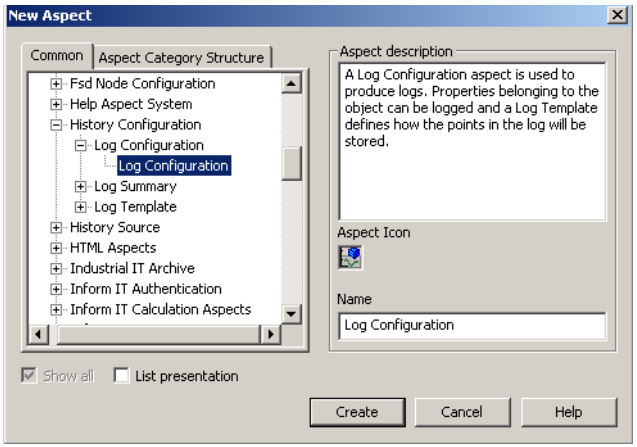


Figure 130. Selecting the Log Configuration Aspect

This adds the Log Configuration aspect to the object's aspect list.

Opening the Log Configuration Aspect

To display the Log Configuration aspect view, select the object, then click the Log Configuration aspect in the object's aspect list, Figure 131.

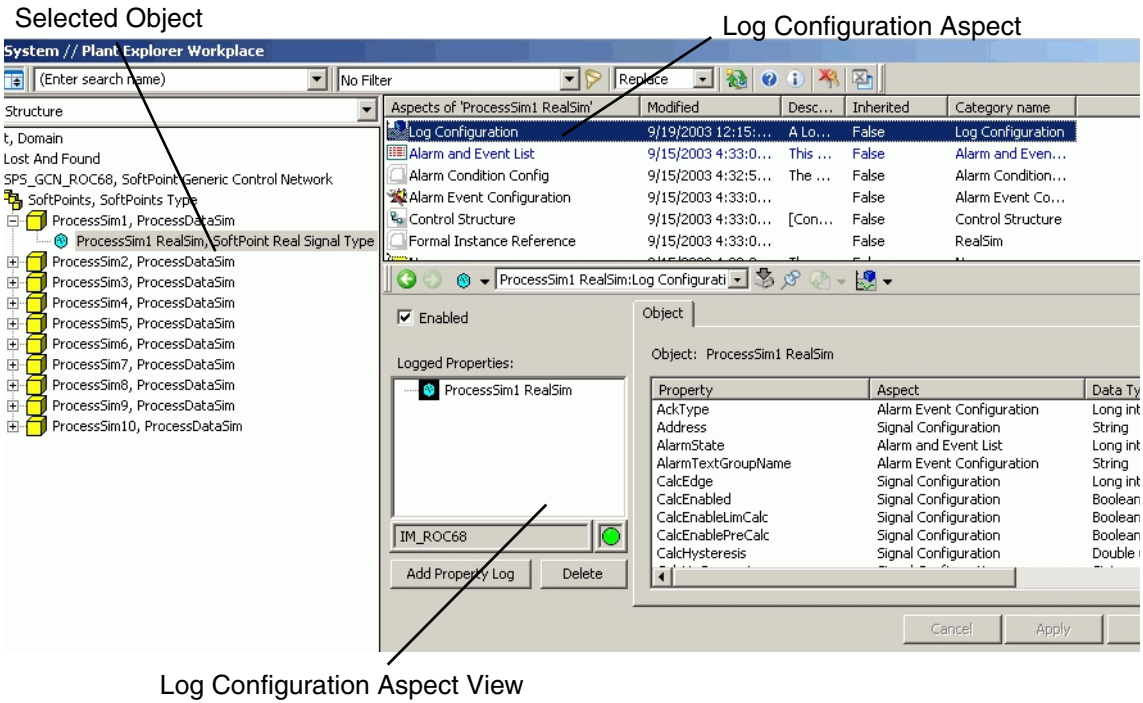


Figure 131. Log Configuration Aspect Added to the Object's Aspect List

The Logged pane in the Log Configuration view shows the name of the object with one or more property logs being added. A property log must be added for each property whose value will be logged.

Adding a Property Log for an Object Property

To add a property log for an object property:

1. Click the **Add Property** button, or right-click on the object name and choose **Add Property Log** from the context menu, Figure 132.

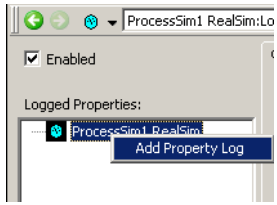


Figure 132. Adding a Property Log

This displays the New Property Log dialog, Figure 133. This dialog is used to select one of the object's properties for which to collect data, and apply the Log Template which meets the property's data collection requirements. If necessary, change the default data type specification for the selected property.

2. From the Property list, select the property for which data will be collected. Then select a template from the Template list.

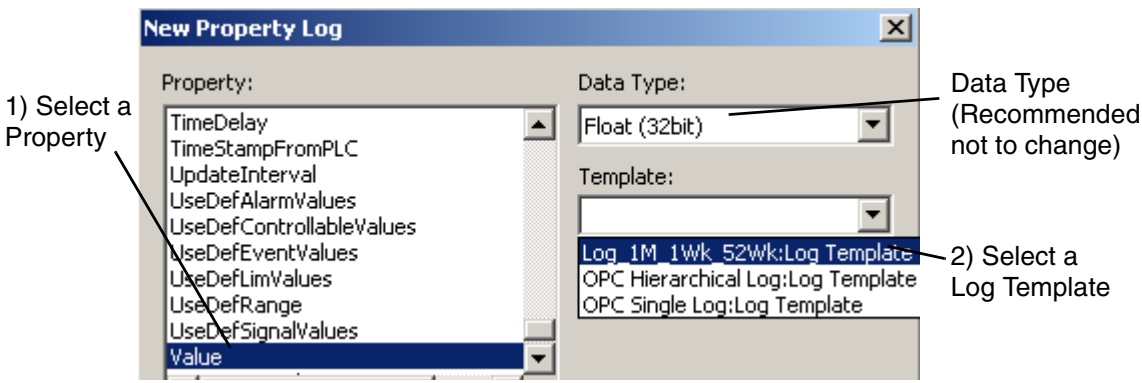


Figure 133. New Property Dialog

The supported data types for property logs are listed in Table 22. When a property is selected, the default data type is automatically applied. This is the native data type for the property as determined by the OPC server. The OPC/DA data type is mapped to the corresponding 800xA data type. DO NOT change the data type. When the storage type for the History log is configured as TYPE5 (recommended), the log is sized to accommodate the data type.

Table 22. Supported Data Types

OPC/DA Data Type(1)	800xA Data Type(2)	Size
VT_UI1	Byte	1 Byte
VT_I1	Char	1 Byte
VT_BOOL	VARIANT_BOOL	2 Bytes
VT_I2	Short Integer	2 Bytes
VT_UI2	Unsigned Short	2 Bytes
VT_I4	Long Integer	4 Bytes
VT_UI14	Unsigned Long	4 Bytes
VT_INT	Integer	4 Bytes
VT_UINT	Unsigned Integer	4 Bytes
VT_R4	Float	4 Bytes
VT_R8	Double	8 Bytes
VT_CY	Very Long Integer	8 Bytes

(1) As indicated on the Control Connection Aspect
(2) As indicated on the Log Configuration Aspect

Reviewing the Log Configuration

The instantiated log configuration for ProcessSim1 based on the selected property (value) and log template is shown in Figure 134.

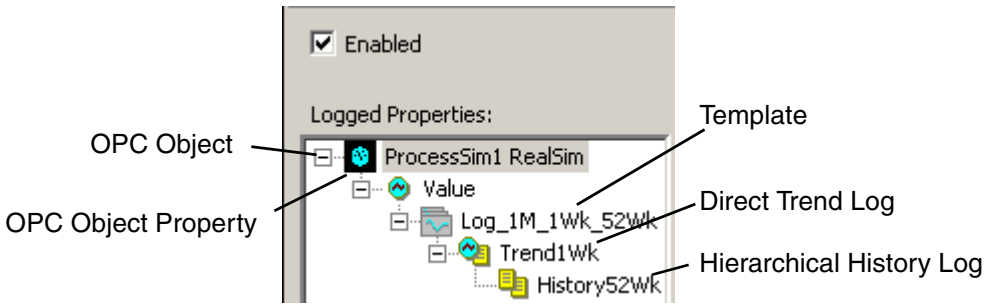


Figure 134. Instantiated Log Configuration

This aspect does not require any further configuration; however, some data collection parameters may be adjusted via the tabs provided for each component log. These tabs are basically the same as the tabs provided for log template configuration. Some configuration parameters are read-only and cannot be modified via the log configuration aspect. Also some parameters that were not accessible via the log configuration template are accessible via the log configuration aspect.

Two additional tabs are provided for the log configuration aspect. The **Presentation** tab is used to configure presentation parameters for viewing log data on a trend display. The **Status** tab is used to view log data directly via the log configuration aspect.

Post-configuration Requirements

History logs will not collect data until they are activated. Refer to Starting and Stopping Data Collection on page 439.

Test the operation of one or more property logs. Use the **Status** tab on the Log Configuration aspect (Status on page 262), or use one of the installed desktop tools such as DataDirect or Desktop Trends.

What to Do Next

This concludes the tutorial for configuring a basic property log. To create a large quantity of similar logs, use the Bulk Configuration utility as described in Bulk Configuration of Property Logs on page 270. To learn about common property log applications, refer to Property Log Applications on page 211. For more information

regarding the configuration of log attributes, refer to Property Log Attributes on page 220.

Property Log Applications

The following applications are demonstrated in this section:

- Lab Data Logs for Asynchronous User Input on page 211.
- Event-driven Data Collection on page 212.
- History Logs with Calculations on page 219.

In addition to these applications, historical process data can be integrated and consolidated from remote history servers. Data may be consolidated from remote History servers in the 800xA system that reside in other systems.

This functionality is configured using the History Access Importer tool and is described in Section 12, Consolidating Historical Data.

Dual Logs

As an option, configure a *dual log* where the same trend log feeds two history logs on two different history server nodes. This may be required when the application cannot wait for history data to be back-filled in the event that a history server goes off line. For example, shift reports may be required to execute at the end of each 8-hour shift, and cannot wait for the hardware to be repaired and for the data to be back-filled.

To implement this functionality, create a log template with two identical primary history logs, with the only difference being the Service Group definition.

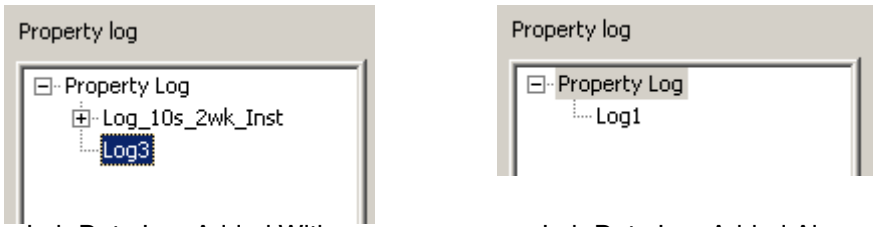
Lab Data Logs for Asynchronous User Input

The lab data log is a special type of property log. This log type is used to collect asynchronous data entered manually, or by an external application such as a User API program. Asynchronous logs must be Oracle-based (refer to Storage Type in Data Collection Attributes on page 227).

Lab Data logs are added in the property log hierarchy at the same level as direct logs. Select **Lab Data** as the Log Type in the New Log Template dialog. A lab data log can be added as a trend log, or as a history log (IM History Log link). The applicable configuration tabs will be displayed accordingly. The source field is

disabled since this is not applicable for lab data logs. Add a Lab Data log alone, or as part of a log hierarchy with a direct log, Figure 135.

When added with synchronous logs, the asynchronous lab data log does not receive data from the data source. The connection is only graphical in this case.



Lab Data Log Added With a Direct Log

Lab Data Log Added Alone

Figure 135. Representation of Lab Data Log in Property Log Hierarchy

When configuring a history-type lab data log, most of the tabs specific to history logs are not applicable. The only attribute that must be configured for a Lab Data log is Log Capacity as described in Data Collection Attributes on page 227.

A lab data log CANNOT be the source for another log. Secondary hierarchical logs cannot collect from lab data logs.

Event-driven Data Collection

Data collection for property logs may be event-driven. The event which triggers data collection is specified as a job description object configured via the Application Scheduler, and an attached data collection action aspect.

TCP/IP must be enabled for OMF if the job is to trigger data collection for logs on a different History server. This setting is configured via the Communication Settings dialog. To check, launch this dialog, from the Windows task bar. Choose **Start>Settings>Control Panel>Administrative Tools> PAS>Settings**. For further information, refer to Appendix A, Extending OMF Domain to TCP/IP.

The event-driven property log is configured much like any other property log as described in Building a Simple Property Log on page 196. The only special requirements when configuring the log for event driven data collection are:

- The direct trend log must always be active in order to collect data before the event actually triggers data collection.
- The event-driven log must be a history log, and it must be the (first) primary log connected to the direct trend log, Figure 136. Also this log must be configured to start deactivated. The log will go active when the event trigger occurs.

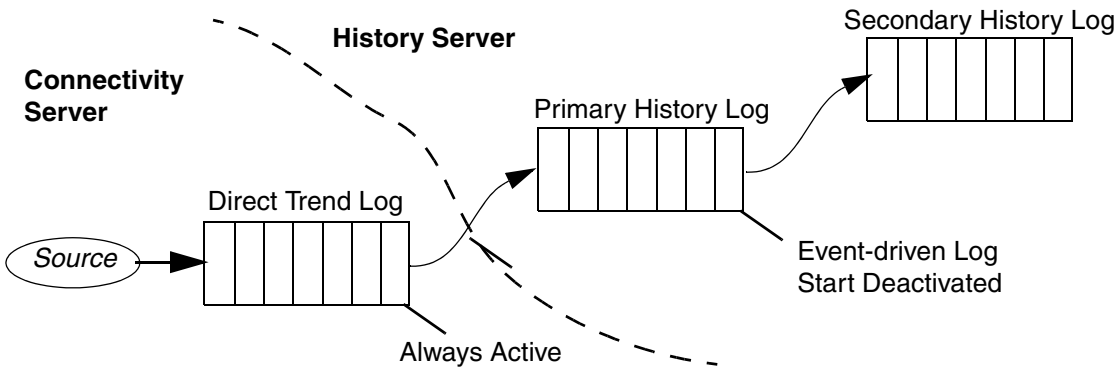


Figure 136. Example, Event-driven Data Collection

- Logs that reside on a consolidation node cannot be event-driven.

Disk throughput requirements for history collection need to be based on having the event-driven logs perpetually active. This way, when the event-driven logs are activated, it will not overload the disk.

Create a Job

To schedule event-driven data collection, create a job with a Data Collection action in Application Scheduler. The Scheduling Definition aspect will be set up to have specified logs collect for a one-hour period, every day between 11:30 AM and 12:30 PM. Jobs and scheduling options are also described in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

To create a job in the Scheduling structure:

1. In the Plant Explorer, select the **Scheduling Structure**.
2. Select **Job Descriptions** and choose **New Object** from the context menu.

3. In the New Object dialog, Figure 137, select **Job Description** and assign the Job object a logical name (for example StartLogging1).

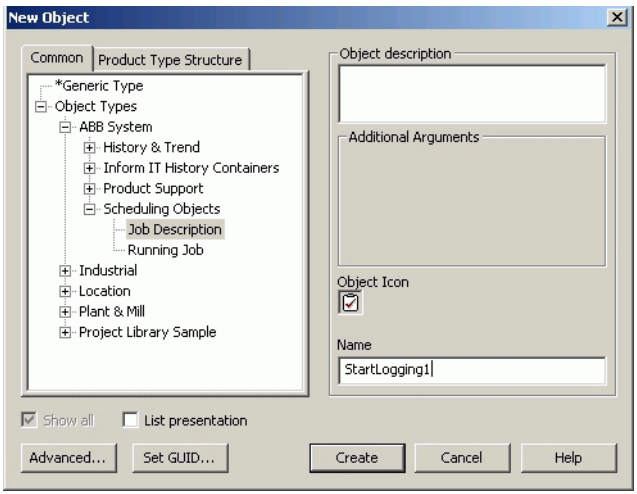


Figure 137. New Object Dialog

4. Click **Create**. This creates the new job under the Job Descriptions group, and adds the Schedule Definition aspect to the object's aspect list.

5. Click on the **Scheduling Definition** aspect to display the configuration view, Figure 138. This figure shows the scheduling definition aspect configured as a periodic schedule. The trigger event will occur once every day, starting July 3rd at 12:00 PM, and continuing until August 3rd at 12:00 PM.

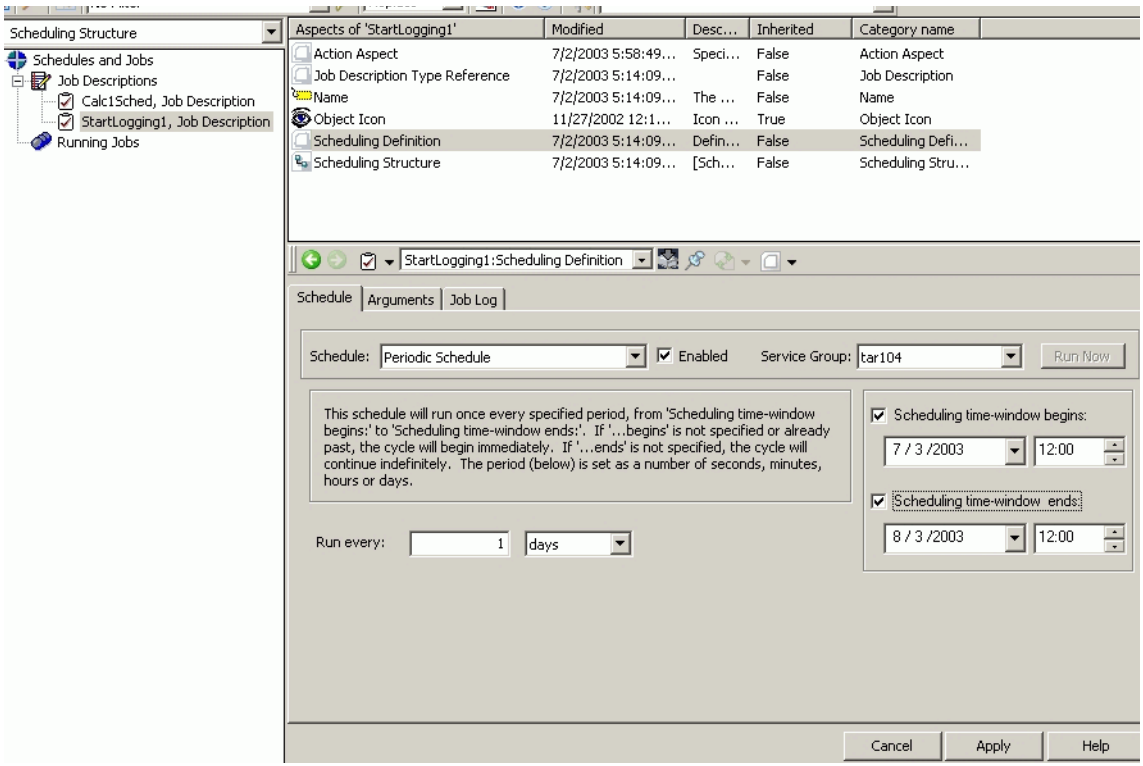


Figure 138. Scheduling Definition Configuration View

Adding and Configuring the Data Collection Action

Actions are implemented as aspects on an object which is on or under a job description in the scheduling structure.

To add an action:

1. From the Job object (for example StartLogging1), choose **New Aspect** from the context menu.
2. In the New Aspect dialog, browse to the Scheduler category and select the Action aspect (path is: **Scheduler>Action Aspect>Action Aspect**). Use the default aspect name, or specify a new name.

3. Click **Create** to add the Action aspect to the job.
4. Click on the Action aspect to display the configuration view.
5. Select **Data Collection Action** from the Action pull-down list. This displays the Data Collection plug-in, Figure 139.

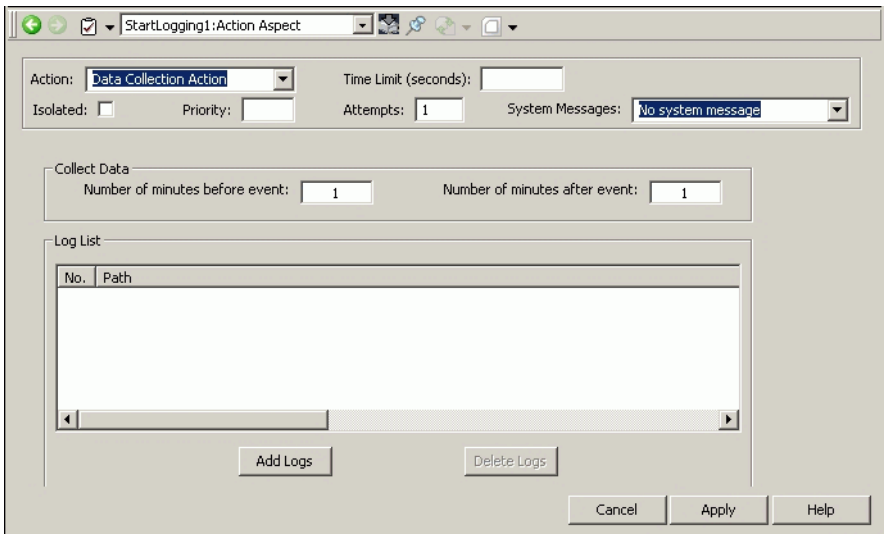


Figure 139. Plug-in for Event-driven Data Collection

6. Specify the logs to be activated with this event:

Event-driven data collection is limited to the primary history log which is connected directly to the direct trend log. Neither the trend log, nor secondary history logs can be event-driven. Also, event-driven data collection cannot be applied to logs on a consolidation node.

   a. Click the **Add Logs** button. This displays a browser dialog similar to the Plant Explorer. Use this browser to find and select the logs in the Plant Explorer structures, Figure 140.

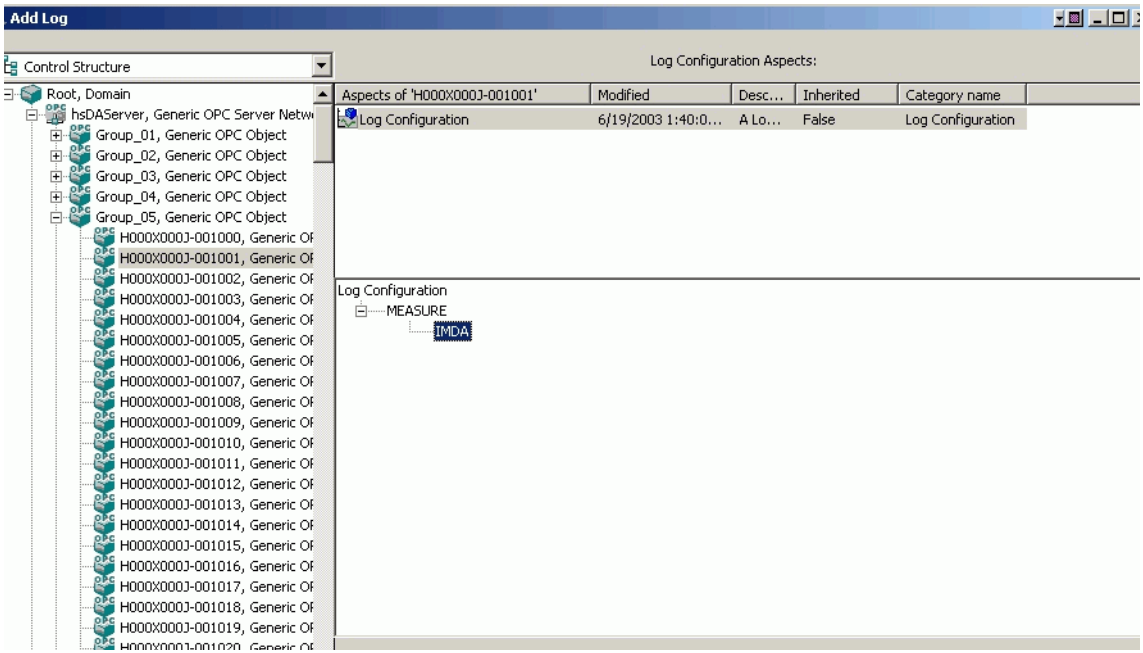


Figure 140. Adding Property Logs to the Log List

b. Select the object whose log will be added to the list, then select the log in the log hierarchy, and click **Add Log To List**. This adds the selected log to the log list, Figure 141.

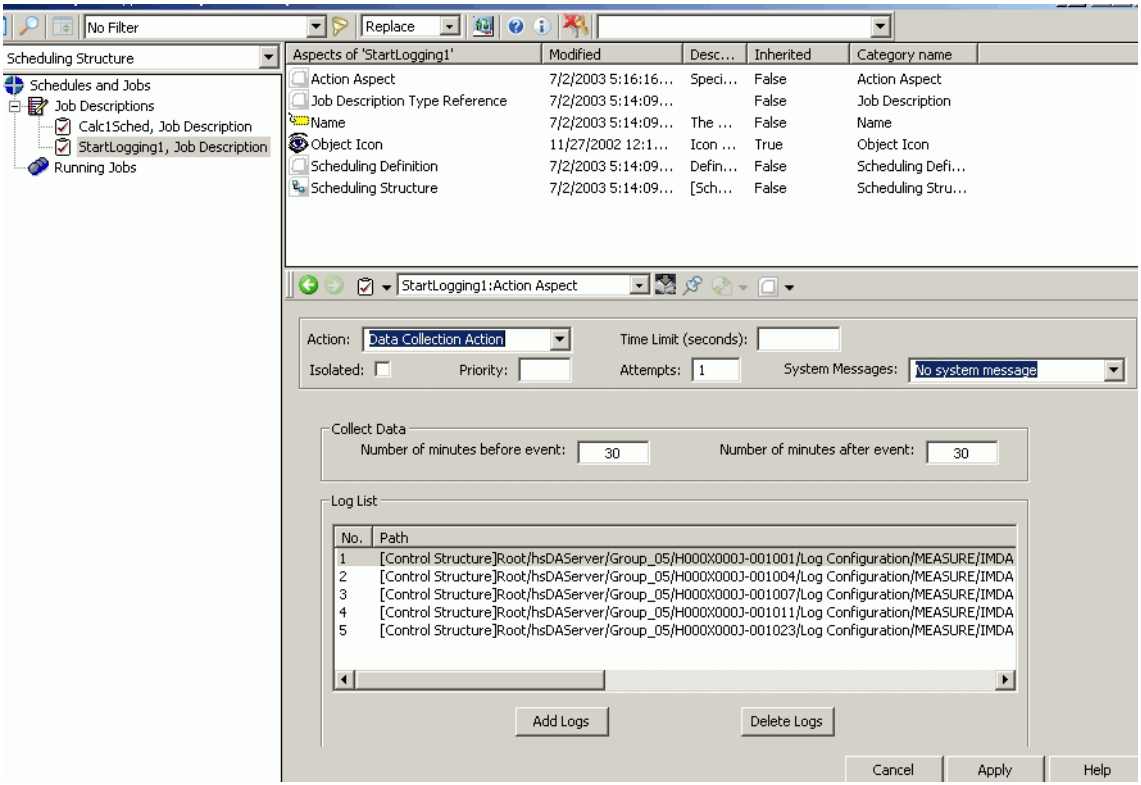


Figure 141. Logs Added to the Log List

c. Repeat steps a and b for as many logs as required.

7. Use the **Number of minutes before event** and **Number of minutes after event** fields to specify the time range for the history logs to collect from their respective trend logs. The maximum range is 10,000 minutes before plus 10,000 minutes after the job starts.

The data collected before the event comes from the trend log. Therefore the number of minutes specified here CANNOT EXCEED the configured Storage Size for that log.

History Logs with Calculations

Desktop tools can perform calculations during data retrieval so it is generally not necessary to use them for data collection.

When calculations are implemented for data collection, the calculation is performed before the data is stored in the log. This allows a single value to be stored that represents a larger time span of values, or key characteristics of the data. For example, a property value can be sampled every minute and the values put into a trend log. History logs can then calculate and store the hourly average, minimum, and maximum values. This is illustrated in Figure 142. When applying calculations for data covering the same time span, add the history logs at the same level as shown in Figure 142 (all collecting directly from the trend log).

In addition to any mandatory attributes that must be configured, the Calculation Algorithm must also be configured. Change the calculation algorithm before specifying the Storage Interval. Refer to Calculation Algorithm in Data Collection Attributes on page 227.

Retain the defaults for the remaining attributes, or configure additional attributes as may be required by the application. Refer to Property Log Attributes on page 220 for details.

More detailed examples are described in Property Log Configuration Reference on page 266:

- Example 3 - Storing Calculated Values.
- Example 4 - Storing Calculated Values in Logs.

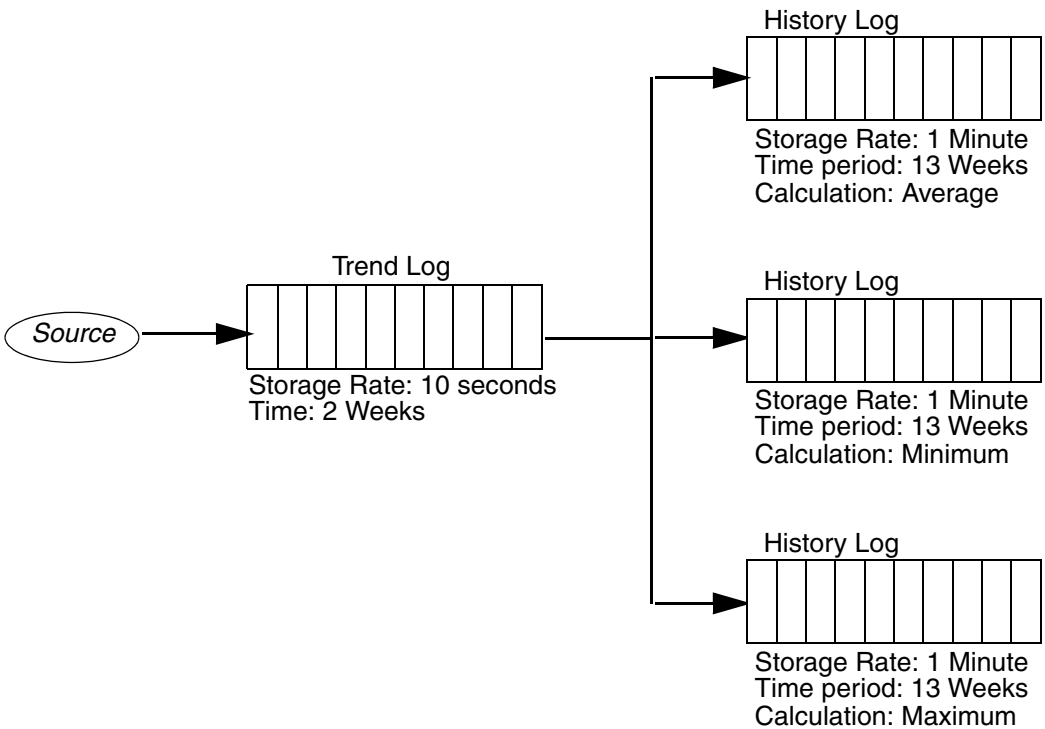


Figure 142. Example, Property Log using Calculations

Property Log Attributes

This section covers attributes for all property logs including trend, lab data, and history log types.

Attributes for individual logs within a property log hierarchy are accessible under their respective tabs. Some attributes are configured when the log is added to the property log hierarchy via the New Log Template dialog. For further information regarding a log attribute category, refer to:

- New Log Template Attributes on page 221.
- IM Definition Attributes on page 223.
- Data Collection Attributes on page 227.
- Deadband Attributes on page 242.

- Presentation on page 261.

Limitations are:

- Most log attributes can be edited while configuring a log template and before saving it. Exceptions are: Collection Mode and Log Capacity (for synchronous logs). Log capacity must be configured for lab data logs.
- The configuration of a Log Template that has Log Configuration aspects instantiated cannot be modified. To modify the template, delete the instantiated Log Configuration aspects.
- To modify certain attributes on Log Configuration aspects, the log must be deactivated.
- After saving the log configuration aspect, the following attributes can not be edited (in addition to the ones above): IM Data Source, Storage Interval, and Storage Type.

New Log Template Attributes

The New Log Template dialog is displayed when the Add button is used to add a component log in the property log hierarchy. This dialog is used to specify: Source., Log Type., and Collector Link.. The preferred method is to use the predefined selections using the **Add Log** context menu.

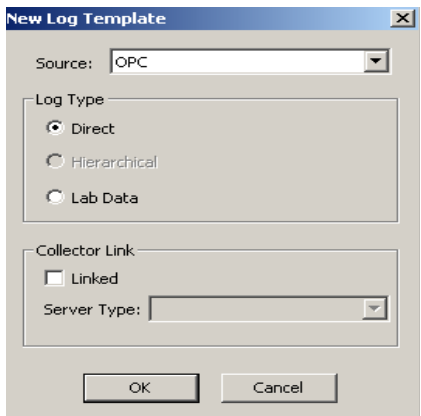


Figure 143. New Log Template Dialog

- **Source.**

  This specifies the type of data source.

  – For direct trend logs, set the source to **OPC** or an appropriate plug-in such as TTD.

  – For all hierarchical logs the source automatically defaults to the name of the log from which the hierarchical log collects (either a direct log or another hierarchical log).

  – Source is not applicable for lab data logs, or for direct history logs that collect from remote (Enterprise Historian-based) logs (Linked).

- **Log Type.**

  When adding a log, specify the type:

  – **Direct** - Log collects directly from the data source. Use this log type for the first component log in the property log hierarchy when building a log that collects from an OPC object property. This log type is used to collect from a remote (Enterprise Historian-based) log when Linked is checked.

  – **Lab Data** - Log entries are asynchronous (manually entered or generated from user program). Lab data logs must be added directly to the Property Log placeholder in the log hierarchy.

  – **Hierarchical** - Log collects either from a direct log, or another hierarchical log. This selection is made automatically and cannot be changed when adding a log to direct log, or another hierarchical log.

- **Collector Link.**

  The Collector Link specification is required for history logs. Check the **Linked** check box, then select the **IM History Log** for the **Server Type** from the pull-down list. This establishes the log as a history type for hierarchical and lab data logs.

Log Definition Attributes

There are two versions of this tab depending on whether a trend log (Direct) or history log (hierarchical) was added, Figure 144. This tab is used to specify: Log

Name. and Service Group. (hierarchical logs only). The other attributes on this tab are read-only. They are defined via the New Log Template dialog.

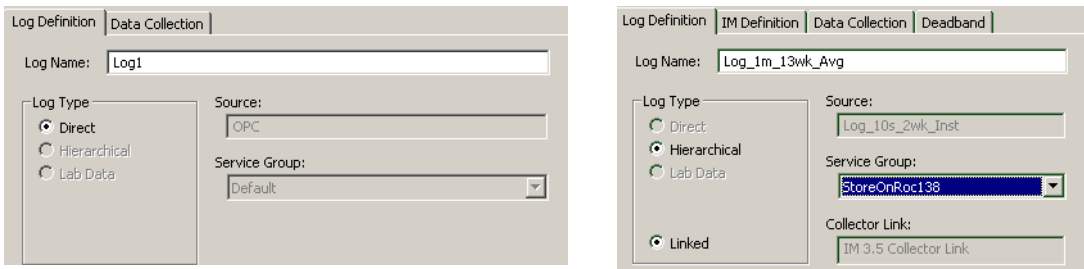


Figure 144. Log Definition Tab

- Log Name.

  This is the name by which each component log is identified within the property log hierarchy. Use a meaningful name. One way is to describe the data collection scheme in the name, for example: **Log_1m_13wAvg** (log for data collected @ 1-minute sample rate and storing the 13-week calculated average).

- **Service Group.**

  The Service Group pull-down list contains all History servers defined in the system. Use this to specify the server where this log will reside. The list defaults to the local node.

The **Default (History Source Aspect)** is only applicable for trend logs.

IM Definition Attributes

This tab is only available for linked history logs using the **IM History Log** server type. It is used to configure Information Management history functionality, Figure 145. This includes: Log Set, Archive Group, Start State, Collection Type, IM Historian Log State, and Dual Log.

Collection Mode is a read-only attribute.

The IM Data Source attributes is not accessible for log template configuration. This attribute is defined via the **IM Definition** tab on the Log Configuration aspect.

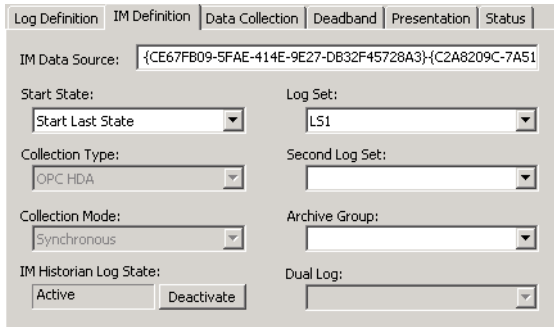


Figure 145. IM Definition Tab

IM Data Source

The IM data source field is activated on the Log Configuration aspect view. This field is dimmed on the log template view. The IM data source defaults to a unique ID for the object name:property name, log template name and its path in the structure. For hierarchical logs connected to a direct log, the IM data source is fixed and cannot be changed.

For applications where History collects data from a remote history log configured in an existing history database, the Collection Type is set to **API**. In this case enter the name using the following format: **$HS***object*,attribute-*n-oIPaddress*

The IP address is only required when collecting from multiple sites and identical log names are used on some sites. Refer to Consolidating Historical Process Data on page 368.

Log Set

This is the primary Log Set to which this log belongs. The **Log Set** pull-down list includes all log sets configured on the local history server. Logs can be assigned to a second log set via the Second Log Set field. To delete a log from a log set, clear the Log Set field. For details on configuring log sets, refer to Section 6, Configuring Log Sets.

Archive Group

This is the archive group to which this log belongs. The **Archive Group** pull-down list includes all archive groups configured on the local history server. To delete a log from an archive group, clear the Archive Group field. Logs may also be assigned to an archive group via the Archive Group aspect. This is described in Configuring Archive Groups on page 341.

If available archive groups are not listed in the Archive Group combo box, click Apply changes to the Log Template aspect. This makes the archive groups available for selection via the combo box.

Start State

This is the initial log state when History Services is restarted via PAS. Choices are:

START_INACTIVE
START_ACTIVE
START_LAST_STATE (default) - restart in state log had when node shut down

Hierarchical logs do not start collecting and storing data until the direct log becomes active. The entire property log with all of its component logs can be activated and de-activated as a unit via the **Enable** check box in the Log Configuration aspect.

Collection Mode

The collection mode determines how data is processed when it is received by History. Collection Mode also determines the valid choices for Collection Type and Storage Type. Collection mode is a read-only attribute whose value is fixed based on the selected Log Type.. The mode can be either synchronous or asynchronous:

SYNCHRONOUS - periodic *Time Sequence* data. This is the collection mode for logs which collect from a specified object property. The time interval between samples received by History must always be the same. Calculations and deadband can be applied to synchronously collected data.

If Collection Mode is **Synchronous**, all Storage Types are allowed. However, the data must be entered in time forward order. This means that once an entry is stored at a particular time, the log will not allow entries with a date before that time. This is true for all Collection Types: API, OPC HDA, and USER_SUPPLIED.

Also, entries are received and stored at the configured storage rate. For example, for a user supplied, synchronous log, attempting to add an entry every second for a log with a 1 minute storage interval, only one entry over a one-minute period will be stored. If only one entry is received every day for a one minute log, history will insert at least one "NO DATA" entry between each value received.

ASYNCHRONOUS - data is entered by an application such as DataDirect, or Display Services. This mode is for *Lab Data* logs. The Storage Type must be **Oracle** and the Collection Type is undefined. The time between entries sent to History does not have to be the same every time. When the log reaches its capacity, the oldest values are deleted. Retrieval is time based.

Collection Type

This determines the application for which History collects data. Collection type is only configurable for primary history logs. The options are:

- **OPC HDA**- The History log collects OPC data from a trend log.
- **API** - History periodically collects data from a remote history log.
- **USER_SUPPLIED**: Samples are *sent* to History from a user application such as DataDirect. The time between samples sent is expected to be equal to the sample rate of the log.

When Collection Mode is Asynchronous, the Collection Type is undefined; however, the log operates as if the collection type was defined as User Supplied.

IM Historian Log State

Hierarchical logs do not start collecting and storing data until the direct log becomes active. Use the **push** button to toggle the state of the IM History log.

Dual Log

This configuration parameter is not used.

Show Log List

Use the **Show Log List** button to refer to the Inform IT History Log List aspect (refer to Viewing Log Runtime Status and Configuration on page 442).

Data Collection Attributes

The Data Collection tab is used to specify how data will be collected for each component log in a property log hierarchy. This includes specifying sample intervals, log capacity, and so on. There are two versions of this tab depending on whether a trend log, or a history log linked via the IM 3.5 Collector is added. Refer to:

- Collection Attributes for Trend Logs on page 227.
- Collection Attributes for History Logs on page 231.

Collection Attributes for Trend Logs

The Data Collection tab for trend logs is shown in Figure 146. Use this tab to configure Aggregate (hierarchical logs only), Storage Interval, and Storage Size. This tab also provides an indication of Min and Max Capacity.

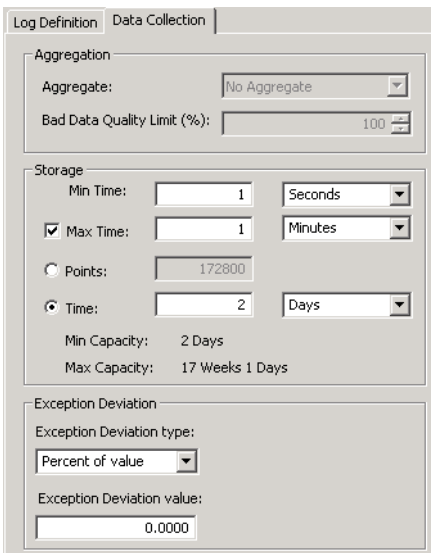


Figure 146. Data Collection Tab - Trend Logs

Aggregate

For hierarchical trend logs apply any one of the aggregates described in Table 23. Aggregates are not applicable for Direct-type logs.

Although hierarchical trend logs support this functionality, it is a good practice to collect and store raw data, and then perform the required calculations using the data retrieval tool, for example, DataDirect.

Table 23. Aggregate Options

Aggregate	Description
Interpolated	Interpolated values.
Total	Time integral of the data over the re-sample interval.
Average	Average data over the re-sample interval.
Timeaverage	Time weighted average data over the re-sample interval.
Count	Number of raw values over the re-sample interval.
Standard Deviation	Standard deviation over the re-sample interval.
Minimum Actual Time	Minimum value in the re-sample interval and the time stamp of the minimum value (in seconds).
Minimum	Minimum value in the re-sample interval.
Maximum Actual Time	Maximum value in the re-sample interval and the time stamp of the maximum value (in seconds).
Maximum	Maximum value in the re-sample interval.
Start	Value at the beginning of the re-sample interval. The time stamp is the time stamp of the beginning of the interval.
End	Value at the end of the re-sample interval. The time stamp is the time stamp of the end of the interval.
Delta	Difference between the first and last value in the re-sample interval.
Regression Slope	Slope (per cycle time) of the regression line over the resample interval.
Regression Const	Intercept of the regression line over the re-sample interval. This is the value of the regression line at the start of the interval.
Regression Deviation	Standard deviation of the regression line over the re-sample interval.
Variance	Variance over the sample interval.
Range	Difference between minimum and maximum value over the sample interval.
Duration Good	Duration (in seconds) of time in the interval during which the data is good.

Table 23. Aggregate Options (Continued)

Aggregate	Description
Duration Bad	Duration (in seconds) of time in the interval during which the data is bad.
Percentage Good	Percentage of data in the interval which has good quality. One (1) = 100%
Percentage Bad	Percentage of data in the interval which has bad quality. One (1) = 100%
Worst Quality	Worst quality of data in the interval.

Storage Interval

The storage interval is the rate at which samples are collected and stored. This is configured using **Min Time**, **Max Time** and **Exception Deviation (%)** properties.

Set **Min Time** to the base rate at which samples are to be collected. Samples will be stored at this rate unless data compaction is implemented using Exception Deviation (%). The shortest possible time between two samples is one (1) sec.

Exception Deviation (%) establishes a deadband range by which to compare the deviation of a sample value. If the current sample value deviates from the previous sample value by an amount equal to or greater than the specified Exception Deviation (%), then the sample is stored; otherwise, the sample is not stored.

Max Time establishes the maximum time that may elapse before storing a sample. This way, if a process value is very stable and not changing by the specified Exception Deviation (%), samples are stored at the rate specified by the Max Time.

To store all samples (no data compaction), set the Exception Deviation (%) to 0, and set the Max Time equal to the Min Time.

It is always recommended to have Max Time greater than Min Time. When Max Time equals Min Time, extra values can be inserted into the log when the next sample point is delayed from the source.

Storage Size

Storage size can either be specified as number of points stored in the log or as the time range covered in the log. If the number of points is specified, the time covered in the log will be calculated and displayed in the time edit box.

Min and Max Capacity

The values shown in these fields are an approximation of the number of points multiplied by the Min and Max Times respectively.

Collection Attributes for History Logs

The data collection function for history logs is illustrated in Figure 147. This function collects values and buffers them internally for a period of time according to the sample blocking rate. It then performs the specified calculation on the buffered data. A value is calculated for every storage interval. These calculated values are passed to the deadband function which checks to refer to if the value is within or outside the deadband range. If deadband is not active, the value is sent to storage. If deadband is active, the value is sent to storage if any one of the following are true:

- The value is outside the deadband range.
- The value is within the deadband range, but the deadband storage interval time has elapsed since the last value was stored.
- The value status changes (possible statuses are: no data, bad data, good data).

If none of these criteria are met, the value is not stored.

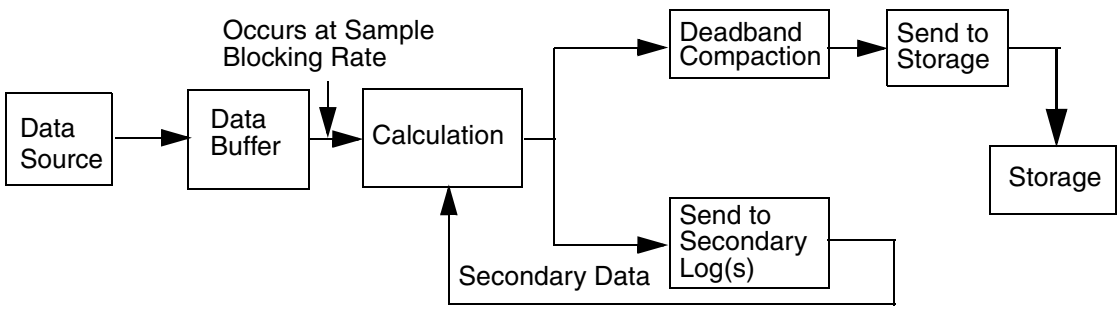


Figure 147. Data Collection, Functional Diagram

Figure 147 shows that data sent to a secondary log does not undergo data compaction, even if the primary log has deadband configured. The secondary log gets all values that result from the calculation. Thus, the calculation performed by the secondary log is more accurate than if done using data stored after deadband compaction.

History can be configured to perform calculations on collected data before the data is actually stored in the log. When a calculation is active, the specified algorithm, such as summing or averaging, is applied to a configured number of data samples from the data source. The result of the calculation is stored by History.

For example, configure History to collect five values from the data source, calculate the average of the five values, and then store the average. This calculation is actually configured by time, not the number of values. The calculation is applied to the values collected over a configured time period.

Data compaction can also be configured. This uses a mathematical compaction algorithm to reduce the amount of data stored for properties. Both calculations and deadband can be used concurrently for a log. The calculation is done first. Then the deadband algorithm is applied to see if the result should be stored.

For examples of data collection applications, refer to Property Log Configuration Reference on page 266.

The **Data Collection** tab for history logs is shown in Figure 148. Use this tab to configure: Log Period, Sample Interval, Storage Interval, Sample Blocking Rate, Calculation Algorithm, Storage Type, Entry Type (when storage type = 5), Start Time, Log Capacity (for lab data log only), Bad Data Quality Limit.

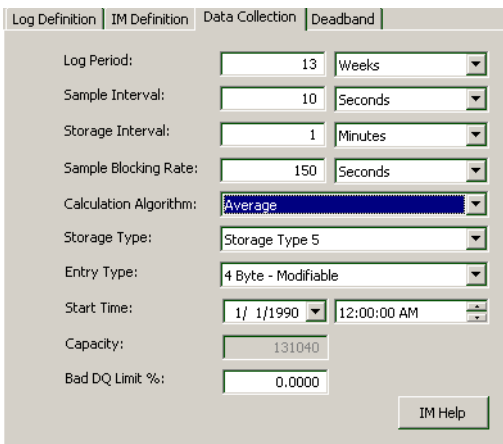


Figure 148. Data Collection Tab - History Logs

Log Period

Disk space is allocated to logs according to their log capacity, which is a function of the log time period and storage interval. The log period can be adjusted on an individual log basis to keep log capacity at a reasonable size for each log. For all synchronous logs with or without data compaction, log capacity is calculated as: log capacity = (log period/storage interval).

When using compaction (or Store As Is), the log period is effectively increased so the same number of samples (as determined by the log capacity) are stored over a longer period of time.

Enter the log period as an integer value ≤ 4095 with time unit indicator for **Weeks**, **Days**, **Hours**, **Minutes**, **Seconds**. The maximum is **4095 Weeks**.

Sample Interval

This is the rate at which data is sampled from a data source. For USER_SUPPLIED, this is the expected time interval between samples sent to History. Whether or not the sample is stored depends on whether or not a calculation algorithm is specified, and whether or not a compaction deadband (or Store As Is) is specified.

Enter the sample interval as an Integer value with a time unit indicator: **Weeks**, **Days**, **Hours**, **Minutes**, **Seconds**. Monthly is NOT a legal sample interval.

If data is not received within 50% of the interval prior to the expected time, or 100% of the interval after the expected time, the entry value is marked with status NO_DATA. Time stamps may drift due to load in the system. An example of a case where time drift can cause NO_DATA entries to be stored is illustrated in Figure 149.

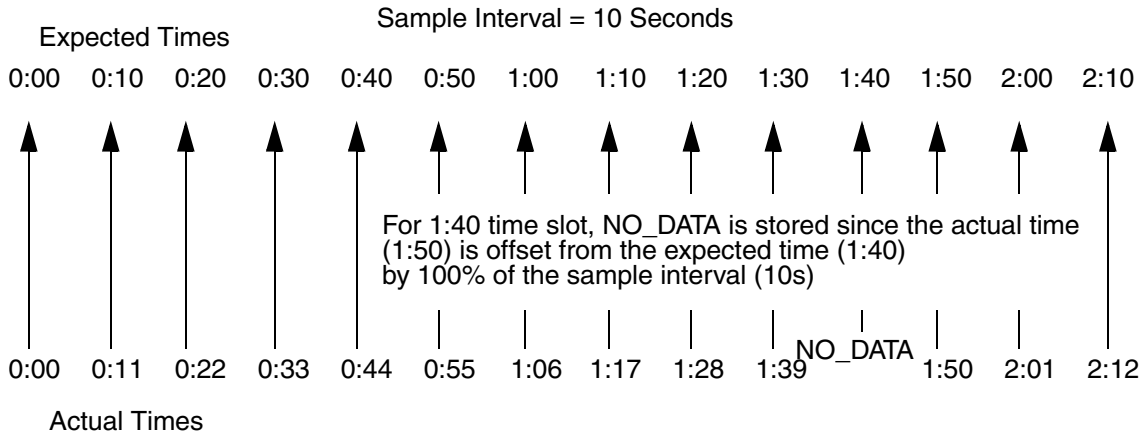


Figure 149. Example of Time Drift

Storage Interval

This is the time interval at which data is stored in the log, except when deadband is active (or Store As Is is used). Enter the storage interval as an Integer value with a time unit indicator: **Weeks**, **Days**, **Hours**, **Minutes**, **Seconds**. The maximum is **4095 Weeks**. The storage interval must be a multiple of the sample interval. For example, if the sample interval is 5, the storage interval can be 10, 15, 20, 25, and so on. When using Store As Is or Instantaneous as the calculation algorithm, the storage interval must be set equal to the sample interval.

A special storage interval is monthly. This causes a value to be stored at the end of the month. The timestamp of a monthly log depends upon the storage interval and alignment of its data source:

- If the data source is a daily log which is aligned to 06:00, then the monthly log values will have a time of 06:00.
- If the data source is hourly or less, then the monthly log will be around midnight (depending on the alignment of the data source). For example, an hourly log at 30 minutes after the hour will produce a monthly value at 00:30.
- If the data source is an ‘x’ hours log aligned to 06:00 (for example, 8 Hours log with values stored at 06:00, 14:00, 22:00, and so on), then the monthly log will have a time of 06:00.

Sample Blocking Rate

This controls the frequency at which data is collected from the source trend log. The sample blocking rate is only used in primary history logs (connected directly to the direct log in the property log hierarchy).

Enter the Sample Blocking rate as an Integer value with a time unit indicator: **Weeks**, **Days**, **Hours**, **Minutes**, **Seconds**. The Sample Blocking Rate must be a multiple of the storage interval. The maximum is **4095 Weeks**.

Sample Blocking Rate is used based on the type of the data source. History Logs that get data from Trend Logs use OPC/HDA. The blocking rate represents frequency that data is requested from the Trend Log. If the blocking rate is 60 minutes, once an hour the data will be requested from the Trend Log. For logs that are collected from an API source, the blocking rate results in a memory buffer allocation that can hold the blocking rate worth of values as they are received. When the buffer is full, the array is processed and the data is stored. For all collection types, blocking rates affect the rate that numeric data is stored to the IM disks. Blocking rate has an additional meaning for connectivity servers that have Trend Logs. When looking at an entire configuration, the blocking rates and the number of logs defined represent a rate that HDA requests are made toward connectivity servers in the system. This rate should be tuned each time the configuration is modified using the Stagger Collection and Storage on page 472.

Sample blocking rate is based on the storage interval of the log and is set to an appropriate value for most configurations. After new logs are created, the stagger functionality should be used to adjust stagger and phasing for the entire IM server. It is not recommended to attempt to adjust stagger and phasing on a per template or per log basis. It is recommended that all IM severs run hsDBMaint, the stagger after any configuration changes. Refer to Stagger Collection and Storage on page 472.

Sample blocking rate is NOT used under the following conditions:

- if the storage interval is Monthly.
- if the Collection Type is USER_SUPPLIED. In this case the values are stored as they are sent to History.

Storage Type

Log entries can be stored in Oracle tables or in files maintained by History. File storage is faster and uses less disk space than Oracle storage. File-based storage is only applicable for synchronous property logs. The default storage type is file-based TYPE5 which supports variable size based on the type of data being collected. ORACLE is mandatory for Asynchronous logs. This type supports collection for floating point data only. Other storage type are available. For further information refer to File Storage vs. Oracle Tables on page 255.

Entry Type

For TYPE5 Storage Type the Entry Type can be specified to conserve the disk space required per entry. The Entry Type attribute specifies whether or not to save the original value with the modified value when a client application is used to modify a stored history value.

Disk space is allocated for both the original and modified value whether or not the value is actually modified. Therefore, only choose to save the original value when the modify history value function is being used and original values need to be saved.

The choices are:

- **Do Not Save Original Value** - If a log entry is modified, the original value is not saved.
- **Save Original Value** - If a log entry is modified, the original value is saved along with the modified value. This allocates disk space for both values, whether or not the value is actually modified.
- **Not Applicable** - For storage types other than TYPE5.

Log Capacity

This is the maximum number of entries for a log. If a log is full, new entries replace the oldest entries.

For asynchronous collection, the log capacity is configurable. Enter an integer value. The maximum log capacity is different for each Storage Type as indicated in Table 24.

For synchronous collection, this field is not configurable. Log capacity is calculated as follows:

Log Capacity = (Log Period/ Storage Interval).

If the specified log period or storage interval result in a calculated log capacity that exceeds the maximum log capacity (Table 24), an error message will be generated and either the log period or storage interval must be adjusted. For further information regarding these log attributes, refer to:

- Log Period on page 233.
- Storage Interval on page 234.

Table 24. Maximum Log Capacity for Storage Types

Storage Type	Maximum Log Capacity
TYPE1	5,483,352
TYPE2	32,638,500
TYPE3	16,449,804
ORACLE	50,000
TYPE4	The maximum file size is 65278*32K + 257*2K (slightly less than 2G) for the 32K data blocks or 65278*1M + 257*2K (slightly less than 64GB) for the 1meg datablocks.
TYPE5	315,000,000(1) when Entry Type is Do Not Save Original Value 267,255,200 when Entry Type is Save Original Value

(1) This is equivalent to 520 weeks of data collected at one-second storage rate.

Calculation Algorithm

Although both trend and history logs can perform calculations on collected data prior to storage, it is a good practice to collect and store raw data, and then perform the required calculations using the data retrieval tool, for example, DataDirect. To do this, use the STORE_AS_IS calculation algorithm, Figure 150.

This calculation causes data to be stored only when it is received from the data source. For OPC type logs, this calculation can use the OPC server's exception-based reporting as a deadband filter. This effectively increases the log period so the

same number of samples stored (as determined by the log capacity) cover a longer period of time. Refer to STORE AS IS Calculation on page 238 for further details.

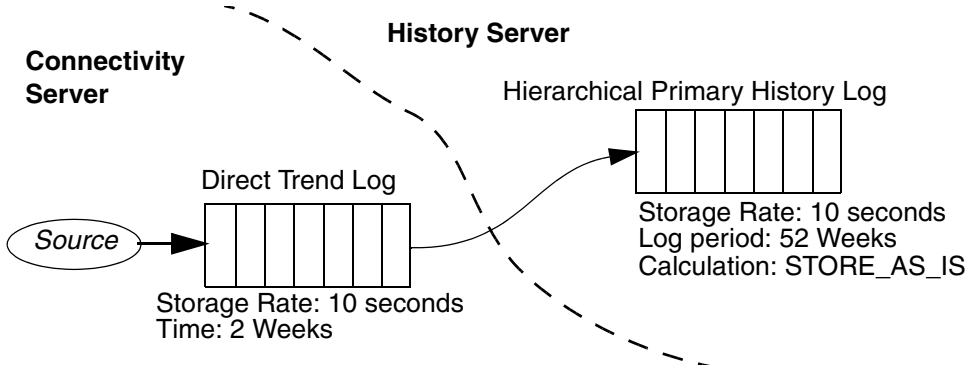


Figure 150. Property Log Configuration

Other calculations may be implemented for certain property log applications. These are described in Other Calculation Algorithms on page 240.

STORE AS IS Calculation

This calculation causes data to be stored only when it is received from the data source. This functionality has three different applications, depending on whether it is applied to an OPC type log, or and API (OMF) type log.

Collecting from Operator Trend Logs (OPC Data)

STORE_AS_IS is an option in the Calculation Algorithm section on the Data Collection tab of the Log Configuration aspect. For OPC type logs, this calculation uses the OPC server's exception-based reporting as a deadband filter. When collecting from an OPC/DA source, OPC returns data at the subscription rate, only if the data is changing. This effectively increases the log period so the same number of samples stored (as determined by the log capacity) cover a longer period of time.

For all calculations other than STORE_AS_IS, if data is not changing and samples are not being received, History will insert previous values with the appropriate time stamps. This is required for calculations to occur normally.

With STORE_AS_IS, no other calculations are performed (including INSTANTANEOUS); therefore, data does not need to be stored at the specified

sample rate. Samples will only be recorded when a tag's value changes, or at some other minimum interval based on the log configuration.

Data Forwarding of Deadbanded Data to a Consolidation Node

STORE_AS_IS can implement data compaction on the collection node, before data forwarding to the consolidation node. This saves disk space on the collection node.

When data forwarding deadbanded data to a consolidation node, be sure to use STORE_AS_IS rather than INSTANTANEOUS as the calculation algorithm for the hierarchical log on a consolidation node. This eliminates insertion of NO_DATA entries for missing samples.

Time Drift in Subscription Data

When running synchronous logs (API or USER_SUPPLIED) at a fast rate, it is possible for time stamps to drift over time, eventually forcing a sample interval to be missed. For all calculations other than STORE_AS_IS, History adds an entry with a status of NO_DATA for the missing sample interval. This is a false indication of missing data, and can cause a spike or drop in a trend display (refer to Figure 149). By using STORE_AS_IS, rather than INSTANTANEOUS, these spikes will not occur.

The STORE AS IS calculation is generally applied to the primary (first) hierarchical log. The log where the STORE AS IS calculation is applied can be the data source for a hierarchical log, if the hierarchical log is also STORE AS IS (for example data forwarding to a consolidation node). Do not use the STORE AS IS calculation when further calculations are required.

Deadband Storage Interval

In the case where a process is very stable and values are not changing, there may be very long periods where no samples are stored. This may result in holes in the trend display where no data is available to build the trends. Use the deadband storage interval on the **Deadband Attributes** tab to force a value to be stored periodically, even if the value does not change. This ensures that samples are stored on a regular basis. Refer to Deadband Compaction % on page 245.

Shutdown and Startup

Since data is not being stored if it is not changing, History will insert the current value and current timestamp when it is shutdown. This guarantees that the data can be interpolated correctly up until the time History was shutdown.

The following is applicable for API (OMF-based), but NOT for operator trend logs or user-supplied logs.

At startup, NO_DATA entries will be inserted for STORE_AS_IS type logs to show that the node was down, and that data was not collected during that time. This will allow interpolation algorithms to work correctly, and not presume that the data was OK during the time the node was down.

Other Calculation Algorithms

The following calculations may be implemented for history logs:

AVERAGE
MAXIMUM
MINIMUM
INSTANTANEOUS - no calculation
SUM
SUM_OF_ SQRS
STANDARD DEVIATION
NUM_OF_ VALUES

Calculations can be done on data before it is stored in a hierarchical log, Figure 151. This allows storage of a single value to represent a larger time span of values, or key characteristics of the data. For example, to sample a variable every minute and put those values into a trend log. Then, the hourly average, minimum, and maximum on data in the trend log can be calculated and values stored in hierarchical history logs.

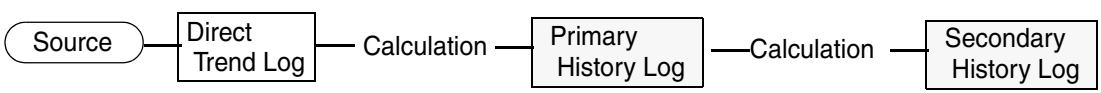


Figure 151. Calculations

If a calculation algorithm other than INSTANTANEOUS is specified for the log, the algorithm is performed at each storage interval. The storage interval must be a

multiple of the sample interval. For example, if the sample interval is 5 Seconds, the storage interval can be 10, 15, 20, or 25 Seconds, and so on.

When the calculation is performed, the specified number of data samples are entered in the algorithm, and the result is passed to the deadband function. If the calculation algorithm is INSTANTANEOUS (no calculation), the sample and storage intervals must be equal, and each data sample is treated as a result.

Start Time

This is the earliest allowable system time that the log can become active. If the specified start time has already past, the log will begin to store data at the next storage interval. If the start time is in the future, the log will go to PENDING state until the start time arrives. Enter the start time in the following format:

1/1/1990 12:00:00 AM(default)

The order of the month, day, and year and the abbreviation of the month is standardized based upon the language being used.

Use the start time to distribute the load. Both the blocking rate and start time may be defined initially when configuring the History database. If the attributes are not configured initially, or if adjustments are required, use the Stagger Collection function in the hsDBMaint utility as a convenient means for adjusting these parameters. Refer to Stagger Collection and Storage on page 472.

It is recommended NOT to use small differences (less than one minute) to distribute the load. Managing log collection at slightly different times will result in significant overhead.

The time when a log is scheduled to start collecting data is referred to as the log's *alignment time*. Alignment time is the intersection of the log's start time and its storage interval. For instance if the start time is 7/3 2003 10:00:00 AM, and the storage interval is 15 Seconds, the log will store data at 00, 15, 30, and 45 seconds of each minute starting at 10:00:00 AM on July 3, 2003. In this case, this is the alignment time.

If the start time for a log has already passed, it will be aligned based on the storage interval. For example, a log with a storage interval of 1 Minute will store data at the beginning of each minute. So its alignment time is the beginning of the next minute.

The log stays in PENDING state until its alignment time. For the log whose storage interval is 1 minute, it will be PENDING until the start of the next minute.

The default is: **1/1/1990 12:00:00 AM**. The order of the month, day, and year and the abbreviation of the month is standardized based upon the language being used.

Bad Data Quality Limit

When the percentage of samples with bad data quality or no data status used in a calculation exceeds the Bad Data Quality Limit, the data quality of the calculated result is bad. For example, 20% means that 2 out of 10 data samples can be bad.

This attribute can be used to avoid having data stored with BAD_DATA quality. When sampling remote objects, the interval between values may vary depending upon the load in the source node. In this case the sampled data will have status BAD_DATA. Use of this attribute allows some of the sampled values to be bad and still get a stored value with ok status.

For example, set sample interval to **3 Seconds**, storage interval to **30 Seconds** and make the calculation **Average**. Set Bad Data Quality to **30**%. In this case, at least three of the ten values must have bad status in order for the calculation result to be bad. Enter the bad data quality limit as a percent (**10.500** = 10.5%). The default is 0.

Deadband Attributes

The Deadband attributes support data compaction. The recommended method for achieving data compaction is to use the STORE_AS_IS calculation algorithm (STORE AS IS Calculation on page 238). The attributes on the Deadband tab may be used in certain circumstances, for example to support upgrades from earlier installations that used deadband, or when configuring calculation algorithms such as minimum, maximum, and average for history logs.

Deadband compaction has two advantages:

- It reduces the load on the system by filtering out values that do not have to be stored since there is not a significant change in the value.
- It allows data to be stored online for a longer period of time since the data is not replaced (overwritten) as quickly.

Compaction is specified on an individual log basis. It may be used for some logs and not for others. Compaction should not be used indiscriminately in all cases.

- Use compaction if an accurate representation of the process over time is needed, and it is not required that every data sample be stored.
- Do not use compaction if the application requires that every data sample be stored as an instance in the log.

Deadband can be used on digital data sources so that only state changes are saved.

The Deadband filters out insignificant changes in the process variable.The deadband defines the minimum change in value or in rate of change for the change to be significant. The deadband is defined as a percent of the engineering units range.

For example, if a process variable is scaled from 200 to 500, the range is 300. If the deadband is 2%, the minimum significant change is 6 (2% of 300). With these values, if the last stored sample was 400, and the next sample is 397, then 397 is within the deadband and will not be stored. If the next sample is 406.2, it will be stored. A small deadband results in more stored samples than a large deadband.

Deadband is activated by configuring a compaction deadband greater than zero. When deadband is active, the deadband algorithm determines whether the calculation result, or raw value if calculation algorithm is instantaneous, is within or outside the deadband. If the most recent result is outside the deadband, the previous result is stored. If the most recent result is within the deadband, it is held until the next result is calculated, and the previous result is discarded. If the compaction deadband is zero, so compaction not activated, all results are stored.

A deadband storage interval is defined on an individual basis for each log. The deadband storage interval is a multiple of the log's storage interval. It is the maximum time that can elapse without storing a sample when compaction is activated. Even if the process is so stable that there are no samples outside the deadband, samples will still be stored at the deadband storage interval. The deadband storage interval is always measured from the time the last sample was stored. The deadband storage interval should be at least 10 times larger than the storage interval to ensure a high compaction ratio.

When deadband is active, a sample may be stored for any one of the following reasons:

- sample outside the deadband.

- value status changed (possible statuses are: no data, bad data, good data).
- deadband storage interval elapsed since last stored sample.

Figure 152 shows how samples are stored when deadband is active. The deadband function holds each sample until it determines whether or not the sample is required to plot the trend. Samples that are not required are discarded.

A sample is stored when deadband is activated at 0s. The sample at 10s is held until the next sample at 20s. Since there is no change in slope between 0s and 20s, the 10s sample is discarded and the 20s sample is held. This pattern is repeated at each 10-second interval until a change in slope is detected at 50s. When this occurs, the samples at 40s and 50s are stored to plot the change. At 90s another change in slope is detected, so the samples at 80s and 90s are stored.

Even though there are spikes between 80s and 90s, these values were not sampled so the slope remains constant (as indicated by the dotted line). Another change in slope is detected at 100s. The sample at 100s is stored to plot the slope. The sample at 90s, also used to plot the slope, was already stored. The slope remains constant thereafter. The next sample stored is 140s since the Deadband Storage Interval (40s) elapses between 100s and 140s. The sample at 180s is also stored due to the Deadband Storage Interval.

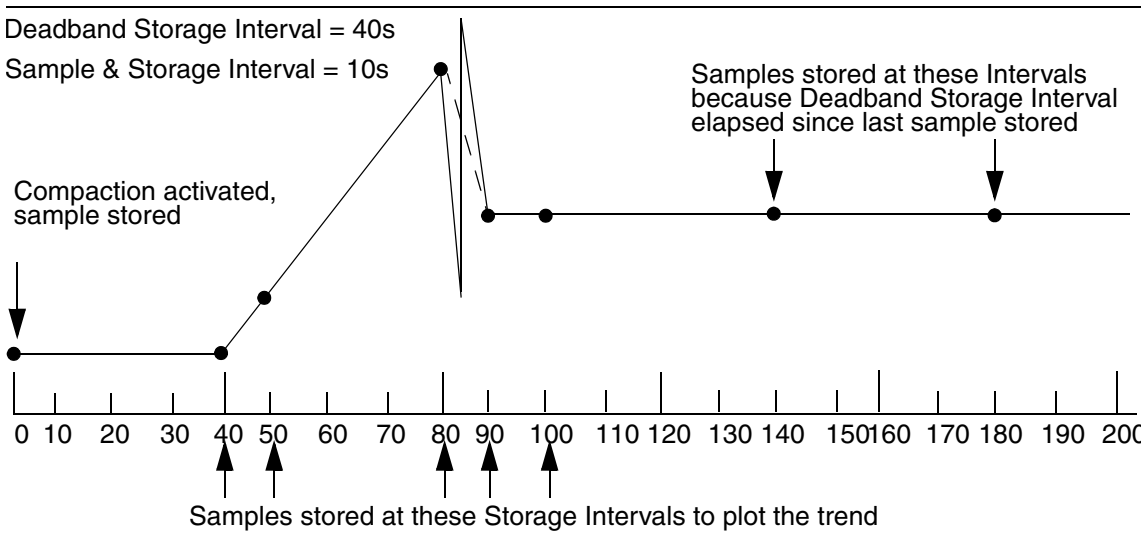


Figure 152. Data Compaction Example

Compaction effectively increases the log period so the same number of samples stored (as determined by the log capacity) cover a longer period of time. If the process is very unstable such that every sample is stored and there is no data compaction, the effective log period will be equal to the configured log period. If the process is perfectly stable (maximum data compaction, samples stored only at deadband storage interval), the effective log period is extended according to the following formula:

effective log period = (deadband storage interval/storage interval) * configured log period

In most cases, process variation is not at either extreme, so the effective log period will fall somewhere in between the configured (minimum) and maximum value.

The calculated effective log period is stored in the EST_LOG_TIME_PER attribute. This attribute is updated periodically according to a schedule which repeats as often as every five days. If necessary, manually execute the update function via the hsDB-Maint utility. This is described in Update Deadband Ratio Online on page 463.

When requesting by name, EST_LOG_TIME_PER determines which log to retrieve data from (seamless retrieval). For instance, consider a case where the trend log stores data at 30 second intervals over a one-day time period, and the history log stores data at one-hour intervals over a one-week time period. The data compaction ratio is 5:1 which extends the effective log period of the trend log to five days. Making a request to retrieve three-day-old data causes the History retrieval function to get the data from the primary log rather than the hierarchical log, even through the configured log period of the primary log is only one day.

When EST_LOG_TIME_PER is updated, both internal History memory and the history database are updated with the new values of EST_LOG_TIME_PER and COMPACTION_RATIO. This will ensure the information will be available after a re-start of the History node.

Before using compaction, refer to examples 2 & 5 in Property Log Configuration Reference on page 266.

Deadband Compaction %

This is the minimum change between a sample and previously stored value that causes the new sample to be stored. Enter this as a percent of engineering units range (Range Max. - Range Min.). For example, 1 means 1% of range.

Deadband Storage Interval

When deadband is active, this is the maximum time elapsed between storing entries. This assures that some data is stored, even if the process is stable.

Enter deadband storage interval as an Integer value with a case time unit indicator: **Weeks**, **Days**, **Hours**, **Minutes**, **Seconds**. It must be a multiple of the storage interval.

The deadband storage interval defaults to ten times the storage interval. When the storage interval is less than or equal to 10m, the deadband storage interval should be much greater than ten times the storage interval. For example:

- If the storage interval = 5 Seconds, make the deadband storage interval at least 1 Hour.
- If the storage interval = 10 Minutes, make the deadband storage interval at least 8 Hours.

When the storage interval is greater than 10m, reduce the deadband storage interval to have less time elapse without storing a value.

Estimated Period

This attribute indicates the time between the oldest and newest stored values. This is filled in by running hsDBMaint. If necessary, use the History Utility to manually update this attribute. Refer to Update Deadband Ratio Online on page 463.

Compaction Ratio

This is a read-only attribute that indicates the number of entries sampled for each entry stored. It is filled in by running hsDBMaint. Refer to Update Deadband Ratio Online on page 463.

Expected Ratio

This attribute indicates the expected ratio for number of entries sampled/each entry stored.

IM Imported Tab

This tab, Figure 153, is provided for log configurations that have been imported from other systems (Importing Remote Log Configurations on page 369).

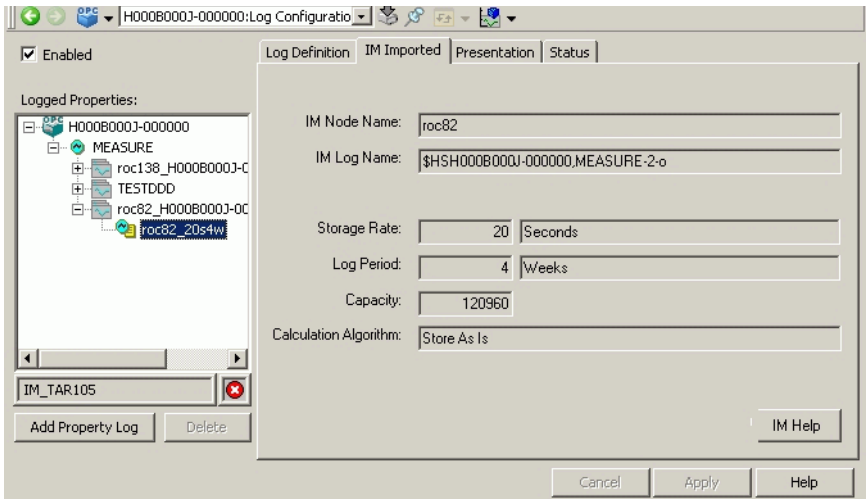


Figure 153. IM Imported Tab

This tab indicates:

- The name of the node from which the configurations were imported.
- Log Name.
- Storage Interval.
- Log Period.
- Log Capacity.
- Calculation Algorithm.

Data Collection Examples

These examples further illustrate common data collection applications:

- Example 1 - Storing Instantaneous Values, No Compaction.
- Example 2 - Storing Instantaneous Values Using Compaction.
- Example 3 - Storing Calculated Values.
- Example 4 - Storing Calculated Values in Logs.
- Example 5- Storing Calculated Values with Compaction.

Example 1 - Storing Instantaneous Values, No Compaction

When the calculation algorithm is INSTANTANEOUS (no calculation), the sample and storage interval must be the same. If no compaction is specified, a data instance is stored every storage interval. Figure 154 shows the relation between these attributes of the Primary Log. In this case a sample is stored every five seconds. The blocking rate is 60 seconds. The data samples to be stored are held in a buffer, and then written to disk in blocks of 12 samples at each 60-second interval. While the data is held in the buffer, it is not available for data retrieval.

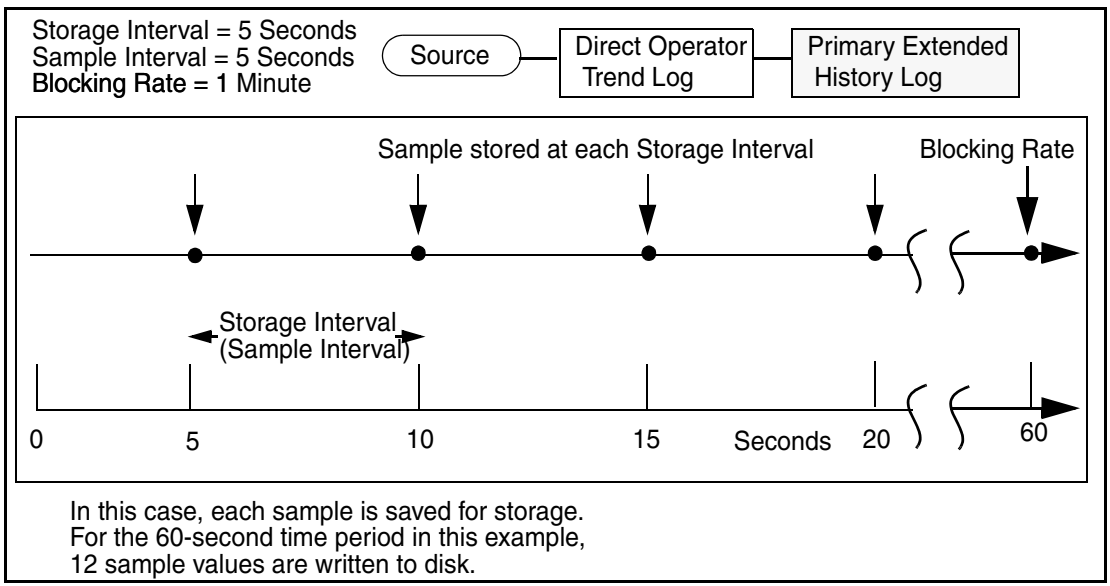


Figure 154. Storing Instantaneous Values, No Compaction

Example 2 - Storing Instantaneous Values Using Compaction

In this example, the process variable is scaled from 200 to 500, the range is 300. The deadband is 2%, the minimum significant change is 6 (2% of 300). Again, since there is no calculation, the sample and storage interval must be the same. At each storage interval the data is checked by the deadband function. If a significant change in slope is detected, the sample is stored. Samples are discarded as the slope remains within the deadband. However, if the deadband storage interval (in this example, 25s) elapses since the last stored sample, another sample is stored on the deadband

storage interval whether or not a change in slope is detected. In Figure 155, the data is sampled every five seconds. Of the 17 samples taken, only 7 are stored. A summary is provided in Table 25 below.

Table 25. Storing Instantaneous Values w/Compaction

Interval	Value	Stored	Reason
0	406.7	Yes	Deadband activated, first sample stored
5	404.5	No	No significant change in slope between 5s & 10s
10	406.2	Yes	Change in slope, next sample outside deadband
15	399.8.	No	No significant change in slope between 15s & 20s
20	394.9	Yes	Change in slope from 20 to 25, sample @ 25 outside deadband
25	409.0	Yes	Change in slope from 25 to 30, sample @ 30 outside deadband
30	404.2	Yes	Change in slope from 30 to 35, sample @ 35 outside deadband
35	404.0	No	No significant change in slope between 35s & 40s
40	402.8	No	No significant change in slope between 40s & 45s
45	403.3	No	No significant change in slope between 45s & 50s
50	403.2	No	No significant change in slope between 50s & 55s
55	404.1	Yes	Deadband storage interval (25s) elapsed since last stored sample (at 30s)
60	404.1	No	No significant change in slope between 55s & 60s
65	403.8	No	No significant change in slope between 60s & 65s
70	404.1	No	No significant change in slope between 605 & 70s
75	402.9	No	No significant change in slope between 70s & 75s
80	405.0	Yes	Deadband storage interval elapsed since last stored sample

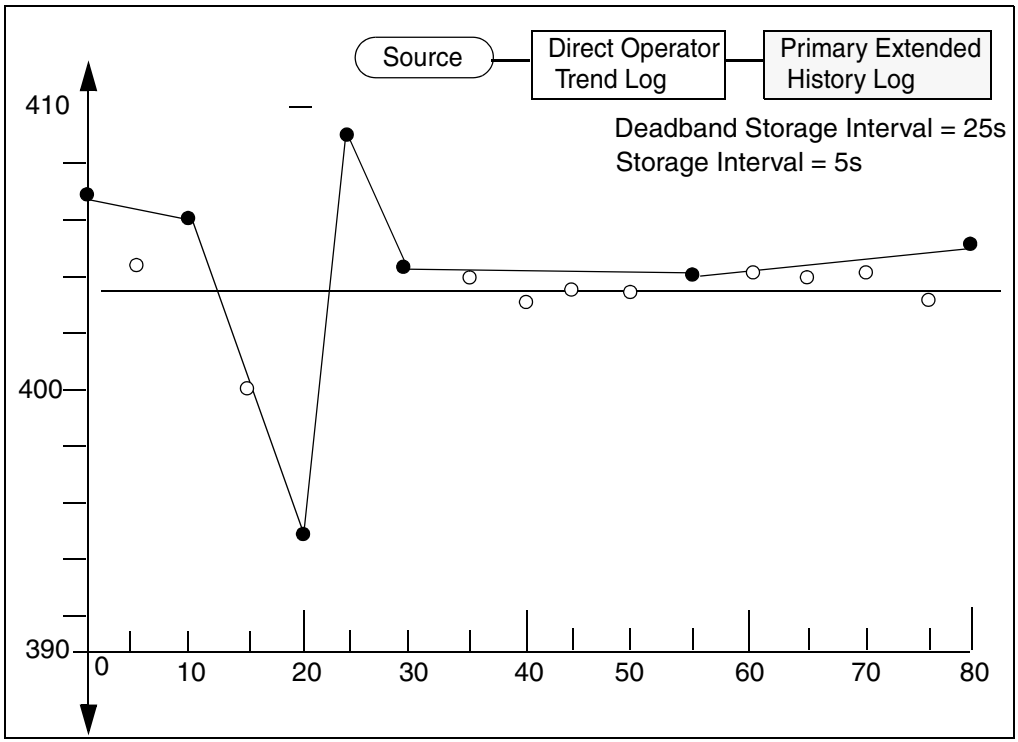


Figure 155. Storing Instantaneous Values w/Data Compaction

Example 3 - Storing Calculated Values

This example shows how to do a calculation before storing a value in a primary log, Figure 156. In this example, an average is calculated over an eight-hour shift.

The storage and sample intervals determine how many samples to use in the calculation (storage interval/sample interval). The storage interval must be an integer multiple of the sample interval. In this case, eight samples are used in each calculation.

Eight samples at hours 1,2,3,4,5,6,7, and 8 are averaged and the average is stored in the primary log at hour 8 (storage interval). Similarly, eight samples for two other sample groups are averaged and stored. The results of these three average

calculations are written to disk at 24 hours (blocking rate). The process is repeated at 48 hours and averages are stored at 32, 40, and 48 hour.

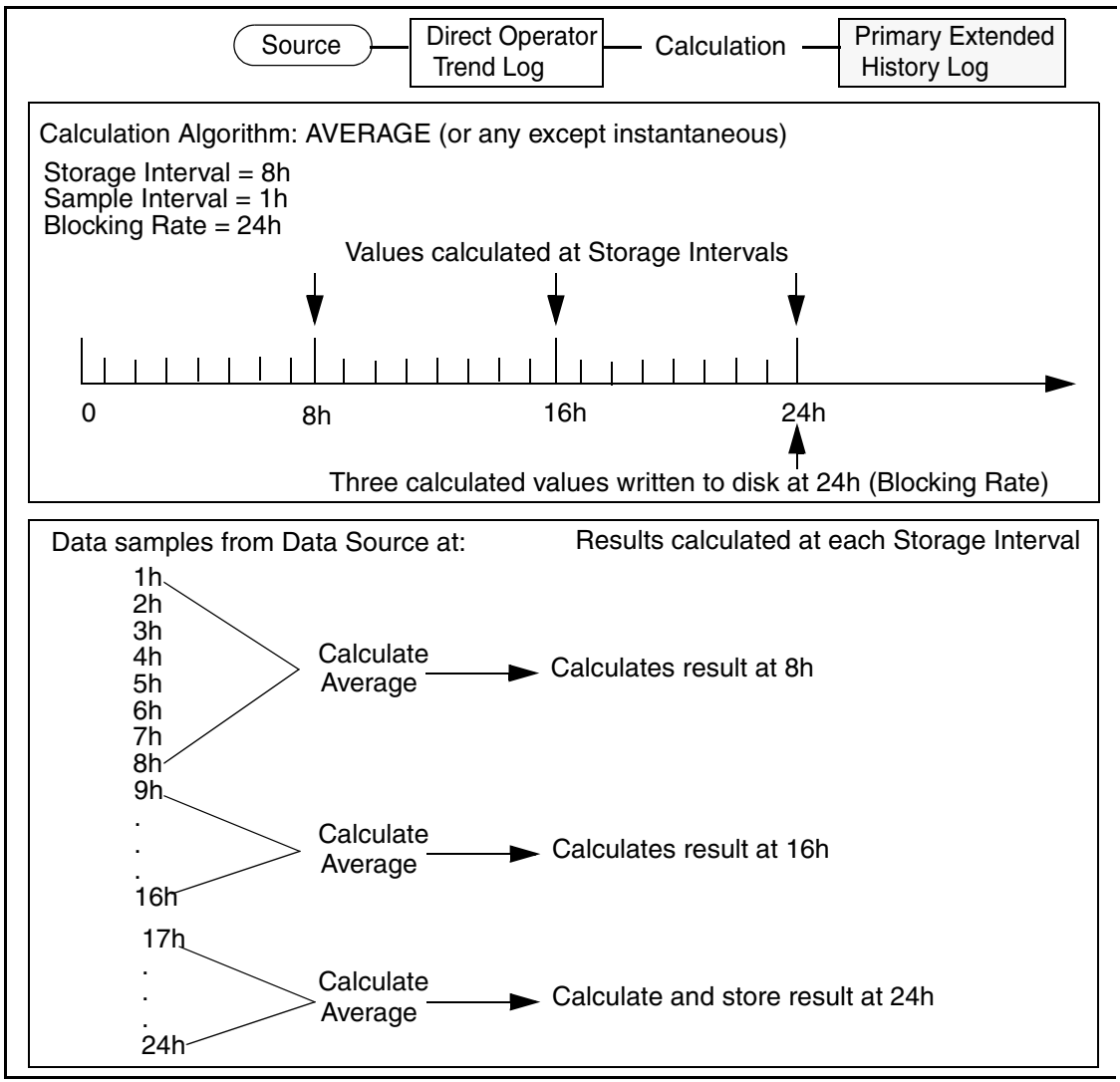


Figure 156. Primary Log Example: Calculation being Used

Example 4 - Storing Calculated Values in Logs

This example shows how to do a calculation before storing a value in a secondary log. The primary log calculates the shift average as described in Example 3, and shown in Figure 156. The secondary log then calculates a daily average as illustrated in Figure 157.

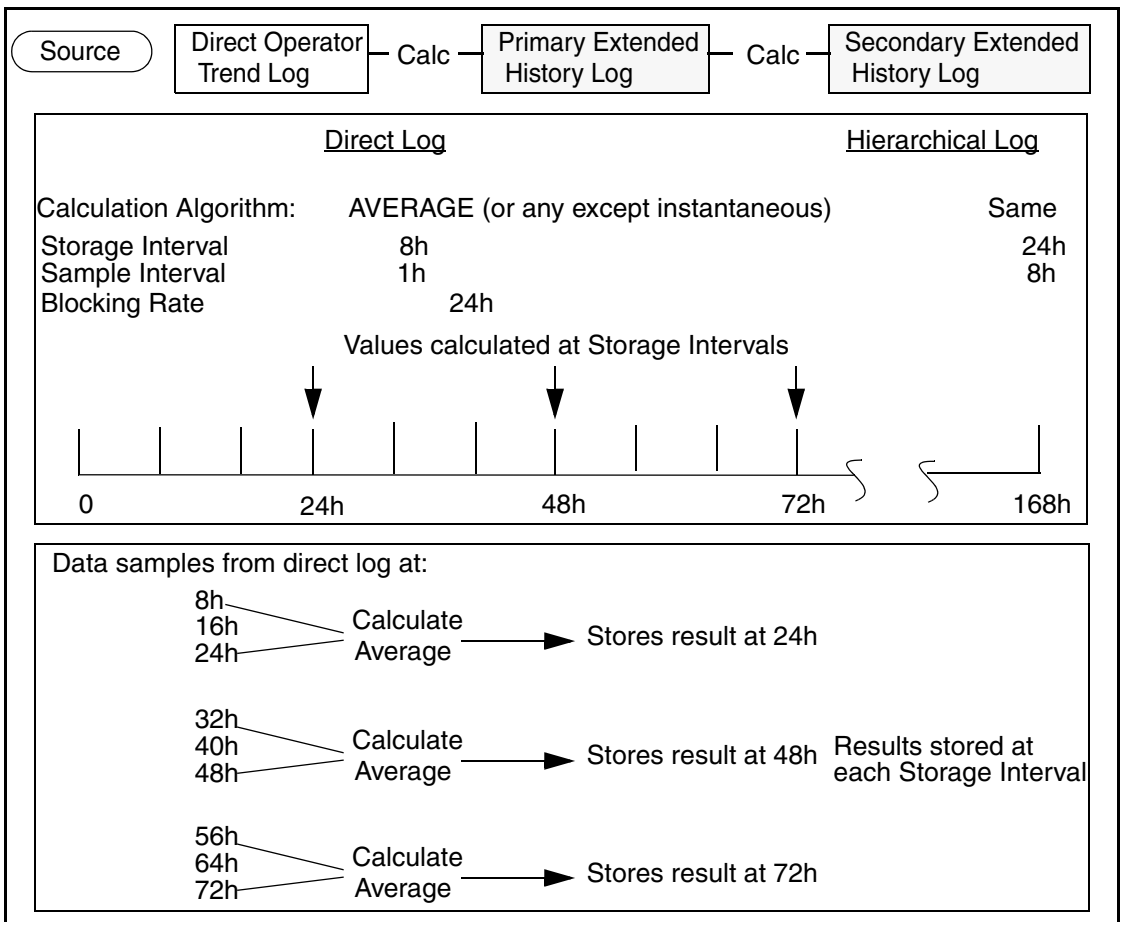


Figure 157. Hierarchical Log Example: Calculation being Used

The sample blocking rate is used for primary history logs that are connected directly to the direct trend log. Subsequent (secondary) history logs do not use the sample blocking rate. The sample blocking rate from the primary log determines when values are stored for all history logs in the property log.

Example 5- Storing Calculated Values with Compaction

This example uses the same data as Example 2 - Storing Instantaneous Values Using Compaction. The process variable is scaled from 200 to 500, the range is 300. The deadband is 2%, the minimum significant change is 6 (2% of 300). Since a calculation is being performed at each storage interval, the storage interval must be an integer multiple of the sample interval. In this case, data samples are collected at 5-second intervals, and four samples are used to calculate an average at 20-second intervals. At each storage interval the calculated value is checked by the deadband function. If a significant change in slope is detected, the calculated value is stored. Calculated values are discarded as the slope remains within the deadband. However, if the deadband storage interval (in this example, 60s) elapses since the last stored sample, another sample is stored at that time whether or not a change in slope is detected. In Figure 158, calculations are performed at 20-second intervals. Of the 4 calculations performed, only 2 are stored. A summary is provided in Table 26. Notice the contrast in the trends between Figure 155 and Figure 158, even though the same data is used to plot the trends.

Table 26. Storing Calculated values w/Compaction

Interval	Value	Calculated Average	Stored	Reason
0	406.7	404.3	Yes	First calculated average
5	404.5			
10	406.2			
15	399.8.			
20	394.9	403.0	No	No significant change in slope between 1st & 2nd calculation
25	409.0			
30	404.2			
35	404.0			

Table 26. Storing Calculated values w/Compaction (Continued)

Interval	Value	Calculated Average	Stored	Reason
40	402.8	403.4	No	No significant change in slope between 2nd & 3rd calculation
45	403.3			
50	403.2			
55	404.1			
60	404.1	403.7	Yes	Deadband storage interval (60s in this example) elapsed since last stored sample (0s)
65	403.8			
70	404.1			
75	402.9			

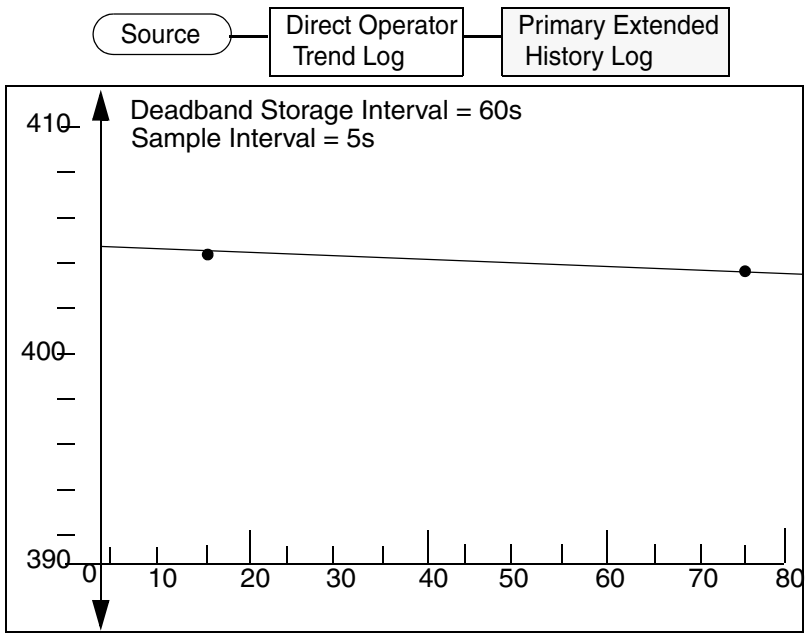


Figure 158. Storing Calculated Values w/Data Compaction

File Storage vs. Oracle Tables

Property log entries can be stored in Oracle tables or in files maintained by History. File storage is faster and uses less disk space than Oracle storage. Whether entries are stored in Oracle tables or in files maintained by History is determined by the STORAGE_TYPE log attribute:

- TYPE5 (default) - file storage variable size based on the collected data type.

This storage type is recommended for all history logs that collect from operator trend logs.

- ORACLE - storage in Oracle tables @ 100 bytes per entry.
- TYPE1 - file storage with full status information @ 24 bytes per entry.
- TYPE2 - file storage with partial status information @ 4 bytes per entry.
- TYPE3 - File storage, 8 bytes per entry. Identical to TYPE2, except that an additional 4 bytes are required per entry for precision time stamps.
- TYPE4 - Profile Log.

Refer to the Comparison of File-based Logs below to determine the type which is appropriate for the application. For disk space requirements, refer to:

- Disk Requirements for Oracle-based Logs on page 257.
- Disk Requirements for File-based Property Logs on page 258.
- Disk Requirements for Profile Logs on page 259.

Comparison of File-based Logs

The default storage type for property logs is TYPE5. This storage type maintains the data type of the collected data, when the STORE AS IS Calculation is used. For all other storage types (and for TYPE5 when any calculation other than STORE_AS_IS is applied), the collected data is converted to floating point.

TYPE1 uses 24 bytes per entry and provides complete status and related information for each entry, as well as full precision (microsecond resolution) timestamp. TYPE2 (4-byte) entries require less storage space; however, some status information is sacrificed, as well as time stamp accuracy (one-second resolution). Also, for TYPE2 entries, if a value is modified, the original value is not saved, and no status bits are set to indicate that the value has been modified.

Even though TYPE2 storage provides a reduction in disk usage, it may not be applicable in all cases. TYPE2 storage is recommended when:

- The storage interval is 1 hour or less and is divisible into 1 hour, and the data source is TTD. TTDs, do not have object status, and times are always aligned.
- The storage interval is one minute or less when collecting from process objects. A time stamp is recorded every 500th storage. During the interim, values are assumed to be exactly one storage interval apart (as noted below). Of course larger storage intervals can be configured, but they should be confirmed as acceptable. Since TYPE2 logs only correct their alignment after the 500th storage, they should only be used on *fast* logs. For example, a one-day storage log, if not properly aligned, will take 500 days for the alignment to correct.

TYPE2 logs should not be used for logs with storage intervals greater than one hour. Use TYPE3 logs instead.

TYPE3 (8-byte) is similar to TYPE2. TYPE3 provides full precision time stamp (like TYPE1), and stores a status when a value is modified. TYPE3 does not store the original value when the value is modified. TYPE3 logs grow to 16-bytes when the storage interval is greater than 30 minutes.

The complete information provided for TYPE1 entries is described below. The differences between TYPE1 and other types are described where applicable.

Time Stamp

Each data entry in a log is time stamped with Universal Time (UTC) Coordinate (Greenwich Mean Time). It is not affected by the time zone, nor daylight saving time. This gives consistency throughout all external changes such as Daylight Saving Time and any system resets that affect the Local time.

TYPE1, TYPE3, TYPE5, and Oracle-based logs support microsecond resolution for timestamps. For TYPE2 logs (file-based 4-byte entries), the time is an estimate of the actual time for the entry. The time for the entries are all defined to be a storage interval apart. A reference time is stored for each block of 500 values. The time for subsequent values after the reference time = Reference time + (*n* * storage interval) where *n* is the sequential number of the value in the block (0 to 499). So if a log is configured to store a value every 15 seconds, the entries will have timestamps at 0, 15, 30, 45s of each minute.

Data Value

The data value format for all types is floating point. If the data being collected is not floating point (for example, integer), the data will be converted to floating type and then stored in that format. This may result in some decimal places being truncated when the data is stored.

The only exception is when TYPE5 is used in combination with the STORE AS IS Calculation. In this case the TYPE5 log maintains the data type of the collected data.

Status

Status information is stored according to OPC DA specifications.

Object Status

Object status is the status of the data source when the value was sampled. This field is not used when the data source is TTD. The format of the status (what is represented and how) is dependent upon the data source object type. When a hierarchical log is created, the object status is obtained from the last primary log entry used in the calculation.

If a User Object that inherits a Basic Object is the source for a log, its status will also be collected and stored.

Object status is not stored for TYPE2 or TYPE3 logs.

Disk Requirements for Oracle-based Logs

Each property Oracle-based log requires approximately 100 bytes per entry:

- 50 bytes for the Inform_HS_RunTime tablespace.
- 50 bytes for the HS_Indexes tablespace for logs whose capacity is > 5000 entries, or 20 bytes for the HS_Indexes tablespace for logs whose capacity is < 5000 entries.

After calculating the requirements for Oracle-based logs, add about 10% to 15% for overhead.

OPC Message Log

Each OPC/AE message entry can range in size from 2k to 6k bytes. The size is dependent on the strings in the message. The OPC message log can hold 4K of text per message and each attribute for a single message can hold a 4K string. It is not typical for the messages to have such large strings. When calculating the required space for a message log, use these metrics based on capacity of the message log.

Table 27. Message Entry Range

Capacity Range	Average Per Message	Average in HS_RUNTIME	Average in HS_INDEXES
10K to 1 million	2500	2000	500
1Million to 4 Million	3000	1000	2000
4 Million to 8 million	3500	2000	1500
8 million to 12 million	4000	2000	2000

Disk Requirements for File-based Property Logs

For file-based logs, the actual disk space may be greater than the calculated capacity if the log is very small. For TYPE1 (24-byte) logs, the minimum capacity allocated on disk is 84 entries. For TYPE2 (4-byte) logs, the minimum space allocated on disk is 500 entries.

Each TYPE1 log requires about 24 bytes per entry. To calculate exact file size for TYPE1:

#data_blocks = (#entries + 83)/84
#index_blocks = (#data_blocks + 253)/254
If (#index_blocks > 3), Then #map_blocks = 1, Else #map_blocks = 0
file size = (#data_blocks + #index_blocks + #map_blocks) *2048

Each TYPE2 log requires about 4 bytes per entry. To calculate exact file size for TYPE2:

#data_blocks = 1+ (#entries + 499)/500
#index_blocks = (#data_blocks + 253)/254

If (#index_blocks > 3), Then #map_blocks = 1, Else #map_blocks = 0
file size = (#data_blocks + #index_blocks + #map_blocks) *2048

Each TYPE3 log requires about 8 bytes per entry (or 16 bytes if storage interval > 30 minutes). To calculate exact file size for TYPE2:

#data_blocks = 1+ (#entries + 251)/252 (when storage interval < 30 minutes)
OR
#data_blocks = 1+ (#entries + 125)/126 (when storage interval > 30 minutes)
#index_blocks = (#data_blocks + 253)/254
If (#index_blocks > 3), Then #map_blocks = 1, Else #map_blocks = 0
file size = (#data_blocks + #index_blocks + #map_blocks) *2048

For file-based storage, Windows allocates 5% overhead per file.

Disk Requirements for Profile Logs

The following information is needed to calculate the size of a profile file:

- LogPeriod: Expected duration of the log, 4w, 10w, etc.
- Storage Rate (from OPC data source, 30s, 1m, etc.)
- Array Size: Maximum number of samples per scan (max allowed 32768).

Based on the storage Rate and Log Period, calculate the capacity as follows:

LogPeriod/StorageRate = Capacity.

Using Capacity and ArraySize, the file size can be calculated with the following equation:

```
if arraySize is odd, adjustedArray = arraySize+1 else adjustedArray =
arraySize;
if (arraySize > 2000)
{
    #data_blocks =
    (1+(capacity*((adjustedArray+2)*4+32)+1048543)/1048544);
    block_size = 1048544
}
else
```

```
{
    #data_blocks = (1+(capacity*((adjustedArray+2)*4+32)+32735)/32736);
    block_size = 32736

    if (#data_blocks > 65278 )
    {
        #data_blocks =
        (1+(capacity*((adjustedArray+2)*4+32)+1048543)/1048544);
        block_size = 1048544
    }
}
#indexBlocks = (#dataBlocks+253)/254;
if #indexBlocks > 3, then #map_blocks = 1 else #map_blocks = 0;
fileSize = (#map_blocks + #index_blocks)*2048 + #data_blocks*block_size
```

#data_blocks cannot exceed 65278. For the other storage types, this means a known maximum limit. However, since the arraySize is variable, the maximum capacity is based on the arraySize. The maximum file size is 65278*32K + 257*2K (slightly less than 2G) for the 32K data blocks or 65278*1M + 257*2K (slightly less than 64GB) for the 1meg datablocks.

For a 600 value profile, the maximum capacity would be: (65278*32768)/((600+2)*4+32) which is approximately 875000 values. This is about 300 days worth of data or more.

Additional duration of data can be stored because the start and ends of the array typically do not have valid data points. Invalid values are not stored. Therefore, values 1-5 and 590-600 were always bad, 15*4 bytes are not stored in the file, saving that space for more entries. Additionally, if the not array values are passed back, only a timestamp is stored, saving 600*4 bytes per entry. If the machine only runs 10 percent of every day, then only 10% of the data to be stored is stored.

Presentation and Status Functions

The log configuration aspect provides two tabs which are not available on the log template view. These are:

- Presentation for formatting history presentation on trend display.
- Status for viewing log data directly from the log configuration aspect.

Other sources of status information are the Property Log Tab, and Status Light.

Presentation

Configure presentation attributes for the Trend Display on this tab, Figure 159. The default settings for these attributes are set in one of the following ways:

- Object Type (i.e. in the Property Aspect).
- Object Instance (i.e in the Property Aspect).
- Log (in the Log Configuration Aspect).
- Trend Display.

These are listed are in order from lowest to highest precedence.

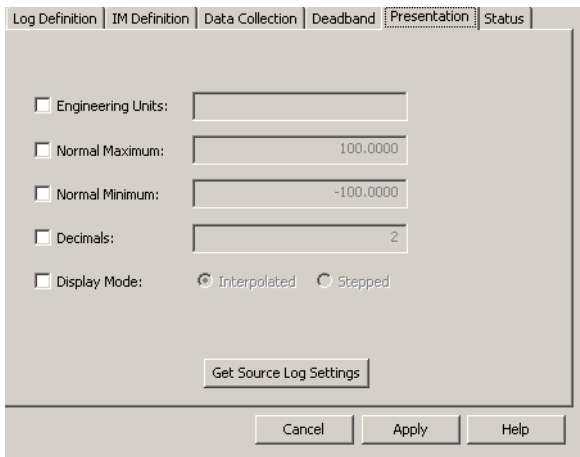


Figure 159. Presentation Tab

For example, to override a value set in the Object Type write a value in the Object Instance. A value set in the Object Type, Object Instance, or the Log can be overridden in the Trend Display.

To override a presentation attribute the check box for the attribute must be marked. If the check box is unmarked the default value is displayed. The default value can be a property name or a value, depending on what is specified in the Control Connection Aspect.

Engineering Units are inherited from the source. It is possible to override it. Normal Maximum and Normal Minimum are scaling factors, used for the scaling of Y-axis in the Trend Display. The values are retrieved from the source but are possible to override.

The Number of Decimals are retrieved from the source but are possible to override.

The Display Mode can be either **Interpolated** or **Stepped**. Interpolated is appropriate for real values, Stepped is for binary.

Click **Apply** to put the changes into effect.

Status

This tab is used to retrieve and view data for a selected log, Figure 160. As an option, use the **Settings** button to set viewing options. Refer to:

- Data Retrieval Settings on page 263.
- Display Options on page 264.

Click **Read** to retrieve the log data. Use the arrow buttons to go to the next or previous page. Page size is configurable via the Data Retrieval Settings. The default is 1000 points per page.

Click **Action** to:

- Import Data from File. Opens csv file to Insert, Replace, or Insert/Replace data.
- Export Data to File. Opens wizard to export csv file.
- Copy Data to Clipboard.
- Restore Log from Backup. Backup object needed in Maintenance structure and file on backup server.

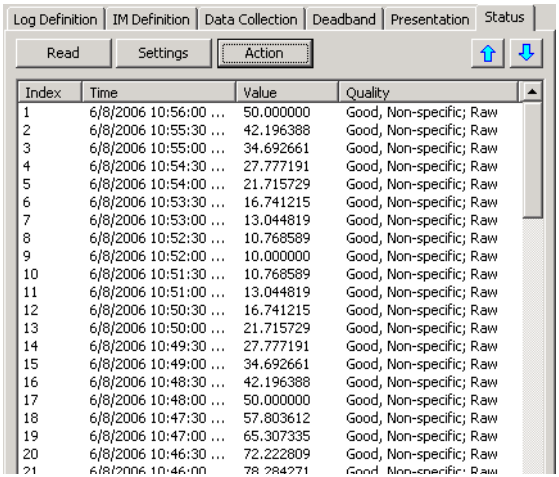


Figure 160. Status Tab

Data Retrieval Settings

The **Settings** button displays the Data Retrieval Setting dialog used to change the time frame and the number of data points to be retrieved, Figure 161.

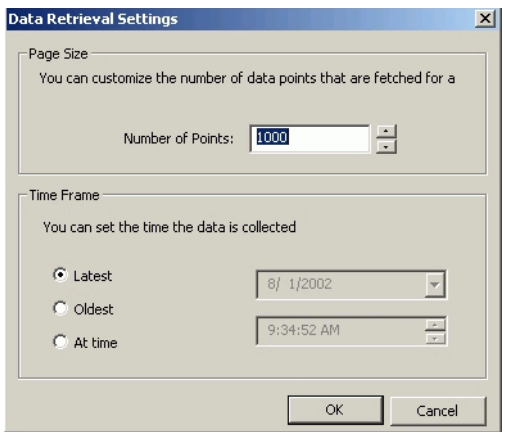


Figure 161. Data Retrieval Settings

Set the number of points per page in the Page Size area. Default is 1000 and maximum is 10,000.

Set the time range for retrieval of data in the Time Frame area.

- **Latest** retrieves one page worth of the most recent entries.
- **Oldest** retrieves one page worth of the oldest entries.
- **At time** is used to specify the time interval for which entries will be retrieved. The data is retrieved from the time before the specified date and time.

Display Options

The **Display** button displays the Display Options dialog used to change the presentation parameters data and time stamp, Figure 162.

Values are displayed as text by default. Unchecking the check box in the **Quality** area, displays the values as hexadecimals.

Timestamps are displayed in local time by default. Unchecking the first check box in the **Time & Date** area, changes the timestamps to UTC time.

Timestamps are displayed in millisecond resolution by default. Unchecking the second check box in the Time & Date area changes the timestamp resolution to one second.

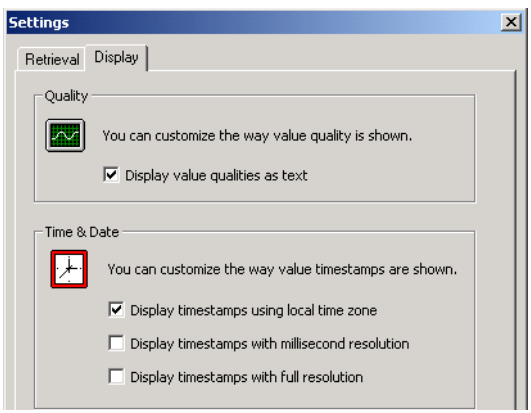


Figure 162. Display Options Dialog

Property Log Tab

The Property Log tab, Figure 163, displays log information. This tab is displayed by clicking the Log Configuration Template placeholder in the Log Configuration hierarchy.

The Data Size is the size of the Property Log on disk. It is the sum of the size of all logs. The size of each log file is the sum of the file header and the data storage size.

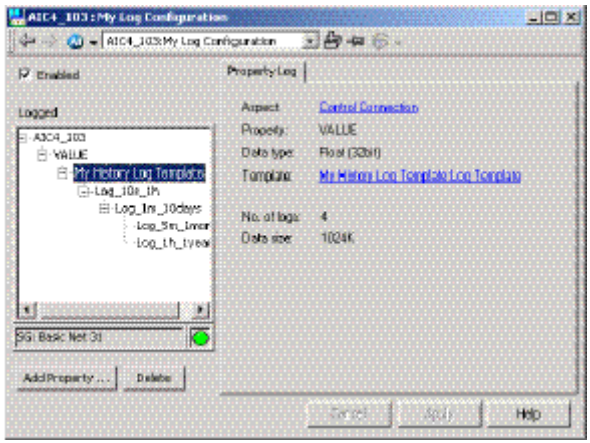


Figure 163. Property Log Tab

Status Light

The Service Status row shows status information for the configured Service Group. The name of the configured group is presented in the row. If no history sources have been defined (refer to Configuring Node Assignments for Property Logs on page 189), the text ‘Not Specified’ is displayed. Status for configured Service Group may be:

- OK: All providers in the service group are up and running.
- OK/ERROR: At least one service provider is up and running.
- ERROR: All providers in the service group are down.

Property Log Configuration Reference

The basic configuration procedure is covered in Building a Simple Property Log on page 196. The following sections provide further details:

- Modifying Log Configurations on page 266.
- Adding a Direct Log on page 266.
- Adding a Hierarchical Log on page 267.
- Deleting an Individual Log from the Log Template on page 267.
- Assigning a Log to a Log Set on page 267.
- Assigning Logs to an Archive Group on page 268.
- Activating/Deactivating a Property Log on page 268.

Modifying Log Configurations

When a log template with IM History logs defined is instantiated, the IM logs in that template cannot be modified. Trend logs in the same template can be modified. If an IM log must be modified, the IM log can be deleted from the template. Once applied and the delete operation is completed, the desired IM log can be re-added to the log template. The new logs will not have any data from the old logs and will start collecting data from the time they are activated.

In some cases, it is recommended to delete all instances of the log template and recreate everything based on the new configuration.

It is also possible to modify some IM History Log attributes on the log configuration aspect.

Adding a Direct Log

A direct log gets its data directly from the Data Source, which can be an aspect object property, or another History Log. The procedure is illustrated in the tutorial. Refer to Adding a Trend Log on page 199.

More than one direct log can be in the log hierarchy. When collecting directly from a process object, DO NOT have more than one direct log. In this case, each direct log in the hierarchy log increase the load on the control network.

Adding a Hierarchical Log

A hierarchical log gets its data either from a direct log, or another hierarchical log. The procedure is illustrated in the tutorial. Refer to Adding a History Log on page 201.

Deleting an Individual Log from the Log Template

To cut a log from the log hierarchy defined in a log template, right-click on the log to be cut, and then choose **Delete Log** from the context menu, Figure 164. An individual log that has children can not be cut. The children logs must be deleted first.

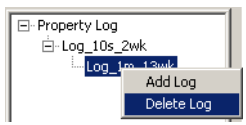


Figure 164. Deleting a Log

Assigning a Log to a Log Set

Log Sets group related logs so that they can be activated or deactivated as a unit. The only limitation is that the logs must reside on the same node. Logs sets are configured as described in Section 6, Configuring Log Sets.

To assign a log to a log set:

1. Select the log in the log hierarchy.
2. Click the **IM Definition** tab.
3. Select the Log Set from the Log Set pull-down list.

To assign a log to a second log set, simply repeat the procedure, using the **Second Log Set** pull-down list.

It is not recommended to use Log Sets in most configurations

Removing a Log from a Log Set

To remove a log from a log set, make the Selected Log Set field blank, then click **OK**.

Assigning Logs to an Archive Group

The Archive Group structure groups related logs for timed (periodic) archival. The only limitation is that the logs must reside on the same node. Archive groups are configured as described in Configuring Archive Groups on page 341.

To assign a log to an archive group:

1. Select the log in the log hierarchy.
2. Click the **IM Definition** tab.
3. Select the Archive Group from the Archive Group pull-down list.

As an option, assign logs to an archive group via the Archive Group aspect. Refer to Configuring Archive Groups on page 341.

Removing a Log from an Archive Group

To remove a log from an Archive Group, make the Selected Archive Group field blank, then click **OK**. As an option, delete logs from an archive group via the Archive Group aspect. Refer to Configuring Archive Groups on page 341.

Activating/Deactivating a Property Log

Activate/deactivate can be used on all the logs in a property log as a unit, or on individual logs within the hierarchy, or for logs on a log set basis.

When activating a log, it may be activated immediately, or activation may be delayed, depending upon how the log's Start Time attribute is defined. Start Time determines the earliest allowable time that the log can be activated. Start time also determines the hour, minute, and second when the first sample is collected.

To activate a property log as a unit:

1. Go to the structure where the logged object resides and click the Log Configuration aspect.
2. Use the **Enabled** check box to activate (checked) or deactivate (unchecked) all logs in the property log, Figure 165.

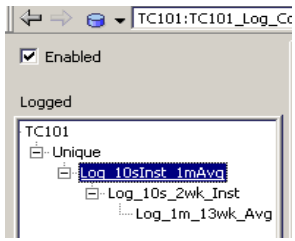


Figure 165. Activating/Deactivating a Property Log

The enabled check box is only for Property logs (basic history).

To activate an individual log within a property log, select the log in the hierarchy, click the **IM Definition** tab, and then click **Activate** under the IM Historian Log State control, Figure 166.

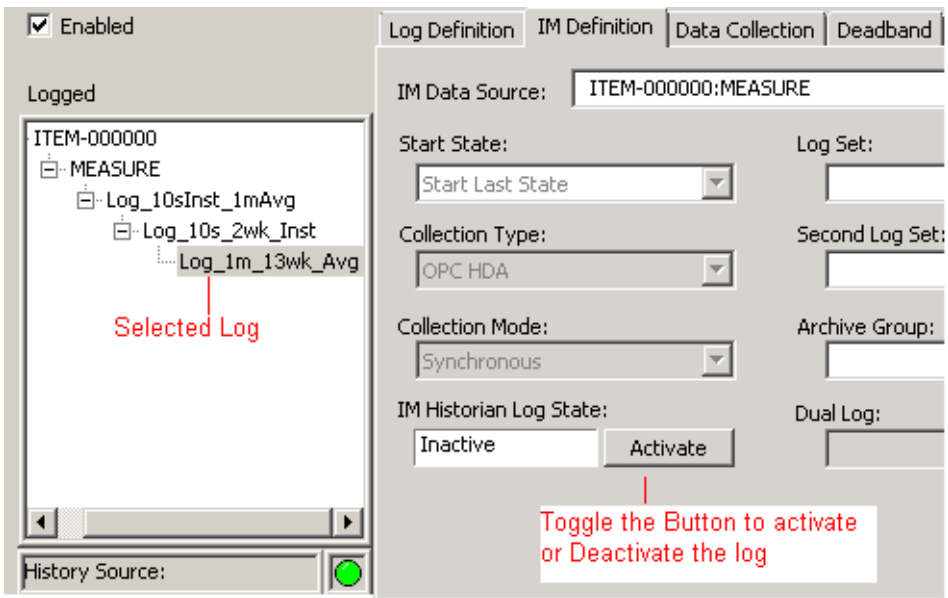


Figure 166. Activating an Individual Log in the Property Log Hierarchy

To activate or deactivate property logs on a log set basis, refer to Activate/Deactivate Logs in a Log Set on page 147.

Bulk Configuration of Property Logs

The Bulk Configuration tool is used to configure and instantiate property logs for a large quantity of object properties without having to add the logs on an object-by-object, and property-by property basis in the Plant Explorer. This tool is an add-in for Microsoft Excel.

Add-in tools for Bulk Import are installed in Microsoft Excel on a user basis. If the add-in tools for Bulk Import are not available when Excel is opened, use the procedure in Installing Add-ins in Microsoft Excel on page 295 to add them now.

The Bulk Configuration tool is used to generate a load report which calculates the disk space occupied by existing file-based property logs, and additional disk space which will be required as a result of instantiating new property logs via this tool. Use this report to determine whether to adjust the disk space allocation for file-based logs. This tool also is used to verify whether or not the logs have been created when finished.

Import filters in the Bulk Configuration tool can be used to shorten the time it takes to read information from aspect objects into the Excel spreadsheet. The most effective filter in this regard is an import filter for some combination of object types, properties, and data type. Since some objects may have hundreds of properties, the time savings can be significant.

A list of properties can be created for which logs have already been created in the event that the log configurations for those properties need to be modified or deleted.

Extensive use of the Bulk Configuration tool may cause disks to become fragmented. This, in turn, may impair the response time and general performance of the system. Therefore, check the system for fragmented files after any such procedure, and defragment disks as required.

History configuration impacts not only the History server disk(s), but also the disk(s) on any Connectivity Servers where trend logs are configured. Therefore, check the disks on any Connectivity Server where trend or history logs are located.

Work Flow

Configure all other history objects, including log templates, before using this tool. The recommended work flow is as follows (instructions for all steps with the exception of 2, 3 & 12 are provided in this section):

1. To start with a fresh (empty) database, refer to the procedure for Dropping the Existing Database on page 272 and recreate the History database before adding any new logs. If this step is not performed, the new logs will be in addition to any existing logs in the current History database.
2. Configure log sets, message logs, report logs as required by the application. If archival is required, create an Archive Device and Archive Groups. Refer to:

   Section 6, Configuring Log Sets
   Section 7, Alarm/Event Message Logging
   Section 8, Historizing Reports
   Section 11, Configuring the Archive Function
3. Configure the Log Templates as described in Building a Simple Property Log on page 196. DO NOT create Log Configuration aspects at this time.
4. Create a new workbook in Excel and initialize the workbook to add the Bulk Configuration add-ins.
5. Create a list of object properties whose values are to be logged.
6. For each object property, specify the Log Template that meets its data collection requirements, and specify the name of the Log Configuration aspect.
7. Configure presentation attributes if necessary.
8. Generate a load report to see whether or not the disk space allocation for file-based logs needs to be adjusted. Make any adjustments as required.
9. Create new property logs based on the spreadsheet specifications.

10. Verify that all property logs and their respective component logs have been created.

If multiple property log configurations are created using the Bulk Import tool and Industrial IT Basic History and Information Management logs stop collecting follow these steps:

a. Locate the Basic History Service Providers for the log configurations just created.

b. Restart the Basic History Service Providers located in step a.

11. Run the Load Report function again. This time, the report should indicate that all logs have been created (existing space used = proposed space requirement).
12. Create a backup copy of the database. Refer to Backup and Restore on page 408.
13. On all nodes where either trend or history logs are configured, check system for fragmented files, and defragment the applicable disks as required.

Dropping the Existing Database

This step is only required when starting with a fresh (empty) database. In this case, drop and recreate the History database before running the Bulk Configuration tool to instantiate the Log Configuration aspects. If this step is not performed, the new logs will be in addition to any existing logs which are currently defined in the History database.

Before drop the database, use the Plant Explorer workplace to delete the Log Configuration aspects from the Aspect Directory. Then refer to the procedure for dropping and recreating the History database in Create or Drop a Product Database on page 481.

When finished, create the other History objects as described in steps 2 and 3 above, and then continue with bulk configuration by Initializing a New Workbook on page 272.

Initializing a New Workbook

This procedure sets up the Excel spreadsheet for Bulk Import. As a prerequisite, the Bulk Import add-ins must be available on the spreadsheet menu bar. These add-ins

are added on a user basis. The add-ins are automatically added for the user that installed the Information Management software. Before running Microsoft Excel as a different user, install the add-ins for that user. Instructions are provided in Installing Add-ins in Microsoft Excel on page 295.

To create and initialize a new workbook:

1. Launch Microsoft Excel and create a new workbook, Figure 167.

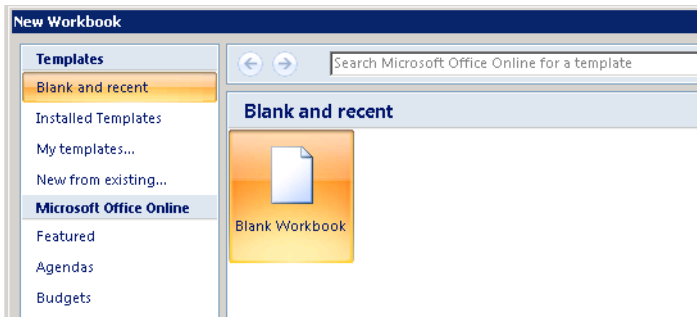


Figure 167. Creating a New Workbook in Excel

2. Choose **Bulk Import>Initialize Sheet**, Figure 168.

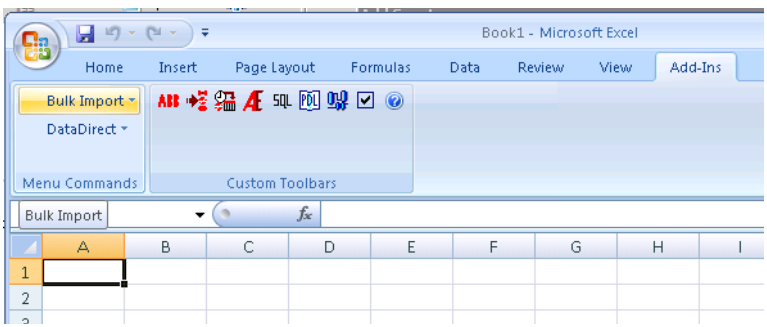


Figure 168. Initializing the Workbook for Bulk Configuration

This sets up the workbook for bulk configuration by adding the applicable row and column headings. Also, a dialog is presented for selecting a system (system created via the 800xA Configuration Wizard), Figure 169.

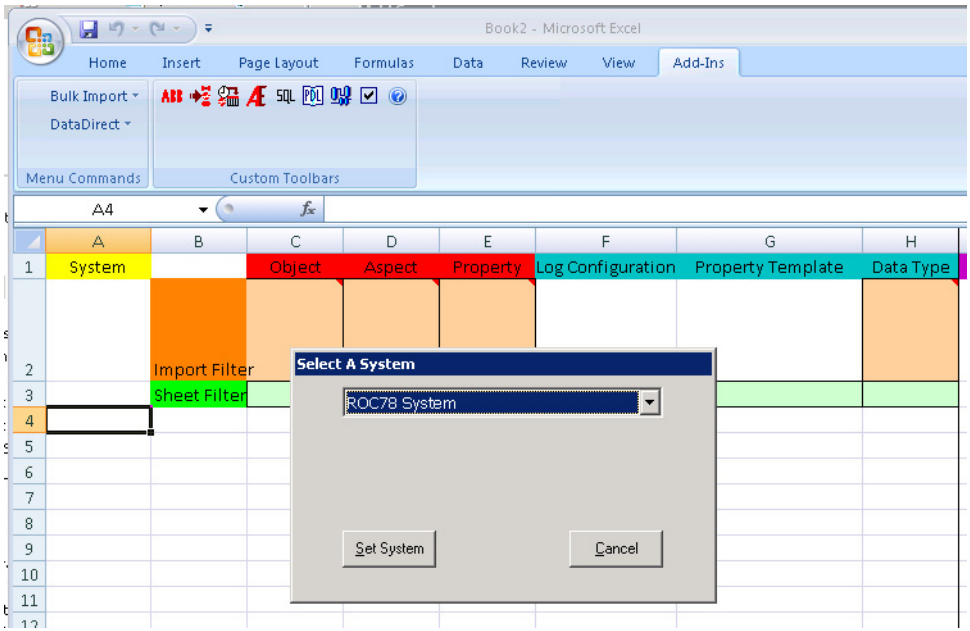


Figure 169. Initialized Workbook

Excel sheet columns from A through E may disappear hiding any log names. This can be corrected by unfreezing panes and refreezing panes.

3. Click **Set System** to connect the spreadsheet to the system indicated in the Select a System dialog. Use the pull-down list to change the system if needed.

When finished initializing the worksheet, continue with Creating a List of Object Properties on page 275.

Creating a List of Object Properties

This procedure imports a list of object properties whose values need to be logged. Be sure to apply an import filter so that only those objects and properties that meet a specific criteria are imported. This reduces the time required to complete the import, and helps keep the list at a manageable size. Make further adjustments to the resulting list using the filtering and sorting tools available in Microsoft Excel.

Filtering may be implemented on two levels.

- First, the Bulk Import add-in provides five filters within the spreadsheet. Spreadsheet cells near the top left corner of the spreadsheet are used to specify object name, object type, aspect, property, and data type as filtering criteria. Generally, a combination of object types, properties, and data type makes the most effective filter.
- Second, import properties for a selected object or for the selected object and its children objects can be specified. These properties can be included or excluded for logs that have already been created. These specifications are made via check boxes in the Select an Object dialog which is described later (step 5) of this procedure.

The following example shows how to specify an import filter using a combination of two object types and three properties. To do this:

1. Enter the object type specification in the *Object* column of the *Import Filter* row (cell C2), Figure 170. Specify more than one object type if needed. If the object type names are known, enter the names directly in the cell using the following syntax:

   TYPE:*object type1***:***object type2*

   where:

   TYPE: causes the following text strings to be interpreted as object type names. Each object type name is separated by a colon (**:**).

To interpret the text as a pattern to match object names (for example: *TC100*), omit the **TYPE:** keyword.

The example specification in Figure 170 will import all objects of the type ggcAIN and ggcPID (**TYPE:ggcAIN:ggcPID**).

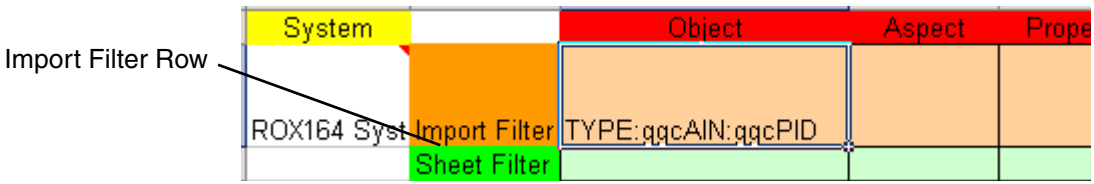


Figure 170. Object Type Specification Completed

Use the Object Type pick list if the spelling of the object type names is not known. To do this, right click in the Object Column for the Import Filter row (cell C2) and choose **Object Type List** from the context menu, Figure 171.

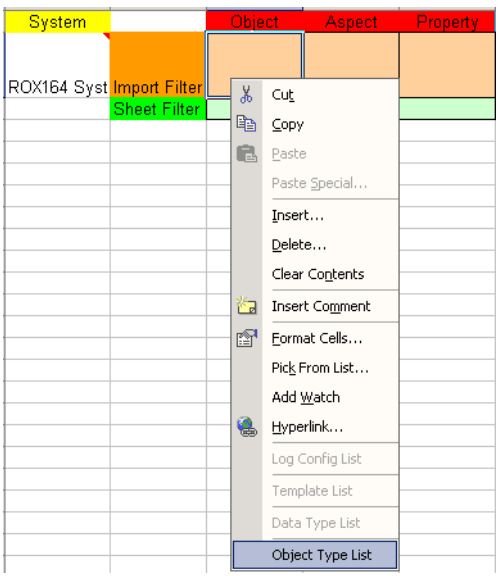


Figure 171. Opening the Object Type List (Using Context Menu on Object Column)

This displays a dialog for selecting one or more object types, Figure 172. Scroll to and select the applicable object types (in this case ggcAIN and ggcPID).

Select as many object types as required, although selecting too many will defeat the purpose of the filter. Select an object type again to unselect it.

Click **OK** when finished.

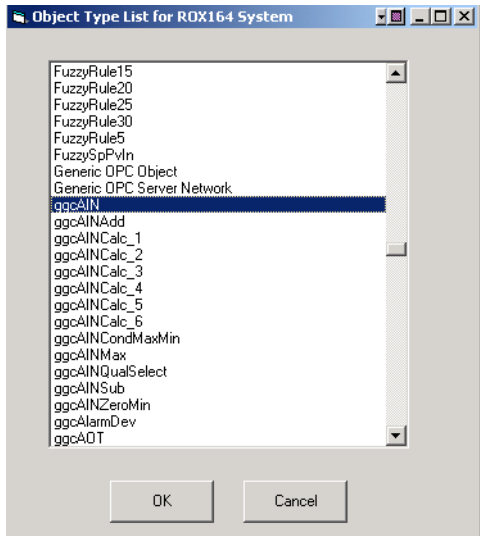


Figure 172. Selecting an Object Type

2. Enter the property specification in the *Property* column of the *Import Filter* row. Specify one or more properties using the following syntax: **NAME:***property1***:***property2*, where each property is separated by a colon (**:**). For example: **NAME:Link.PV:Link.SP:Link.OUT**, Figure 173.

To interpret the text as a pattern to match property names (for example: *value*), omit the **NAME:** keyword.

A	B	C	D	E
System		Object	Aspect	Property
ROX164 Syst	Import Filter	TYPE:ggcAIN:ggcPID		NAME:Link.PV:Link.SP:Link.OUT
	Sheet Filter			

Figure 173. Property Specification Completed

3. Click outside the cell to register the entry in the spreadsheet.

Optionally, use a Data Type filter to make the object search more closely match your requirements. The Data Type filter is a pattern filter. Enter the data type manually (for example: **Float**), or use Data Type pick list which is available via the Data Type cell's context menu, Figure 174.

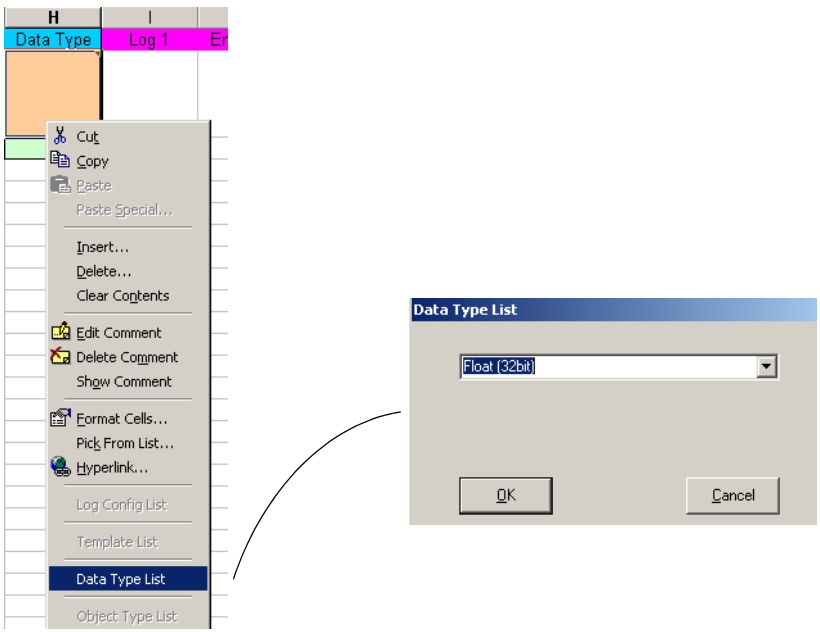


Figure 174. Specifying a Data Type Filter

4. When finished with the import filter specification, choose **Bulk Import>Load Sheet from System**, Figure 175.

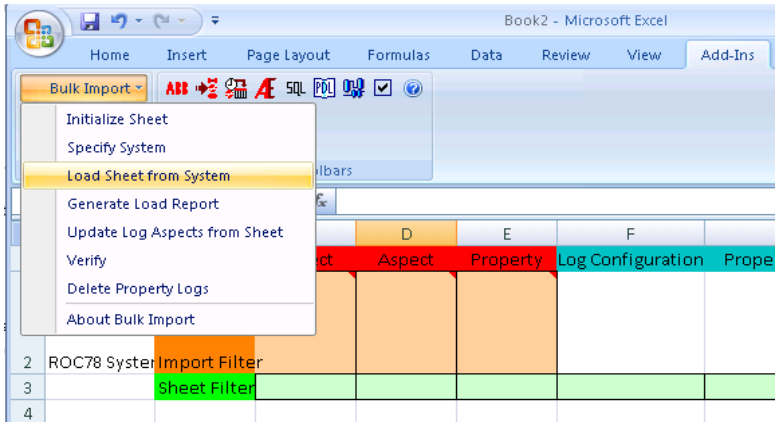


Figure 175. Loading the Spreadsheet with Object Properties

This displays a dialog for selecting the root object in the Aspect Directory under which the importer will look for objects and properties that meet the filtering criteria, Figure 176.

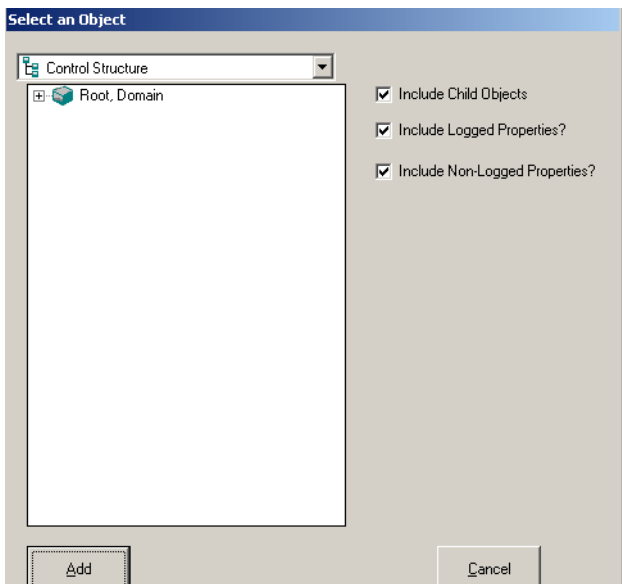


Figure 176. Select an Object Dialog

5. Use the Select an Object dialog to specify the scope of properties to be imported.

   Use the browser to select the root object whose properties will be imported. Typically, the **Include Child Objects** check box to the right of the browser is checked. This specifies that properties of the selected object AND properties of all child objects under the selected object will be imported. If the box is not checked, only properties for the selected object will be imported.

   The other two check boxes specify:

- Whether or not to limit the import to properties that already have logs.
- Whether or not to enter the log configuration information in the spreadsheet when importing properties that have logs.

  ALWAYS check one or both of these check boxes; otherwise, no properties will be imported. The operation of these check boxes is described in Table 28.

Table 28. Logged and Non-Logged Properties Check boxes

Include Logged Properties	Include Non-Logged Properties	Operation
X	X	Import all properties according to specified filter. For those properties that already have logs, fill in the Log Configuration Aspect and Log Template specifications.
	X	Import all properties according to specified filter. For those properties that already have logs, do not fill in the Log Configuration Aspect and Log Template specifications.
X		Import only properties that already have logs, and fill in the Log Configuration Aspect and Log Template specifications.
		This will result in no properties being imported.

For example, the selections in Figure 177 will import the properties for objects in the Group_01 Generic OPC Object branch based on the filter set up in steps 1&2. It will import the log configuration and log template specifications for any properties that already have log configurations.

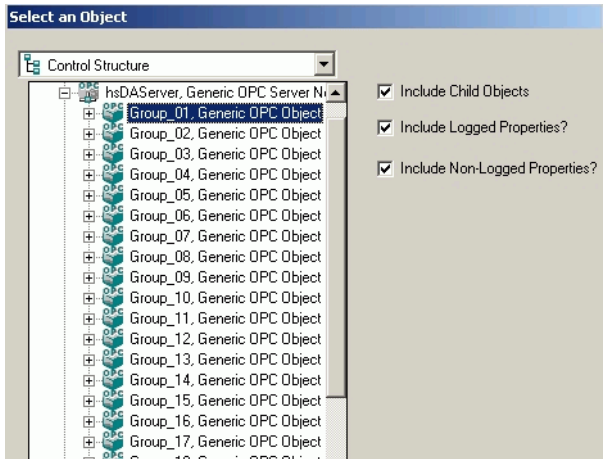


Figure 177. Selecting a Parent Object

Only objects and properties that meet the filtering criteria will be imported.

6. Click **Add** when finished. This runs the importer.

   A partial listing of the import result based on the specified filter is shown in Figure 178. (The complete list is too lengthy to show.)

C	D	E
Object	Aspect	Property
TYPE:ggcAIN:ggcPID		NAME:Link.PV:Link.SP:Lin
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/FI423_03	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/FI423_04	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI414_01	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI414_03	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI414_04	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI425_04	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI428_02	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI428_03	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI428_04	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI434_02	Control Module	Link.PV
[Control Structure]Root/Control Network/PVCPlant/Applications/Dispersant/Control Modules/LI434_03	Control Module	Link.PV

Figure 178. Import Filter Result

Use the Sheet Filter (row below Input Filter), or the sorting and filtering tools in Microsoft Excel to make further adjustments to the list of imported objects and properties.

Once the list is refined, continue with the procedure covered in Assigning Log Templates below.

Assigning Log Templates

This procedure assigns a Log Template to each object property according to the property's data collection requirements. This procedure also is used to specify a name for the Log Configuration aspect for each property log, or specify that the property log be added to an existing Log Configuration aspect (if one exists for the object where the property log is being added).

To do this:

1. Select a cell in the *Property Template* column. For example, Figure 179 shows the cell for the first object property being selected.

   If the name of the template is known, enter it directly in the spreadsheet cell. If the name is not known, right-click and choose **Template List** from the context menu, Figure 179.

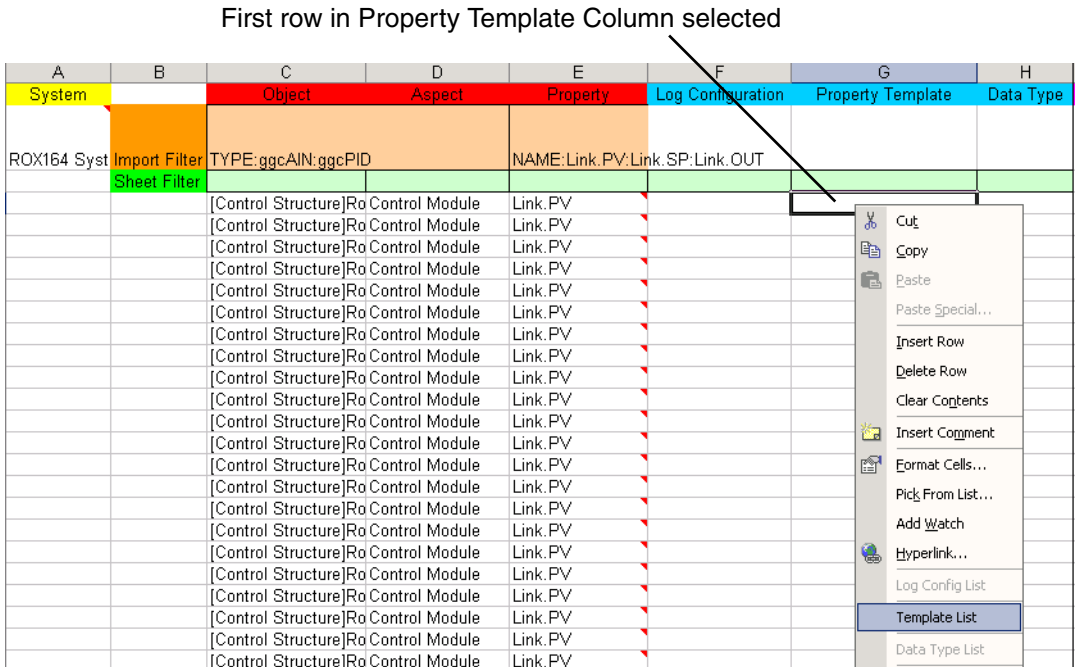


Figure 179. Selecting a Property Template

This displays a pull-down list for all property log templates that exist in the system, Figure 180.

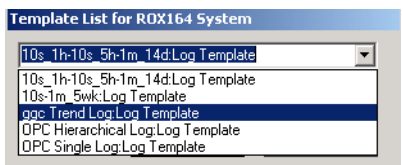


Figure 180. Template List

2. Select the template from the list then click **OK**.

To use the same template for additional object properties in contiguous rows under the currently selected object property, click on a corner of the cell, and pull the corner down to highlight the Property Template column for as many rows as required, Figure 181. The template specification will automatically be entered in the highlighted cells when the mouse button is released.

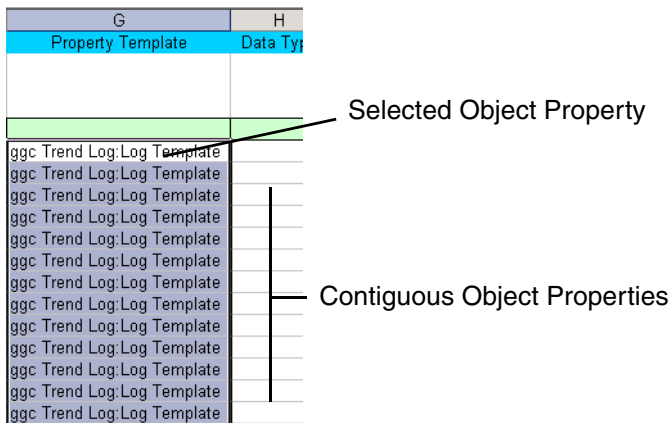


Figure 181. Applying Log Template Specification to Objects in Contiguous Rows

3. The Log Configuration aspect name is specified in much the same manner as the property template. To create a new Log Configuration aspect for the property log because when one does not yet exist, enter a Log Configuration aspect name directly in the spreadsheet cell.

   If no Log Configuration aspect names have been specified yet in the entire column, typing the letter **L** in a cell will automatically fill in the cell with: **Log Configuration** (matching the Column Name). This is a suitable aspect name. It is not required for these names to be unique.

   To add the property log to an existing Log Configuration aspect, right-click and choose **Log Config List** from the context menu. This displays a list of Log Configuration aspects that exist for that particular object. If no Log Configuration aspects exist for the object, the list will indicate **None**.

   As with the Property Template specification, to use the same Log Configuration aspect name for contiguous object properties in the list, click on

corner of the cell, and pull the corner down to highlight the Log Configuration column for as many object properties as required.

When finished assigning Log Templates and Log Configuration Aspect names, continue with the procedure described in Configuring Other Log Attributes below.

Configuring Other Log Attributes

The spreadsheet has additional columns for log attributes that are configurable via the **Presentation** tab on the Log Configuration aspect. For details, refer to Presentation on page 261. When finished configuring other log attributes, continue with the procedure covered in Generating a Load Report on page 285.

Generating a Load Report

The Load Report function calculates the disk space occupied by existing file-based property logs, and additional disk space which will be required as a result of instantiating new property logs as specified in the spreadsheet list. This report aids in determining if the disk space allocation for file-based logs needs to be adjusted.

Run this report after creating the object property list and assigned the log templates. To run the report, from the Excel menu bar choose **Bulk Import>Generate Load Report**, Figure 182.

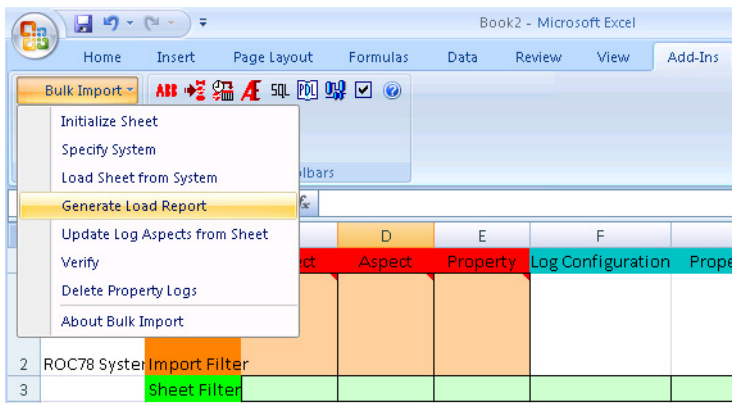


Figure 182. Generate Load Report

Then click the **Load Report** tab at the bottom of the spreadsheet, Figure 183.

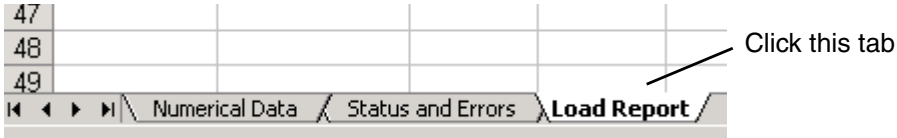


Figure 183. Load Report Tab

An example report is shown in Figure 184. The report shows the disk space requirements on a per-server basis. The information for each server is presented in a separate column. For the example in Figure 184 there is only one server.

4		
5		Load Per Server
6		ROC2HFYP01
7	Existing	0
8	Proposed	3,891,200
9	Total	3,891,200
10		
11	Grand Total	3,891,200
12		

Figure 184. Example, Load Report Result Before Creating Logs

The information provided in this report is described below.

Load Per Server	The report shows the disk space requirements on a per-server basis. In this example, there is only one server (in this case ROC2HFVP01).
Existing	This row indicates the disk space (in bytes) used by existing property logs. In this case, 0 (zero) indicates that no space is used since there are no existing logs.
Proposed	This row indicates the space requirement (in bytes) for the property logs which will be created based on this spreadsheet. In this case 3,891,200 bytes will be required.
Total	This row indicates the sum of the Existing and Proposed disk space requirements.
Grand Total	This row indicates the sum of the totals for all servers.

Compare the results of this report with the current disk space allocation for file-based logs. Use the Instance Maintenance Wizard or hsDBMaint function as described in Directory Maintenance for File-based Logs on page 467. Make any adjustments as required, following the guidelines provided in the Directory Maintenance section. When finished, continue with the procedure covered in Creating the Property Logs on page 287.

Creating the Property Logs

This procedure updates the History database by adding Log Configuration aspects and property logs as specified in the spreadsheet. Save the spreadsheet before running the Bulk Configuration function. Excel can then be exited without losing the configuration data in the event of a problem.

The update function skips those object properties in the list that already have a property log existing in the database. To replace the existing property logs, delete those property logs before running the update. To do this choose **Bulk Import>Delete Property Logs**. This deletes all property logs for all object properties represented in the spreadsheet.

To run the Bulk Configuration function,

1. Choose **Bulk Import>Update Log Aspects from Sheet**, Figure 185.

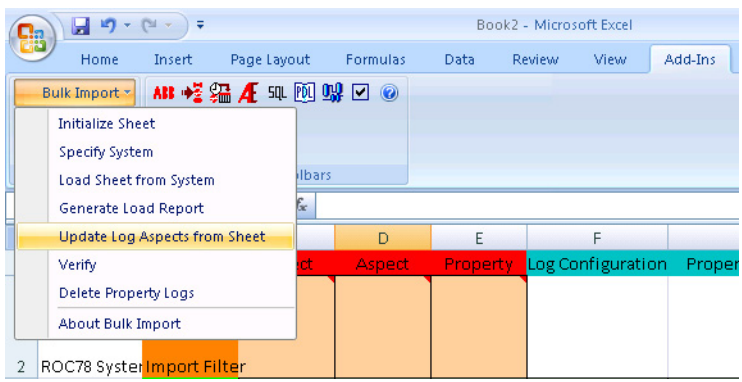


Figure 185. Updating Log Aspects from the Sheet

2. When prompted to confirm, click **OK** to run the update.

If the error message in Figure 186 is shown, then refer to Handling Error Conditions on page 288. Otherwise, continue with Verifying the Update on page 288.

Handling Error Conditions

During the update process, an error message such as the one shown in Figure 186 may be encountered. This message indicates the Bulk Configuration tool has failed to create the log for the property at a specified row in the spreadsheet. It also indicates the parameters specified for creating the log including object, aspect, and property name, log configuration aspect name, and template. Review these parameters.

If it is determined that these parameters are correct, and that the log should be created according to these parameters, then click **Yes** to continue the update. This message will not be displayed again, even if additional rows are skipped. When the update process is finished, continue with Verifying the Update on page 288. This procedure provides guidelines for handling the skipped rows (properties whose logs were not created).

If it is determined that one or more of the parameters are incorrectly defined, for example, the specified log template does not exist, then click **No** to abort the update. Fix the problem, then retry the Bulk Configuration procedure starting at Initializing a New Workbook on page 272.

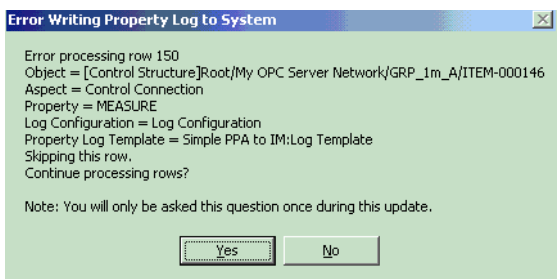


Figure 186. Error Writing Property Log To the System

Verifying the Update

It may take considerable time for History Services to create the property logs, depending on the size of the object property list. Therefore, wait some time after the

update is finished before attempting to verify the update (as a general rule, about 30 minutes for 2000 logs). After a reasonable delay, run the Verify function to verify all property logs have been created. To do this, from the Excel menu bar choose **Bulk Import>Verify**, Figure 187.

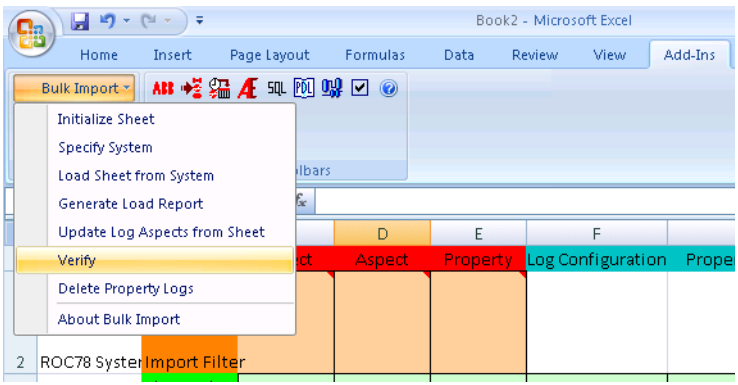


Figure 187. Running the Verify Function

An example result is shown in Figure 188. The presence of a red *not verified* message in the far left (A) column indicates a problem within the corresponding row. The item that caused the failure, for example the log template, will also be shown in red. Verified rows are shown in black.

Some rows may be marked as not verified because the verification function was run before History Services was able to create the logs. Clear these rows by re-running the verification function. Rows that were not verified on the previous run, but are verified on a subsequent run are shown in green. The next time the function is run, the green rows will be shown in black.

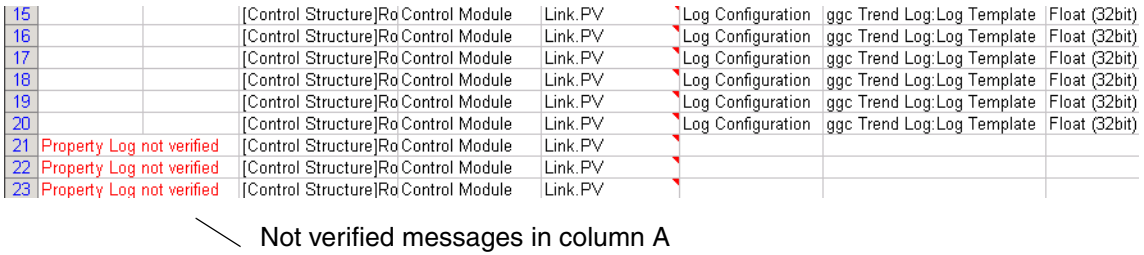


Figure 188. Example, Verify Function

Guidelines for handling not verified rows are provided in Handling Not Verified Rows on page 293.

To get a summary for each time the verification function is run, click the **Status and Errors** tab near the bottom of the spreadsheet, Figure 189. A new entry is created each time the verify function is run. The summary is described in Table 29.

Table 29. Verify Function Summary

Item	Description
Date and Time	Date and time when the verify function was run.
Rows Processed	Number of rows that History Services finished processing at the time this verification instance.
Rows Verified	Number of processed rows that have been verified at the time of this verification instance.
Rows Not Verified Due to Property Logs	Number of rows not verified due to a problem with the Log Configuration aspect or property log.
Rows Not Verified Due to Component Logs	Number of rows not verified due to a problem with component logs. The log configuration aspect was created, but the component history logs do not have item IDs.
Logs Processed	Number of component logs (within a property log structure) processed.
Logs Verified	Number of component logs (within a property log structure) verified.

Table 29. Verify Function Summary

Item	Description
Logs Not Verified	Number of component logs (within a property log structure) not verified.
Total Time	Time taken to run the verify function.

When using the Bulk Configuration tool in Microsoft Excel, the Update System Status information will be incorrect when a sheet has been filtered. After filtering, only the visible rows are processed during the **Update Log Aspects from Sheet** operation. However, the Status and Errors shows the number of rows written to system as the total (unfiltered) number of rows. This is the normal mode of operation.

A	B
	79 rows processed 41 Rows Verified 0 Rows not verified because of Property Logs 38 Rows not verified because of Logs 160 Logs Processed
11/6/2002 12:40:38 PM	57 Logs verified 103 Logs not verified Total Time = 8.6 seconds
11/6/2002 12:40:55 PM	Started Property Log Verify 79 rows to process
	79 rows processed 64 Rows Verified 0 Rows not verified because of Property Logs 15 Rows not verified because of Logs 160 Logs Processed
11/6/2002 12:41:01 PM	119 Logs verified 41 Logs not verified Total Time = 5.3 seconds
11/6/2002 12:41:13 PM	Started Property Log Verify 79 rows to process
	79 rows processed 79 Rows Verified 0 Rows not verified because of Property Logs 0 Rows not verified because of Logs 160 Logs Processed
11/6/2002 12:41:18 PM	160 Logs verified 0 Logs not verified Total Time = 5.6 seconds

Numerical Data | Status and Errors | Load Report

Figure 189. Verify Report

To further confirm that property logs have been instantiated for their respective object properties, go to the Control structure and display the Log Configuration aspect for one of the objects. An example is shown in Figure 190.

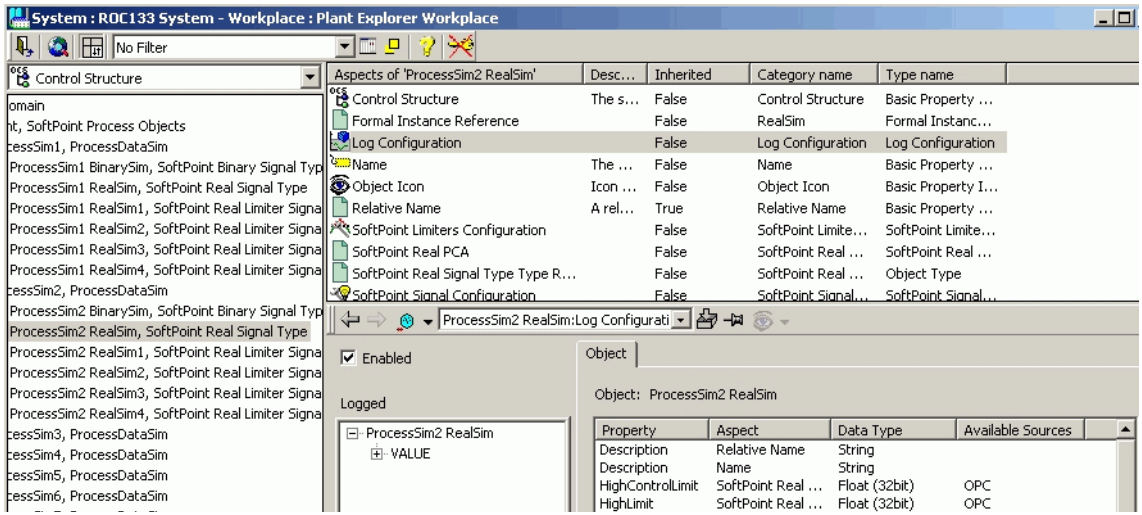


Figure 190. Confirming Instantiation of Property Logs

Handling Not Verified Rows

If after several iterations of the verification function, rows continue to be shown as not verified, the problem may be due to errors in the Log Template, Log Configuration aspect, or typographical errors in the spreadsheet. Another possibility is that History Services failed to access the Aspect Directory to get data required to build the log.

During the update procedure, if an error message is encountered indicating the Bulk Configuration tool failed to create the log for the property at a specified row in the spreadsheet (Figure 186), then at least some not verified rows are due to History Services failing to access the Aspect Directory. This problem may be corrected by simply filtering the property list to show only those rows that are not verified, and then re-running the update process. Further guidelines for this procedure are provided in Troubleshooting on page 294.

If there is a potential problem with a specification in the property list, the problem must be corrected, and then the Bulk Configuration tool must be run again starting with Initializing a New Workbook on page 272.

Troubleshooting

On occasion, the Bulk Configuration Tool may fail to create logs for one or more properties in the property list because the tool was temporarily unable to access the Aspect Directory for data. The rows for these properties will have a red *not verified* message in column A of the spreadsheet after running the verification function (Verifying the Update on page 288). Also, the item that caused the failure (for example, the log template) will be shown in red. If this occurs, use the following procedure to modify the property list to show not verified rows, and then re-run the Bulk Configuration tool to create the logs for those rows:

1. Select column A, and then choose **Data>Filter>Auto Filter** from the Excel menu bar, Figure 191.

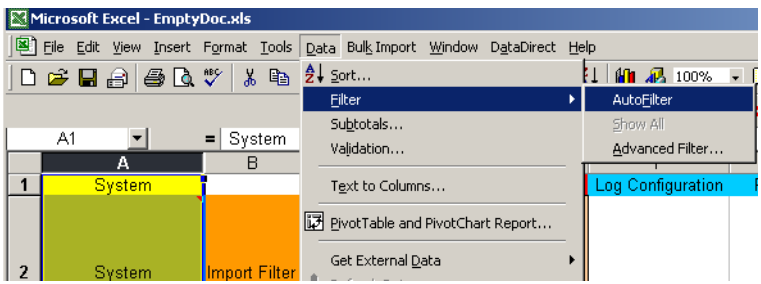


Figure 191. Applying the Auto Filter

2. Choose **NonBlanks** from the Auto Filter menu, Figure 192.

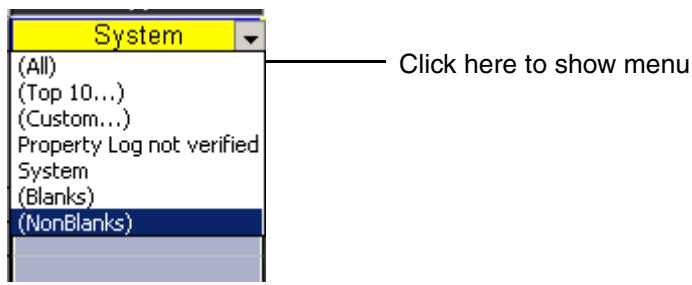


Figure 192. Auto Filter Menu

This will hide all rows that have no content in Column A. These rows have been verified. Only rows that have not been verified will be shown, Figure 193.

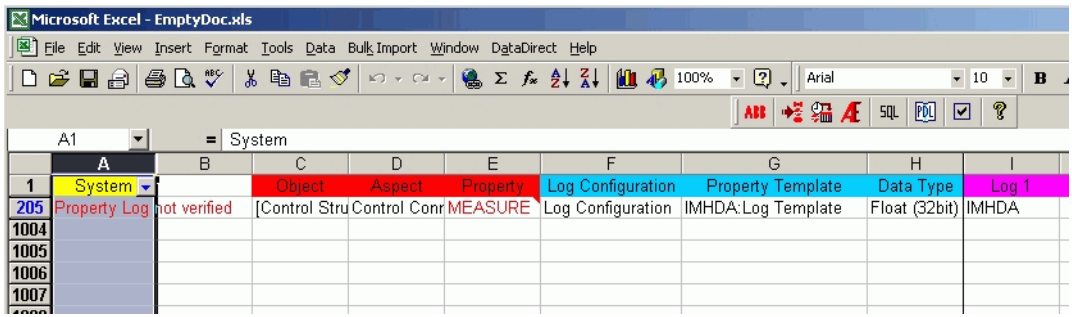


Figure 193. Result After Querying for No Blanks

3. Re-run the bulk configuration process again by choosing **Bulk Import>Update Log Aspect from Sheet** (refer to Creating the Property Logs on page 287).

Depending on the quantity of rows that were not verified, steps 1-3 may need to be repeated more than once. Continue to repeat these steps until the error message stops. Also run the verification function again as described in Verifying the Update on page 288.

Installing Add-ins in Microsoft Excel

Add-ins are automatically installed in Microsoft Excel for the user that installed the Information Management software. If other users will be using these applications, the add-ins must be installed in Microsoft Excel specifically for those users or they will not be available.

To install the add-ins for a different user (bulk import tool for example):

1. Log on to the Information Management server computer as the user requiring the add-ins.
2. Launch Microsoft Excel.
3. Click on the office button in the upper right hand corner.
4. Select Excel Options from the bottom of the menu.
5. Select Add-ins from the pane on the left.
6. Click the go button in the right pane.

7. Click **Browse,** then browse to **C:\Program Files\ABB Industrial IT\Inform IT\History\bin**. Select the file named **Inform IT Bulk Import.xla**.
8. Click **OK**. This adds the Bulk Import add-in to the Add-ins tool, Figure 194.

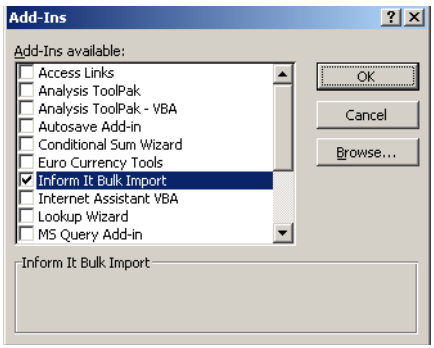


Figure 194. Bulk Import Added

9. Verify that the **Inform IT Bulk Import** box is checked in the Add-ins tool, Figure 194, then click **OK**.
10. Verify the Bulk Import add-in tools are added in Excel, Figure 195.

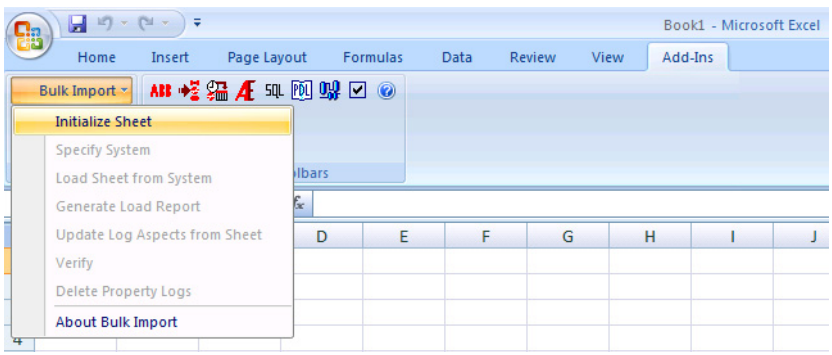


Figure 195. Verify Add-in Tools Available in Excel

Unless Archive Macros are signed, Excel Macro Security must be set to low for the macros to function. When the Bulk Import Tool or the Archive import tool is used, the macro security must be set to LOW.

What To Look For When Logs Do Not Collect Data

- **No History Data**. Only the operator trend data is seen when a log is first activated and not the history (Information Management-based) data. A lag may exist in the sample blocking rate configured on the Data Collection tab of the Log Template. The blocking rate controls the frequency at which data is collected by the history log from the source trend log. If this lag happens, check the sample blocking rate to determine whether or not that is the cause.
- **Inform IT Logs Not Collecting**. A history source aspects needs to be added in the structures where the logs reside when not collecting data for operator trend logs. Refer to Configuring Node Assignments for Property Logs on page 189.
- **History configuration functions can not be used**. Another web service may have been installed over and shut down the Internet Information Service. Use the Windows Service utility in the Windows Control Panel to make sure this service is enabled.
- **History Source aspect points to a connectivity server that does not exist**. Log Configuration aspects use the basic history service group ID on the Information Management server and carry that specification to the destination system when importing or synchronizing from an engineering system to a production system. The correct linkage between a Log Configuration aspect and its respective basic history service group can be fixed by making a quick change on the applicable log template(s).

  1. Find the log template used to create the log configuration aspect as indicated on the Log Configuration aspect.

  2. Go to the Log Definition tab for the Information Management log on the Log Template configuration aspect and select the service group.

  3. Ignore the contents of the Log Configuration update message (...*Partly Succeeded*...) and click **OK** to acknowledge.

Section 10 Profile Data Collection

This section provides an overview of the Profile Historian function in the 800xA system, and instructions for configuring this functionality.

Profile Historian Overview

History profiles are used to monitor critical quality measurements in flat-sheet manufacturing applications. For instance, in the Pulp and Paper industry history profiles may be used to monitor *basis weight* and *moisture content*. Collection, storage and retrieval of the profile data is supported by the Profile Historian option for Information Management.

Profile Historian consists of three basic components - AccuRay Object Server, Profile Historian Server, and Profile Historian Client. Figure 196 illustrates how these components are integrated into the manufacturing application.

Quality measurements associated with reel turn-ups and grade changes are recorded by frames (scanners) on one or more machines. These measurements are routed via each machine's dedicated controller to the **AccuRay Object Server**, where the measurements are consolidated in an OPC Database.

The **Profile Historian Server** contains the History Server and History database where the profile data is stored. History Services is used to configure profile logs which collect and store quality measurements from the Accuray Object Server.

Reel-turn ups, grade changes, and dayshift events for each machine are processed according to the Reel/Grade report which is created via the Reel Report Configuration Tool. The names, time stamps and other information associated with these events are stored in Production Data Logs (PDLs) in History. This information may be accessed by Information Management client applications such as DataDirect and Display Services, and by other reporting applications that support SQL queries.

The **Profile Historian Client** is used to view the quality measurements with industry-standard Contour Maps, Machine Direction (MD) graphs, and Cross Direction (CD) graphs.

The AccuRay Object Server, Profile Historian Server, and Profile Historian Client applications all run on the Windows platform. These applications may be installed on the same computer or dedicated computers. The Profile Historian Server must be installed on the Information Management Server.

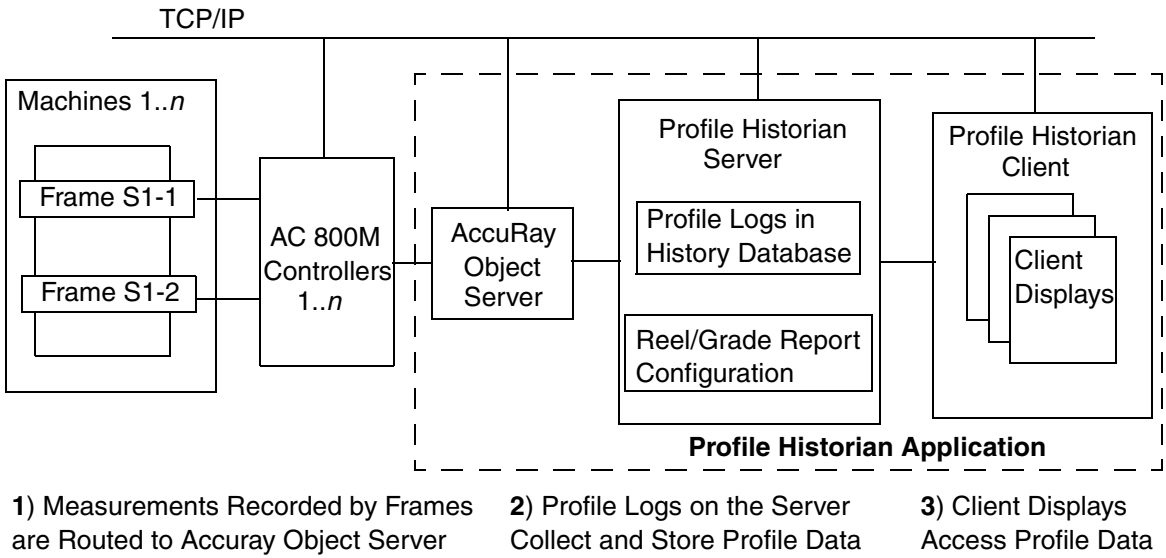


Figure 196. Profile Historian Architecture

Profile Historian Configuration Overview

There are four basic steps for configuring data collection for Profile logs:

- Configuring a Log Set for Profile Logs on page 301.
- Configuring Profile Logs on page 301.
- Configuring Reports on page 311.
- Exporting/Importing a Reel Report Configuration File on page 321.

Configuring a Log Set for Profile Logs

Every profile log MUST belong to a log set. This section provides specific requirements when configuring a log set for profile logs. Configure the log set before configuring a profile log. The name of the log set must be known to configure the profile log.

When specifying the log set name, use the name of the machine where the sensors corresponding to the OPC tags for the profile log are located. This same machine name will be used when configuring the Reel Report as described in Configuring Reports on page 311. An example is shown in Figure 197. Refer to Section 6, Configuring Log Sets for instructions on configuring log sets.

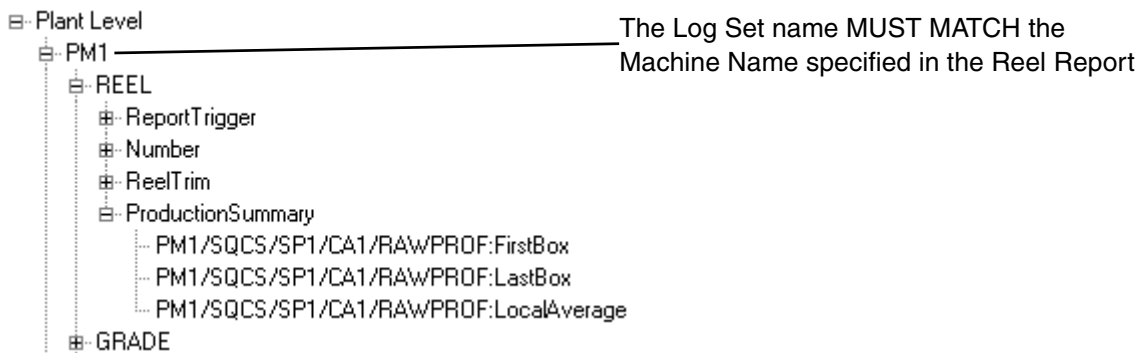


Figure 197. Machine Name to be used for Log Set Name

Configuring Profile Logs

Profile logs are added as objects in the Node Administration structure. To do this (reference Figure 198):

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node being added to the Profile Log (for example, ENG110 in Figure 198).
3. In the object tree for the selected node, navigate to **InformIT History**_*nodeName* **Service Provider > InformIT History Object**.

Under the InformIT History Object find containers for each of the Inform IT History object types. The History objects (in this case, a Profile Log) must be instantiated under the corresponding container.

4. From the **Profile Logs** group**,** choose **New Object** from the context menu. This displays the New Object dialog with the **Inform IT Profile Log** object type selected.

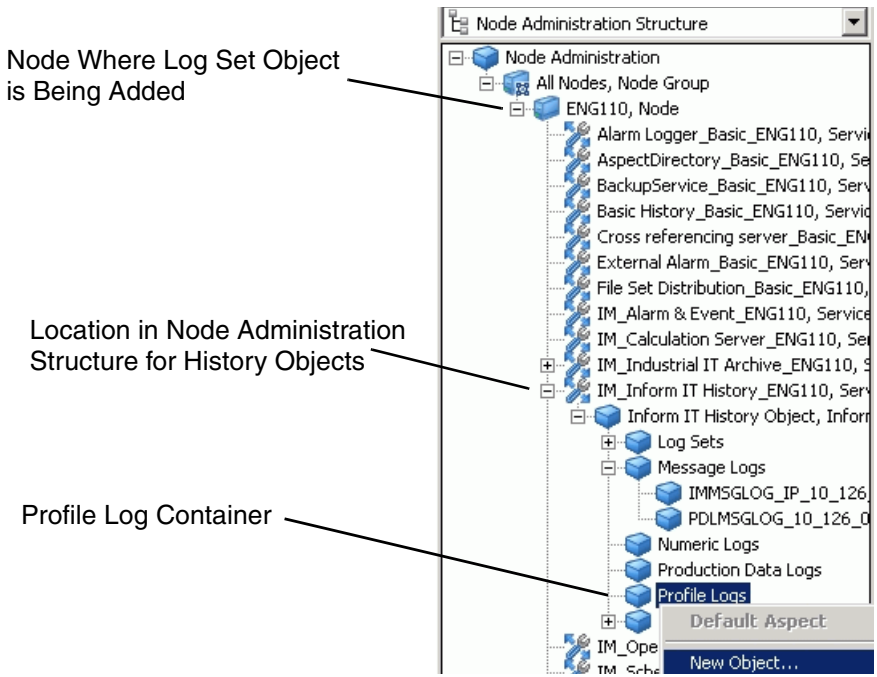


Figure 198. Adding a Profile Log in the Node Administration Structure

5. Enter a name for the object in the Name field, for example: CA1Profile, then click **Create**. This adds the object under the Profile Logs group, and creates a corresponding Profile Log Aspect. This aspect is used to configure the profile log attributes.

6. To display this aspect, select the applicable Profile Log object, then select the Inform IT History Profile Log aspect from the object's aspect list.

7. Use this aspect to configure the Profile Log Attributes.

Profile Log Attributes

The profile log attributes are configured via the Inform IT History Profile Log aspect, Figure 199. The attributes are organized under three tabs. Only a few attributes need to be configured. Most are automatically configured based on the configuration of another attribute, or may be left at their default value. Configure the mandatory attributes as described below. To configure other profile log attributes, refer to Profile Log Attributes Summary on page 309.

The following profile log attributes MUST be configured:

- On the **Main** tab configure:
  - Data Source on page 304.
  - Log Set on page 306.
- On the **Collection** tab configure:
  - Array Size on page 306.
  - Log Period on page 307.
  - Storage Interval on page 308.
- On the **Data Source** tab configure: Machine Position on page 308.

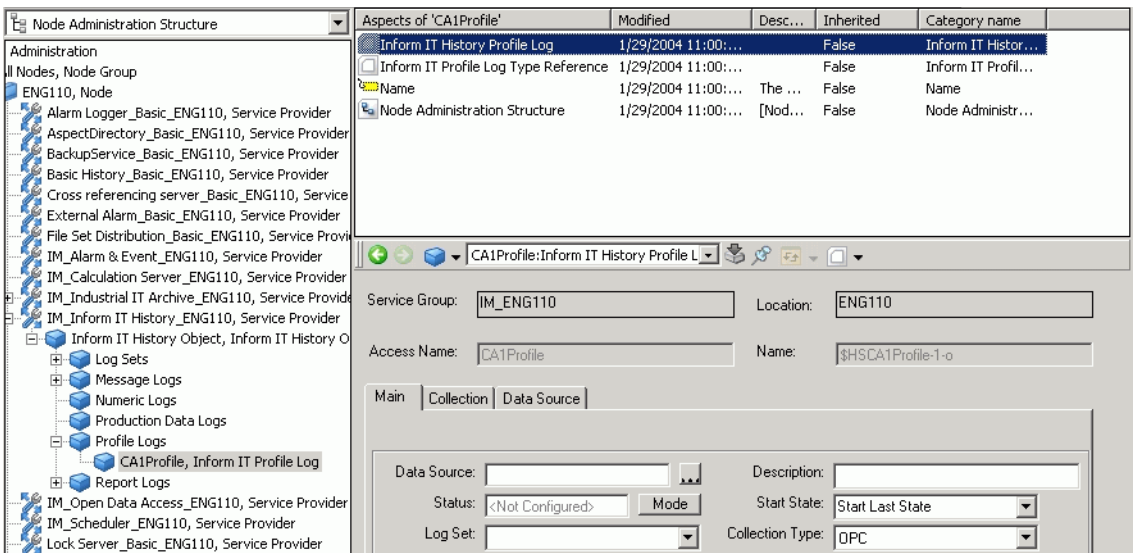


Figure 199. Displaying the Inform IT History Profile Log Aspect

Data Source

The profile log has multiple data sources - one for each of the quality measurements being stored in the profile log. The Data Source entered on the **Main** tab specifies the root portion of the OPC tag. The full data source tags for each of the quality measurements are automatically filled in on the **Data Source** tab based on the root portion of the OPC tag.

To select the OPC tag:

1. Click the browse button for Data Source, Figure 200.

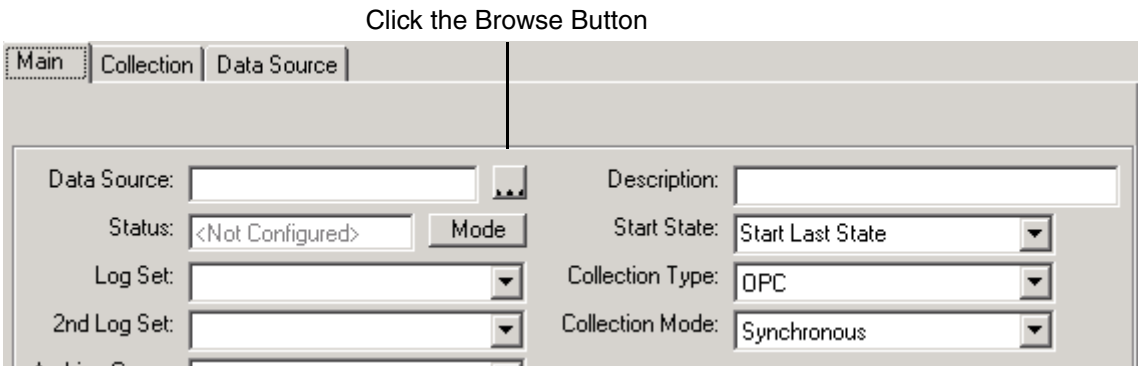


Figure 200. Browse Button for Data Source

   This displays the Select Data Source dialog for browsing the OPC server for the applicable object.

2. Use this dialog to select the OPC object which holds the profile data as demonstrated in Figure 201. Typically, the object is located in the Control structure, under the Generic OPC Network object for the Profile data server.

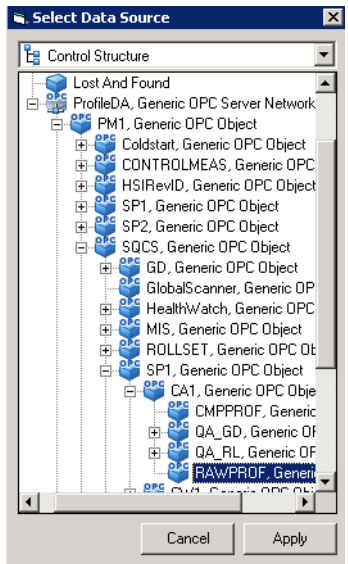


Figure 201. Selecting the OPC Tag

3. Click **Apply** when done. This enters the object name in the Data Source field, Figure 202.

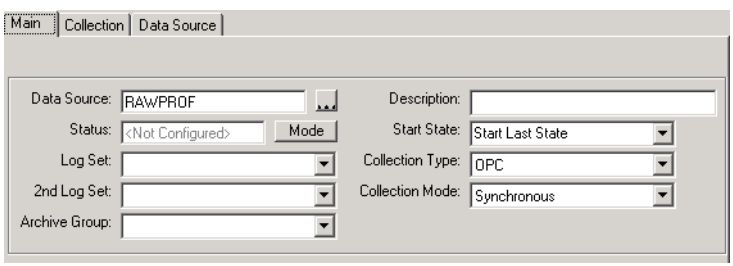


Figure 202. Data Source Defined

In addition, the data source tags for the quality measurements are automatically entered under the Data Source tab, Figure 203.

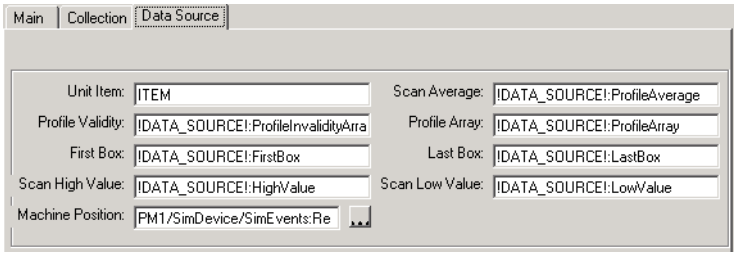


Figure 203. Profile Log Data Source Specifications

The**!DATA_SOURCE!** string represents the root of the OPC tag which is common for all data sources. The string following the colon (:) specifies the remainder of the tag for each measurement, for example **!DATA_SOURCE!:FirstBox**, **!DATA_SOURCE!:ProfileAverage**, and so on.

Log Set

Use the Log Set pull-down list on the **Main** tab to select the log set which was configured specifically for the profile logs. The name of this log set must match the machine name, for example PM1 as shown in Figure 204. Log Sets should not be used for xPAT configurations.

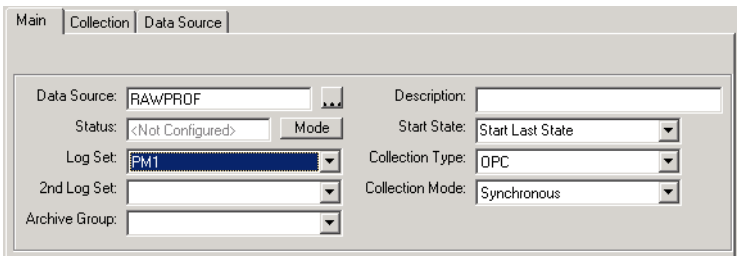


Figure 204. Selecting the Log Set

Array Size

The Array Size attribute is located on the **Collection** tab, Figure 205. Array size must be configured to match the number of data points per scan for the machine from which the data points are being collected. The default is **100**.

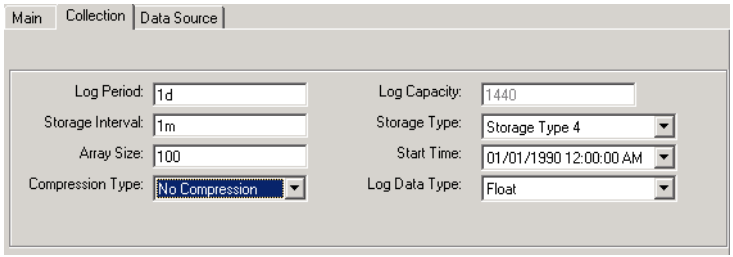


Figure 205. Collection Tab

To determine the correct array size for the profile log via the Control Connection aspect of the selected OPC tag, Figure 206, select the Control Connection aspect, check the **Subscribe to Live Data** check box, then read the value for ProfileArray (for example, 600 in Figure 206).

If Array Size is configured too large or too small, the file will not be appropriately sized for the expected log period. For instance, consider a case where the log period is eight days and the machine generates 200 data points per scan. If the array size is set to 100, the log will only be capable of holding four days worth of data. If the array size is set to 400, the log will store 16 days worth of data, and needlessly consume more disk space.

Log Period

Disk space is allocated to logs according to their log capacity, which is a function of the log time period and storage interval. The log period can be adjusted on an individual log basis to keep log capacity at a reasonable size for each log. For profile logs, log capacity is calculated as: log capacity = (log period/storage interval). Enter the log period as an integer value ≤ 4095 with time unit indicator for **Weeks**, **Days**, **Hours**, **Minutes**, **Seconds**. The maximum is **4095 Weeks**. For example: **40d** = 40 days, **8h** = 8 hours. The maximum is **4095 Weeks**.

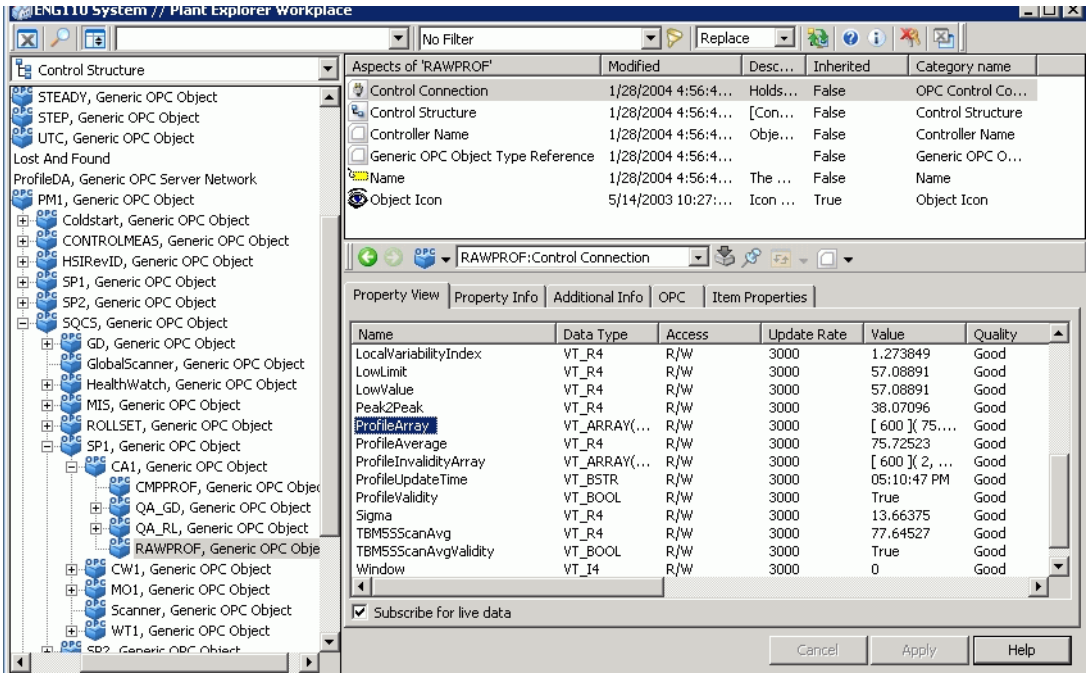


Figure 206. Determining the Proper Array Size

Storage Interval

This is the time interval at which data is stored in the log. Enter the storage interval as an Integer value with a time unit indicator: **W**eeks, **D**ays, **H**ours, **M**inutes, **S**econds. For example: **40d** = 40 days, **8h** = 8 hours. The maximum is **4095 Weeks**.

Machine Position

This attribute is located on the **Data Source** tab, Figure 207. Enter the name of the object which holds the Reel Footage value. An example is shown in Figure 208. Using this example, the Machine Position value would be PM1/SQCS/MIS/RL/ACCREP:ReelFootage.

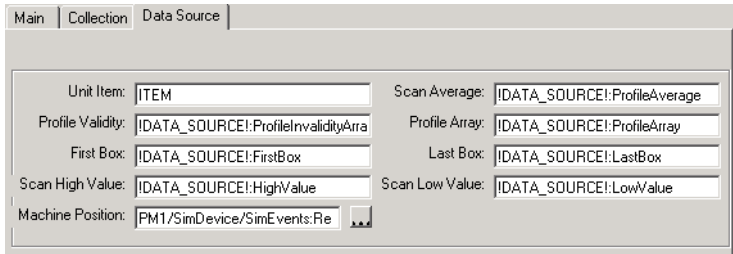


Figure 207. Data Source Tab

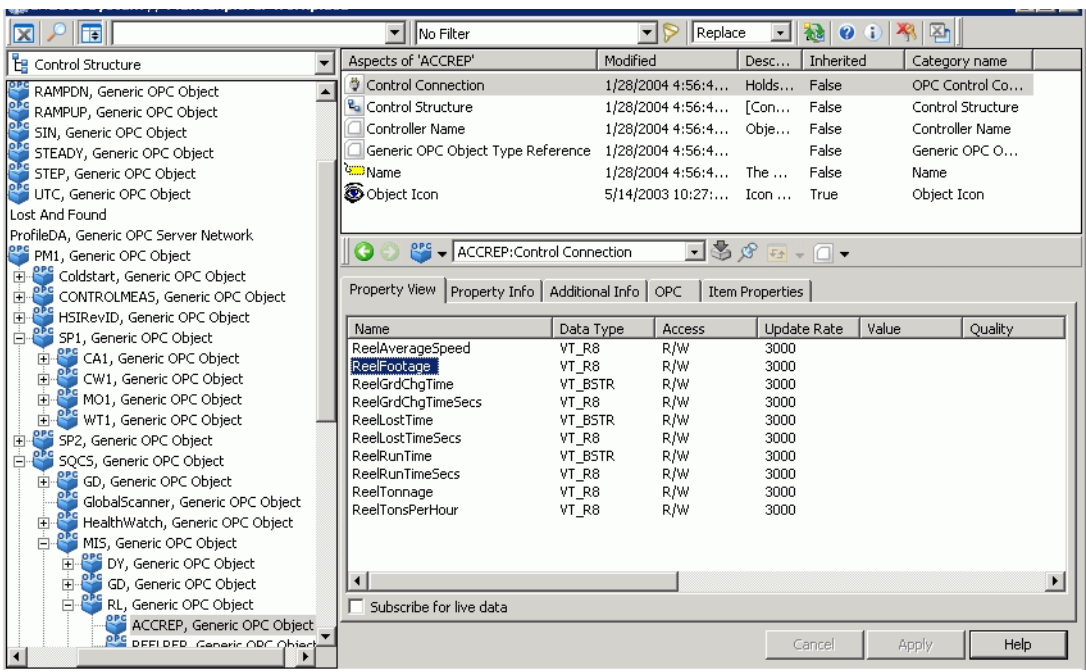


Figure 208. Finding the Object Which Holds the Reel Footage Value

Profile Log Attributes Summary

The profile log configuration attributes are described in:

- Table 30 - Main Tab.

- Table 31 - Collection Tab.
- Table 32 - Data Source Tab.

Table 30. Main Tab

Attribute	Description
Data Source	Refer to Data Source on page 304.
Log Set	Refer to Log Set on page 306.
2nd Log Set	Optional second log set.
Archive Group	Typically, to archive the profile log, assign the log to an archive group. For details regarding archive groups, and the archive function in general, refer to Section 11, Configuring the Archive Function.
Description	Enter an optional descriptive text string for this log.
Start State	This is the initial log state when History Services is restarted via PAS. Choices are: START_INACTIVE START_ACTIVE START_LAST_STATE (default) - restart in state log had when node shut down
Collection Type	This defaults to OPC and CANNOT be changed.
Collection Mode	This defaults to Synchronous and CANNOT be changed.

Table 31. Collection Tab

Attribute	Description
Log Period	Refer to Log Period on page 307.
Storage Interval	Refer to Storage Interval on page 308.
Array Size	Refer to Array Size on page 306.
Compression Type	This attribute is not implemented in this version of History Services and may be left at its default value (No Compression).

Table 31. Collection Tab (Continued)

Attribute	Description
Log Capacity	This is the maximum number of entries for a log. If a log is full, new entries replace the oldest entries. This field is not configurable. Log capacity is calculated as follows: Log Capacity = (Log Period/ Storage Interval).
Storage Type	Storage Type defaults to type 4 which is the only valid storage type for profile logs. This attribute cannot be modified.
Start Time	This is the earliest allowable system time that the log can become active. If the specified start time has already past, the log will begin to store data at the next storage interval. If the start time is in the future, the log will go to PENDING state until the start time arrives. Enter the start time in the following format: 1/1/1990 12:00:00 AM(default) The order of the month, day, and year and the abbreviation of the month is standardized based upon the language being used.
Log Data Type	Data type defaults to Float which is the only valid data type for profile logs. This attribute cannot be modified.

Table 32. Data Source Tab

Attribute	Description
Machine Position	Refer to Machine Position on page 308.
All other attributes on this tab.	These are the names for the objects which hold the quality measurements for the profile log. The root object name is specified via the Data Source attribute on the Main tab, and is represented as !DATA_SOURCE! in the respective fields on this tab.

Configuring Reports

History Profiles may be configured to collect data for: reel turn-up events, product grade change events, day shift, and roll setup. This functionality is configured via the Reel Report Configuration Tool.

The Reel Report Configuration Tool is used to build these four reports for each machine in the plant. An example is shown in Figure 209.

For each machine specified, the Reel Report Configuration Tool provides a template to facilitate report configuration. This template provides place holders for each of the signals/variables which must be specified. Add signals/variables as required.

To collect data for a particular report, specify the tag corresponding to each signals/variable. The signals/variables are read from OPC data items stored in an OPC server.

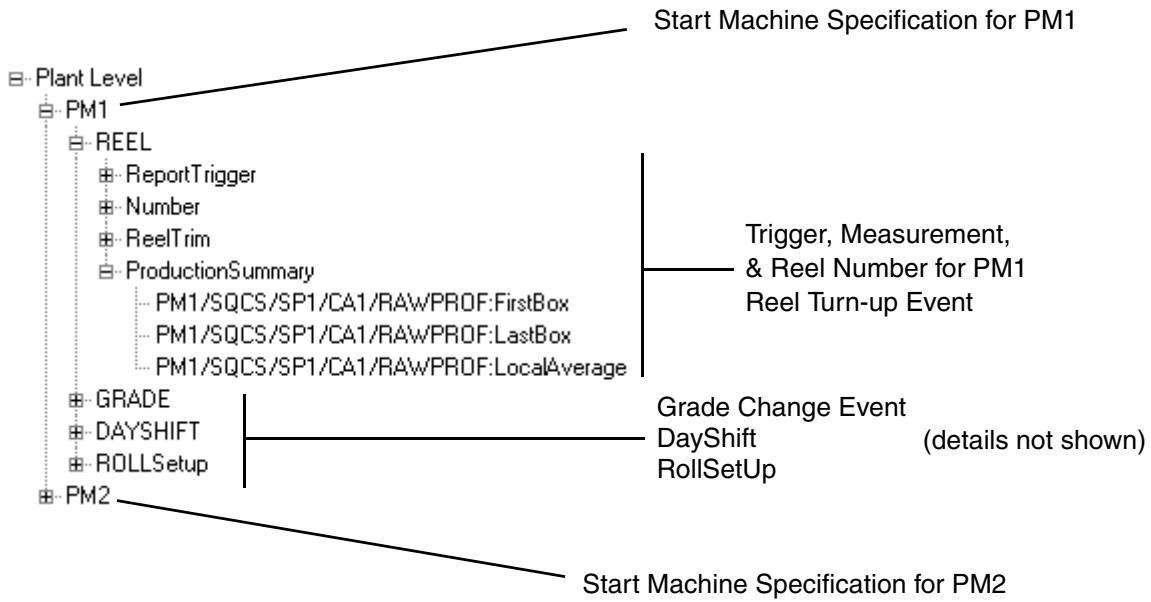


Figure 209. Example Tree Model for Reel Reports

Launching the Reel Report Configuration Tool

To launch the Reel Report Configuration Tool, from the Windows task bar, choose: **Start>Programs>ABB Industrial IT 800xA>Information Mgmt>Profiles>Reel Report Configuration.**

When opening this window for the first time, the tree is empty except for the Plant Level object, Figure 210.

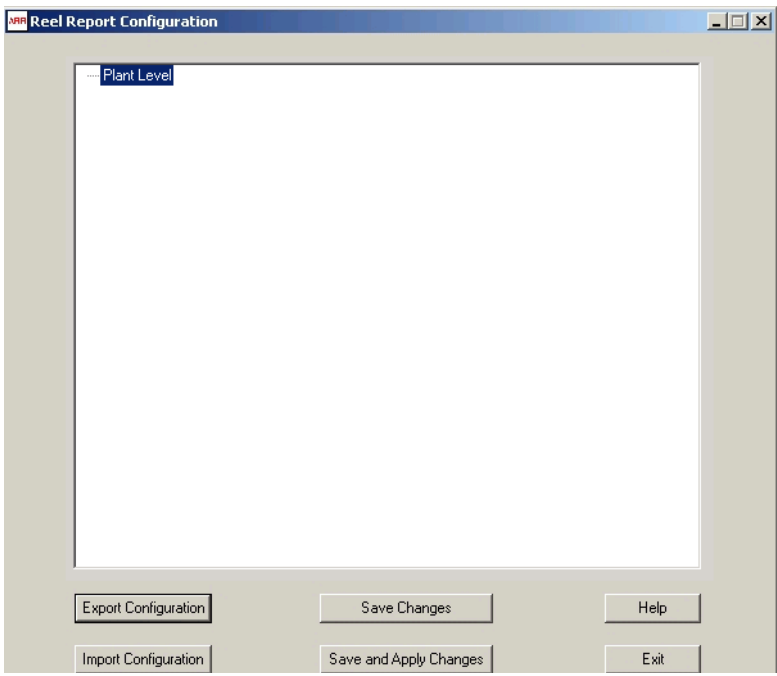


Figure 210. Reel Report Configuration Tool

Start by adding a machine, and then specify the OPC data items to be read for that machine.

Adding a Machine to the Hierarchy

To add a machine:

1. Select **Plant Level**, right-click and choose **Add New Tag** from the context menu, Figure 211.

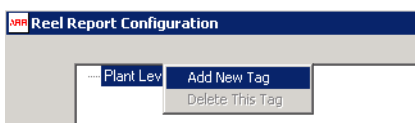


Figure 211. Adding a New machine

This adds a new machine with the default structure as shown in Figure 212. The default machine structure provides a default specification for REEL turn-up, GRADE change, DAYSHIFT, and RollSetup.

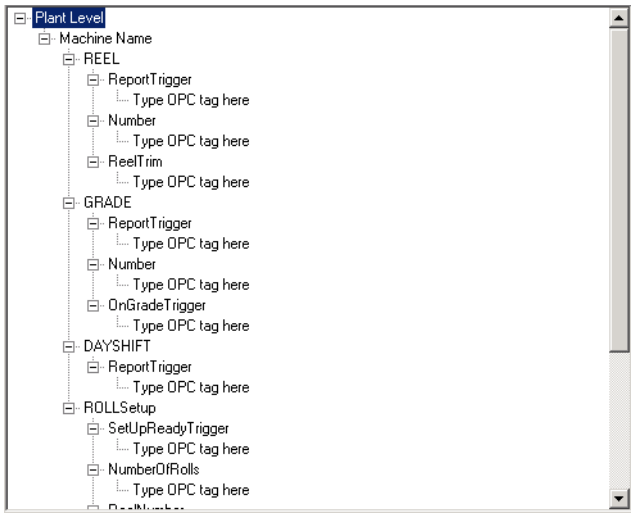


Figure 212. Default Machine Structure

2. Specify the machine name. To do this, click on **Machine Name**, and then enter a new name, for example: **PM1**, Figure 213. This machine name must also be used as the name for the log set to which all profile logs for this machine will be assigned.

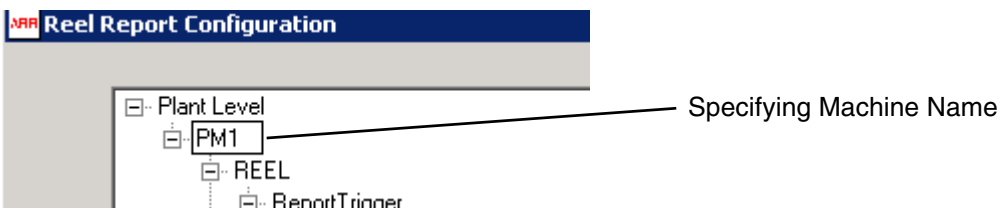


Figure 213. Specifying the New Machine Name

Saving Changes while Working

Periodically saving changes is recommended. Either save after every change or when done. To continue editing, click on the next item that to be changed.

Click the **Save Changes** button to periodically save changes. This saves the configuration, but does not apply the changes to the Profile server. Exiting without saving displays a prompt to save the changes.

When finished, apply the changes to the Profile server by clicking **Save and Apply Changes**.

Pressing **Enter** after every change requires responding to two prompts after every change. This may be annoying when configuring a large report.

Specifying Signals for Reel Turn-up

To collect data for a particular report specify the tag corresponding to each signals/variable. If a tag is not specified for a particular report, the data will not be collected and the report will not be generated. Continue by specifying the signals for Reel Turn-up section. This section is required for all reports. The other sections (Grade, DayShift, and RollSetUp) are optional.

The default specification for Reel Turn-up (REEL) provides place holders for the Report Trigger, reel number counter, and ReelTrim. The Report Trigger signals the occurrence of each reel turn-up on the machine. The Number counter increments the reel number for each reel turn-up that occurs on the machine.

These signals are mandatory in order to generate a Reel Turn-up report. They cannot be deleted. One or more additional measurements can be added to collect on an optional basis. To specify the OPC tags for these signals:

1. Click on the OPC tag place holder for the REEL Report Trigger, Figure 214.

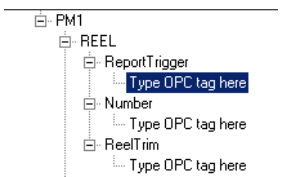


Figure 214. Selecting the REEL Report Trigger Place Holder

2. Enter the OPC tag for the trigger that indicates the occurrence of a reel turn-up. An example is shown in Figure 215.

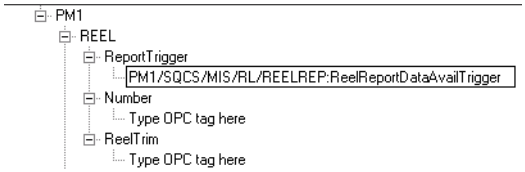


Figure 215. OPC Tag for REEL Trigger Specified

3. Repeat this procedure to specify the OPC tag for the REEL Number counter and ReelTrim. An example completed specification is shown in Figure 216.

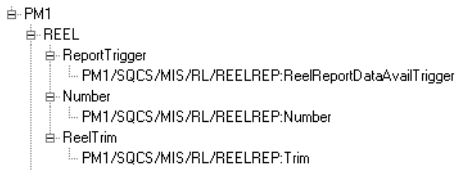


Figure 216. Mandatory Signals for Reel Turn-up Completed

4. Add additional signals for this report as follows:

   a. Select **REEL** and choose **Add New Tag** from the context menu. This adds a new Report Heading in the tree under REEL, Figure 217.

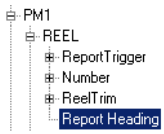


Figure 217. New Report Heading Added

   b. Select the Report Heading place holder and enter the new report heading, for example: **ProductionSummary**. There are no specific requirements for the heading, except that it must be unique within the event (in this case, within the REEL for PM1).

c. Right-click on the report heading and choose **Add New Tag** from the context menu. This adds an OPC tag place holder for the new measurement.

d. Specify the OPC tag for the measurement. This procedure is the same as for configuring any other signal. An example is shown in Figure 218.

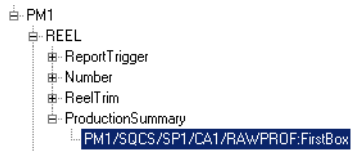


Figure 218. New Measurement Specified

Multiple report headings can be added under REEL. Multiple measurements under each report heading can also be added. An example is shown in Figure 219. Remember to save periodically.

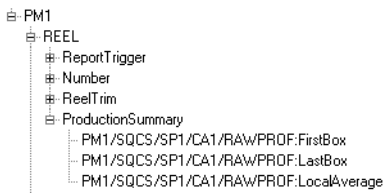


Figure 219. Example Completed Report

Specifying Signals for Grade Change

The default specification for Grade Change (GRADE) provides place holders for the Report Trigger, On Grade Trigger, and Reel Number Counter, Figure 220. The Report Trigger signals the end of the current grade. The On Grade Trigger signals that product quality has reached the new grade. The Number counter increments the reel number for each reel turn-up that occurs on the machine.

These signals are mandatory in order to generate a Grade Change report. They cannot be deleted. One or more additional measurements can be added to collect on an optional basis. The procedure for specifying the signals is the same as the procedure described in Specifying Signals for Reel Turn-up on page 315.

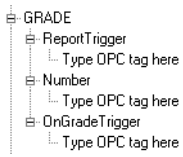


Figure 220. Grade Change Default Specification

Configuring the DAYSHIFT

The default specification for DAYSHIFT has a place holder for the Report Trigger, Figure 221. This signals when to generate a new DAYSHIFT report. This signal is mandatory in order to generate the DAYSHIFT report. The procedure for specifying this trigger, and for adding other signals is the same as for specifying the reel turn-up report as described in Specifying Signals for Reel Turn-up on page 315.

Figure 221. Example DAYSHIFT Report

Configuring ROLLSetup

This report is used to examine sets and rolls within a reel. This setup divides the reel lengthwise into a specified number of sets. Each set is divided by width into a specified number of rolls. This is illustrated in Figure 222.

The default specification for ROLLSetup is shown in Figure 223. The variables are described in Table 33. These signals are mandatory in order to generate a ROLLSetup report. They cannot be deleted. One or more additional measurements can be added to collect on an optional basis. The procedure for specifying the signals is the same as the procedure described in Specifying Signals for Reel Turn-up on page 315.

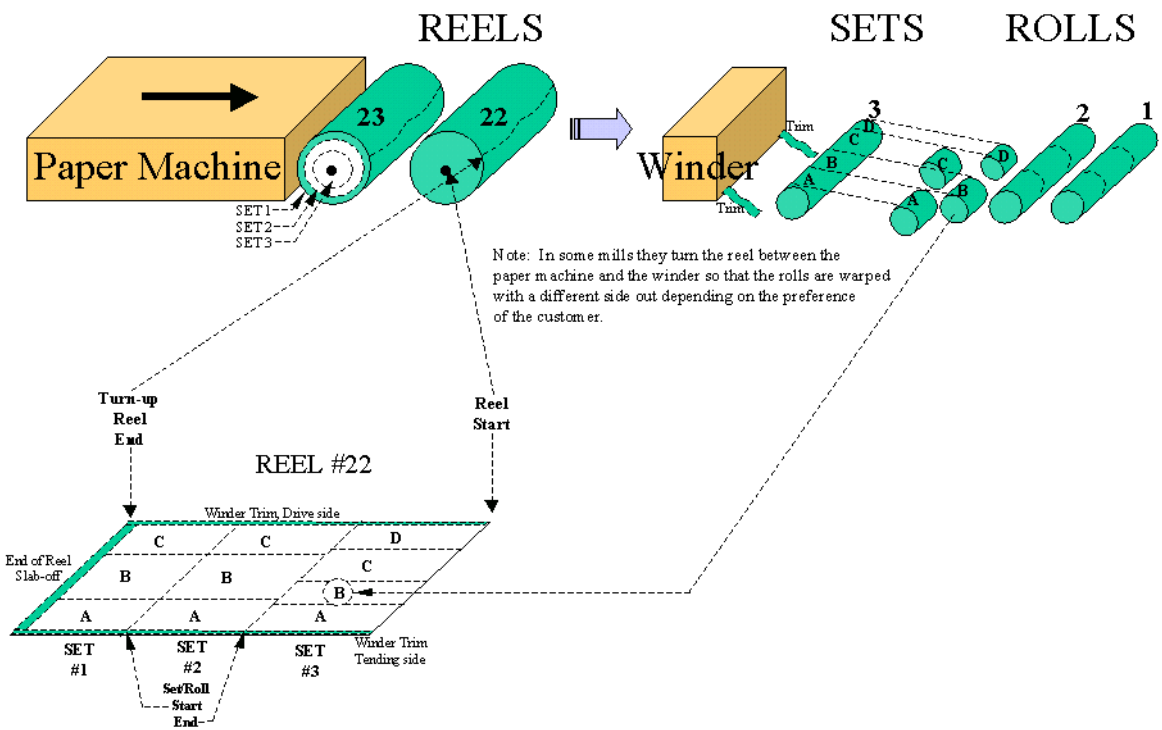


Figure 222. Roll/Set

Table 33. ROLLSet Report Specification

Signal	Description
NumberOfRolls	Indicates the number of rolls for the current set being processed.
ReelNumber	Indicates the reel number for the current reel.
ReelTrim	Indicates the width of the reel.
RollTrimArray	One value for each roll in the current set indicates the width of each roll.
SetFootage	Indicates the actual length of the current set.

Table 33. ROLLSet Report Specification

Signal	Description
SetLength	Indicates the recommended or target length of any set for this reel.
SetNumber	Indicates the current set number.
SetupReadyTrigger	Indicates when the current set has been completed.

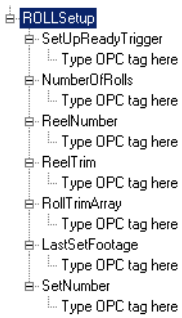


Figure 223. Default Specification for ROLLSetup

Deleting Objects from the Reel Report Configuration Tool

Individual measurements can be deleted that were added under machine objects, or an entire machine can be deleted. Deleting a machine deletes the entire machine structure. Individual Trigger or Number specifications can not be deleted for a machine, nor are Reel or Grade events permitted to be deleted.

To delete an object,

1. Select the object and choose **Delete This Tag** from the context menu, Figure 224. A confirmation message is displayed in the AdvHistProfCfg dialog: “Selected item and all subitems will be deleted”.

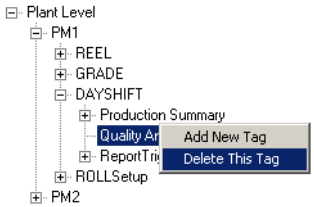


Figure 224. Example, Deleting a Machine

2. Click **Yes** to confirm, or **No** to cancel the delete operation.

Exporting/Importing a Reel Report Configuration File

A reel report configuration can be exported to a binary (.dat) file. This is used to save and re-use a reel report configuration. To export the reel report configuration click Export Configuration. The configuration is saved to a file named **PDLConfig.dat** in C:\Documents and Settings\All Users\Application Data\ABB\IM\History\tmp. Typically, one export file at a time is maintained. Attempting to export a new configuration when an export file already exists in the tmp directory will cause a prompt for overwriting the existing file.

Maintain multiple export files by renaming the existing files before exporting a new one. Keep in mind that the import function will always import the file named **PDLConfig.dat**.

A valid export file can be used to create reel report configurations on other Profile Historian servers. To do this, click **Import Configuration**. If a reel report configuration already exists on the server, a prompt for overwriting the existing configuration will be displayed.

Archiving Profile Log and Reel Report Data

Reel and grade events are stored as level 1 PDL tasks. Along with the PDL task, History also creates a History Association for each profile log in the log set corresponding to the machine (Section 6, Configuring Log Sets). When a PDL task having History Associations is archived, the archive function automatically archives the profile log data pointed to by the PDL History Association. PDLs are described in detail in *System 800xA Information Management Profile Historian Operation (3BUF001121*).*

Since profile log data is automatically archived with the PDL data, only the PDL archive function for the Profile Historian application will need to be configured. For details regarding this procedure, refer to PDL Archive Configuration on page 362.

Activating Profile Logs

The same methods for activating and deactivating numeric logs also apply to Profile logs. The basic method is to use the Composite Log window. For details, refer to Starting and Stopping Data Collection on page 439.

Section 11 Configuring the Archive Function

The archive function supports permanent offline storage for:

- Numeric process data stored in history logs.
- Finished reports scheduled and executed via Application Scheduler, and stored in report logs, or as completed report objects in the Scheduling structure.
- Production data from batch control applications and stored in Production Data Logs (PDLs).
- Alarm messages, event messages (including audit trail events), and system messages generated by 800xA system applications and stored in message logs.

Alarm, event, and system messages can also be archived directly from the 800xA System alarm/event message services, although these messages are more typically archived from the message logs in History Services.

Without the archive function, when a history log becomes full, the oldest entries will be overwritten by newer entries. When archiving is used, the contents of specified logs are copied to a designated archive media to prevent the loss of critical historical data. Any archive created from a previous version of Advant Historians or System 800xA Historians can be read in the latest version of System 800xA Information Manager.

Archive Media Supported

Archive Devices use a portion of the IM's disk space. An off-line storage media such as CD, DVD, and Blue Ray disks are used as the allotted disk space becomes full to make backups that can be stored in a safe location for playback later or on a different system. A network share can also be used to create off-line backups as the disk space is consumed.

Magneto-optical (MO) disks are supported to allow Windows Enterprise Historians (EH) archives to be accessed in the latest Information Manager. The most common use case for this is when an Advant System with EH is migrated to System 800xA. While MO disks can be used for archiving System 800xA data, the media size is small when compared to the typical data a System 800xA Information Manager stores in a typical system. If MO disks are used they require manual formatting and supervision to remove media when it is full. In typical systems, this could happen every few hours and is not practical.

Archive Configuration

Archiving is managed by one or more archive device objects which are configured in the Node Administration structure. An archive device is a logical entity that defines where and how archive data is written. Every MO or disk drive used for archiving must have at least one archive device aspect configured for it. A single drive may have several archive devices configured for it to satisfy different archive requirements. For example, more sensitive data may be archived through a separate device which is configured to prevent automatic overwriting of stored data.

Archiving may be scheduled to occur on a periodic or event-driven basis through the Application Scheduler, or execute manual archive operations on demand. For manual archives, if the specified time range has no samples, no data will be archived. For scheduled archives, even if no new samples were collected, at least one sample (the last valid point) will be archived. Each archive operation is referred to as an archive entry.

Scheduled archiving is implemented through archive groups. These are user-defined groups of logs which are archived together as a single unit. Scheduling instructions for archive groups are specified in job description objects created in the Scheduling structure. The schedules are associated with their respective archive groups through an Archive Action aspect attached to the job description object. Manual archiving may be done on an archive group basis, or by selecting individual logs.

Accessing Archived Data

Archive volumes support viewing of archive data (through the corresponding archive volume aspect). Partition the hard disk media into any number of archive

volumes. MO media will only have one archive volume. Archive volumes are automatically created for all removable ROM drives (DVD, Blu-ray and CD drives) to support viewing of archive data from them. Further, additional read-only volumes can be created for reading archive volumes that have been copied to a mapped network drive, or for viewing archive files that have been copied to the local drive.

Archived historical data may be viewed via desktop applications such as DataDirect, Desktop Trends, and 800xA system tools. Archived data may also be incorporated into reports which are created with a report building package such as DataDirect or Crystal Reports, and then scheduled and executed via the Application Scheduler.

In order for client applications to access archived log data, the archived logs must be restored from the archive media to the restored history database, or the applicable archive volume must be published. The method used depends on the type of entry. Oracle-based log types (reports and PDL) and completed report objects must be restored.

For numeric (property) logs, the archive volume where the archived data resides must be published. The contents of a complete volume, or even multiple volumes can be published in relatively few steps. This method does not require the archive data to be written to Oracle tables in the restored database. Since Oracle tablespace is limited, the publishing method is used to expose a larger amount of archive data for client applications at any one time.

Message log data stored on the published volume is also published. This is used to access the message log data via alarm/event lists configured in the 800xA System. Archive message logs must be restored to access the archive data via the SQL-based tools in DataDirect, Desktop Trends, and Crystal Reports.

Planning for Reliable Archive Results

For each IM and its configuration, the amount of archive data is dependent on the Information Management configuration and how much of the data is being archived. It is not required to archive all IM runtime data. The selection of which logs (Numeric, Message, Report, Profile and Product Data Log) are archived is based on any requirements to preserve the data for future review. If the runtime system has six months of online data and there is a requirement to have data for the last five years. Archive must be configured to store the data for review when necessary.

In order to ensure that the archives are created that will meet the any long term storage requirements, follow the guidelines specified on Archive Configuration Guidelines on page 326.

Archive Topics

- Configuring Archive Devices on page 330.
- Configuring Archive Groups on page 341.
- Setting Up the Archive Schedule for an Archive Group on page 352.
- Adding a Read-Only Volume for a Mapped Network Drive on page 361.
- PDL Archive Configuration on page 362.
- Accessing and managing archive data is described in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

Archive Configuration Guidelines

Take the time to plan the archive function to ensure reliable and predictable operation. Consider the quantity and storage rate for History logs, and the size of the archive media. The following general steps should be used to configure archive:

1. Determine how much data will be archived per day.
2. Determine if an extra disk is needed for the archive devices, based on disk I/O.
3. Determine the best archive volume size based on the archive frequency and desired final medium (ISO or disk).
4. Configure archive groups and timed archive interval.
5. Configure timed archive.
6. Maintain the generated archive volumes.

Example History Database

The following example configuration will clarify these guidelines.

- 500 logs @ 2-second storage rate (15,000 points per minute), 20-day log period.

- 3000 logs @ 15-second storage rate (12,000 points per minute), 20-day log period.
- 2000 logs @ 60-second storage rate (2000 points per minute), 20-day log period.

This example covers numeric data only.

To proceed with planning for this configuration, follow the procedures described in Determining the Quantity and Size of Archive Groups on page 327.

Some calculation results are approximated or rounded. For instance, a calculation that yields 3.3 archive groups is rounded to 4.

Determining the Quantity and Size of Archive Groups

Carefully plan archive group configuration with regard to quantity and size (number of logs). This will help to avoid accumulating excessive amounts of archive data that exceed the capacity of the archive media.

The size of an archive entry and the amount of data written to an Archive Volume each time an Archive Group is archived, is determined by the number of logs assigned to the archive group, the archive interval (time elapsed between archive operations for the archive group). The amount of data archived at any one time must be limited based on the individual archive volume size. The archive volume size should be at least three times the maximum size for an archive group entry.

Use the following procedure to calculate a reasonable size for an archive entry. This procedure must be done for each unique log configuration (storage rate/log period). To illustrate, the following calculation is for the log configuration that has 500 logs @ 2-second storage rate, and 20-day log period:

1. Calculate the archive interval. It is recommended that an interval which is no greater than 1/4 the log time period be used.

A smaller archive interval is recommended for large log periods (one year). This limits the amount of data that may be lost in the event of a hardware failure. Do not make the archive interval too small. This causes a greater percentage of the disk to be used for overhead.

For a 20-day log period, the recommended archive interval = 1/4 * 20 = **5 days**.

2. Calculate the amount of archive data that one log generates over the archive interval. The following calculation is based on the 2-second storage interval.

   21 bytes per archive entry * 30 points per minute * 60 minutes * 24 hours * 5 days = **4.5 megabytes**[1]

For systems using deadband, the amount of data per archive interval will be reduced.

3. Calculate the maximum archive entry size for any archive group supporting this log configuration:

   size of one platter side / (reasonable estimate for number of missed archives + 1) = 2.2 gigabytes/4 = **550 megabytes**.

   By limiting each scheduled archive operation to 550 megabytes, archive data for up to three missed archive attempts can be accumulated before exceeding the capacity of the 2.2 gigabytes archive volume in an archive device. If the cumulative size of the current archive entry and the previous failed attempts exceeds the capacity of the archive media, the archive entry will be split into some number of smaller entries such that one entry will not exceed the available space.

4. Calculate number of archive groups needed for the 500 logs @ 2-second storage rate. This is done in two steps:

   a. # of logs per archive group = result of step 3 / result of step 2 = 550 megabytes / 4.5 megabytes = **125 logs per group**

   b. # archive groups = total # logs of this type / # logs in one archive group= 500 / 125 = **4 archive groups**

Steps 1-4 must be repeated for each of log configuration (storage rate, log period combination).

1. 4.5 Megabytes = $4.5*10^6$

Deadband Considerations

Deadband[1] decreases the rate at which the platter is filled.[1] For example, if deadband results in only 70% of the samples being saved, the rate at which the platter is filled will decrease as shown below:

Time = 2.2 gigabytes / (29K ppm *** 0.7** * 60 *24 * 21) **= 3.6 days**

Summary of Results for Other Log Configurations

Rather than show detailed calculations for the other log configurations, the results for those configurations are summarized in Table 34.

Table 34. Archive Group Calculations Summary

Configuration	One Log's Data/ Archive Interval	Logs/Group	No. of Archive Groups
500 @ 2-sec.	21bytes***30**ppm*60m*24h*5d = **4.5Mbytes**	550/4.5 = **125**	500/125 = **4**
3000 @ 15-sec.	21bytes***4**ppm*60m*24h*5d = **605Kbytes**	550/.605 = **909**	3000/909 = **4**
2000 @ 60-sec.	21bytes***1**ppm*60m*24h*5d = **151Kbytes**	550/.151 = 3642	2000/3642 = **1**

Configure Archiving

Configure the archive application according to the results of your calculations:

1. Configure the archive device as described in Configuring Archive Devices on page 330.

   The recommended Device Behavior for MO Drive is **Stop When Full**. This is because the platter must either be turned over, or replaced when the current side is full.

   For Disk Drive, the recommended Device Behavior is **Wrap When Full**. This will support the archive backup scenario whereby the contents of a volume is written to an ISO Image file, or a shadow copy is created on a network file server when the volume becomes full. The Disk Drive device can be configured to re-initialize itself when all volumes are full, and the Overwrite Timeout has expired.

1. Deadband is defined in percentage of engineering units range.

2. Configure archive groups as described in Configuring Archive Groups on page 341. Configure the required number of archive groups for each log configuration as indicated in Table 34.

   Guidelines for Message Logs

   Create Message Archive Groups as needed. If a message log is created with a capacity of 10,000 and expect 2000 messages per day, the message log will have approximately 5 days of messages. Therefore, an archive group time interval of one or two days is reasonable. The one-day interval should be sufficient.

3. Create a job to schedule archival according to Setting Up the Archive Schedule for an Archive Group on page 352. In this case, schedule archiving to occur once every five days.

Configuring Archive Devices

This section describes how to configure archive devices to define where and how data will be archived on a specific archive media. Two device types are supported:

- **MO Drive** - requires removing and replacing platters as they become full.
- **Disk Drive** - The hard disk may be partitioned into multiple volumes which are sized to match ROM media.

Several disk archive devices can be configured on the same partition to satisfy several different archive schemes.

Removed archive volumes don't recapture used disk space. The second step of manually deleting the extra or deleted volumes ensures archive data is not deleted as a result of a configuration mistake.

When deleting volumes from a Disk Drive archive device, delete the folder manually. Look for a folder under the Device File with the name nnArch where nn is the volume number. Delete folders that match the volumes that were deleted from the device.

Refer to the computer's documentation for instructions on connecting storage devices to the computer where the archive service runs.

The operating parameters for each archive device are specified in the corresponding archive device aspect. This requires adding one or more archive device objects for each archive media (MO or disk drive), and then configure the archive device aspects. For instructions, refer to Adding an Archive Device on page 331.

Adding an Archive Device

Archive Device objects exist in the Node Administration Structure. Each node, where the archive service runs, has an **Industrial IT Archive** object under the **Industrial IT Archive** service provider. The aspect list for this object has an **Archive Service Aspect** which facilitates adding Archive Device objects (as well as Archive Group and Archive Volume objects).

To add an Archive Device object (reference Figure 225 for steps 1-4):

1. In the Plant Explorer, select the Node Administration structure.
2. Expand the object tree for the node where the archive device is being added (for example, TAR105 in Figure 225).
3. In the object tree for the selected node, expand the **Industrial IT Archive Service Provider** and select the **Industrial IT Archive** object.
4. Select the **Archive Service Aspect** from this object's aspect list.

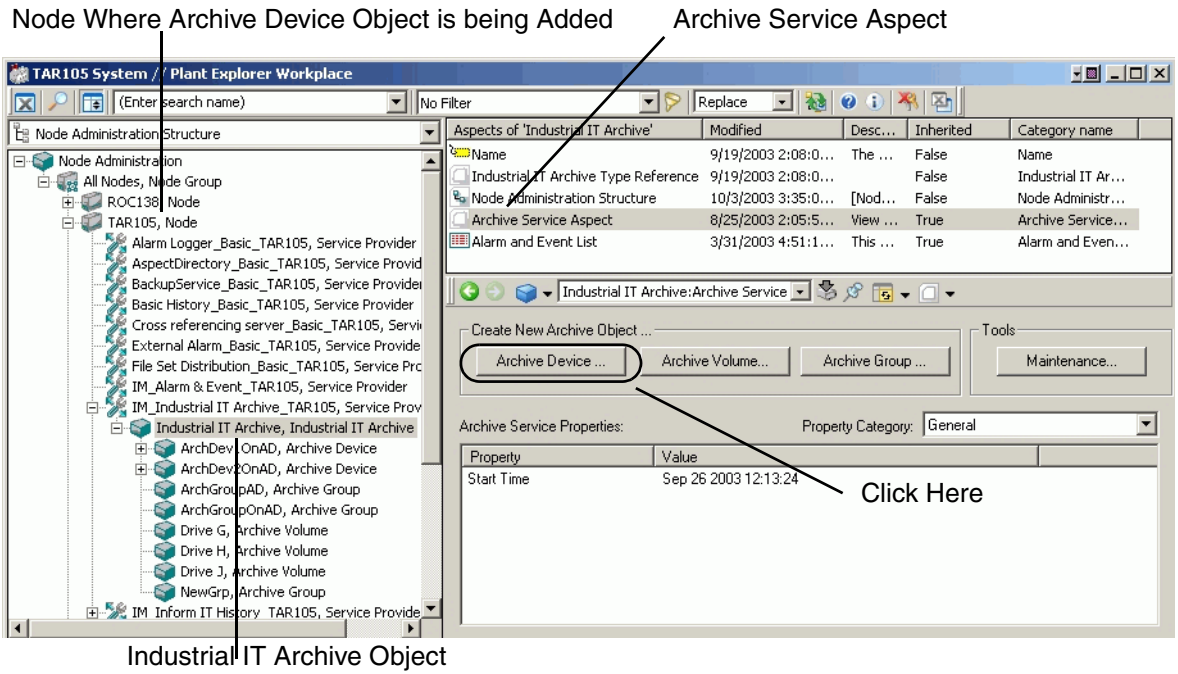


Figure 225. Adding an Archive Device in the Node Administration Structure

5. Click **Archive Device** in the Create New Archive Object section. This displays the New Archive Device dialog, Figure 226.

6. Enter a name for the object in the Name field, for example: ArchDev2OnAD, then click **OK**.

Keep the **Show Aspect Config Page** check box checked. This will automatically open the configuration view of the Archive Device aspect.

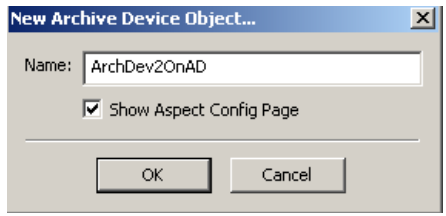


Figure 226. New Archive Device Object Dialog

This adds the Archive Device object under the Industrial IT Archive object and creates an Archive Device Aspect for the new object. Use this aspect to configure the device as described in Archive Device Aspect below.

Archive Device Aspect

The Archive Device aspect Config view is shown in Figure 227.

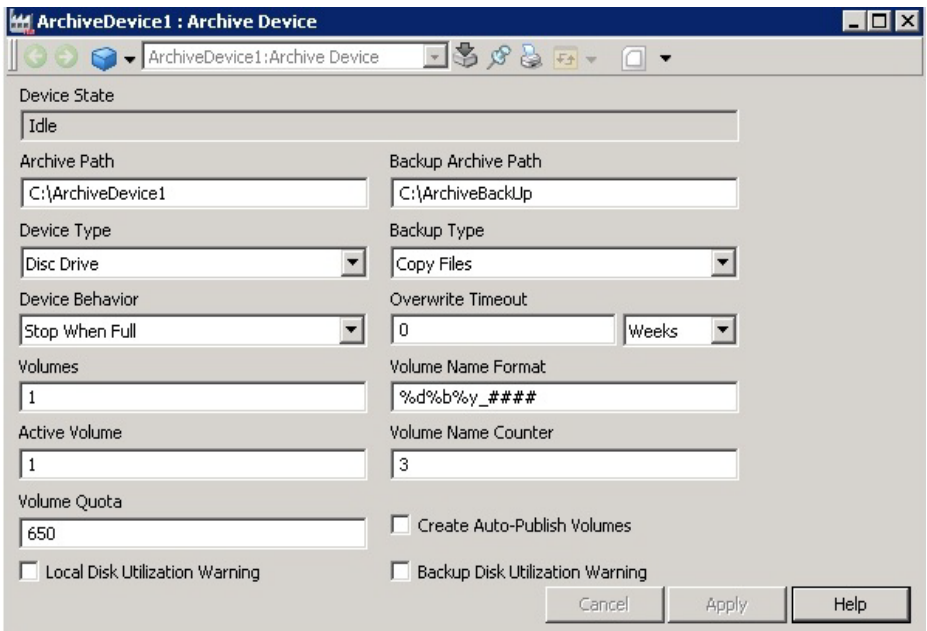


Figure 227. Archive Device Configuration View

This aspect also has a main view for managing the archive volumes on the archive device. This is described in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

The archive device operating parameters which must be configured, and the manner in which they are configured depends somewhat on the type of device (MO or hard disk). Review the guidelines below. Details are provided in the sections that follow. When done, click **Apply** to apply the changes.

Guidelines

To configure an archive device, first specify the type of media for which the device is being configured, and then configure the operating parameters according to that media.

The Archive Path specifies the drive where archive data will be written. If the archive media is a hard disk, specify the directory.

The Device Behavior and Overwrite Timeout fields are used to specify how archiving will proceed when the current archive media (volume) becomes full.

When using a disk drive, partition the disk into one or more volumes. This is not applicable for MO drives. To configure volumes, specify the number and size of the volumes, and set up volume naming conventions. Optionally, specify whether or not to have archive volumes automatically published when the volumes have been copied to a removable media (CD or DVD) and the media is remounted.

For disk-based archiving, it is recommended that automatic backup of archive data to another media be configured. This way when a volume becomes full, an ISO image or shadow copy, or both are created on another media. This is not applicable for MO media which are removable and replaceable.

Optionally, specify whether or not to generate an alarm when the archive media (or backup media) exceed a specified disk usage limit. If these features are enabled, the alarms recorded in the 800xA System message buffer.

Attempting to apply an invalid parameter setting causes the Device State field to be highlighted in red to indicate an invalid configuration.

Device Type

Specify the type of media for which the archive device is being configured:

MO Drive (single magnetic/optical drive).
Disk Drive

Archive Path

This specifies the location where data will be archived. If the Device Type is an MO drive, enter the drive letter, for example: **D**: or **E**: If the Device Type is Disk Drive, specify the full path to the directory where data will be archived for example: **E:\archive** (root directory is not a valid entry for disk drive).

Device Behavior and Overwrite Timeout

Device Behavior and Overwrite Timeout fields determine how archiving will proceed when the current archive media (volume) becomes full. For Device Behavior the choices are:

- **Stop When Full** (default, and recommended for MO drives) - Return device full error when the device is full.
- **Advance When Full** - Advance the active volume until one of the following is found: a valid media with storage capacity, or a media that is not initialized. If a volume that is not initialized is found, initialize the volume and start archiving to it. If not, return device full error.
- **Wrap When Full** (Recommended for Hard Disk devices) - First look for a valid or not-initialized media. If a valid media is found, start archiving to it. If a not-initialized media is found, initialize it and then start archiving to it. If neither of these is found, look for a full media whose Overwrite Timeout has expired, and with the earliest last archive time. Then reinitialize that media and start archiving to it. Wrap When Full requires a finite overwrite timeout.

The **Overwrite Timeout** specifies the delay between the time when a media becomes full and the time when the media can be reinitialized (overwritten). Set this to the duration for which the data must be preserved. For example, to preserve data for one year, set the Overwrite Timeout to 365 days. This means 365 days after the LAST data has been written to the media, the media can be automatically initialized by the archive service if the media is in the drive.

Use the pull-down list for this field to select the units: Hours, Days, or Weeks. Then enter the number of units. The default is **0** which means infinite delay. To get

virtually no delay, enter a very small delay, for instance: 1 second. The Overwrite Timeout is stored on the media, so removing the media from the drive and then replacing it will not affect the Overwrite Timeout. Overwrite Timeout can be changed via the Initialize dialog when a media is initialized for manual archiving.

When using archive backup (Configuring Archive Backup on page 338) follow these guidelines for configuring Device Behavior and Overwrite Timeout.

When using archive backup, it is recommended that the device behavior be set to Wrap When Full, and set the overwrite timeout to allow for periodic checking of the backup destination to make sure its not full. The overwrite timeout should be two to three times the interval that is checked for the backup destination. If checking once a week, then set the overwrite timeout to two or three weeks. Also, the number of volumes and volume size must be configured to hold that amount of data (in this case two to three weeks). This is covered in Configuring Volumes on page 336. Following these guidelines will ensure reliable archival with ample time to check the backup destination and move the archive files to an offline permanent storage media.

Configuring Volumes

Hard disks can be partitioned into multiple volumes. This involves configuring the number of volumes, the active volume, volume quota, volume name format, and volume name counter. Specify whether or not to have archive volumes automatically published when the media where the volumes have been copied are remounted. Configuring volumes is not applicable for MO drives.

The **Volumes** field specifies the number of volumes on the media that this archive device can access. The maximum range is **64**. The quantity specified here will result in that number of Archive Volume objects being created under the Archive Device object after the change is applied, Figure 228.

Figure 228. Specified Number of Volumes Created Under Archive Device Object

Volume Quota is the partition size for each volume in megabytes. For example, a 20 gigabyte hard disk can be partitioned to five 4000-megabyte partitions where 4000 (MB) is the Volume Quota and five is the number of volumes as determined by the Volumes field. Typically, size volumes on the hard disk to correspond to the size of the ROM media to which the archive data will be copied, for example 650 MB for CD ROM, or 4000 MB for DVD. The minimum Volume Quota value is 100M. The maximum Volume Quota is 100 GB.

The **Volume Name Format** is a format string for generating the volume ID when initializing a volume during timed archive. Enter any string with/without format characters (all strftime format characters are accepted). This can be changed when manually initializing a volume. The default value is '%d%b%y_####':

%d = day of month as a decimal number [1,31].
%b = abbreviated month name for locale, for example Feb or Sept.
%y = year without century as a decimal number [00,99].

is replaced by the Volume Name Counter value to make it a unique volume ID, the number of #'s determines the number of digits in the number.

The **Volume Name Counter** is used to generate a unique volume id when the volume is initialized. The default value is 1. This number is incremented and appended to the Volume ID each time the volume is initialized.

The **Active Volume** field is the volume number of the current volume being archived, or to be archived.

The **Create Auto-Publish Volumes** check box is not operational in this software version.

The **Local Disk Utilization Warning** check box is not operational in this software version.

The **Backup Disk Utilization Warning** check box is not operational in this software version.

Configuring Archive Backup

It is recommended that the archive backup feature be used when archiving to a disk drive. There are two backup methods: ISO image and copy files. With ISO image, when a volume becomes full, the contents are written to an ISO Image file at the specified Backup Destination. The ISO files can then be burned onto CD ROM or DVD for permanent storage. As an alternative, specify that a shadow copy of the volume be created on a network file server. Both methods can be used.

When archive backup is configured, as volumes are backed up, the volumes are marked *backed up*. Volumes cannot be overwritten until they are marked backed up. If necessary, override this feature and mark volumes as backed up even if they are not. This is described in the section on managing archive media in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

When using Archive Backup, ensure that there is always space available in the destination device to fit the size of the Archive Volume(s) to be copied to it. Regular maintenance to make offline backups of copied files allowing for the deletion of copied Archive Volumes is highly recommended.

Also, follow the guidelines for configuring Device Behavior and Overwrite Timeout as described in Device Behavior and Overwrite Timeout.

When ISO backups are selected, individual archives are limited to 2GB. The ISO file format does not support files greater than 2GB. If ISO backups are not used, the maximum individual archive can be 50 GB and the maximum archive volume size can be 100 GB.

Archive backup is configured with the Backup Archive Path and Backup Type fields.

The **Backup Archive Path** specifies the destination for ISO Image files and/or archive copies when backing up archive data. The path must specify both the disk and directory structure where the archiving is to take place, for example: **E:\archivebackup**.

The ISO image and shadow copy can be made on a remote computer which does not require Information Management software. The directory on the remote node must be a shared folder with the correct privileges (the write privilege must be one of the privileges). For example: **\\130.110.111.20\isofile**.

There are two requirements for the computer where the destination directory is located:

- The destination directory must exist on the computer BEFORE configuring the archive device.
- The computer must have a Windows user account identical to the account for the user under which the Archive Service runs. This is typically the 800xA system service account.

If these conditions are not met, the error message shown in Figure 229 will be displayed after applying the archive device configuration. The object will be created (*Apply Successful* initial indication); however, the archive device will not be operational, and the error message will appear when the message bar updates.

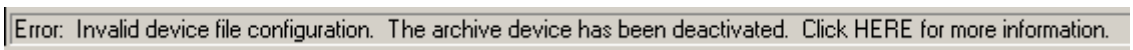


Figure 229. Backup Destination Error Message

The **Backup Type** specifies the backup method. The options are:

ISO Image creates an ISO image file for the currently active volume when the volume becomes full. The ISO Image file can then be copied to a ROM media for permanent storage.

When ISO backups are selected, individual archives are limited to 2GB. The ISO file format does not support files greater than 2GB. If ISO backups are not used, the maximum individual archive can be 50GB and the maximum archive can be 100GB.

Copy Files creates a shadow copy rather than an ISO Image file.

BOTH creates both an ISO image and a shadow copy.

Activate/Deactivate an Archive Device

An archive device must be active in order to archive data to, or restore data from the device. When the archive media is MO Drive, the corresponding archive device must be deactivated to remove and replace a platter. To activate/deactivate an archive device, go to the Archive Device aspect main view, click Actions and choose Activate or Deactivate.

Configuring Archive Groups

Archive groups support scheduled or manual archiving for a group of logs as a single unit. This is done through an archive group aspect. One or more archive groups may be added to this aspect, and each group specifies a set of items (logs and/or aspect objects) to be archived as a unit.

To configure scheduled archive operations, configure a job in the Scheduling structure. The association between the schedule and a specific archive group is made via an Archive Action aspect which must be added to the job in the Scheduling structure.

This section describes how to create an archive group and configure a job to schedule archive actions. Start with Adding an Archive Group on page 341. To configure scheduled archiving, refer to Setting Up the Archive Schedule for an Archive Group on page 352. The **Maintenance...** button tool is used when archive groups are improperly created and the system needs to be cleaned up, refer to Delete/Create Archive Logs using Maintenance Button on page 360.

Adding an Archive Group

Archive groups are configured and managed via the archive group object. Typically, this object is added under the Industrial IT Archive object in the Node Administration structure (using the Archive Service Aspect as described in Adding an Archive Device on page 331). To do this:

1. In the Plant Explorer, select the Node Administration Structure.
2. Navigate to and expand the object tree for the node being added to the archive group.
3. In the object tree for the selected node, expand the **Industrial IT Service Provider** tree and select the **Industrial IT Archive** object.
4. Select the Archive Service Aspect from this object's aspect list.
5. Click **Archive Group,** Figure 230.

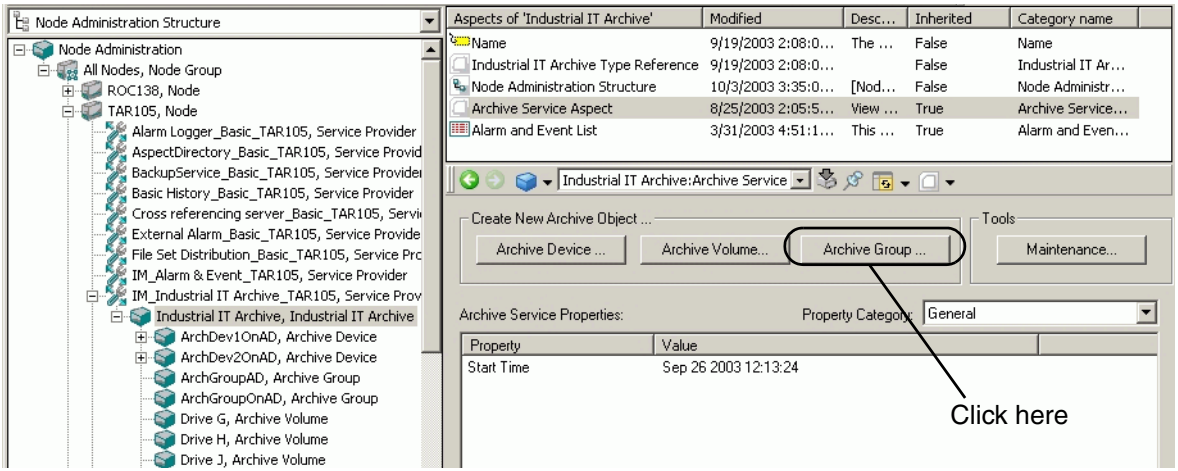


Figure 230. Creating a New Archive Group

This displays the New Archive Group dialog, Figure 231.

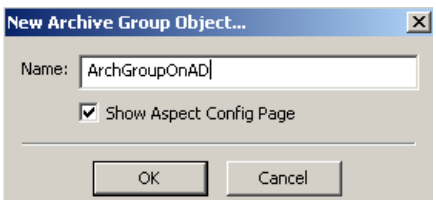


Figure 231. New Archive Device Object Dialog

6. Enter a name for the object in the Name field, for example: ArchGroupOnAD, then click **OK**.

Keep the **Show Aspect Config Page** check box checked. This will automatically open the configuration view of the Archive Device aspect.

This adds the Archive Group object under the Industrial IT Archive object and creates an Archive Group Aspect for the new object. Use this aspect to configure one or more archive groups as described in Archive Group Aspect on page 343.

Archive Group Aspect

The Archive Group aspect is shown in Figure 232.

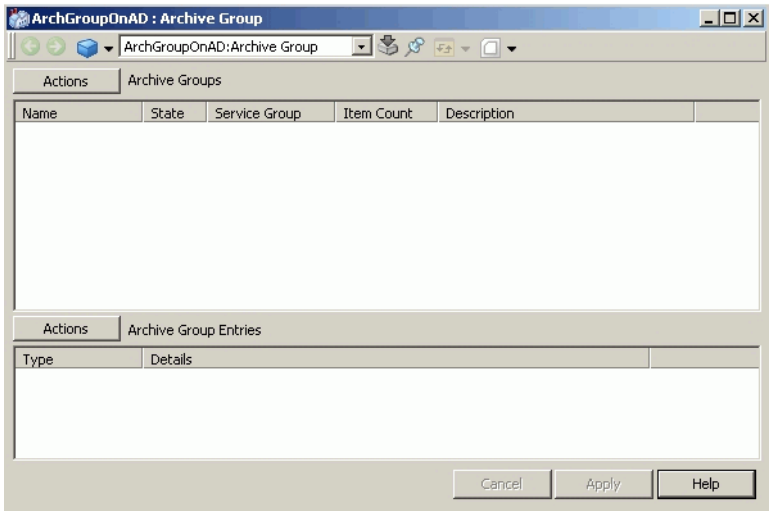


Figure 232. Archive Group Aspect

When configuring the History database, the primary function of this aspect is for adding and configuring archive groups. This is described in Adding and Configuring an Archive Group on page 343.

After configuring an archive group, make changes to the group, or to the archive entries added to the group. This is described in Adjusting Archive Group Configurations on page 351.

This aspect also is used to invoke manual archive operations on an archive group basis. This is described in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

Adding and Configuring an Archive Group

To do this:

1. Right click inside the archive groups window (or click **Actions**) and choose **New Group**, Figure 233.

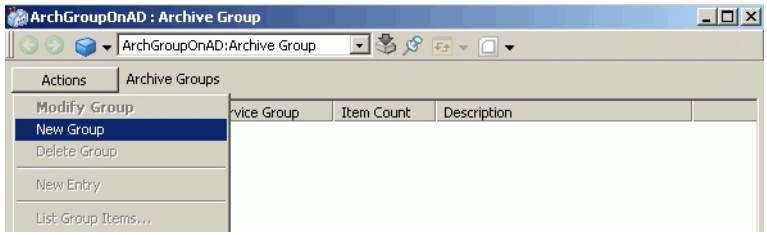


Figure 233. Adding an Archive Group

This displays the Add Archive Group dialog, Figure 234.

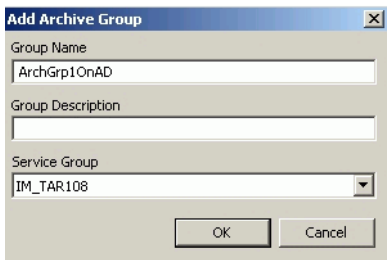


Figure 234. Add Archive Group Dialog

2. Use this dialog to specify the Group name, description (optional), and the Industrial IT Archive Service Group whose service provider will manage this archive group.
3. Click **OK** when finished.
4. Click Apply to initialize the new group.

Repeat this to add as many groups as required. Then specify the contents of each archive group as described in Adding Entries to Archive Groups.

Adding Entries to Archive Groups

This procedure specifies the logs (and/or objects) to be archived as part of this archive group. Different entry types can be mixed in the same group. To add entries:

1. Select the archive group from the list of archive groups, right click and choose **New Entry** from the context menu, Figure 235.

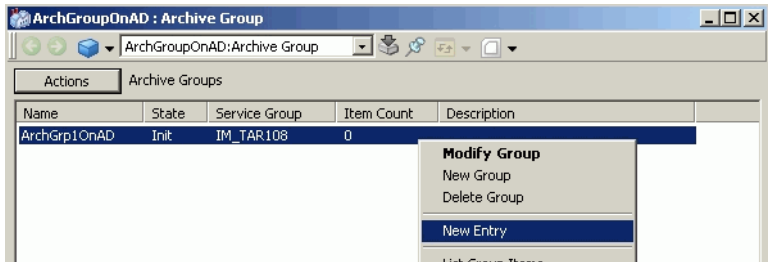


Figure 235. Adding a New Entry

This displays the Add Group Entry dialog.

2. Use this dialog to select an entry type, Figure 236, then click **OK**. The options are described in Table 35.

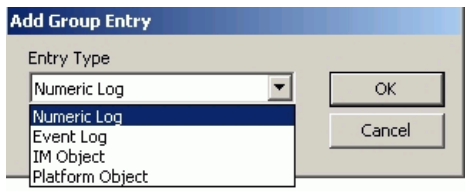


Figure 236. Add Archive Group Dialog

Table 35. Archive Group Entry Options

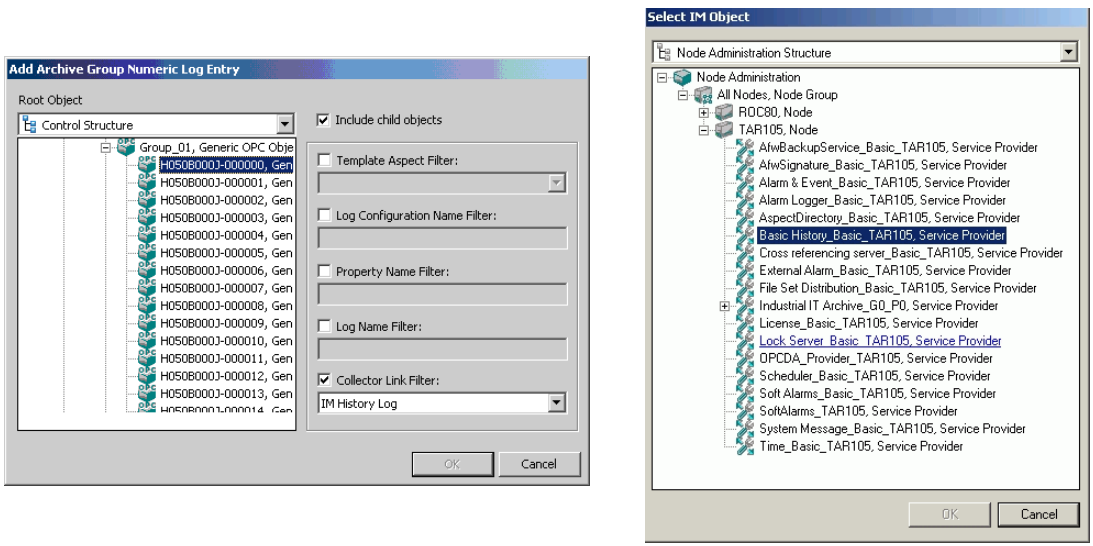
Entry Type	Description
Numeric Log	Creates an entry for numeric (property) logs. Selecting this option displays the Add Archive Group Numeric Log Entry dialog, Figure 237. Use this dialog to browse the aspect directory for the object whose property log(s) to be included in the archive group. The **Include Child Objects** check box is selected by default to include logs for all child objects of the selected object. If this check box is not checked, only logs for the selected object will be included. A filter based on the following criteria may also be specified: • **Template Aspect** - Include logs whose log template matches the specified filter. • **Log Configuration Name** - Include logs whose log configuration aspect name matches the specified filter. Accepts wildcard characters. • **Property Name** - Include logs whose property matches the specified filter. Accepts wildcard characters. • **Log Name** - Include logs whose name matches the specified filter. Accepts wildcard characters. • **Collector Link** - Use this to filter on **IM History Log** (default prevents all PPA logs in an archive from being included), **IM Importer Link**, **All Collector Link Logs** (adds PPA logs), **No Collector Link Logs** (just PPA logs). Wildcard characters are * and ?. Asterisk (*) means 0 or more of any character. Question Mark (?) means a single character.
Event Log	Creates an entry for alarm/event messages stored in the 800xA System alarm/event buffer. There is no dialog associated with this option. Typically, this option is used when the system does not have Information Management History Server function. If the History Server function exists, the 800xA System alarm/event messages are stored in a message log, and the messages may be archived by creating an entry for IM Object types as described below.

Table 35. Archive Group Entry Options (Continued)

Entry Type	Description
IM Object	Creates an entry for message and/or report logs. Selecting this option displays the Select IM Object dialog, Figure 237. Use this dialog to browse the aspect directory for a report or message log object.
Platform Object	Creates an entry for file viewer aspects which contain finished reports executed via the Application Scheduler, or any other object type. For details, refer to Adding Entries for Platform Objects on page 348.

If adjustments are needed to an archive group configuration, refer to Adjusting Archive Group Configurations on page 351.

To set up a schedule for an archive group, refer to Setting Up the Archive Schedule for an Archive Group on page 352.



Numeric Log Entry **IM Objects Entry**

Figure 237. Add Archive Group Numeric Log Entry, IM Objects Entry

Adding Entries for Platform Objects

This dialog is used to select file viewer aspects which contain completed reports. These aspects are attached to Completed Report objects in the Scheduling structure. Other object types may also be selected.

The dialog has two preset configurations used to select Completed Reports - one for signed report objects, and one for unsigned report objects. To use a preset configuration, use the browser to go to the Scheduling structure and select the Reports object, Figure 238. Then use the Preset Configurations pull-down list to select either **Report Objects** (unsigned reports), or **Signed Report Objects**. In either case, the filters and other settings are automatically configured for the selection made. Figure 238 shows the preset configuration for signed report objects. The configuration for unsigned reports is essentially the same, except that the Require Signatures area is disabled for unsigned reports.

Make sure the reports are digitally signed. Do not let the archive utility build up to the maximum number of reports before unsigned reports are deleted.

When the preset configuration is **Signed report Objects**, and if the completed reports (FileViewer aspects in the completed reports objects) are not signed, the Scheduled Platform object archiving skips the archiving of these report objects. When the aspects are signed, these skipped reports will be archived with the next archive entry.

There is a possibility that the unsigned reports missed by the archive will be deleted before they are archived due to the **Report Preferences** setting in the `Reports` tree of the scheduling structure. Make sure that reports are digitally signed. Do not let the archive utility build up to the maximum number of reports before unsigned reports are deleted.

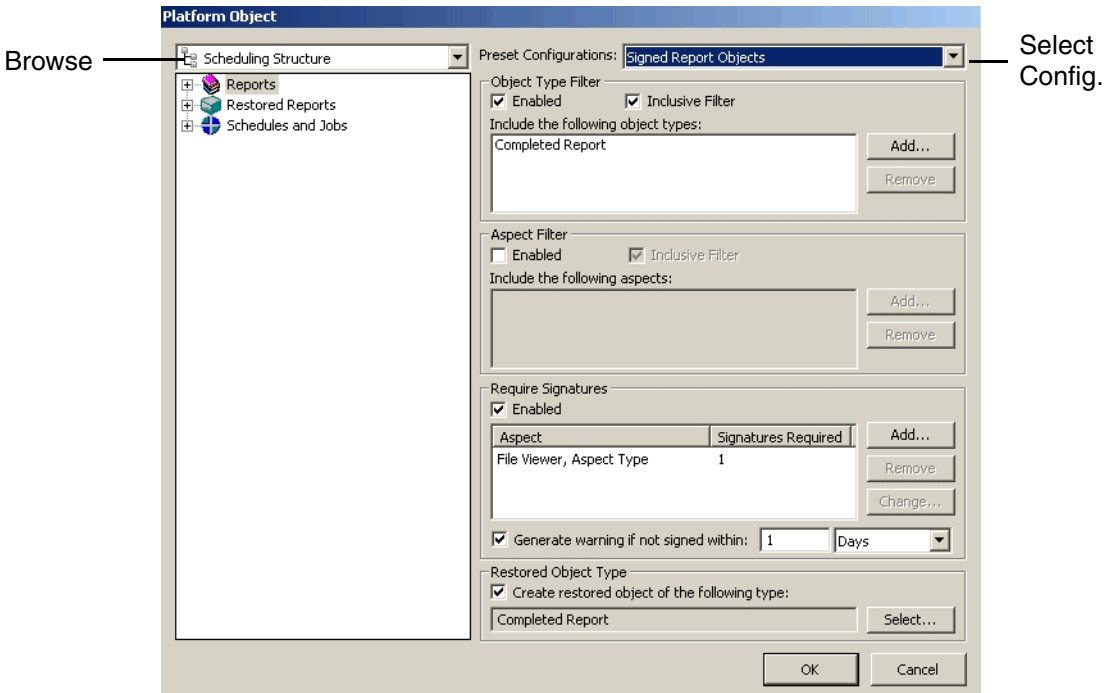


Figure 238. Platform Object Dialog

If another type of object must be selected, adjust one or more settings, or simply understand what each setting does, refer to Table 36.

Table 36. Platform Object Settings

Feature	Description
Browser	Left-hand side of dialog provides a browser similar to Plant Explorer for navigating the aspect directory and selecting the root object. Objects and aspects under the selected root will be selected based on the filters.
Object Type & Aspect Filters	These filters are used to specify whether to include or exclude all objects and/or aspects of one or more selected types. To use a filter, the corresponding **Enabled** check box must be checked. The **Inclusive Filter** check box is used to specify whether to include or exclude all object (or aspect) types that satisfy the filter. Inclusive includes the selected aspects/objects in the list, exclusive (check box not checked) excludes them. To add an object (or aspect) to the filter, click the corresponding **Add** button. This displays a search dialog. Set the view in this dialog to hierarchy or list. Any object or aspect type that has been added may be removed from the list by selecting it and clicking **Remove**.
Required Signatures	This area is used to specify whether or not to check for signatures on selected aspects. When Enabled, only aspects which have been signed with the specified number of signatures (default is one signature) will be archived. To select an aspect, click the **Add** button. This displays a search dialog. Set the view in this dialog to hierarchy or list. Any aspect that has been added may be removed from the list by selecting it and clicking **Remove**. The **Change** button displays a dialog used to change the number of required signatures (up to five maximum).
Generate warning if not signed	This check box is used to specify whether or not to generate an event when the archive service does not archive an aspect because the aspect is not signed. If an aspect has missed an archive because it was not signed, once the aspect is signed, it will be archived with the next archive entry.
Restored object type	This area is used to specify the type of object to create when an archived aspect is restored. Check the corresponding check box, then click **Select** to select the object type.

Adjusting Archive Group Configurations

Changes can be made to the archive group configuration on two levels. When selecting an archive group in the archive group list, the context menu, Figure 239, (or **Actions** button) is used to:

- Change the archive group description (**Modify Group**).
- Delete the archive group (**Delete Group**).
- Rescan the archive group (**Rescan Group**) - This rereads the archive group configuration in the event that logs have been added to or deleted from the group.

Reset Last Archive Time, List Group Items, and Manual Archive are related to runtime operations and are described in the archive section in *System 800xA Information Management Data Access and Reports, 3BUF001094*.*

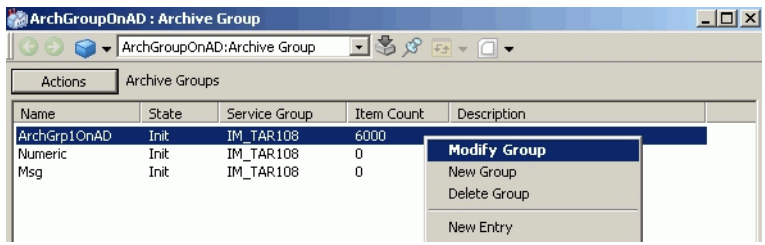


Figure 239. Changing an Archive Group Configuration

When selecting an entry from the entries list for an archive group, the context menu (or **Group Entries** button) is used to modify or delete the entry, Figure 240. The **Modify Entry** command displays the object selection dialog as described in Adding Entries to Archive Groups on page 344.

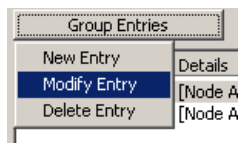


Figure 240. Editing Group Entries

Setting Up the Archive Schedule for an Archive Group

Scheduling instructions for archive groups are specified by adding a job in the Scheduling structure, and configuring the job's scheduling definition aspect. The schedule may be cyclic, periodic, weekly, monthly, a list of dates and times, or conditional. But, periodic schedule is recommended for all the timed archives. The schedule is associated with a specific archive group via the Archive action aspect which must be added to the job object. This section quickly demonstrates how to add a job and use the scheduling definition aspect to set up a periodic schedule. It also describes how to add and use the Archive action aspect. For further information on jobs and scheduling options, refer to the section on scheduling in *System 800xA Information Management Data Access and Reports (3BUF001094*).*

When selecting archive intervals and configuring archive group contents, it is important to remember that archive works most efficiently when the following points are considered:

- Having many small archives is not very efficient. The ratio of overhead to actual data on the archive media is very low.
- Having fewer large archives, both in time interval and quantity of logs, can have a positive performance impact on the IM server. Having too many logs or too much data in a single archive will cause archive to run for a long period of time and create large archive entries. Having too much data archived increases the amount of data that can be lost if a system fails.

It is important to strike a balance between logs and time intervals such that the archive media is used effectively. Some common guidelines are:

1. The archive interval should be between 1/4 of the log period and two weeks. If you have a four day log, you would want to archive it once a day. This gives archive three chances to catch up if the IM was down during an archive. If you have a four year log, having a one year archive interval would create huge archives. It makes more sense to archive once every two weeks. If it is two weeks of 60 second data, that is 20,000 values, which is reasonable. If the log has one second data, that is 1.2 million numeric values, that is a significant amount of data per log and the archive interval should be lowered to keep the numeric entries per archive to 100000 or less.
2. Once the intervals are selected the number of logs in each archive group should be determined. This depends on the size of your media and entries per log per

interval. If you have smaller media, CD size, then you want to keep the number of logs smaller, if you have DVD size, then you have more logs. If using just disk archives and want archive volumes configured at the greatest capacity, then you can use the limits. A practical limit is 2500 log per archive group. Limit should be decided by the overall size of the archive. If 2500 logs with one week of data is 1GB and CD size media was selected (640 Mega), this configuration will not work. For a given media size, it is recommended that 10 or more archives fit on a media.

3. Make sure that the archive created, either ISO images or copies do not fill up the backup destination. After an archive configuration is created, it will create archives at the rate the 800xA system collects data. If a 500/second numeric configuration creates 1GB of archives per day, a 2000/second archive will consume 4 times that, or 4GB per day. The end user must plan for and remove these archives from the backup destination to keep archive running. Once the archive backup destination fills up, archive will stop archiving until more disk space is available.

It is expensive, in terms of physical media, to archive frequently. Therefore, try to maximize the time between archives. General guidelines are provided below. For detailed guidelines, refer to Archive Configuration Guidelines on page 326.

- When specifying the schedule, configure the archive interval between 1 hour and 1/4th the log period. For example, if the log time period is four weeks, archive once a week. This allows for a number of archive failures (due to media full or drive problems) without loss of data. If the archive is scheduled more frequently, any missed archives will have more chances to recover before the source log(s) wrap.
- For Property logs use the following guidelines:

  10,000 logs per archive group maximum, 5,000 recommended
  25,000 entries per log per archive

  For example, for a storage interval of 30s, which is 2 entries per minute, 25,000 entries in 12,500 minutes (208 hours or 8 days), schedule archiving for about once a week.

- For Message log: max of 50,000 messages in a single archive.

If a sample blocking rate for numeric logs is configured, the time stamps for some logs in the archive group may not be aligned with the scheduled archive time. Compensate for this by following the procedure described in Aligning Staggered Time Stamps on page 359.

For scheduled archives, even if no new samples were collected, at least one sample (the last valid point) will be archived.

Adding a Job and Specifying the Schedule

Jobs are created in the Scheduling structure. To create a job:

1. In the Plant Explorer, select the **Scheduling Structure**.
2. Right-click on **Job Descriptions** and choose **New Object** from the context menu, Figure 241.

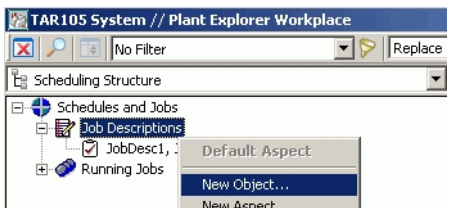


Figure 241. Adding a New Job Object

3. Browse to the Scheduling Objects category and select the **Job Description** object (Object Types>ABB Systems>Scheduling Objects>Job Description), Figure 242.
4. Assign the object a logical name.

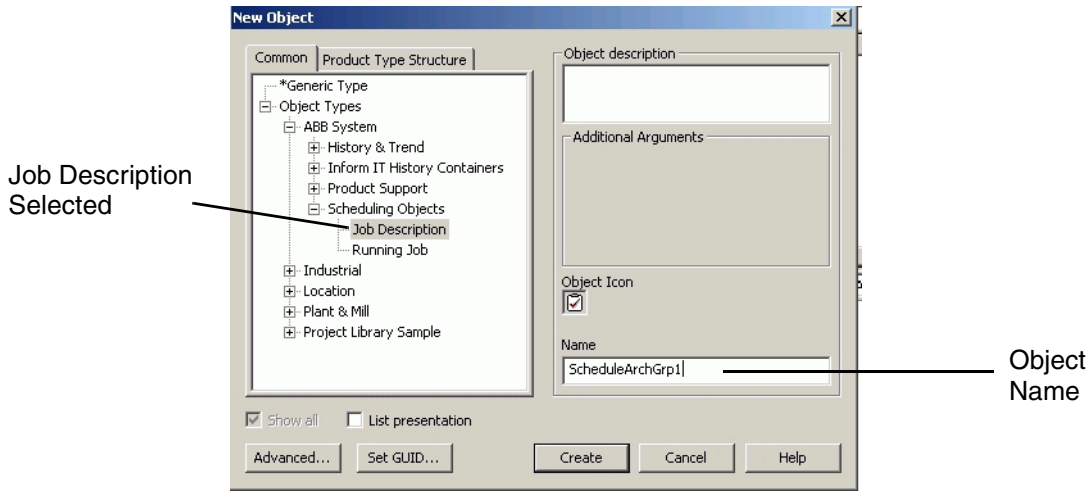


Figure 242. New Object Dialog

5. Click **Create**. This creates the new job under the Job Descriptions group, and adds the Schedule Definition aspect to the object's aspect list.

6. Click on the **Scheduling Definition** aspect to display the configuration view, Figure 243. This figure shows the scheduling definition aspect configured as a periodic schedule. A new archive will be created every three days, starting June 9th at 5:00 PM, and continuing until September 9th at 5:00 PM.

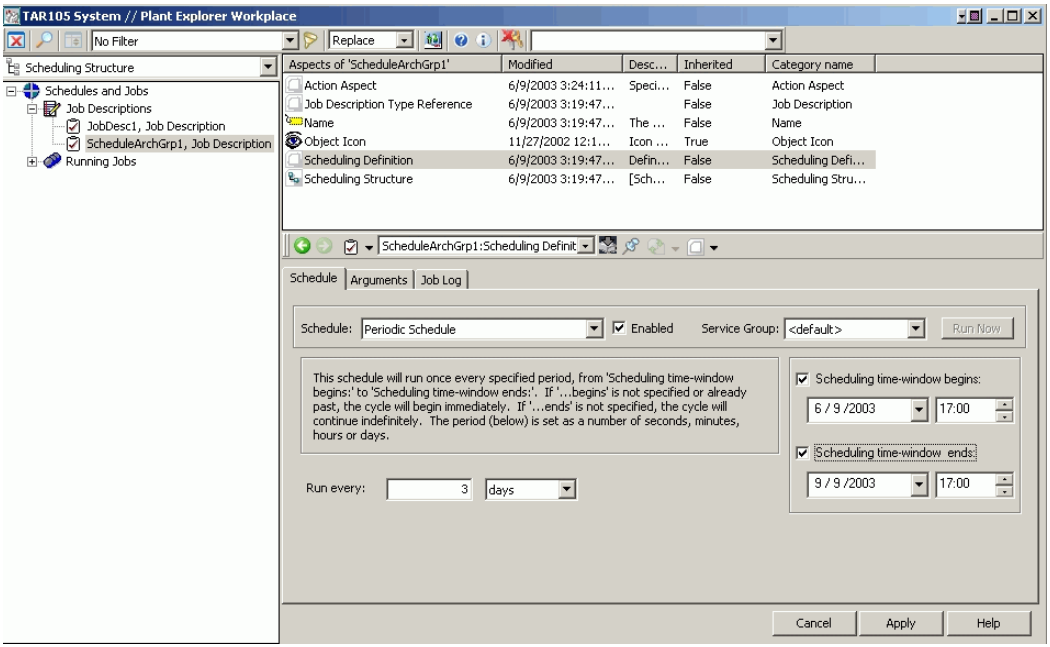


Figure 243. Scheduling Definition Configuration View

Adding and Configuring the Archive Action

Actions are implemented as aspects on an object which is on or under a job description in the scheduling structure.

To add an action:

1. Right-click on the Job object (for example ScheduleArchGrp1 in Figure 244) and choose **New Aspect** from the context menu.

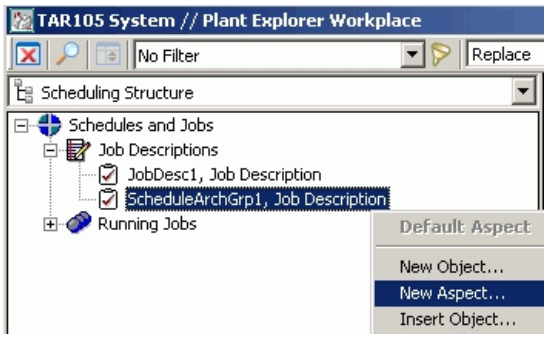


Figure 244. Adding an Action Aspect

2. In the New Aspect dialog, browse to the Scheduler category and select the Action aspect (path is: **Scheduler>Action Aspect>Action Aspect**), Figure 245.

   Use the default aspect name, or specify a new name.

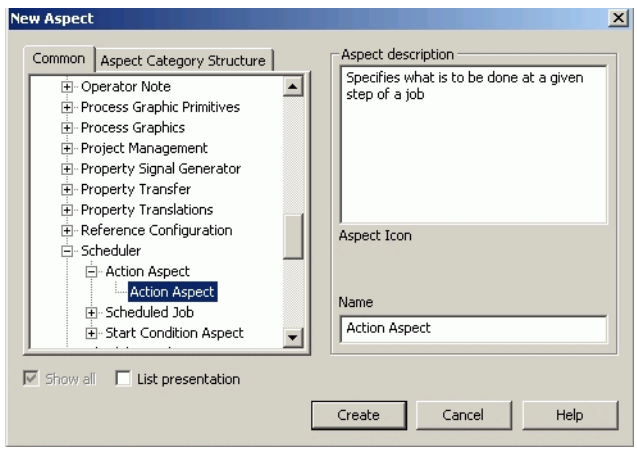


Figure 245. New Aspect Dialog

3. Click **Create** to add the Action aspect to the job.
4. Click on the Action aspect to display the configuration view.

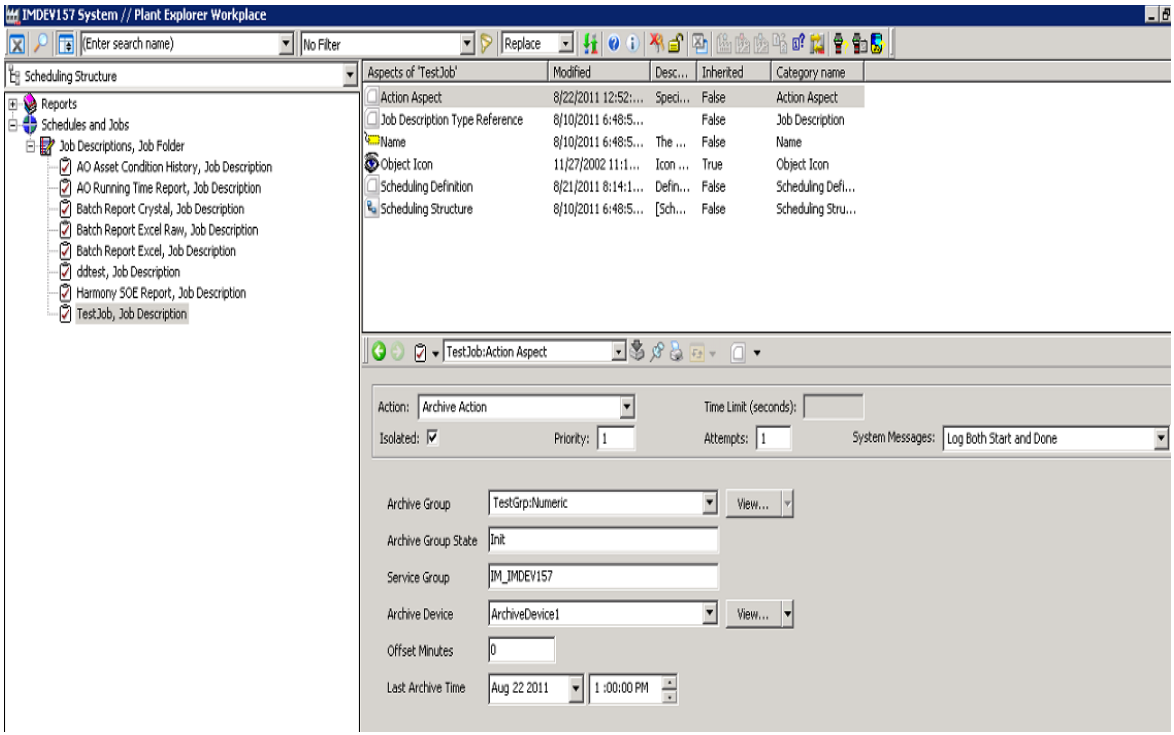


Figure 246. Action Aspect Configuration View

Select the archive group to be scheduled by this action. The Service Group and Archive Devices fields will be automatically filled in according to the selected archive group. If a different archive device needs to be specified, use the corresponding pull down list to display a list of available devices.

The **Last Archive Time** is filled in by History. This is the time the last archive operation occurred for a group. This time is remembered by the system so that the next archive operation will archive data starting at that time. Change (reset) this time to cause the next archive operation to go back farther in time, for example to account for a failed archive, or skip ahead to a later time, for example to stop the archive from collecting data for a certain time period. To do this, change the date and time, then press **Reset**. The last archive time can also be reset from the Archive Group aspect as described in Adjusting Archive Group Configurations on page 351.

Aligning Staggered Time Stamps

Staggering, phasing, and sample blocking rate are methods for balancing CPU load due to data collection and storage (Stagger Collection and Storage on page 472). Using any one of these methods may result in some logs in an archive entry having time stamps not be aligned with the scheduled archive time. This does NOT result in any data being lost; however, if it is important that time stamps for all logs within an archive entry are aligned with the scheduled archive time, then use the **Offset Minutes** function in the Archive Group aspect to accomplish this.

First, configure the schedule for the archive group to delay the archive operation for a number of minutes equal to or slightly greater than the largest sample blocking rate (or phasing span). This allows all logs in the archive group to store their respective values prior to the scheduled archive.

Then use the **Offset Minutes** field in the Archive Action to reset the end time of the archive operation to the desired alignment time. The Offset Minutes value is subtracted from the scheduled Archive time so that data will be collected up to the desired alignment time where all logs will have the same time stamp.

For example: consider the need to archive on a daily basis at midnight (00:00), and data storage for logs in the archive group are phased over a nine (9) minute period. In this case, schedule the archive operation to occur at 10 minutes after midnight (00:10), and set the Offset Minutes to **10**.

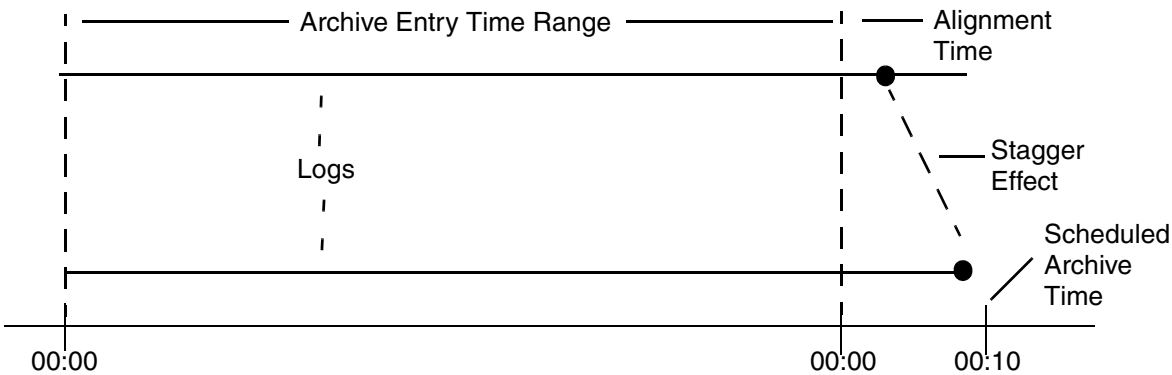


Figure 247. Using Offset Minutes

Delete/Create Archive Logs using Maintenance Button

The **Maintenance...** button tool, reference Figure 230, is used when archive groups are improperly created and the system needs to be cleaned up. The Maintenance tool can also put links back on existing logs. This is used to not wait for the next timed archive. In addition, the Maintenance tool can delete restore (_Rst) logs on all PPA logs that were accidentally configured.

After selecting the **Maintenance...** button, use the Maintenance Dialog, Figure 248, to create and delete archive logs.

Specify parameters as in the numeric log archive group entry dialog. Next, go through the system and create or delete archived log references that match specified criteria.

The **Validate Archive Logs** button verifies the correctness of the references, updates them if they are not, and reports the number of updated references to the user.

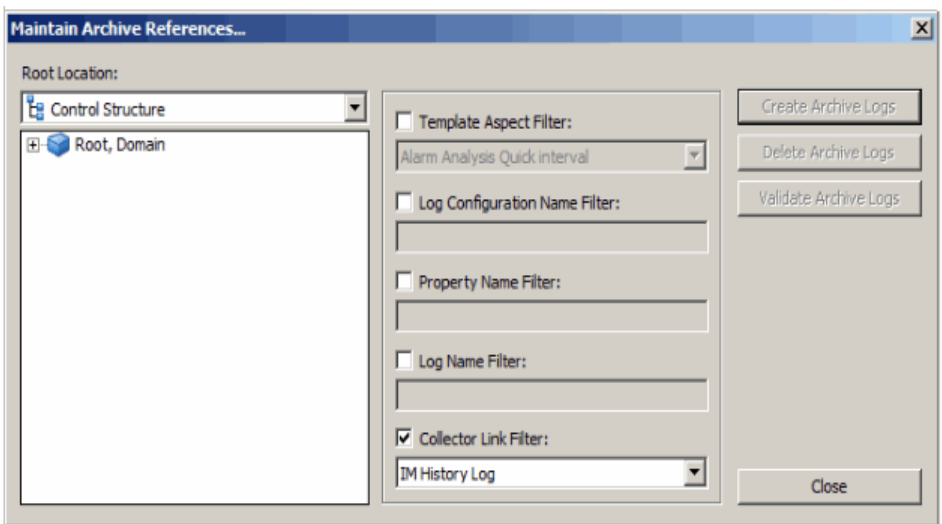


Figure 248. Maintenance Dialog

Adding a Read-Only Volume for a Mapped Network Drive

To access archived history data, either restore the archived logs to the Restored History database, or publish one or more selected volumes for direct viewing. (Publishing applies to property and message logs only.)

The MO media and volumes created for any archive device are read/write volumes. This means archive data can be written to those volumes, and use data access applications such as DataDirect to read the archive data from those volumes.

Read-only volumes are automatically created for all removable disk drives (DVD and CD drives). This is used to read archive data from DVDs and CDs on to which archive volumes are copied and when those DVDs or CDs are mounted.

Additional read-only volumes can be created for reading archive volumes that have been copied to a mapped network drive. These volumes should not be associated with (created beneath) any archive device object. The recommended method is to use the Archive Service Aspect. This is basically the same procedure used to add archive devices and archive groups (Adding an Archive Device on page 331).

To add a volume for a mapped network drive:

1. In the Plant Explorer, select the Node Administration structure.
2. Expand the object tree for the node where the volume is to be added.
3. In the object tree for the selected node, expand the **Industrial IT Archive Service Provider** and select the **Industrial IT Archive** object.
4. Select the **Archive Service Aspect** from this object's aspect list.
5. Click **Archive Volume**. This displays the New Archive Volume dialog.
6. Enter a name for the object in the Name field, for example: Drive H, Figure 249, then click **OK**.

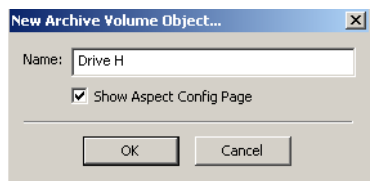


Figure 249. New Archive Volume Dialog

7. This adds the Archive Volume object under the Industrial IT Archive object and creates an Archive Volume aspect for the new object.
8. Use the config view to select the Service Group that will manage archive transactions for this volume, and set the volume path to the mapped network drive.

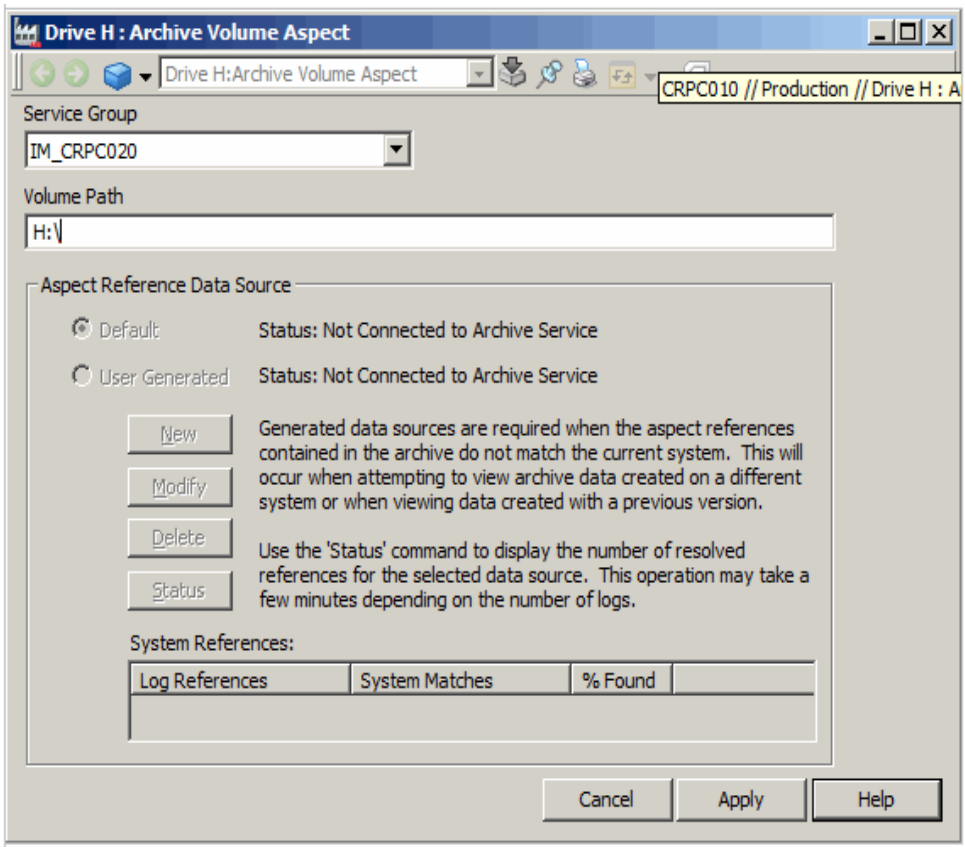


Figure 250. Archive Volume Aspect

PDL Archive Configuration

The PDL Archive aspect is used to configure PDL archive parameters.

1. Select the **Inform IT History Object** under the applicable node in the Node Administration structure.
2. Select the **Production Data Log Container.**
3. Select the **Inform IT PDL Auto Archive** aspect, Figure 251.

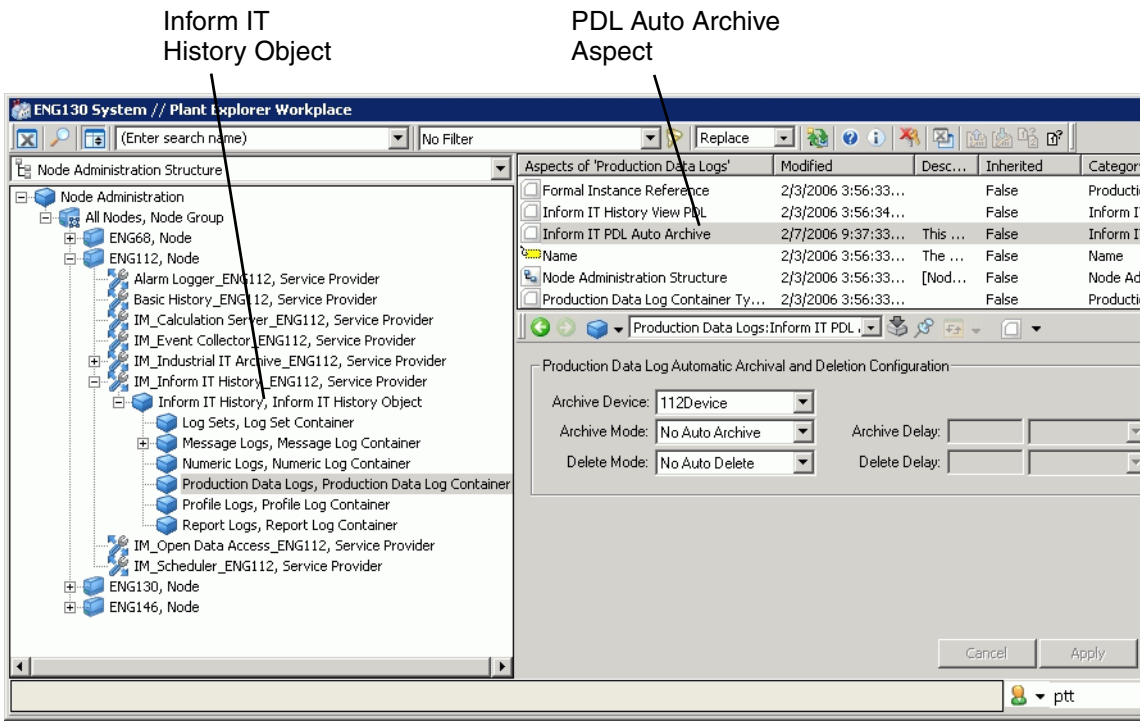


Figure 251. PDL Archive Configuration Window

Archive and Delete Modes

Archive and Delete modes must be configured in unison. Table 37 indicates the legal combinations of archive and delete modes, and the corresponding delay mode.

Table 37. Legal Combinations for Archive Mode, Delete Mode, and Delay Mode

Archive Mode	Legal Delete Modes	Applicable Delay Mode
No_auto_archive	No_auto_delete	Not Applicable
	After_job (campaign)	Delete Delay
	After_batch (reel or grade)	Delete Delay
On_job_end (campaign)	No_auto_delete	Archive Delay
	After_archive	Archive Delay + Delete Delay
On_batch (reel or grade)	No_delete	Archive Delay
	After_archive	Archive Delay + Delete Delay

Use the **Archive Mode** button to select the archive mode. The choices are:

- **No_auto_archive** - Archive PDLs on demand via View Production Data Logs.
- **On Job End** - Archive PDLs automatically after specified delay from end of job (for Batch Management, job = campaign).

If **On Job End** is chosen for a Batch Management campaign with multiple batches in the campaign, Information Management archives data at the end of each batch in the campaign, and at the end of the campaign. For example, if there are three batches in a campaign, and **On Job End** is selected as the Archive mode, there will be three archives: batch 1, batch 1 and batch 2, and batch 1, batch2, and batch 3.

- **On Batch End** - Archive PDLs automatically after specified delay from end of batch. (For Profile Historian, batch = reel or grade.)

For Profile Historian applications, configure the archive mode as ON BATCH END.

Use the **Delete Mode** button to select a delete mode. The choices are:

- **No_auto_delete** - Delete PDLs on demand via PDL window.
- **After Archive** - Delete PDLs from PDL database automatically, a specified time after they have been archived.

- **After Job End** - Delete PDLs after specified delay from end of job (for Batch Management, job = campaign). The PDL does not have to be archived.
- **After Batch End** - Delete PDLs after a specified delay from the end of the batch. The PDL does not have to be archived to do this. (For Profile Historian, batch = reel or grade.)

For Profile Historian applications, configure the delete mode as AFTER ARCHIVE. Also the Delete Delay must be set to a value greater than or equal to the profile log time period.

Archive Delay

A delay can be specified between the time a job or batch finishes and the time the PDLs are actually archived. This field is only applicable if **On Job End** or **On Batch End** for archive mode is selected.

Configure the delay to be at least five minutes (5m). This ensures that all data is ready for archive after the end of a job or batch. A delay of hours (h) or days (d) can be specified if time is needed to enter additional data for archive.

Delete Delay

A delay until PDLs are actually deleted can be specified. This delay is not applicable when the delete mode is configured as NO_AUTO_DELETE. This delay has a different purpose depending on whether the PDL is archived or not.

When the PDL is being archived, this is the amount of time to delay the delete operation after the archive. This is used to archive the data immediately for security, and keep the data online for a period of time for reporting, analysis and so on.

When the PDL is not archived, this is the amount of time to delay after the end of the job or batch before deleting. Configure the delay to be at least five minutes (5m). A delay of hours (h) or days (d) can be specified.

For Profile Historian applications, configure the Delete Delay equal to or slightly greater than the profile log time period.

Archive Device

Enter the archive device name.

Section 12 Consolidating Historical Data

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

This section describes how to consolidate historical process data, alarm/event data, and production data from remote history servers on a dedicated consolidation node. This includes data from 800xA 5.1 or later Information Management nodes in different systems.

The consolidation node collects directly from the History logs on the remote nodes.

For instructions on consolidating historical process data, refer to Consolidating Historical Process Data on page 368.

For instructions on consolidating alarm/event data stored in message logs, and production data stored in PDLs, refer to Consolidating Message Logs and PDLs on page 401.

Consolidating Batch PDL Data with History Associations

History associations are created in the source node for Batch data. The log name typically contains three parts:

- Object name.
- Property name.
- Log name.

These are defined by the History Log template that was used in the Log Configuration aspect. For example: TIC100:MEASURE,IMLog, where TIC100 is the object name, MEASURE is the property name, and IMLog is the name of the log in the Log Configuration aspect.

When the consolidation node is set up, a new Log Configuration aspect is manually created on the consolidation node. If the naming convention for the numeric log does not match the one on the source node, any attempt to retrieve data in a report will fail.

When setting up numeric consolidation on nodes that will also consolidate Batch PDL data with history associations, the naming convention on the Numeric Log template on the consolidation node must match the naming convention for the numeric log template on the source node.

Consolidating Historical Process Data

Process object names may be duplicated from one ABB network to the next. In the event that duplicate log names are used, the remote History node's IP address is used to distinguish one log from the other.

Setting up Historical data consolidation is a three-step process:

- Interaction between two or more History nodes (as is the case with historical data consolidation) requires the OMF domain to be extended to TCP/IP. This must be done prior to performing the other steps described in this section. For details, refer to Appendix A, Extending OMF Domain to TCP/IP.
- The log configurations on the nodes from which historical data is being consolidated must be imported to the consolidation node.

  This step does not establish historical data collection and storage on the local (consolidation) node for the imported logs. The Log Configuration aspects created as a result of this step merely provide a window for viewing the historical data stored in the log configurations on the remote nodes.
- To establish local data collection and storage on the consolidation node for the imported logs, create a second set of Log Configuration aspects on the consolidation node.

When ready to import from IM, start with the procedure for Importing Remote Log Configurations on page 369.

Other considerations are:

- When consolidating message logs, the target (consolidation) log must already exist on the consolidation node. This is not required when consolidating PDLs.

- When consolidating message logs or PDLs from Information Management, a user account must be set up on the source node. This account must use the same user name and password as the installing user for the target consolidation node. Also, this user must be added to the HistoryAdmin group. Detailed instructions for adding user accounts are provided in the section on Managing Users on page 589.

Importing Remote Log Configurations

This procedure shows how to import the log configurations from remote history servers. This is a prerequisite step for creating the logs which consolidate the historical data from the remote history logs. These imported log configurations also support viewing of the historical data using the Information Management desktop tools, as well as tools supported by 800xA System.

Launching the History Access Importer Tool

From the Windows task bar choose **Start>Programs>ABB Industrial IT 800xA> Information Mgmt>History>History Access Importer Tool**.

The History Access Importer is a wizard-like tool for importing remote history log configurations. The first step is to read the log configuration information from the remote history server.

Reading the Remote Log Configurations

Use the dialog in Figure 252 to read log configuration information from a specified remote history server into an xml file on the Information Management node:

1. Specify whether to generate a new file, or use an existing file.

   Always use the **Generate New File** option if the information has not yet been read from the remote history server.

   The **Use Existing Definition File** option is used only when the information from the remote history server has been read and an import operation that is partially completed is being finished.

2. For Import Type select **IM Historian Database**.
3. Enter the computer name and IP address for the remote history server.

4. Enter the port number. as **1521**.
5. Enter the password of history account for the remote history server in the **Password** field or leave the **Password** field as <default>. When <default> is used, the tool will attempt to connect to the remote history with NT authentication. This will work when both nodes are in the same domain. If both nodes are not in the same domain, the password for the history account must be specified.
6. Specify an xml file name.
7. Click **Next** when done.

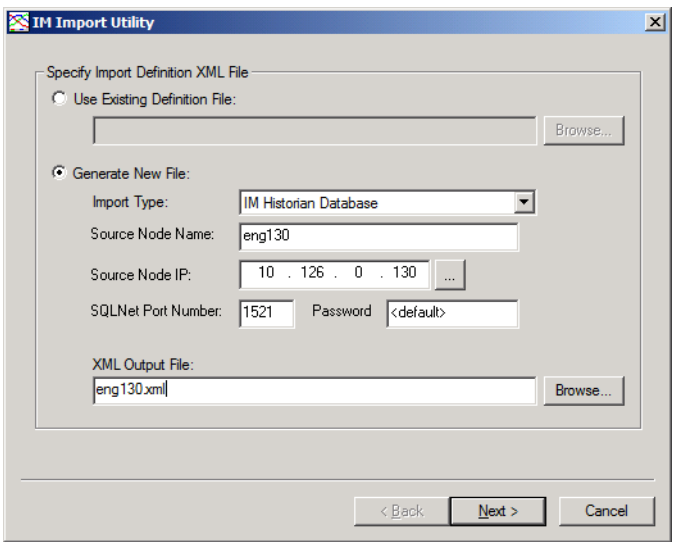


Figure 252. History Access Tool - Initial Dialog

This starts the process of reading the log configuration information from the specified remote history server. The progress is indicated in the dialog shown in Figure 253. When the *Importer Module successfully loaded* message is displayed, the read-in process is done.

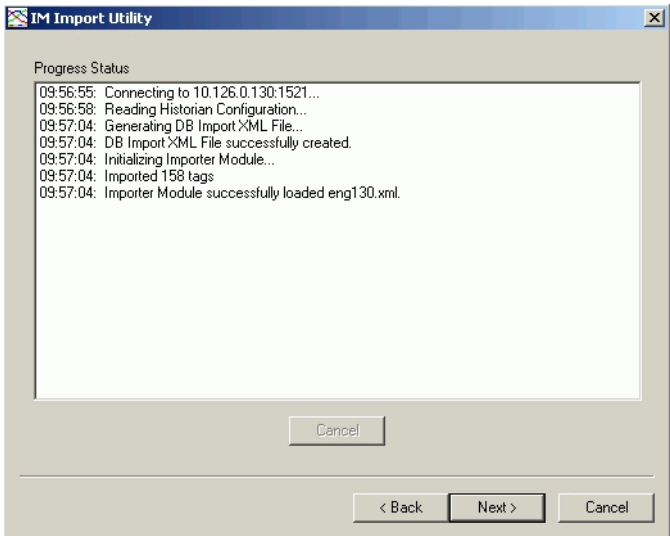


Figure 253. Progress Status Window

8. Click **Next**, then continue with the procedure for Assigning Objects to the Imported Tags on page 371.

Assigning Objects to the Imported Tags

Every Log Configuration aspect must be associated with an object and object property. For consolidation, the IM Import utility is used to create dummy objects whose sole purpose is to provide a place to instantiate the Log Configuration aspects for the imported tags. To configure these log/object associations:

1. Use the dialog in Figure 254 to select the method for associating the imported tags with their respective objects. In most cases, the **Create new objects for imported tags** option should be used. This option is selected by default.

   As an option, the default **Root object name** may also be changed. This is the name of the root object under which the imported log configuration aspects will be located. The object is created as part of the import operation. The

default name is derived from the specified source node name. Use the default, or modify it as required, then click **Next** to continue.

If Multisystem Integration is used, the Match option can be selected. Instead of creating generic objects and properties, the imported log configurations will be associated with the same objects and properties from the source system.

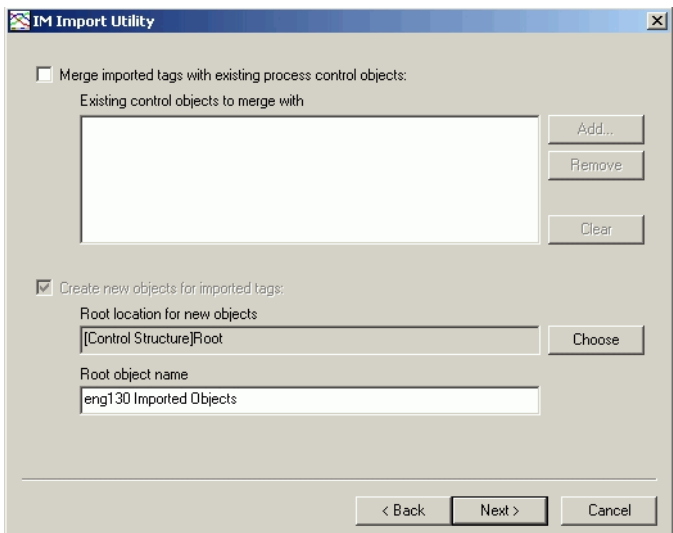


Figure 254. Assigning Objects to Imported Tags

This displays a dialog for specifying the location and naming conventions for the Log Template object, and the Log Configuration aspects, Figure 255.

2. Typically, the default settings can be used. This will create the Log Template in the **History Log Template Library** and provide easy-to-recognize names for the template and Log Configuration aspects. Adjust these settings if necessary. Click **Next** when done.

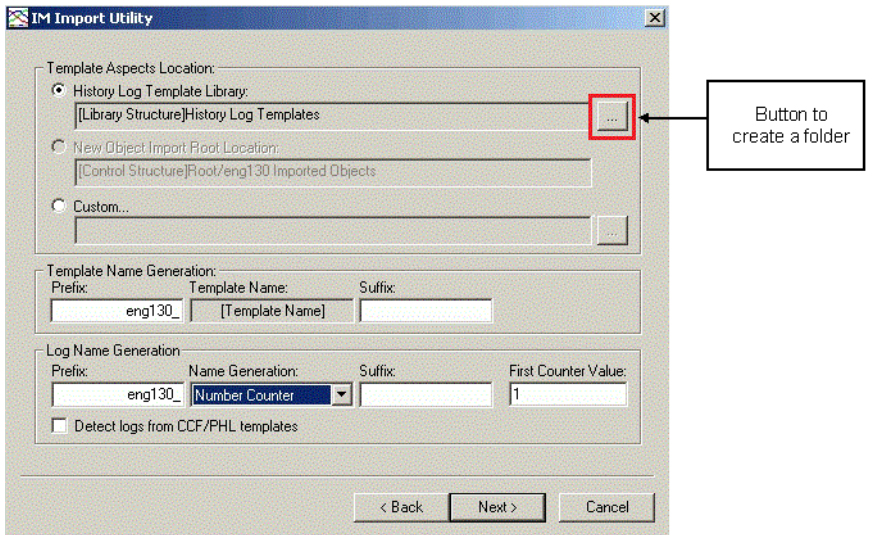


Figure 255. Template Specification

It is recommended to create a new folder for imported templates.

This displays a dialog which summarizes the log/object association process, Figure 256. At this point no associations have been made. There are no tags assigned to new objects.

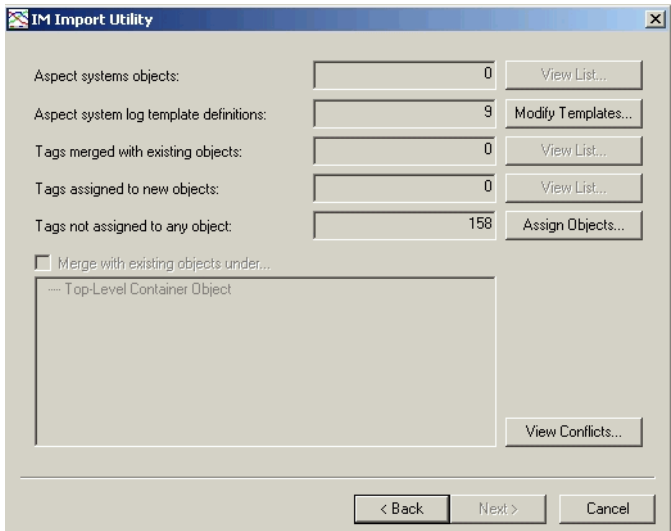


Figure 256. Assign Objects Summary - Before Assign Objects Done

3. Click **Assign Objects**. This displays the Assign Objects dialog, Figure 257. This dialog lists all logs that have been imported from the remote history node and are now in the resulting xml file. The status of each log is indicated by a color-coded icon in the Access Name column. The key for interpreting these icons is provided in the lower left corner of the dialog.
4. Use the Assign Objects dialog to make any adjustments to the imported tag specifications that may be required.

Making Adjustments for Tags

Adjustments to the Object and Property names are required to make the History configurations on the source and consolidation nodes match. More than one property may be logged for a given object. Generally, the property logs for all properties being logged for an object are contained within one Log Configuration Aspect. When importing log configurations from Information Management nodes, by default the importer will create a separate object and Log Configuration aspect for each property being logged. Thus, there will be a one-to-one correspondence for property logs on the source and consolidation nodes; however, additional objects

and Log Configuration aspects may be created on the consolidation node that do not exist on the source node. Use the following procedure to avoid this.

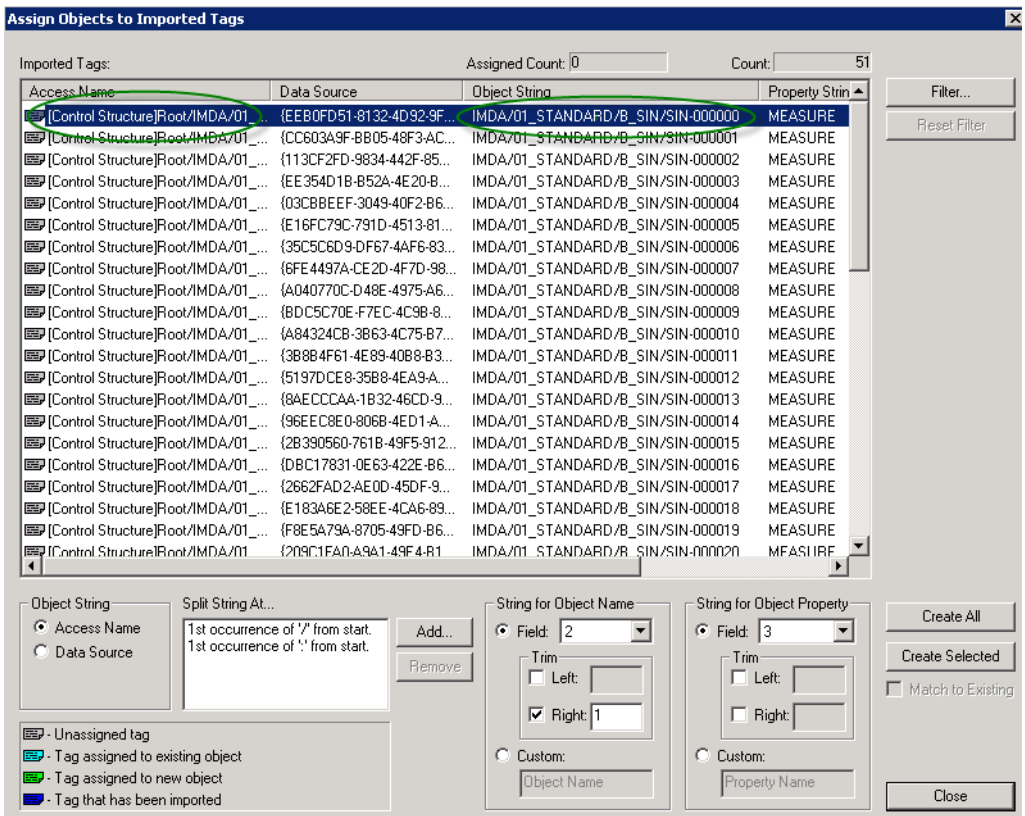


Figure 257. Assign Objects Dialog

To modify the object and property names to make the History configurations match:

1. Split the Access Name into three fields - field 2 will become the object name path, and field 3 will become the property name. In this example, the *[control Structure]Root/* is removed from the object path. If this is not removed from the object path, the import will not work correctly. To split the Access Name:

   a. Make sure the selected Object String is **Access Name**.

b. Click the **Add** button associated with the Split String At field. This displays the Add String Split Rule dialog, Figure 258.

c. For Split Character, enter the character that splits the structure and the object path. This is, the forward slash (/). Leave the other parameters at their default settings: Occurrence = **1st**, Search Direction = **Forward**.

d. Click **OK**. Figure 259

e. Click the **Add** button associated with the Split String At field. This displays the Add String Split Rule dialog.

f. For Split Character, enter the character that splits the object and property names. For Information Management, the colon (**:**) character is used. Leave the other parameters at their default settings: Occurrence = **1st**, Search Direction = **Forward**.

g. Click **OK**.

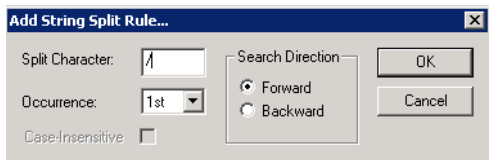


Figure 258. Add String Split Rule

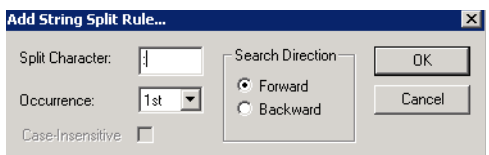


Figure 259. Add String Split Rule

2. Make Field 2 of the split Access Name (text left of the split character) the Object String, and trim (remove) the split character (in this case, the comma) from the Object String. To do this (reference Figure 260):

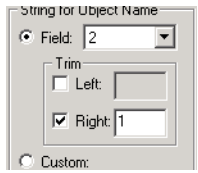


Figure 260. Object String Specification

a. In the String for Object Name section, select **2** from the **Field** pull-down list. This specifies Field 2 (text left of split character) as the Object String.

b. In the Trim section check the **Right** check box and enter **1**. This specifies that one character will be trimmed from the right end of the text string.

As changes are made, the Object String will change accordingly. The resulting Object String is shown in Figure 261.

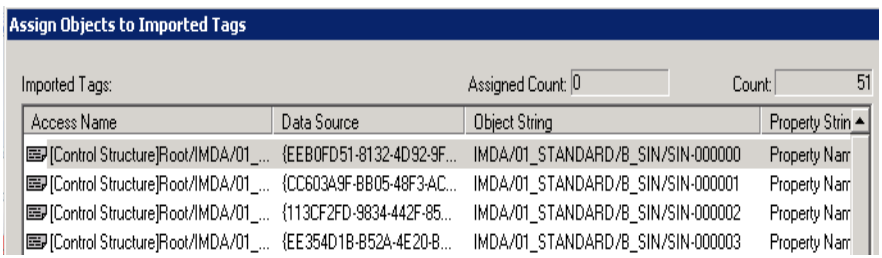


Figure 261. Object String Modified Showing Results of Object String Adjustment

3. Make Field 3 of the split Access Name (text right of the split character) the Property String. To do this, in the String for Object Property section, select **2** from the **Field** pull-down list, Figure 262. This specifies Field 2 (text right of split character in the Access Name) as the Object Property String.

Figure 262. Property String Adjustments

The finished specification is shown in Figure 263.

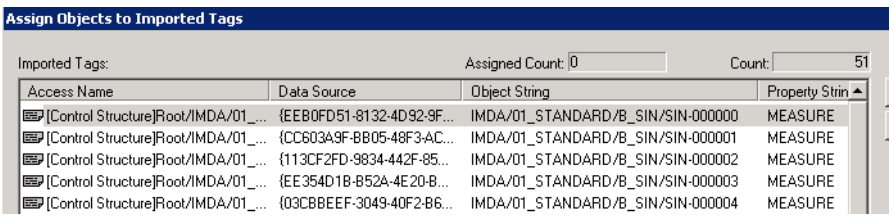


Figure 263. Completed Assign Objects Specification

After any desired adjustments are made, continue with the procedure for Creating the Object/Tag Associations on page 378.

Creating the Object/Tag Associations

At this point, either create object/tag associations for all tags, or select a portion of the tag list for which to create associations. For example, when the list is very large and only a portion of the tag list is to be processed. If the History Access Import tool is exited and then restarted, the **Use Existing Definition File** option (step 1 of Reading the Remote Log Configurations on page 369) can be used to read the log configuration information from the existing XML file rather than re-read the remote history server.

To create the associations:

1. For the entire tag list, click **Create All**. For a partial list, first select the tags, then click **Create Selected**.

   The result (for creating selected tags) is shown in Figure 264. The status indicator for tags that have been assigned to new objects will be green.

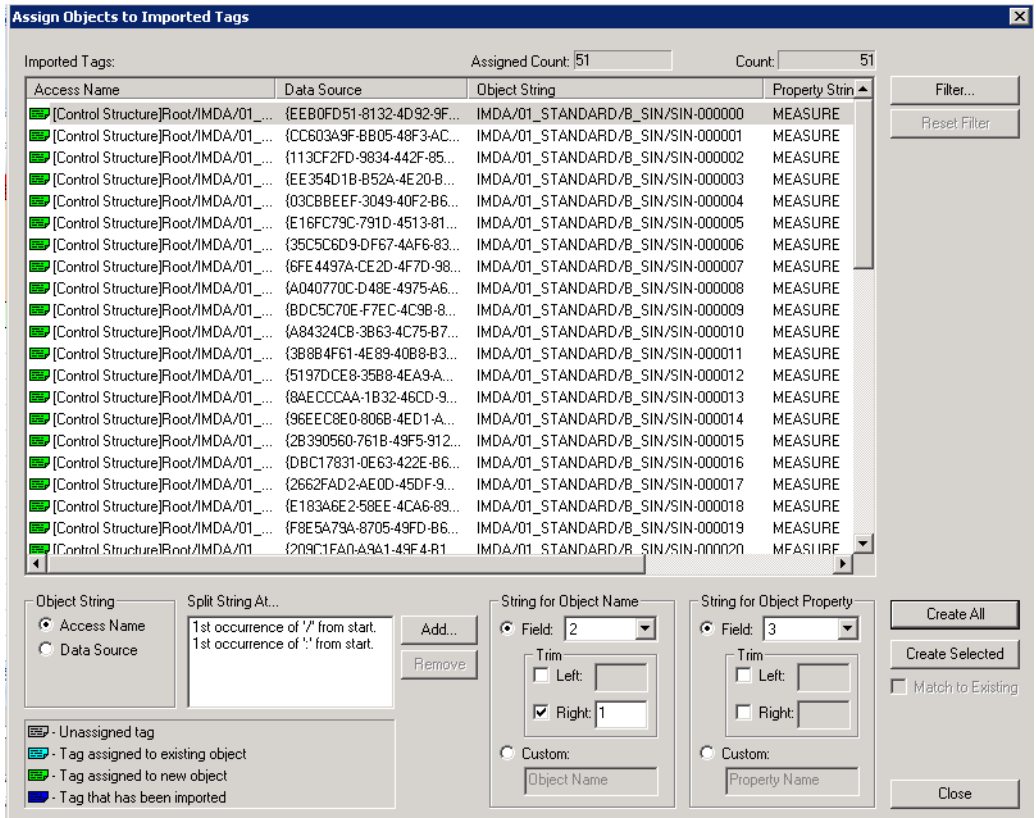


Figure 264. Create Selected Result

2. Click **Close**. This displays the Assign Objects summary again. This time the summary indicates the result of the assign objects process, Figure 265.

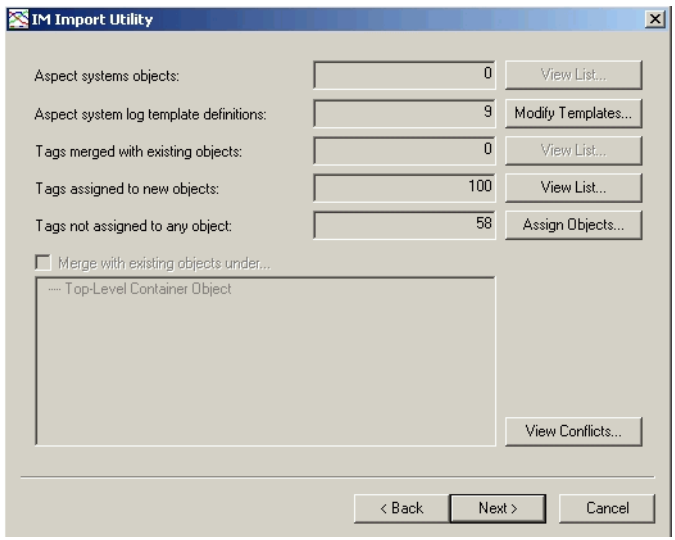


Figure 265. Assign Objects Summary

3. Click **Next**. This displays a dialog for specifying parameters for creating the new Log Template and Log Configuration aspects.
4. Continue with the procedure for Specifying the Service Group and Other Miscellaneous Parameters on page 380.

Specifying the Service Group and Other Miscellaneous Parameters

Use the dialog in Figure 266 to specify the History Service Group for the source node from which the log configurations are being imported. This dialog also is used to group the Log Configuration aspects into folders. This improves system performance when the tag list is very large, because it is time consuming to open a folder with an excessive number of objects.

To use this feature, click the **Enable** check box in the Grouping New Imported Objects section. The default size for each group is 250 tags. Create larger or smaller groups as needed. Do not make groups larger than 1000 tags.

Enter a prefix for each group's folder name. The specified counter type (A-Z or 0-9) will be appended to the prefix to make each folder name unique. For example, by

specifying **eng130_** as the prefix, and **0...9** as the Counter Type, the following group names will be used: **eng130_0**, **eng130_1**, and so on.

Set the Source Type to **Remote Historian**.

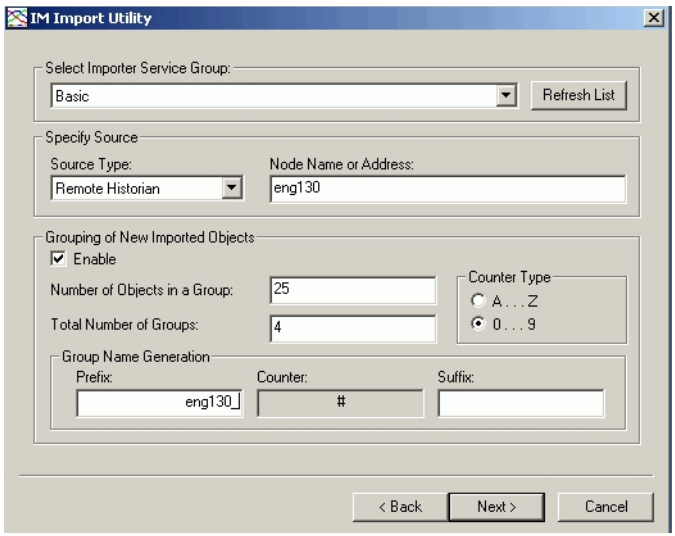


Figure 266. Selecting Service Group

Click **Next** when done. This displays an import preview where settings can be confirmed, Figure 267.

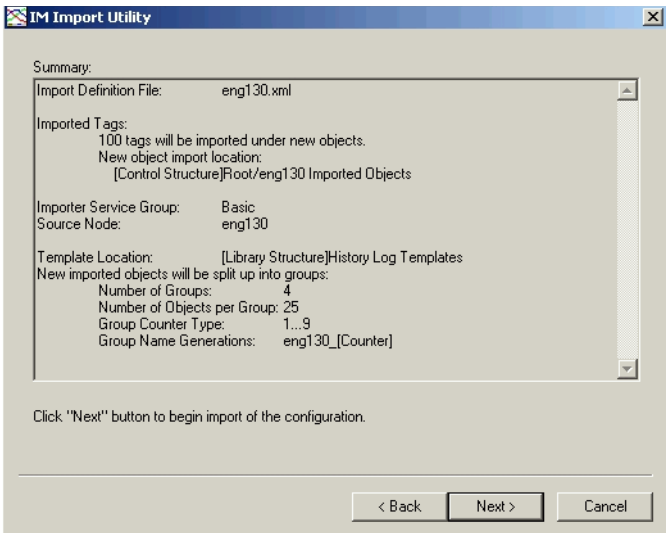


Figure 267. Import Summary

Read the preview. When ready, click **Next** to start the import.

The progress status is displayed while the log configurations are being imported, Figure 268. When the *Import has completed successfully* message appears, click **Finish**.

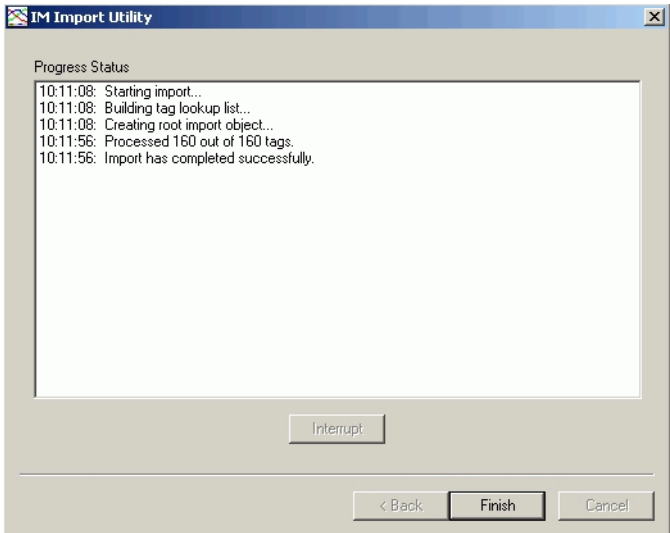


Figure 268. Progress Status

The Log Configuration aspects have now been created for the imported log configurations.

A History Source object must be present above the Log Configuration aspects in the structure where they reside, Figure 269. If the History Source object is not present, refer to the procedure for adding History Source objects in Configuring Node Assignments for Property Logs on page 189.

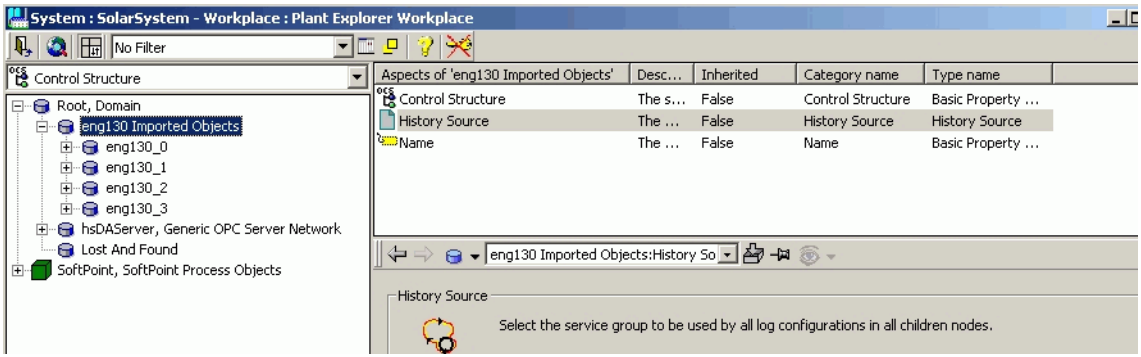


Figure 269. Example, History Source

The Log Configuration aspect's **Status** tab (or any Desktop tool such as DataDirect or desktop Trends) can now be used to read historical data from the remote log. An example is shown in Figure 270.

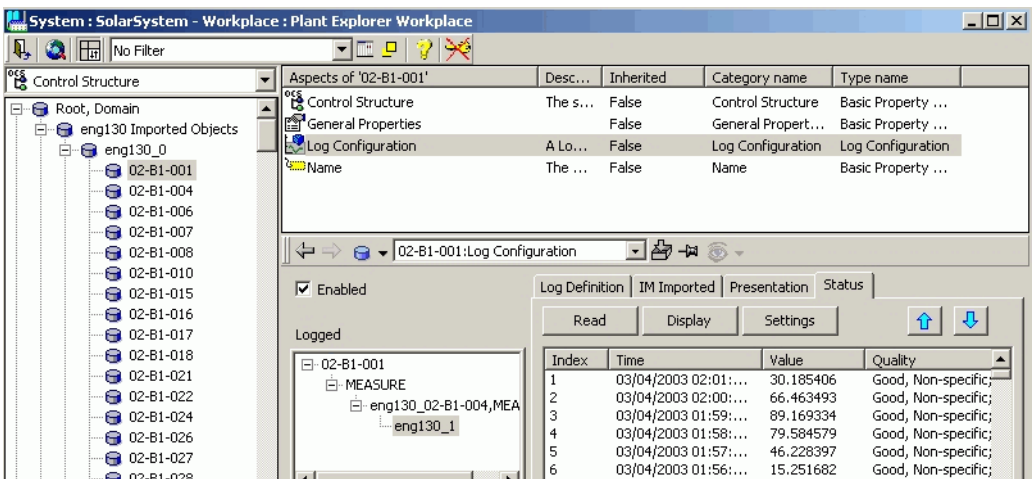


Figure 270. Example, Viewing Historical Data from an Imported Log Configuration

Finish the set-up for historical data consolidation by Collecting Historical Data from Remote History Servers on page 385.

Collecting Historical Data from Remote History Servers

The logs created in the previous procedure simply provide a window for viewing the historical data. They DO NOT perform a data collection function, nor do they actually store data.

The procedure described in this section creates a second set of property logs that will collect historical data from the remote history server, and store the data on the consolidation node. This procedure uses the Microsoft Excel add-in Bulk Import tool. Detailed instructions for this tool are provided in Bulk Configuration of Property Logs on page 270. This section provides specific instructions for using this tool to instantiate Log Configuration aspects to collect historical process data from remote history logs.

The basic steps are described below. Further details are provided in the referenced sections:

- **Determine the storage interval of the remote logs**. The storage interval for the new logs being created should match or be a multiple of the storage rate for the remote logs being collected. To determine the storage rate of the remote logs, refer to a Log Configuration aspect created for an imported log configuration. Refer to Determining the Storage Rate on page 385.
- **Create a new Log Template**. This procedure is basically the same as Building a Simple Property Log on page 196, except that the direct log in the property log hierarchy must be an IM 3.5 Collector link type, rather than a basic history trend log. Also, the storage interval recorded in the previous step will be used. For details refer to Creating a New Log Template on page 386.
- **Use the Bulk Import tool** to instantiate the property logs. When this is done, change the Item IDs for all objects, and apply the new Log Template. Refer to Using the Bulk Import Tool to Instantiate the Property Logs on page 388.

Determining the Storage Rate

The storage interval for the new logs that are being created should match or be a multiple of the storage rate for the remote logs being collected. The storage rate can be found using one of the Log Configuration aspects created for an imported log configuration. The storage rate is indicated on the **IM Imported** tab, Figure 271. Record this storage rate for use when creating the new Log Template as described in Creating a New Log Template on page 386.

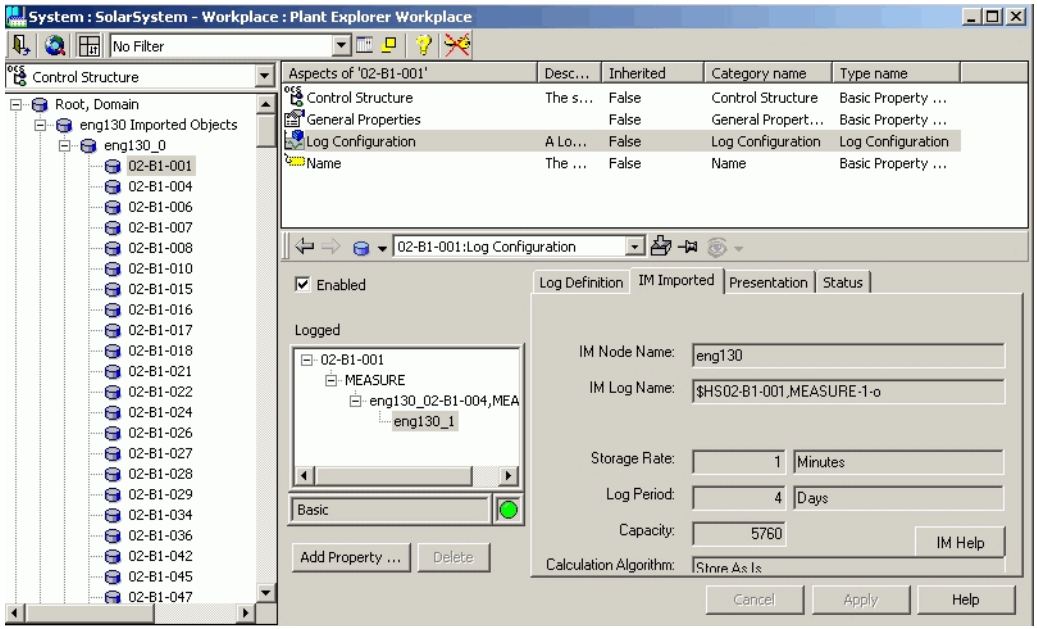


Figure 271. Example, Looking Up the Storage Rate for the Remote Logs

Creating a New Log Template

This procedure is basically the same as Building a Simple Property Log on page 196, except that the direct log in the property log hierarchy must be an IM History Log link type. When adding the direct log in the property log hierarchy, click the **Linked** check box, and then select the **IM History Log link** from the Server Type pull down list, Figure 272.

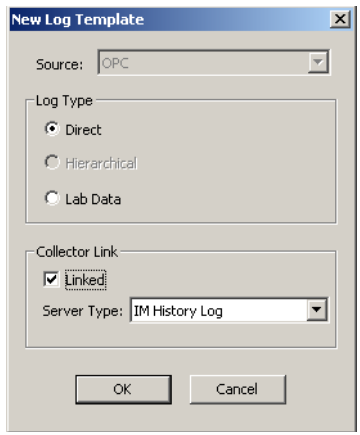


Figure 272. Adding the Direct Log as IM History Log Link

When configuring the Data Collection parameters, set the storage interval to match or be a multiple of the remote log's storage interval, Figure 273.

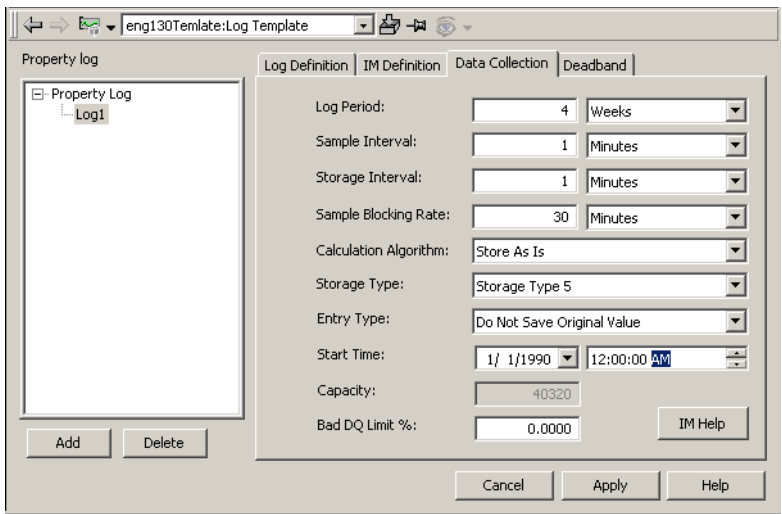


Figure 273. Configuring the New Template

Using the Bulk Import Tool to Instantiate the Property Logs

This section provides specific instructions for using the Bulk Import tool to create property logs that collect from remote history logs. These property logs will be added to the same Log Configuration Aspects that were created for the imported log configurations (Importing Remote Log Configurations on page 369). For further information regarding the operation of the Bulk Import Tool, refer to Bulk Configuration of Property Logs on page 270.

Start by loading the sheet from the system. When the location from which to import objects is specified, select the root object under which the objects created for the imported log configurations reside, Figure 274. Also check the **Include Child Objects** check box.

For this application, specifying a filter prior to importing is not required.

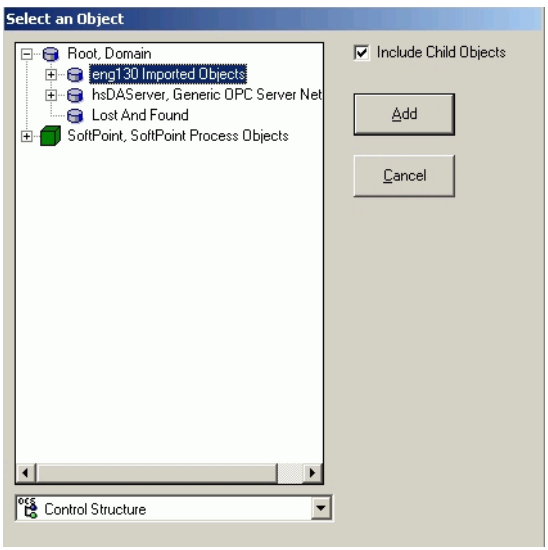


Figure 274. Selecting the Objects to Import

Modify the Item IDs in the spreadsheet to make the IM data source for the direct logs reference the remote history log names. The IM Data Source specification has the following form: **$HS***object*, *attribute*-*n*-**o***IPaddress.*

The IP address is only required when collecting from multiple sites, and some sites use identical log names. The IP address is not required when all data sources are unique.

Scroll to the Item ID column. The remote log name is embedded in the corresponding object's Item ID, Figure 275.

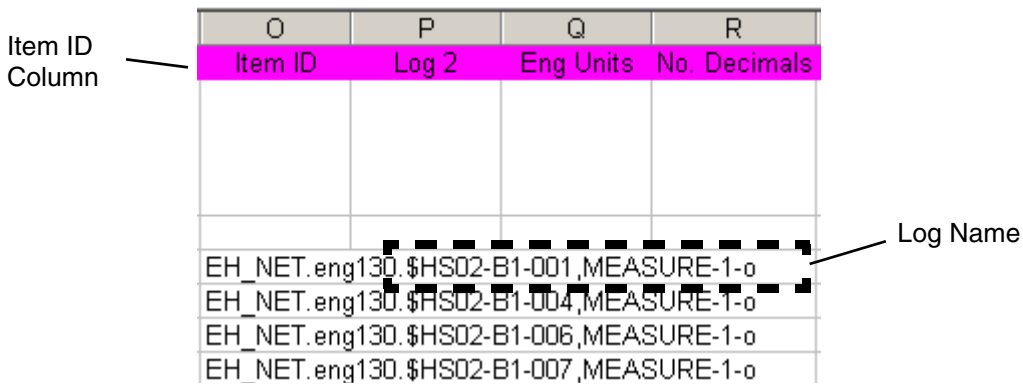


Figure 275. Log Name Embedded in Item ID

The portion of the Item ID text string before the start of the .$HS string must be stripped away and be replaced with **DS,** Figure 276. This instructs the Bulk Import tool to use the remainder of the Item ID (**$HS***object*, *attribute***-***n***-o***IPaddress*) as the log's data source.

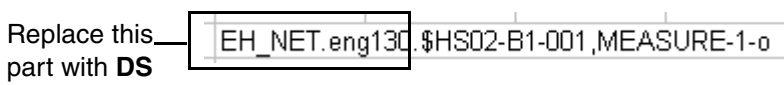


Figure 276. Trimming the Item ID

Use Excel's Find and Replace function to do this, Figure 277.

Be sure to remove the wildcard (*) character from the Find criteria; otherwise, the entire Item ID (not just the prefix) will be replaced.

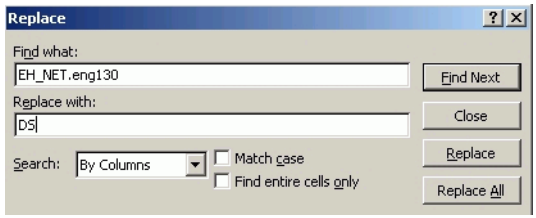


Figure 277. Find/Replace Example

The result of the find and replace operation is shown in Figure 278.

O	P	Q
Item ID	Log 2	Eng Units
DS.$HS02-B1-001,MEASURE-1-o		
DS.$HS02-B1-004,MEASURE-1-o		
DS.$HS02-B1-006,MEASURE-1-o		
DS.$HS02-B1-007,MEASURE-1-o		
DS.$HS02-B1-008,MEASURE-1-o		

Figure 278. DS Prefix Replacing Original Prefixes in Item IDs

When consolidating a dual log configuration, edit the ItemID string to specify BOTH logs in the dual log configuration. The syntax for this is described in Additional ItemID Changes for Dual Logs on page 390.

Additional ItemID Changes for Dual Logs

To consolidate historical data from a dual log configuration, modify the ItemID string to specify as the data source, BOTH history logs in the dual log configuration.

If dual logs are not being consolidated, skip this section and continue with Selecting the Log Template Configuration on page 392.

The correct syntax depends on if it is a simple dual log configuration (no secondary history logs), or a hierarchical configuration with secondary history logs.

Simple Dual Log Configuration

For a simple dual log configuration (two history logs), Figure 279, the syntax is:

DS.$HS*object*,*attribute*-**1-d** or **DS.$HS***object*,*attribute*-**1-d2**

The number following the first dash (-1) identifies the first log added to the log template. The **d** following the second dash identifies the matching dual log, in this case, the second log added to the template. The number following -d defaults to: *first dash number+1*. With this configuration, the matching dual log is always -2; therefore the number does not have to be entered following d. For example:

DS.$HS02-B1-001,MEASURE-1-d = DS.$HS02-B1-001,MEASURE-1-d2

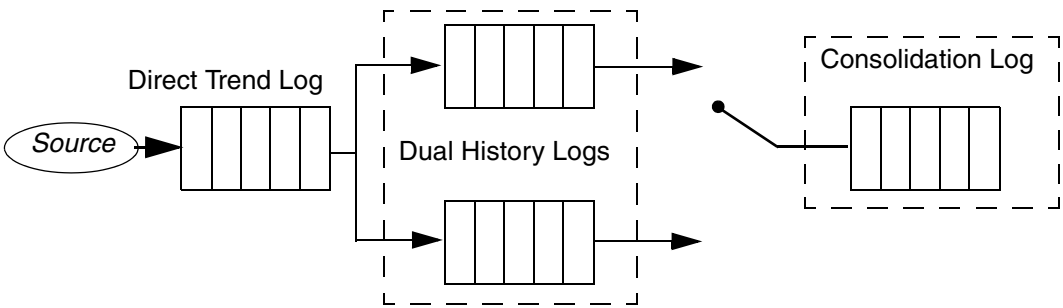


Figure 279. Simple Dual Log Configuration

Hierarchical Dual Log Configuration

When the dual log configuration includes secondary history logs, the syntax may depend on the order in which the individual logs were added to the log configuration template. This is illustrated in Figure 280 and Figure 281.

In Figure 280, since the two source logs (-3 and -4) follow sequentially, the number following -d is optional. In Figure 281, since the two source logs (-2 and -4) are NOT sequential, the number following -d MUST be specified.

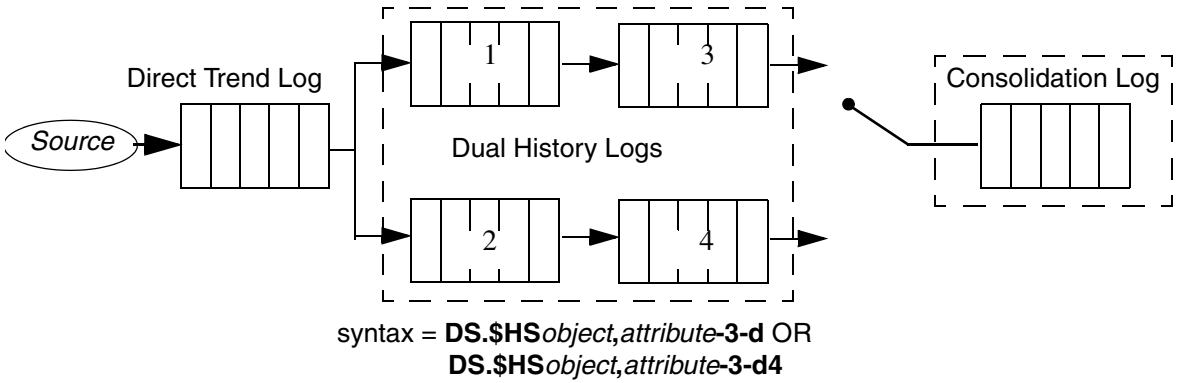


Figure 280. Hierarchical Dual Log Configuration, Example 1

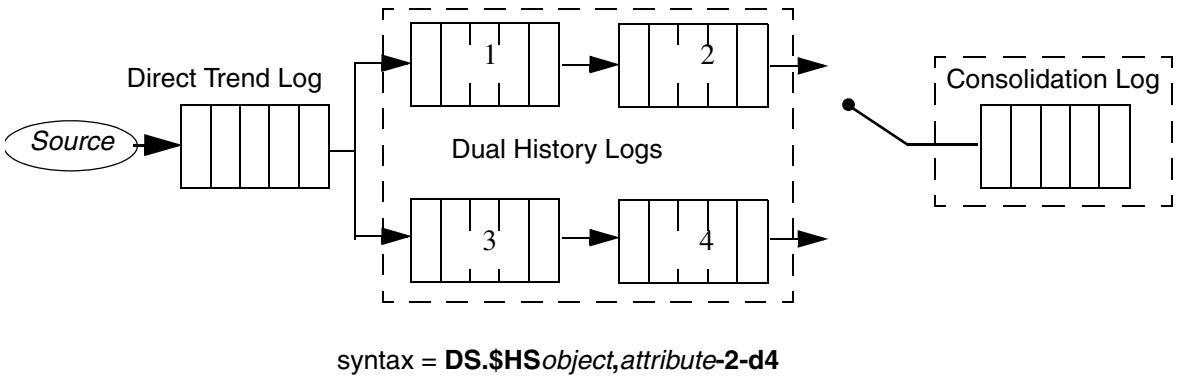


Figure 281. Hierarchical Dual Log Configuration, Example 2

Selecting the Log Template Configuration

Replace the current Log Template specification with the new Log Template.

1. Delete the existing template before replacing it by selecting the **Delete** option on the Bulk Import Menu.

This *Delete* operation deletes all the history data of the logs available in the Excel sheet from the 800xA system.

2. From the Property template Column for the first (top) object, choose **Template List** from the context menu.

3. Select the template from the list provided Figure 282.

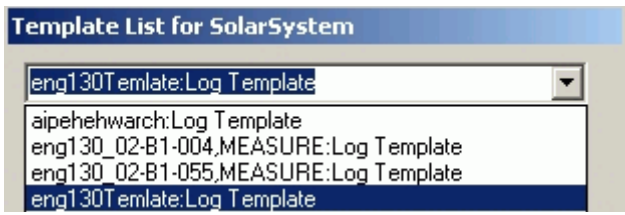


Figure 282. Changing the Log Template Specification

4. Click on a corner of the cell, and pull the corner down to highlight the Property Template column for the remaining objects. The template specification will automatically be entered in the highlighted cells when the mouse button is released, Figure 283.

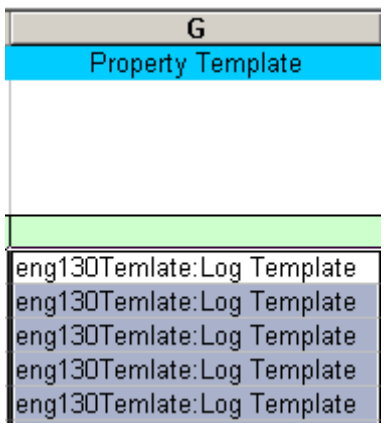


Figure 283. Applying the New Log Template Specification to All Objects

5. When finished, run the Bulk Configuration utility to update the log aspects from the spreadsheet. To do this, choose **Bulk Import>Update Log Aspect from Sheet**.

To check the results, go to the location where the Log Configuration aspects have been instantiated, and use the **Status** tab of a Log Configuration aspect to read

historical data for the log. These logs will be located under the same root object as the imported log configuration aspects. An example is shown in Figure 284.

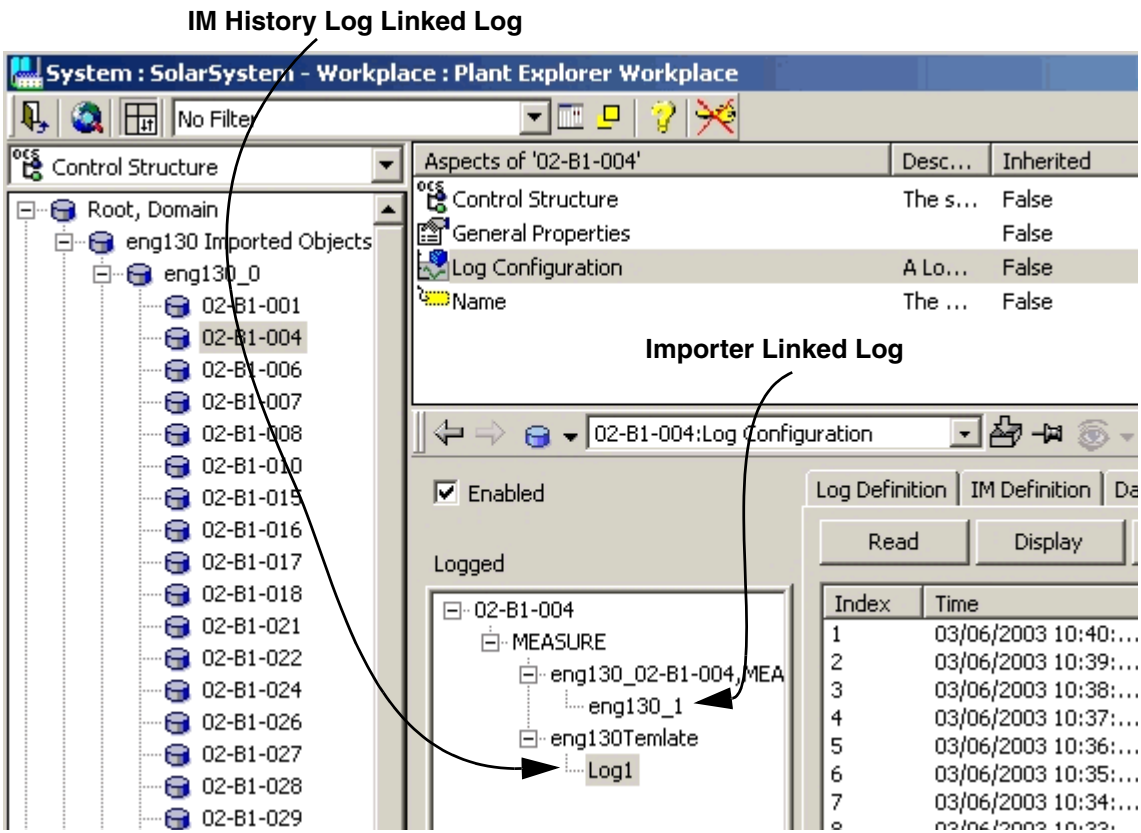


Figure 284. Checking Results

Collecting from TTD Logs

Property logs can be configured on the Information Management node to collect from TTD logs on an AC 400M Controller.

In order to do this the TTD log configurations must be present in the Aspect Directory. Refer to the applicable AC 400 Controller documentation for this procedure. This creates a Log Configuration Template and Log Configuration Aspect for each TTD log group. The Log Configuration Aspects will be located under an object in the Control structure as specified during this procedure.

This step does not establish historical data collection and storage on the local node for the imported logs. The Log Configuration aspects created as a result of this step merely provide a window for viewing the historical data stored in the TTD logs on the remote nodes.

Once the TTD log configurations are present in the Aspect Directory, follow the guidelines in this section to create property logs to collect from the TTD logs. There are two basic steps as described below. Further details are provided in the referenced sections:

- **Create a new Log Template**. This procedure is basically the same as Building a Simple Property Log on page 196, except that the direct log in the property log hierarchy must be an IM Collector link type, rather than a basic history trend log. Also, the Collection Type must be OPC HDA. For details refer to Creating a New Log Template on page 395.
- **Use the Bulk Import tool** to instantiate property logs to collect from the TTD logs. When this is done, specify the Item IDs for all objects to reference the TTD logs, and apply the new Log Template. Refer to Using the Bulk Import Tool to Instantiate the Property Logs on page 397.

Creating a New Log Template

This procedure is basically the same as Building a Simple Property Log on page 196, except that the direct log in the property log hierarchy must be an IM 3.5 Collector link type, and the Collection Type must be OPC HDA.

When adding the **Direct** log in the property log hierarchy, click the **Linked** check box, and then select the **IM History Log** from the Server Type pull down list, Figure 285.

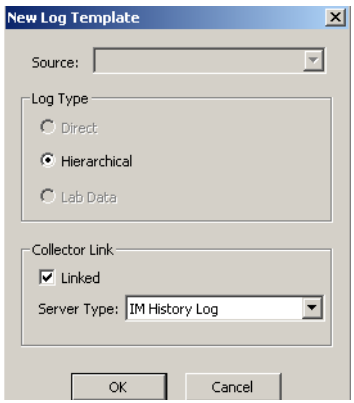


Figure 285. Adding the Direct Log as IM History Log

When configuring the IM Definition parameters, set the Collection Type to **OPC HDA**, Figure 286.

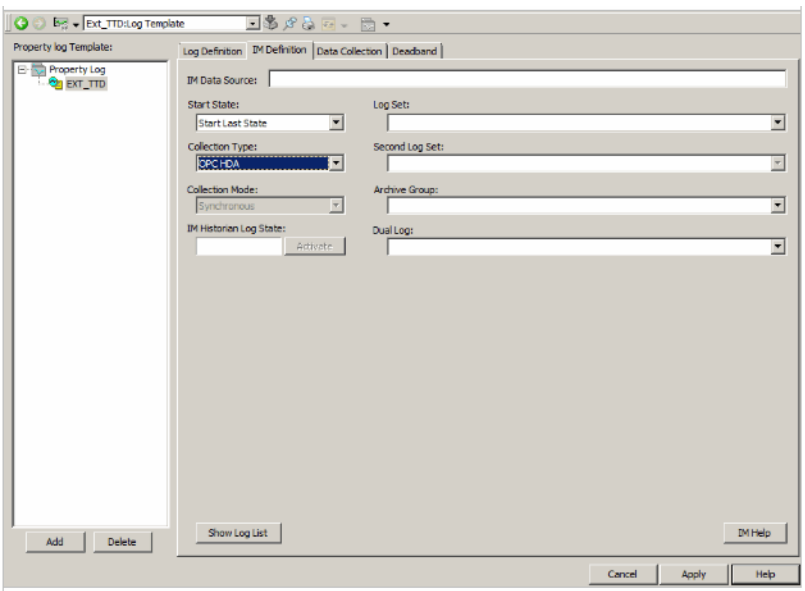


Figure 286. Configuring the New Template

Using the Bulk Import Tool to Instantiate the Property Logs

This section provides specific instructions for using the Bulk Import tool to instantiate the property logs to collect from the TTD logs. The property logs will be added to the same Log Configuration Aspects that were created when the TTD log configurations were imported. For further information regarding the operation of the Bulk Import Tool, refer to Bulk Configuration of Property Logs on page 270.

Start by loading the sheet from the system. When the location from which to import objects is specified, select the root object under which the objects created for the imported TTD log configurations reside, Figure 287. Also check the **Include Child Objects** check box.

For this application, a filter is not required prior to importing.

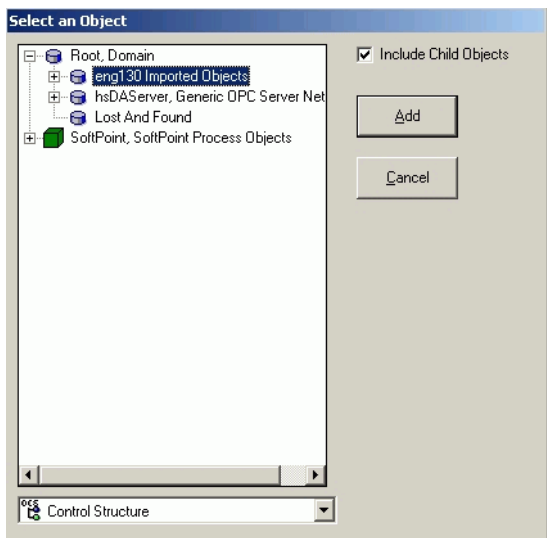


Figure 287. Selecting the Objects to Import

Modify the Item IDs in the spreadsheet to make the IM data source for the direct logs reference the TTD log names. The IM Data Source specification has the following form: **DS.***object***:***property***,***TTDlogname.* To do this, do the following for each TTD log specification:

1. Obtain the object name from the Object column. The object name is the text string that follows the last delimiting character in the full text string, Figure 288.

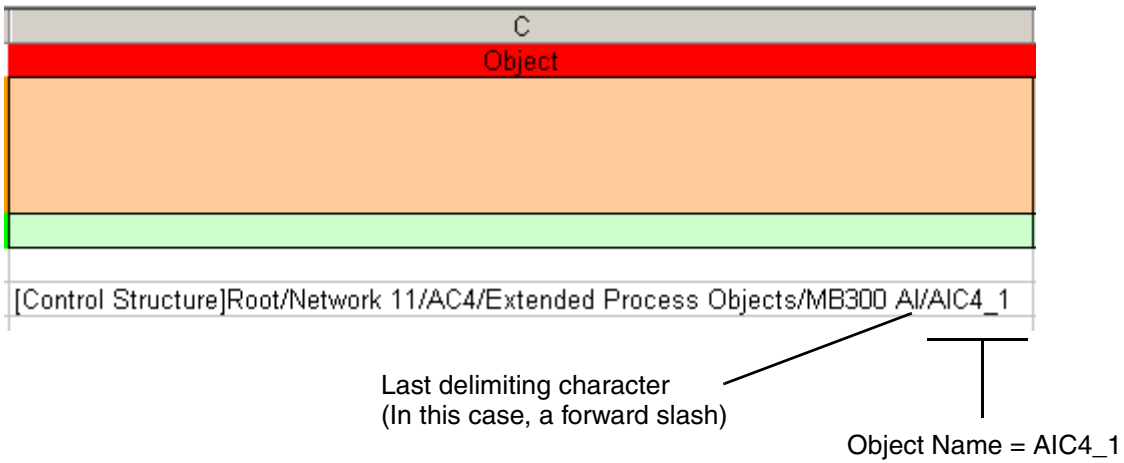


Figure 288. Obtaining the Object Name

2. Obtain the property name from the Property column, Figure 289.

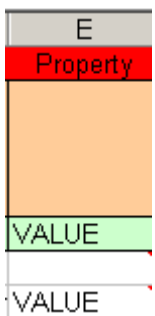


Figure 289. Obtaining the Property Name

3. Obtain the TTD log name from the applicable log template which was created in the Aspect Directory for the TTD log group, Figure 290.

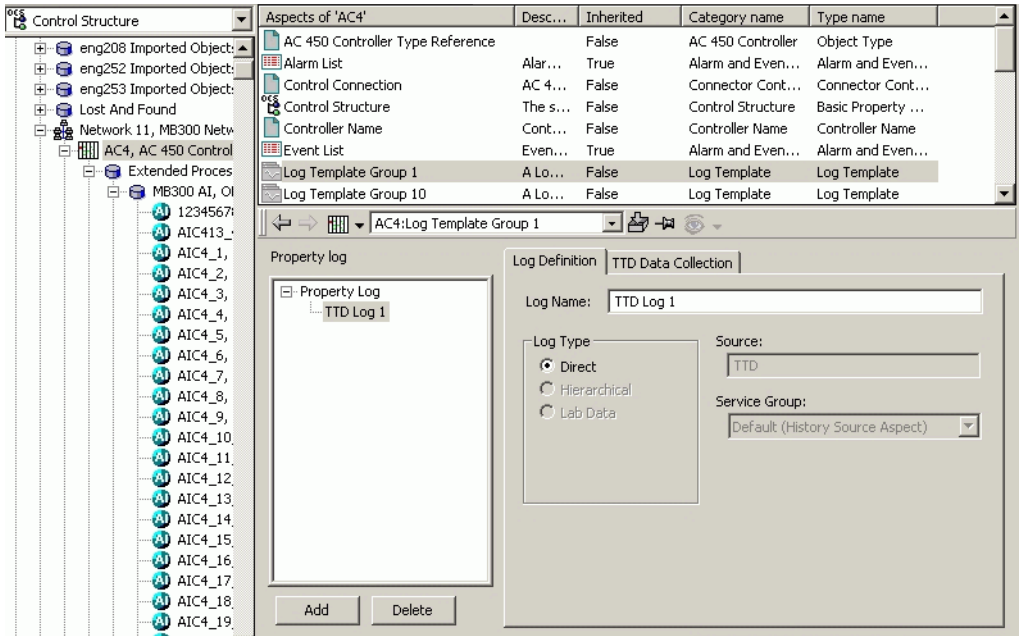


Figure 290. Obtaining the TTD Log Name

4. Enter the Data Source specification in the Item ID column, Figure 291.

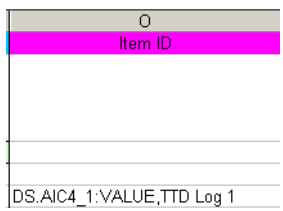


Figure 291. Entering the Data Source Specification

Replace the current Log Template specification with the new Log Template. To do this, right click on the Property template Column for the first (top) object, and choose **Template List** from the context menu. Then select the template from the list provided Figure 292.

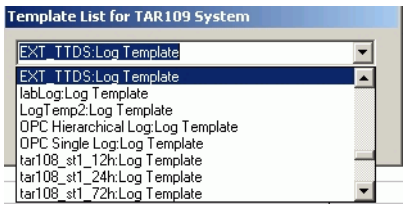


Figure 292. Changing the Log Template Specification

Click on a corner of the cell, and pull the corner down to highlight the Property Template column for the remaining objects. The template specification will automatically be entered in the highlighted cells when the mouse button is released.

When finished, run the Bulk Configuration utility to update the log aspects from the spreadsheet: choose **Bulk Import>Update Log Aspect from Sheet**.

To check the results, go to the location where the Log Configuration aspects have been instantiated, and use the **Status** tab of a Log Configuration aspect to read historical data for the log. An example is shown in Figure 293.

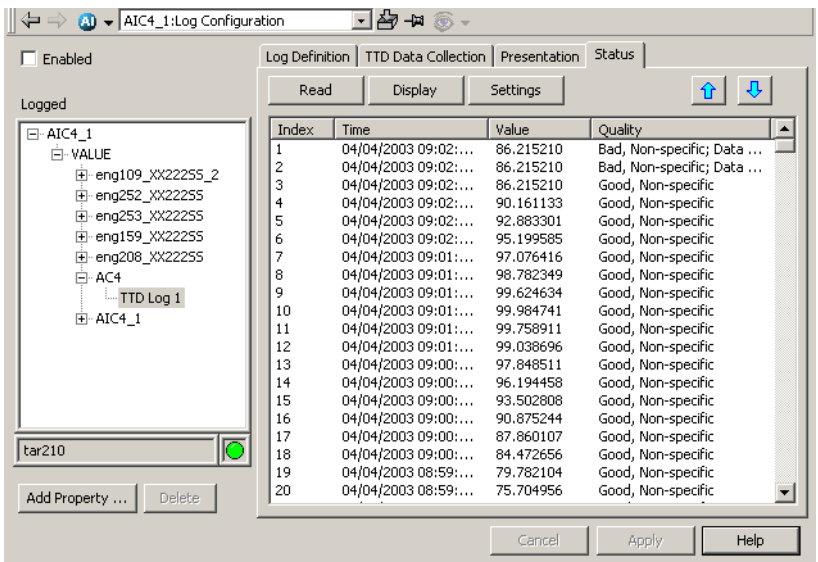


Figure 293. Checking the Results

Consolidating Message Logs and PDLs

All Information Management consolidation functionality is limited to 800xA 5.1 to 800xA 5.1 Systems.

Message logs and PDLs on one or more history server nodes can be consolidated on one central *Consolidation* node. This is done by creating a job with one or more IM Consolidation actions to run on the *target* consolidation node. A separate action is required for each source node from which the consolidation node will consolidate messages and PDL data.

Follow these basic steps and refer to the referenced section in each step for details:

1. When consolidating message logs, make sure the consolidation (LocalName) logs already exist on the consolidation node and that they are correctly configured. To consolidate PDL data for Batch Management applications, create one PDL_MESSAGE log on the consolidation node to consolidate PDL messages from all Batch Management nodes. Procedures for configuring message logs is described in Section 7, Alarm/Event Message Logging.

Avoid consolidating Batch Management created PDL Message Logs if a PDL Message Log does not exist on the Information Management Consolidation node.

Perform the following procedure if a PDL Message Log has not been configured on the Information Management Consolidation node, or if it has been configured incorrectly. Although a PDL Message Log created by Batch Management will be consolidated from the target node, if a PDL Message Log has not been configured on the Information Management Consolidation node, or if it has been configured incorrectly, it will not contain any messages for any of the tasks when viewed on the Consolidation node.

a. Correctly configure the PDL Message Log on the Consolidation node.

b. Use SQLPLUS on the target node to set the Archive Status to 0 for all tasks in the task table.

c. This forces the PDL Message Logs to be reconsolidated at which time the messages will be consolidated.

2. When consolidating message logsm, a user account must be set up on the source node. This account must use the same user name and password as the installing user for the target consolidation node. Also, this user must be added

to the HistoryAdmin group. Detailed instructions for adding user accounts are provided in the section on Managing Users on page 589.

3. On the consolidation node, create the Job Description object and configure the schedule and start condition(s) through the respective aspects. Refer to Setting Up the Schedule on page 402.
4. Add an action aspect to the Job Description object. Refer to Setting Up the Schedule on page 402.
5. Open the action aspect view, and select **IM Consolidate Action** from the Action pull-down list. Then use the dialog to specify the source and target nodes, and the source and target log names. Each action is used to specify logs for ONE source node. Refer to Using the Plug-in for IM Consolidation on page 404.
6. Repeat Step 4 and Step 5 for each source node from which the consolidation node will consolidate message and PDL data.

Reuse of Batch ID (TCL) is supported by PDL Consolidation. A warning is displayed in the log file when processing a remote task entry (TCL batch) that already exists on the local node with a different start time. The user can decide what to do about the duplicate. If nothing is done, the local entry will eventually be removed through the archive process.

Setting Up the Schedule

Consolidation of alarm/event messages and/or PDL data from an Information Management node is scheduled by adding a job in the Scheduling structure, and configuring the job's scheduling definition aspect. The schedule may be cyclic, periodic, weekly, monthly, a list of dates and times, or conditional. The schedule is associated with a specific node via the IM Consolidation action aspect which must be added to the job object. This section quickly demonstrates how to add a job and use the scheduling definition aspect to set up a weekly schedule. It also describes how to add and use the IM Consolidation action aspect. For further information on jobs and scheduling options, refer to the section on scheduling in *System 800xA Information Management Data Access and Reports (3BUF001094*).*

Adding a Job and Specifying the Schedule

Jobs are created in the Scheduling structure. To create a job:

1. In the Plant Explorer, select the **Scheduling Structure**.
2. Right-click on **Job Descriptions** and choose **New Object** from the context menu.
3. Add the Job object as a **Job Description** object. Assign the object a logical name.
4. Click **Create**. This creates the new job under the Job Descriptions group, and adds the Schedule Definition aspect to the object's aspect list.
5. Click on the **Scheduling Definition** aspect to display the configuration view, Figure 294. This figure shows the scheduling definition aspect configured as a weekly schedule. Consolidation for the specified node will occur every Sunday at 11:00 PM, starting July 3rd, and continuing until December 31st at 11:00 PM.

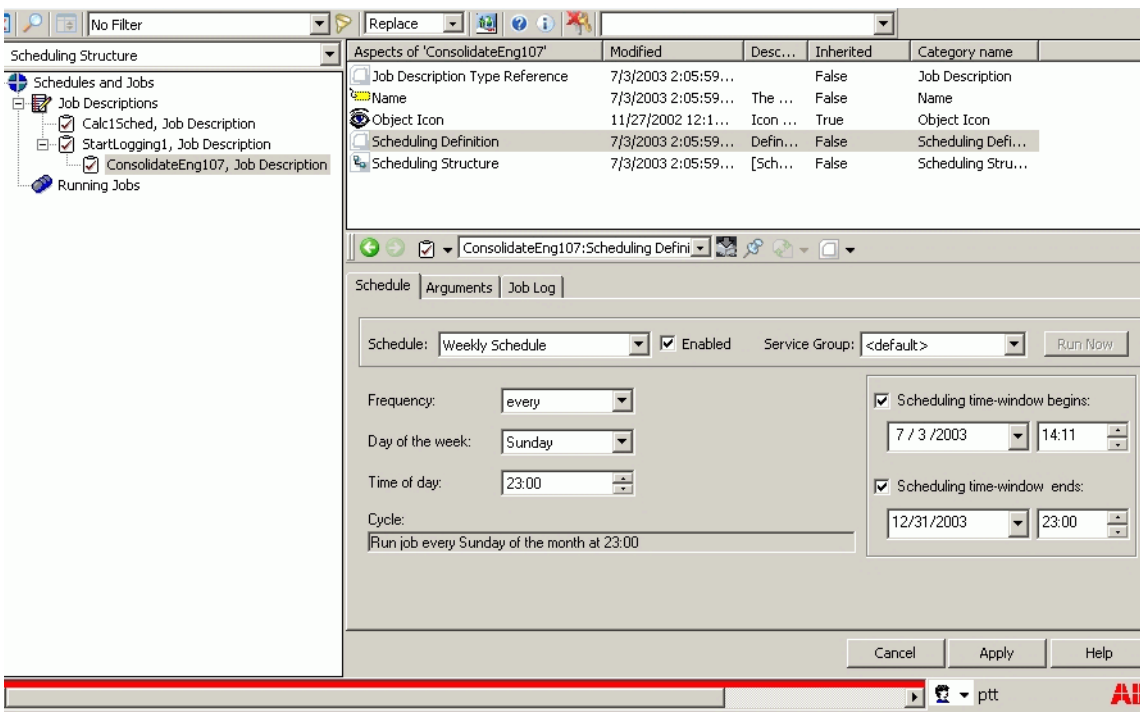


Figure 294. Scheduling Definition Configuration View

Adding and Configuring the Archive Action

1. Right-click on the Job object (for example ScheduleArchGrp1) and choose **New Aspect** from the context menu.
2. In the New Aspect dialog, browse to the Scheduler category and select the Action aspect (**Scheduler>Action Aspect>Action Aspect**).

   Use the default aspect name, or specify a new name.
3. Click **Create** to add the Action aspect to the job.
4. Click on the Action aspect to display the configuration view.

Using the Plug-in for IM Consolidation

To configure the IM Consolidation action:

1. Select **IM Consolidation Action** from the Action pull-down list. This displays the Consolidate Logs plug-in, Figure 295.
2. Specify the node name and IP address for the *source* node from which the consolidation node will consolidate messages and PDL data.

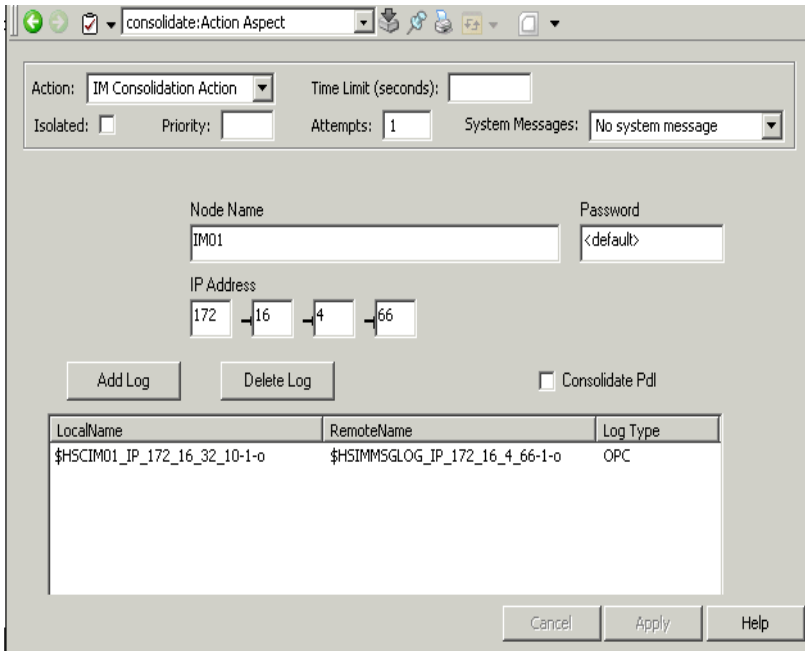


Figure 295. Consolidate Logs Aspect View

3. The password is used to connect to Oracle on the remote IM server. If the source IM node is in the same domain and has the same service account as the destination IM, the field can remain <default>. If the source IM is in another domain, the Oracle Administration password for the remote IM must be entered. If the source IM password is updated, the consolidation action must be updated to reflect the change.

4. To consolidate PDL data, select the **Consolidate PDL** check box. For Batch Management, this includes PDL message logs, so it is not required to specify PDL message logs in the message log list as described in Step 5. However, the PDL_MESSAGE log must exist on the consolidation node.

   When consolidating PDL data for MOD 300 software applications, the corresponding MODMSGLOG must be added to the message log list.

5. Specify the message logs to be consolidated as follows:

a. Click the **Add Log** button.

b. In the Add Log dialog enter the full log name of the consolidation log in the Local Name field, and the full log name of the log from which messages are being collected in the Remote Name field, Figure 296.

the **$HS** prefix and **-n-o** suffix must be used when specifying the local and remote log names.

c. Select the Message Log Type - **OPC**, **DCS**, or **Audit** Trail.

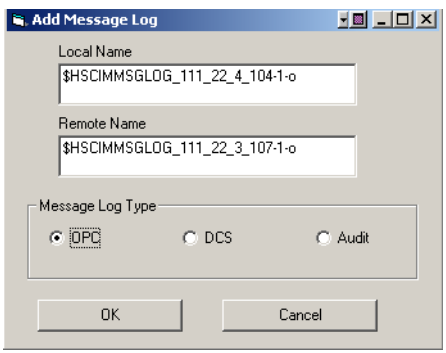


Figure 296. Add Log Dialog

d. Click **OK** when finished. This puts the log specification in the list of message logs to be consolidated, Figure 297.

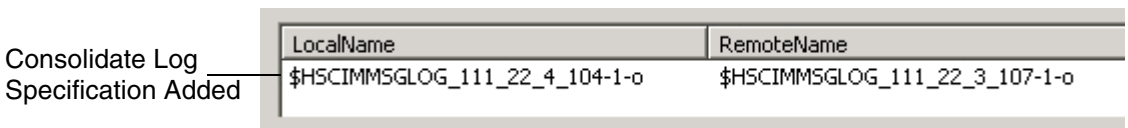


Figure 297. Log Added to List

Repeat Step a through Step d- for as many message logs as required.

Section 13 History Database Maintenance

This section provides instructions for optimizing and maintaining the history database after configuration has been completed. Utilities for some of these maintenance functions are available via the Windows task bar.

The procedures covered in this section are:

- Backup and Restore on page 408 describes how to create backup files for the current History database configuration, restore a history database configuration from backup files, and synchronize the Aspect Directory contents with the current Information Management History database configuration.
- Schedule History Backups on page 438 using the Report Action of the Action Aspect to schedule a History backup.
- Starting and Stopping History on page 437 describes how to stop and start history processes under PAS supervision. This is required for various database maintenance and configuration procedures.
- Starting and Stopping Data Collection on page 439 describes various methods for starting and stopping data collection.
- Viewing Log Runtime Status and Configuration on page 442 describes how to use the Log List aspect for viewing runtime status for logs.
- Presentation and Status Functions on page 447 describes how to use the Presentation and Status tabs on the log configuration aspect for formatting history presentation on Operator Workplace displays, and for viewing log data directly from the log configuration aspect.
- History Control Aspect on page 453 describes how to read status and network information for the History Service Provider for a selected node in the Node Administration structure.

- Database Maintenance Functions on page 455 describes various maintenance operations including extending tablespace, maintaining file-based storage, staggering data collection.

Accessing History Applications

To access maintenance functions related to History from the Windows task bar, choose **Start>Programs>ABB Industrial IT 800xA>Information Mgmt> History>***application*.

Integrating History into the 800xA System Platform

The following links between History and the 800xA system are typically auto-configured as part of the post installation process. Linkage requirements are covered here for reference, and to verify, if necessary:

- Linking Information Management History with the 800xA System Trend Function on page 493.
- Integrating System Message and History Servers on page 495.

The interface between the History server and the 800xA system message service is pre-configured and typically does not require adjustments. The following may need to be adjusted:

- The maximum number of messages which may be stored, and the threshold for when messages are forwarded from the history client to the server.
- How far back in time the Event Collector should get historical events.

Backup and Restore

The Backup and Restore utility provides a graphical user interface for running the command line-based hsBAR function. This utility is used to:

- **Create backup files for your History database**. This procedure creates all the files that are required to completely recreate the Information Management History database. This includes all configuration data, log data from both file-based and Oracle-based logs, and the Aspect System definition file. This does

not include archive data which may be backed up by using the Backup Destination feature as described in Configuring Archive Backup on page 338.

Trend log configurations are backed up by this utility ONLY when they exist in a property log structure in combination with a History log (as defined by the log template). Property logs made up entirely of trend logs are NOT backed up. These log configurations must be backed up via the 800xA system backup.

- **Restore a history database from backup files**. This procedure recreates the Information Management History database using the backup files described above. If the backup was created from an earlier History database version, the restore utility will convert the restored database to the current version.
- **Synchronize the Aspect Directory contents with the current Information Management History database configuration**. This procedure ensures that the Information Management History database is at the current version, and that the Aspect Directory matches the current database configuration.

Considerations

When backing up or restoring the History database, make sure the disk is ready and available on the machine on which the procedure is to occur. Also, these preparatory steps are required before performing a restore operation:

- All services under PAS supervision must be stopped.
- The Inform IT History Service Provider must be stopped.
- There must be NO applications accessing the Oracle database.

The log file should be checked after each backup and restore operation to make sure that each backup or restore operation completed successfully.

Backup and Restore Utility

To start this utility, choose: **Start>Programs>ABB Industrial IT 800xA>Information Mgmt> History>Backup and Restore**. For instructions on using this utility, refer to the applicable section:

- Backing Up the History Database on page 410.
- Restoring the History Database from Backup Files on page 417.
- Synchronizing the Aspect Directory with the History Database on page 424.

Backing Up the History Database

This procedure creates all the files required to completely recreate the Information Management History database. This includes all configuration data, log data from both file-based and Oracle-based logs, and the Aspect System definition file.

Trend log configurations will only be backed up by this utility when they exist in a property log structure in combination with a history-based log.

As an option, choose to backup just the Aspect System Definition file. This provides a file that can be used to recreate the History database configuration in the Aspect Directory.

How Backup Works

During a backup operation, all data in the Oracle database owned by the Oracle History user is exported to the specified destination and compressed into a zipped archive, along with any files that have been created to store file-based property log entries (called flat files).

The History database can be backed up to any drive, including any mapped network drives. The disk type should be NTFS for the backups.

To avoid any ambiguity, the backup operation produces a zipped archive of compressed History database files for each drive that contains at least some portion of the database, where each archive contains only the database files that are stored on the corresponding drive. The backup utility uses the naming convention `name-drive.zip` for the zipped archives that it produces. For example, if the History database is located entirely on the C:\ drive and you wish to back up the database to a zipped archive called hist, the backup operation will compress the database files into a zipped archive named `histDB-C.zip`.

If the data files exceed two gigabytes, or if there are more than 25,000 files, then multiple zip files will be created using the following naming convention:

- First File *name-drive*.zip
- Next File *name-drive0001*.zip
- Next File *name-drive0002*.zip

When backing up the History database, make sure the disk is ready and available on the node on which the procedure is to occur. The log file should be checked after the backup operation to make sure that the backup operation completed successfully.

Make sure the system drive is not getting full. Temp space is required to make the backup. If the log file indicates that the Oracle export failed, use the option to export to a disk with more space.

The following procedures can be used to backup all information related to Information Management software. Information Management software includes:

- Inform IT History (includes Archive).
- Inform IT Display Services.
- Inform IT Application Scheduler.
- Inform IT Data Direct.
- Inform IT Desktop Trends.
- Inform IT Open Data Access.
- Inform IT Calculations.

Not all components require additional steps to backup local files. An Information Management server will have all the software installed while other nodes will have only some of the software used. For each component that is configured and used on a node, all the files should be backed up occasionally. The following procedure provides the steps to perform the backups.

Node Types

The IM software run on three basic node types.

- Information Management Server.
- Information Management Client, with optional Desktop Tools components.
- Desktop Tools installation.

For a given node type, the backup steps depends on the type of software installed and the software used on the node. If software is installed and not utilized, there will be nothing to backup. The following steps identify how to backup an Information Management 800xA Server node. The steps to backup the other node types will reference the server backup procedure.

ABB Process Administration Service (PAS)

Perform the following procedure to run the PAS utility.

1. Run the Process Administration Service (PAS) utility on the Information Management Application Server node. From the Windows Taskbar select:

   Start > Settings > Control Panel > Administrative Tools > PAS > Process Administration

   This opens the Process Administration Service dialog box.
2. Click **Stop All** to stop all processes under PAS supervision.
3. Click **Close** when the dialog box indicates that all processes are stopped.
4. Use standard Windows procedures, via the Services selection from Administrative Tools in Windows Control Panel, to place the ABB Process Administration Service into manual and insure that it is stopped.

Cleaning the History Database

It is recommended that the history database be cleaned before making the backup.

1. Open a Windows Command Prompt and enter `hsDBMaint -checkDB`.
2. If any problems are found, enter **hsDBMaint -clean** to fix them.

Information Management History Backup and Restore Utility

Use the Information Management History Backup/Restore utility to create all the backup files that are required to completely back up the Information Management History database. This includes all configuration data, log data from both file-based and ORACLE-based logs, and the Aspect System definition file.

The IM Server must be connected to an active aspect system at this time. The following steps require acesss to the aspect directory.

1. Select:

   Start > All Programs > ABB Industrial IT 800xA > Information Mgmt > History > Backup and Restore
2. Verify the **Create Backup Files of Current Configuration** option is enabled in the IM Historian Backup/Restore Utility window.

3. Click **Next**. A window for setting up the backup operation is displayed (Figure 299).
4. Specify the location where the backup files are to be created in the New Directory Path for the Backup field. This path must already exist and the directory must be empty. If necessary, click **Browse** to create a new directory (Figure 299). Add a D:\HSDATA\History as an additional option. Figure 299 shows the Browser dialog being used to browse the existing path **C:\Backup**, and then create a folder named **T3** in the Backup directory.

The backup of the History data must be in a directory of its own, not the D:\HSDATA\History directory. If the data is put into the D:\HSDATA\History directory, it will get lost.

a. This option enables an alternate location to be specified for the temporary and oracle DB export files used during history database backup.

   To specify a different folder for the export other than the system drive, enter the following in the additional options dialog:

   `d:\export`

   Where `d:\export` is a folder created on a disk with enough space for the export.

5. Verify the **Only Generate Aspect Definition File** option is disabled.
6. Click **Next**. The HsBAR Output Window is displayed.
7. Select the **Automatically Close Upon Completion** option.

Refer to Specifying Additional hsBAR Options on page 431 to specify additional options for the hsBAR command.

After the HsBAR Output Window closes, monitor the progress in the Progress Status area of the IM Historian Backup/Restore Utility window and click **Finish** when the backup is complete. The Browser dialog is used

If a message appears stating that there are inconsistencies between the log configurations in the Aspect System and the log configurations in Oracle, it may be because the database was not cleaned before running the backup. Use the hsDBMaint -clean function to clean the database and then rerun the backup. If this does not fix the problem, contact ABB Technical Support for further assistance.

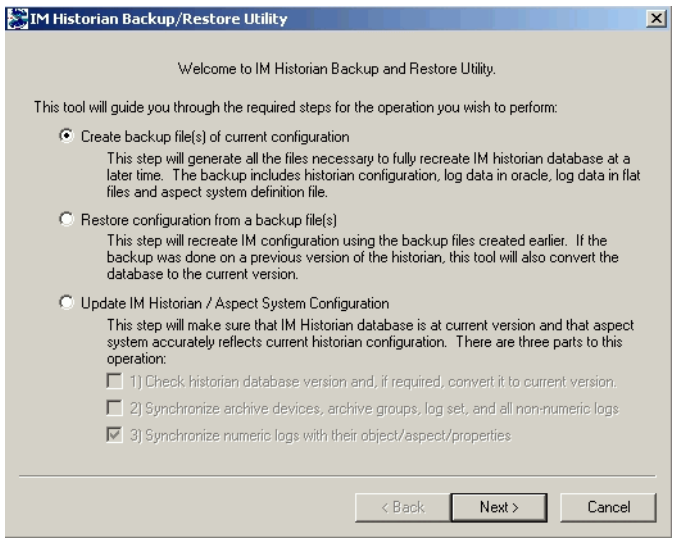


Figure 298. Backup/Restore Utility - Backup Option Selected

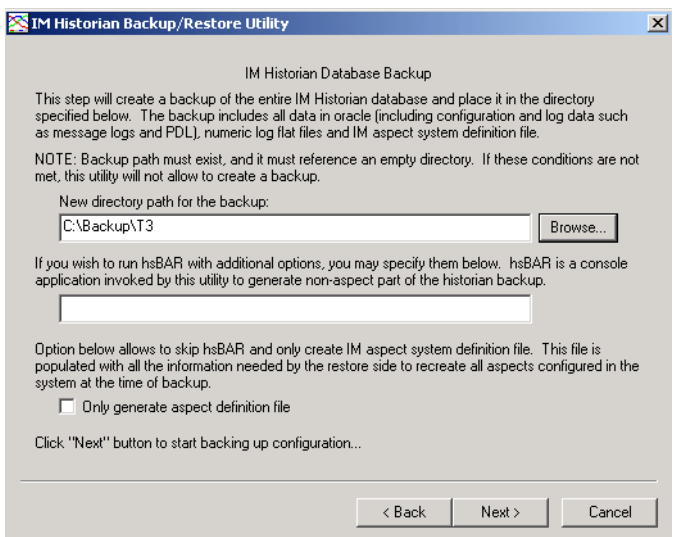


Figure 299. Setting Up the Backup Operation

8. The HsBAR Output window may be opened over and hide the progress status window. There is a check box for specifying that the HsBAR window be closed automatically when finished, Figure 300. This is recommended. When the window is not closed automatically, wait for the **Close** button to be activated on the bottom of the output window, then click the **Close** button.

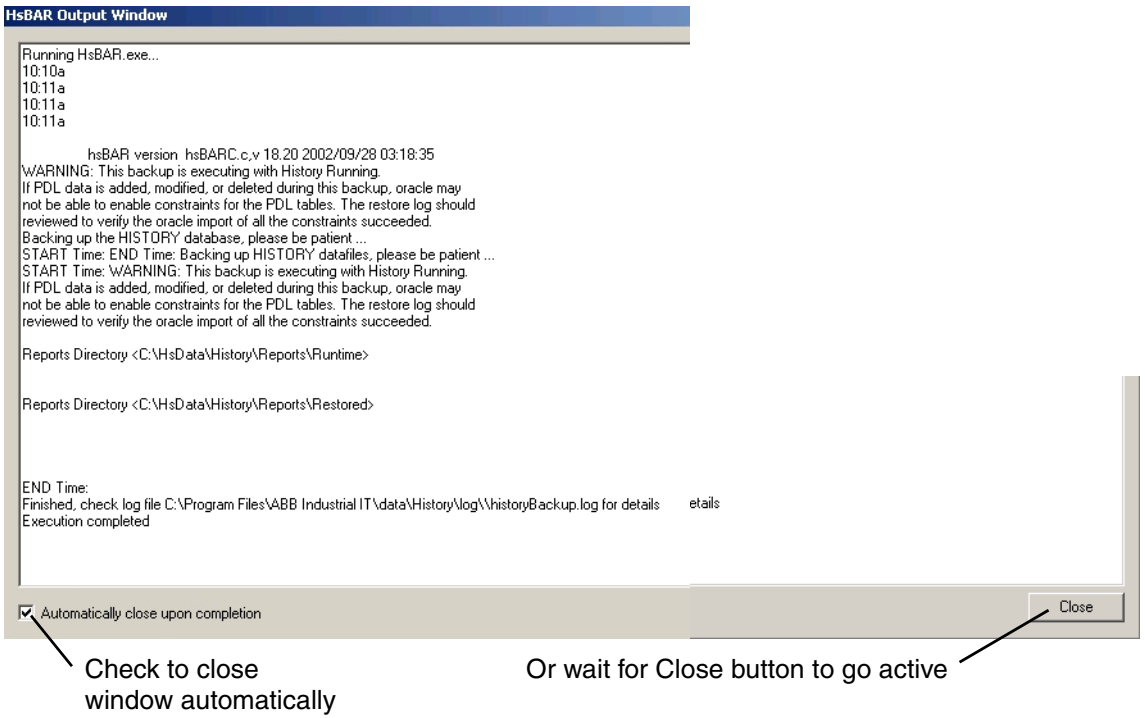


Figure 300. Closing HsBAR Output Window

9. Start all PAS services.

Saving Other Information Management Related Files

There are several other files related to Information Management that need saved as part of total system backup.

More detailed instructions for this procedure can be found in the section on backing up individual Information Manager applications in the appropriate *Information Management* instruction.

- **History Archive Data:** For each archive device, go to the location specified by the Device Filename and copy the folders under that directory to a safe location. Do this even if automatic backup is configured. If automatic backups are for local disks, also transfer them to a safe media at this time. Archive state information must be saved. The Archive folder from the following path must be saved to a safe location:

  ```
  C:\ProgramData\ABB\IM\
  ```

- **Reports:** Save any report template files created in Microsoft Excel, DataDirect, and/or Crystal Reports. Also save report output files created as a result of running these reports via the Scheduling Services.
- **Desktop Trends:** Back up trend display, ticker display, and tag explorer files.
- **Display Services:** Back up the directories for custom users, as well as display and user element definitions.
- **DataDirect:** Back up custom text files for object, object type, and attribute menus used on the DataDirect windows.

All files should be copied to a remote storage medium after the saving is completed.

Use PAS to start all Information Management processes and the Inform IT History service provider for this node.

The procedure can be performed by using the pasgui command in the Run dialog box (**Start > Run**).

a. Use the Windows Taskbar to select:

   Start > Programs > Administrative Tools > PAS > Process Administration

b. Click **Start All** in the PAS dialog box. This will start the Inform IT History Service Provider.

Non-Information Management Server nodes.

Any nodes that are installed and are configured to use Desktop Trends, Data Direct, Application Scheduler (reporting) should backup any application specific files after backup process is completed on the node. The backup files include:

- **Reports:** Save any report template files created in Microsoft Excel, DataDirect, and/or Crystal Reports. Also save report output files created as a result of running these reports via the Scheduling Services.
- **Desktop Trends:** Back up trend display, ticker display, and tag explorer files.
- **DataDirect:** Back up custom text files for object, object type, and attribute menus used on the DataDirect windows.

Restoring the History Database from Backup Files

This recreates the Information Management History database from backup files created with this utility (Backing Up the History Database on page 410). If the backup was created from an earlier History database version, the restore will convert the restored database to the current version. For an overview, refer to How Restore Works on page 417. To perform a restore, refer to Restoring a History Database on page 418.

How Restore Works

During the restore, the existing database is dropped, and a new one is created. Mount points and additional table spaces are created based on the database being restored. Oracle data is imported, and the file-based property logs are copied back into the system. Unless a different mount point is specified, the History database will be restored to its original location (its location prior to being backed up).

The History database can be restored from any drive, including any mapped network drives. The restore utility will first search a specified location for zipped archives matching a specific name and fitting the form *name-drive***.zip** (such as `histDB-C.zip`, `histDB-A.zip`, and `histDB-D.zip`), and will then restore the compressed database files contained within the archives to their respective original locations (their locations prior to being backed up).

For any object in the original database that had more than one Log Configuration aspect, the contents of those aspects will be merged into one Log Configuration aspect per object.

Be sure to follow the preparatory steps (1-4) in Restoring a History Database on page 418 before beginning the actual restore operation.

Restoring a History Database

Complete these preparatory steps before beginning the restore operation:

If Log Configuration aspects are imported via the 800xA System import/export utility prior to this restore, those Log Configuration aspects and corresponding log templates must be deleted.

1. Stop all PAS processes. Refer to Starting and Stopping History on page 437.
2. Make sure no third-party applications are accessing the Oracle database.
3. Stop the Inform IT History Service Provider. To do this (reference Figure 301):
   a. Go to the Service structure in the Plant Explorer and select the **Inform IT History Service Provider**.
   b. Select the **Service Provider Definition** aspect.
   c. Click the **Configuration** tab.
   d. Uncheck the **Enabled** check box then click **Apply**.

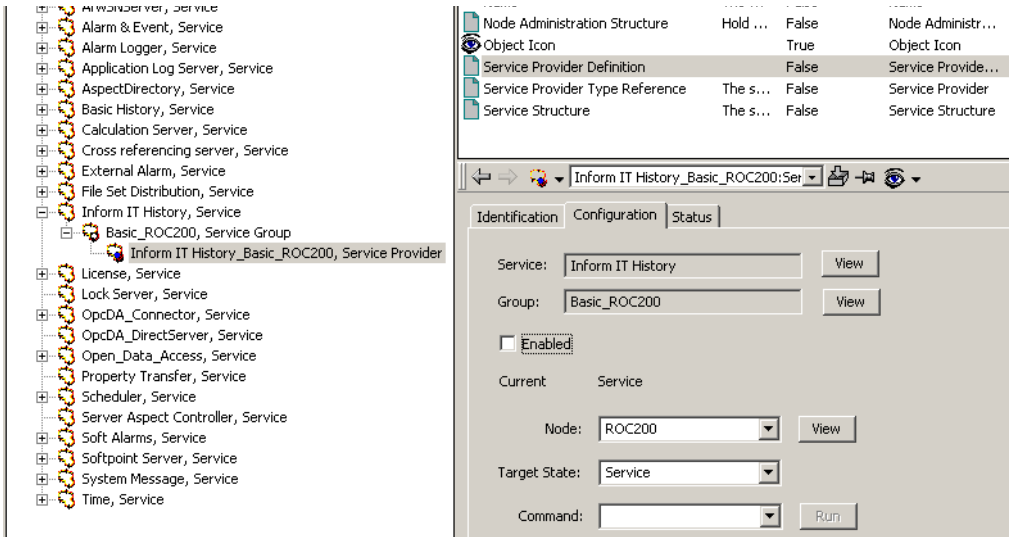


Figure 301. Stopping the Inform IT History Service Provider

When finished with the aforementioned preparatory steps, run the restore utility as described below:

1. Start the Backup/Restore utility. When the Backup/restore utility is started, the Create Backup option is selected by default.
2. Click the **Restore configuration from backup files** option, Figure 302.

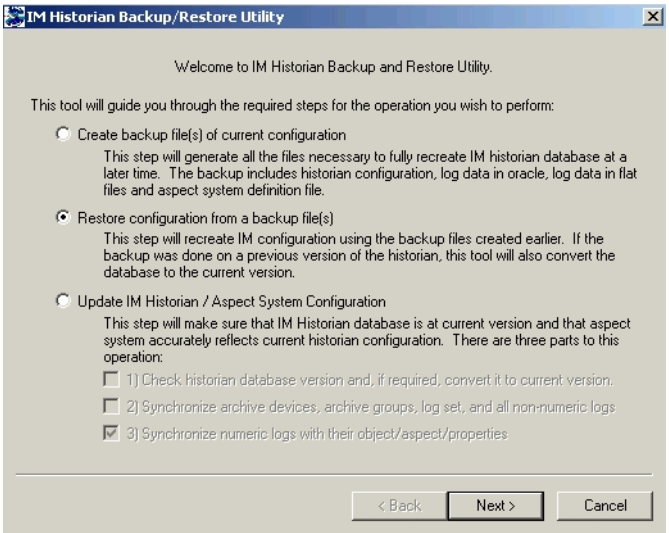


Figure 302. Starting a Restore Operation

3. Click **Next**. This displays a dialog for setting up the restore operation, Figure 303.

4. Specify the location of the backup files from which the History database will be recreated. Enter the path directly in the *Path of IM historian backup* field, or use the corresponding **Browse** button. Figure 299 shows the Browser dialog being used to browse to **C:\Backup\T3**.

   – As an option specify a new mount point path for file-based logs. Use the corresponding **Browse** button and refer to Mount Point on page 434 for further guidelines.

   – Also, specify that a different Oracle tablespace definition file be used. This may be required when restoring the files to a machine that has a different data drive layout than the original backup machine. Use the corresponding **Browse** button and refer to Moving hsBAR Files to a Different Data Drive Configuration on page 435 for further guidelines.

   – Also, specify additional hsBAR options. These options are described in Specifying Additional hsBAR Options on page 431.

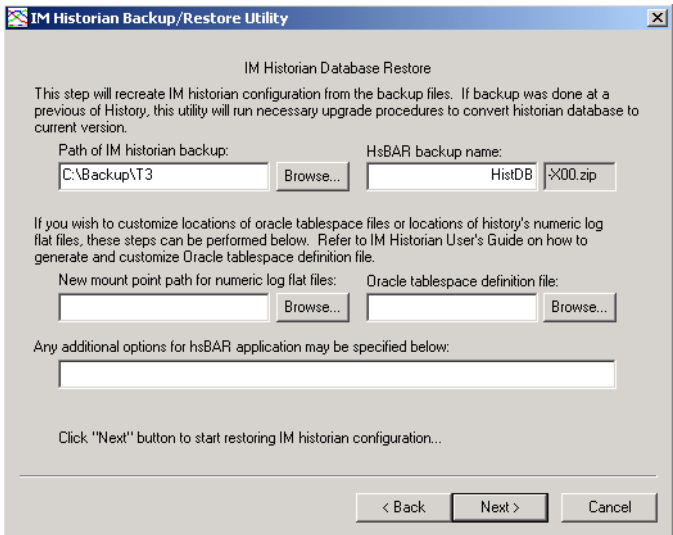


Figure 303. Setting Up the Restore Operation

5. Click **Next** when finished with this dialog. This opens the progress status window and the HsBAR Output Window.

If the restore operation fails with Oracle Error Message 1652 - *Unable to extend tmp segment in tablespace tmp* - it may be due to a large OPC message log which exceeds the tmp tablespace capacity during the restore operation.

Use the Database Instance Maintenance wizard to increase the tmp tablespace. The default size is 300 megabytes. Increase the tablespace in 300-megabyte increments and retry the restore operation until it runs successfully. Refer to Maintaining the Oracle Instance on page 137.

6. The HsBAR Output window may be opened over and hide the progress status window. There is a check box for specifying that the HsBAR window be closed automatically when finished, Figure 304. This is recommended. If the window is not closed automatically, then wait for the **Continue** button to appear on the bottom of the output window, then click the **Continue** button when ready.

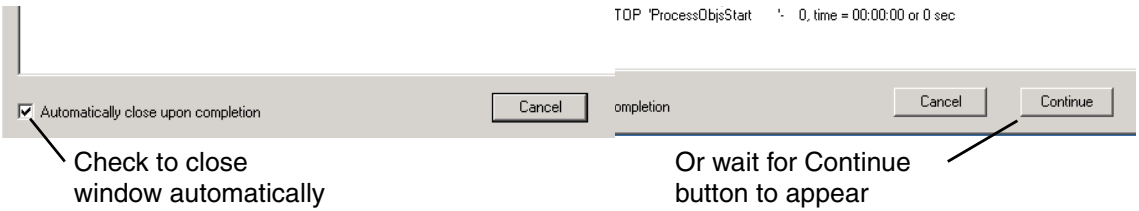


Figure 304. HsBAR Output Window

7. After the HsBAR window is closed, monitor the Progress in the Progress Status window, Figure 305. Ignore the error messages that indicate *error deleting aspect*.

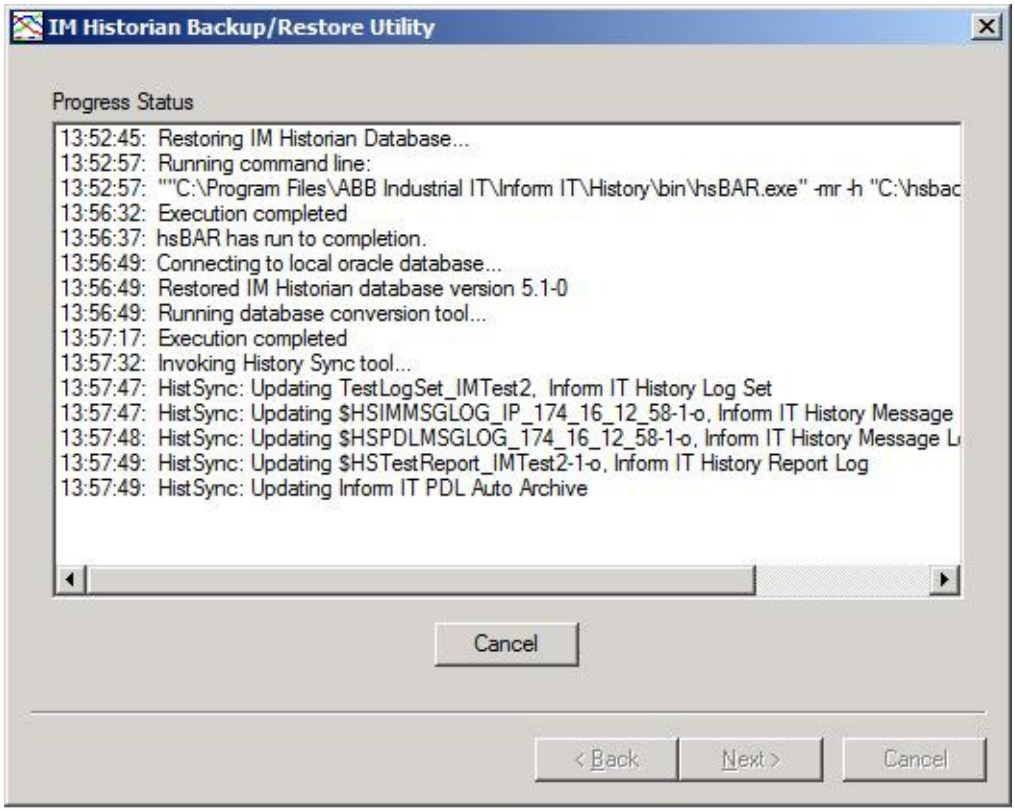


Figure 305. Progress Status Window

8. Click **Finish** when the `Click to Exit the Application` message is displayed.
9. Start all processes under PAS supervision. For further details, refer to Starting and Stopping History on page 437.
10. Start the Inform IT History Service Provider. To do this (reference Figure 301):
    a. Go to the Service structure in the Plant Explorer and select the **Inform IT History Service Provider**.
    b. Select the **Service Provider Definition** aspect.

c. Click the **Configuration** tab.

d. Check the **Enabled** check box then click **Apply**.

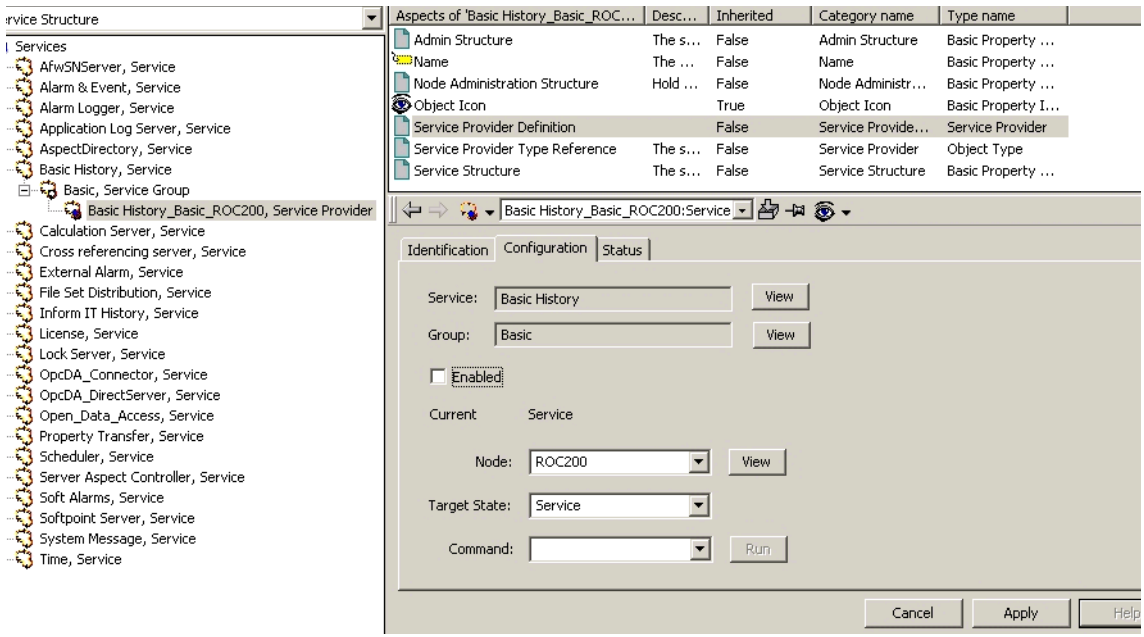


Figure 306. Restarting Basic History

This completes all steps required to restore a history database.

Synchronizing the Aspect Directory with the History Database

This procedure makes sure that the Information Management History database is at the current version, and that the Aspect System matches the current history database configuration.

To use this utility, click Update IM Historian/Aspect System Configuration, and then check one or more of the applicable options, Figure 307.

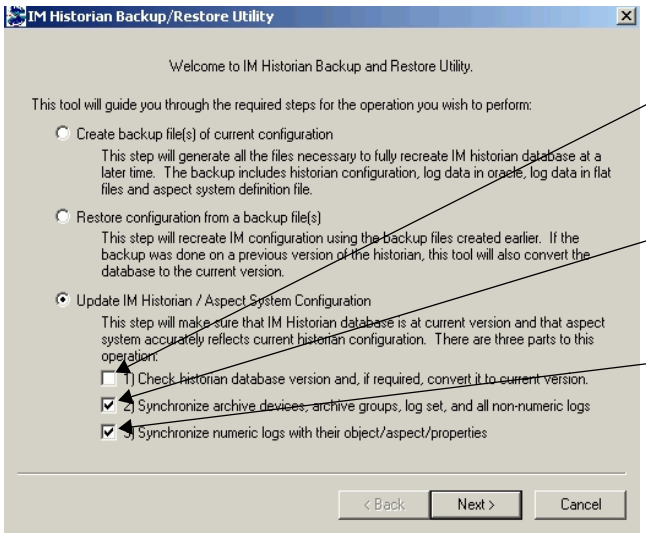


Figure 307. Synchronizing the Aspect Directory with the History Database

An example Synchronization operation is demonstrated in the following steps:

1. Click **Next** to start the Synchronizer which makes sure that the Information Management History database is at the current version, and that the Aspect System matches the current history database configuration.

   The synchronizer automatically accesses the applicable aspect system definition file which was created during the backup, Figure 308.

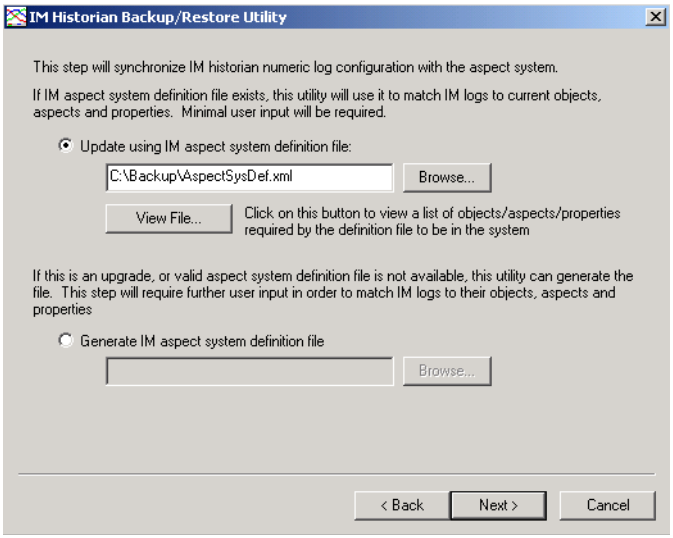


Figure 308. Synchronizer - Step 1

2. Click **Next** to continue. This displays a dialog for mapping controller object locations, Figure 309. It is NOT necessary to make any entries in this dialog unless the locations have changed since the backup was made.
3. Enter the mapping specifications if necessary (typically NOT REQUIRED), then click **Next** to continue. This displays the Progress Status, Figure 310.

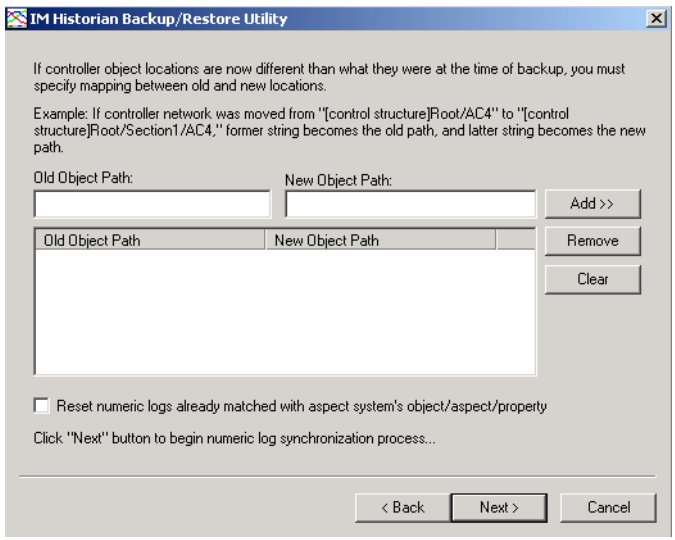


Figure 309. Location mapping Dialog

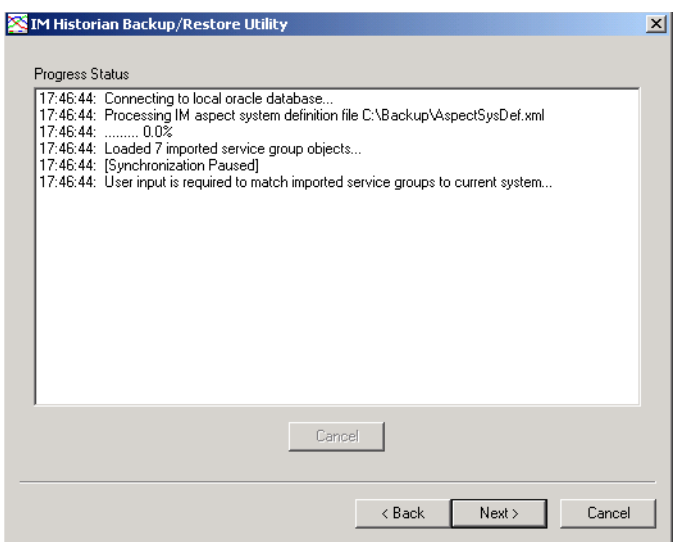


Figure 310. Synchronization Progress Status

4. Click **Next** when ready.
5. Use the dialog in Figure 311 to make sure the imported Service Groups are properly mapped to the Service Groups that currently exist in the Aspect Directory. On rare occasions there may be a problem if Service Group configurations in the Aspect Directory have changed.
   a. To check, click on a group in the Imported Service Groups list. If the Service Providers in the Imported and Current Service Providers lists below match, then the Service Groups are properly mapped.

      If the imported Service Groups do not match the current Service Groups, or if adjustments need to be made for some other reason, refer to Adjusting Service Group Mapping on page 429.
   b. When satisfied that the imported and current Service Groups are properly matched, click **Next** to continue. This displays another Progress Status, Figure 312.

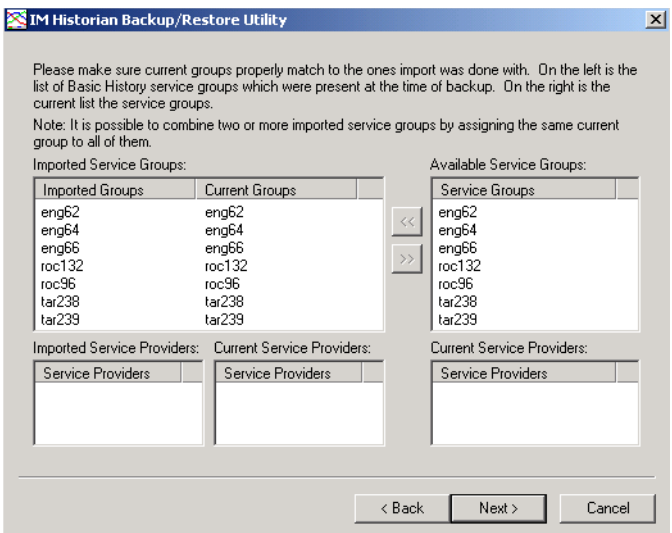


Figure 311. Checking Service Groups

6. Click **Next** when ready to leave Progress Status.

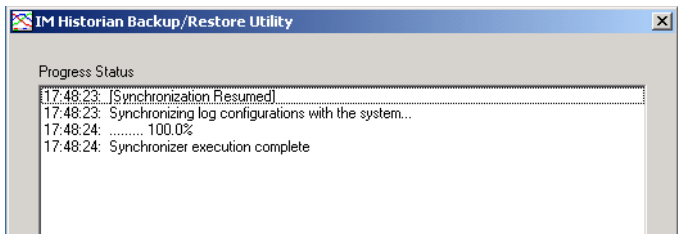


Figure 312. Progress Status

7. When the execution complete message is displayed, Figure 313, click **Finish**.

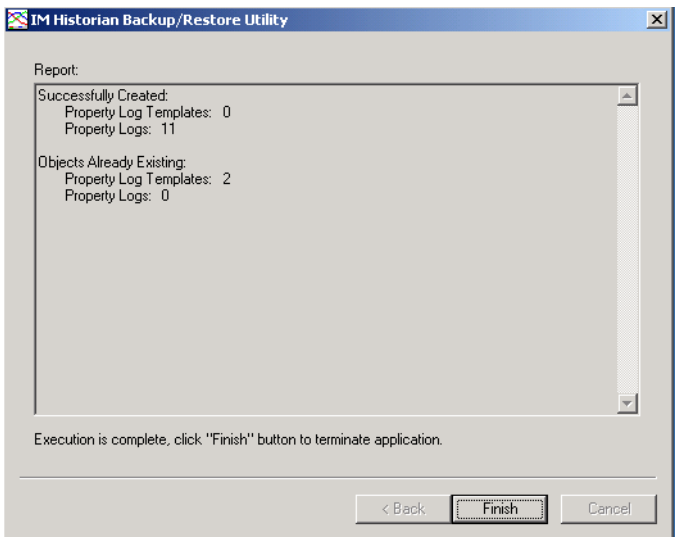


Figure 313. Finish Synchronization

Adjusting Service Group Mapping

During the restore or synchronization of a History database configuration, the Service Group definitions in the configuration being restored or synchronized are matched to their corresponding Service Groups in the Aspect Directory. The Service Group mapping dialog, Figure 314, is displayed to verify that all Service Groups are properly matched, and correct any mapping error that may occur. The dialog may

also be used to make adjustments. For instance, to merge two restored Service Group configurations into one Available Service Group.

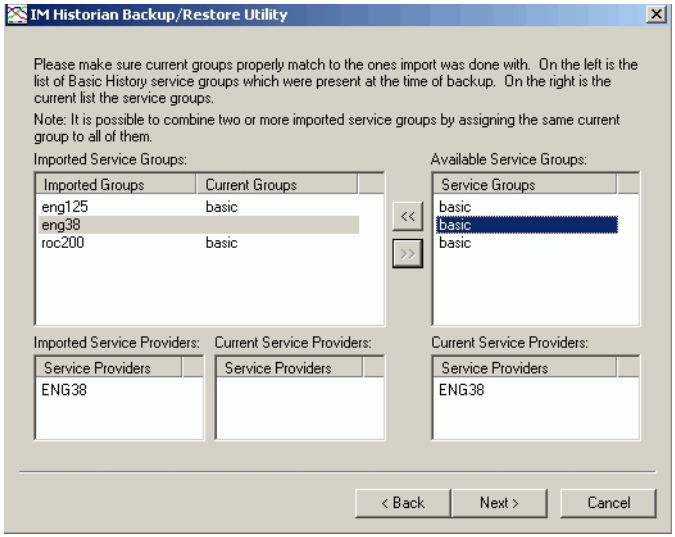


Figure 314. Adjusting Service Group Mapping

In this dialog, Service Group definitions being restored or synchronized are called *Imported* Service Groups. The actual Service Groups which are available in the Aspect Directory are called *Current* or *Available* Service Groups. The backup/restore utility automatically maps Imported Service Groups to an Available Service Group by matching the host name for each Imported Service Group with the host name for an Available Service Group. (The host name is specified in the Service Group's Service Provider configuration.)

If Service Group configurations in the Aspect Directory have changed since the creation of the History database configuration now being restored or synchronized, some Imported Service Groups may not be mapped to any Available Service Group. This condition is illustrated in Figure 314 where the Current Service Group column for Imported Service Group eng38 is blank. If this occurs, use the dialog manually to map the Imported Service Group to an Available Service Group.

To do this:

1. From the Imported Service Groups list (top left pane), select the Imported Service Group whose mapping is to be changed.

   When this is done, the host names for the imported and currently mapped Service Providers for the selected Imported Service Group are indicated in the respective Service Provider lists (bottom right panes).

2. From the Available Service Groups list (top right pane), select the current Service Group to be mapped to the selected imported Service Group. When this is done, the host name for the Service Provider for the selected Available Service Group is indicated in the Current Service Providers list (bottom left pane). This helps when matching the host name for the Imported Service Group with the host name for the Available Service Group.

3. When satisfied that the appropriate Available Service Group was selected, click the left (**<<**) arrow button. This puts the name of the selected Available Service Group in the Current Group column for the Imported Service Group.

4. Check the host names for the Imported and Current mapped Service Providers in the respective Service Provider lists (bottom right panes) to verify that the Imported and Available Service Groups were mapped as intended.

5. Repeat this procedure to map additional Imported Service Groups to the same Available Service Group if needed.

To change the Current Service Group for an Imported Service Group, first unmap the Current Service Group. To do this, select the Imported Service Group, then click the right (**>>**) arrow button. This removes the name of the currently mapped Service Group. At this point the Imported Service Group is not mapped to any Available Service Group. Repeat the above procedure starting at step 2 to do this.

Specifying Additional hsBAR Options

By default the backup operation runs hsBAR with the -m b option (b specifies backup). Other options may be specified. A list of available options is provided below:

`-h` < *storage location on hard drive* > specifies the location to which the History database is to be backed up to or restored from. Refer to Storage on page 433.

`-c` < *compression level* > is used to specify the compression ratio to use for backing up the History database. Refer to Compression Level / Ratio on page 433.

`-f` < *log file name* > is used to specify the file to use for storing program output during a backup or restore procedure. Refer to Log File on page 434.

`-o` indicates flat files are to be excluded when backing up the History database on the hard drive. Refer to Excluding Flat Files on Backup on page 437.

`-a` is used to specify an alternate location for the temporary and oracle DB export files used during history database backup. Default directory for these files is C:\HsData\History\

`-k` This option can be used when restoring oracle-only database backup to skip the creation of empty datafiles. If backup was performed with the datafiles, -k is ignored

`-b` instructs hsBAR to extract oracle data info file and put it in the specified directory. Therefore this option can only be used on existing database backup files and -mr has to be specified.

`-e` Used on restore only. This option tells hsBAR to ignore the Oracle data info file found in backup and use the one specified instead. This includes table information stored in stats. file (as in prior 2.5 versions of History). The stats information from the specified Oracle data info file will be used instead.

Running hsBAR

As an option, use hsBAR from a command line. This backup procedure stores the files that hold file-based logs, and Oracle tables that hold configuration data and any logs that are Oracle-based (message logs and asynchronous property logs).The hsBAR executable resides in %ABB_ROOT%History\bin and is run from the command line. (The default path for %ABB_ROOT% is c:\Program Files\ABB Industrial IT\).

Run hsBAR using the following command:

"%HS_HOME%\bin\hsBAR" <*options*>

The syntax for hsBAR is as follows:

`"%HS_HOME%\bin\hsBAR" -m b|r -h source/destination`
`[ -c` *compression level* `] [ -f` *log file name* `]`
`[ -p [ -n` *new mount point directory* `] ] [ -o ]`

Due to a limitation in the ORACLE import facility, paths specified for hsBAR cannot contain spaces in the directory or file names. This includes the environment variable, %HS_DATA%

hsBAR Explanations and Examples

Operation Mode

In order for the hsBAR program to execute, the operation mode must be specified by entering the `-m` option, followed by either:

`-b` (back up the current History database) or
`-r` (restore a previously backed up History database)

Storage

In addition to specifying the mode (through the `-m` option), in order for the hsBAR program to execute, a storage location must also be specified. On an hsBAR backup operation, the `-h` option must be used to specify the destination to which the History database is to be backed up. On an hsBAR restore operation, the `-h` option must be used to specify the source from which the History database is to be restored. The drive, directory path, and file name must all be specified to use as the backup destination or restore source location, as no default is provided:

For backup "%HS_HOME%\bin\hsBAR" -m b -h C:\Temp\history

For Restore "%HS_HOME%\bin\hsBAR" -m r -h C:\Temp\history

In the above example, the backup operation will produce in the `C:\Temp` directory one or more zipped archives (depending on the number of directories containing History database-related files) of the form history-drive.zip (such as history-C.zip or history-D.zip, depending on the drive containing the database files). The restore operation in the above example will search in the `C:\Temp` directory for zipped archives matching the format history-drive.zip, and will then attempt to restore the files contained within the found archives.

Compression Level / Ratio

The `-c` option is used to specify the intensity level with which to compress the History database files during a backup procedure. This option must be followed by

an integer value ranging from 1 (very low compression) to 9 (very high compression). Using a lower compression ratio results in a relatively quick backup of the History database, but more hard disk space is required for storage. If a higher compression ratio is used, less hard disk space is needed for storing the backed up database, but the backup operation runs slower. If this option is not provided, a default compression level of 6 will be used for the backup operation:

"%HS_HOME%\bin\hsBAR" -m b -h C:\Temp\history -c 8

Log File

All progress messages and errors related to hsBAR are printed to a log file. The default log file for a backup operation is `%HS_LOG%historyBackup.log`, while the default log file for a restore operation is `%HS_LOG%historyRestore.log`. The hsBAR program, however, can write messages to an alternate log file specified by entering the `-f` option, followed by the full directory path and log file name:

For backup "%HS_HOME%\bin\hsBAR" -m b -h C:\Temp\history -f C:\Temp\backup.log

For restore "%HS_HOME%\bin\hsBAR" -m r -h C:\Temp\history -f C:\Temp\restore.log

Mount Point

The mount point specifies the location to which the History database is to be restored on an hsBAR restore operation. The `-p` option is used to restore flat files (file-based logs) to the mount point itself, and not necessarily to a location that is relative to the root. If this option is omitted, all flat files will be restored relative to the root, and not relative to the mount point itself. The `-n` option may be used in conjunction with the `-p` option. This is used to specify a new mount point. If the `-p` option is used alone (without the `-n` option), the default mount point for the History database `C:\HsData\History` (%HS_DATA%) will be used. If the `-p` and `-n` options are used together, the old History database will be deleted from its original location and restored to the new specified mount point. This option is especially useful in that it can take a History database that was backed up on several different drives and restore it all to a centralized location on one drive:

"%HS_HOME%\bin\hsBAR" -m r -h C:\Temp\history -p -n C:\Temp

Moving hsBAR Files to a Different Data Drive Configuration

There are two cases to consider when moving haBAR Files to a data drive with a different configuration:

Backup was Made on Information Management 4.n or Earlier. The -b and -e options are used to move hsBAR files to a machine that has a different data drive layout than the machine where the hsBAR backup was created. For example, if the source machine has Oracle data files on drives C, D, E and F, and the files are being restored to a machine with drives C, E, F and G, the restore will fail when hsBAR attempts to restore files to the D drive.

In order for this restore to succeed, the files that were originally located on the D drive must be restored to the G drive on the new machine. To do this:

1. Run hsBAR with the following options to extract the datafiles_config information:

   hsBAR -mr -h c:\myDB -b C:\\

   where myDB is the file name that was entered when the backup was made and -b is the location to extract the datafiles_config file to.

2. Edit the datafiles_config file. An example is shown in Figure 315.

Edit Drive Specification
In this example, change three instance of **D** to **G**

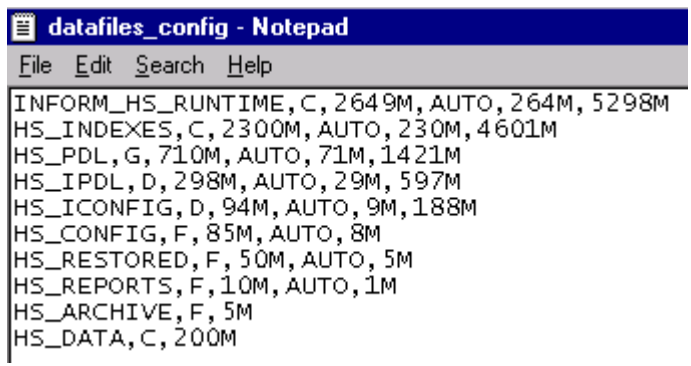


Figure 315. Editing the datafiles_config File

3. Run hsBAR with the following options to use the modified datafiles_config file:

hsBAR -mr -h c:\myDB-C.zip -e C:\datafiles_config

After editing the datafiles_config file, a file extension is added (for example datafiles_config.txt), that extension must be included in the specification for the -e option in hsBAR.

Backup was Made on Information Management 5.0 or Newer. The -b and -e options are used to move hsBAR files to a machine that has a different data drive layout than the machine where the hsBAR backup was created. For example, if the source machine has Oracle data files on drives C and E, and the files are being restored to a machine with drives C and D, the restore will fail when hsBAR attempts to restore files to the D drive.

In order for this restore to succeed, the files that were originally located on the E drive must be restored to the D drive on the new machine. To do this:

1. Run hsBAR with the following options to extract the instance_config information:

   hsBAR -mr -h c:\myDB -b C:\\

   where myDB is the file name that was entered when the backup was made and -b is the location to extract the instance_config.txt file to.

2. Edit the instance_config file. An example is shown in Figure 316

3. Run hsBAR with the following options to use the modified instance_config file:

   hsBAR -mr -h c:\myDB-C.zip -e C:\instance_config.

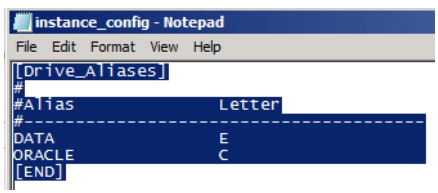


Figure 316. Edit Drive Specification

Excluding Flat Files on Backup

By default, the flat files associated with any property logs are stored during a backup of the History database. The -o option is used to exclude these flat files from the backup procedure. In cases where the History database contains many log files, this option may allow for a significantly faster backup operation, but only the Oracle database files are backed up:

"%HS_HOME%\bin\hsBAR" -m b -h C:\Temp\history -o

Viewing the Contents of the Zipped Archive

The hsUnzip utility can be used to produce a listing of the files that were compressed into a zipped archive during a backup of the History database:

"%HS_HOME%\bin\hsUnzip" -v C:\Temp\history-C.zip

Getting Help with hsBAR

When the hsBAR utility is run with missing or invalid arguments, a brief but informative message gets displayed. Though technically for error handing purposes, the message contains information explaining the correct usage of the hsBAR program, along with a brief description of the utility's options. The simplest way to view this message is by trying to run the hsBAR utility with no arguments:

"%HS_HOME%\bin\hsBAR"

Starting and Stopping History

Certain procedures require History to be stopped and then restarted. The recommended method for stopping and restarting History is to stop and start all ABB processes under Process Administration Services. To do this:

1. From the Windows task bar, choose **Start>Settings>Control Panel> Administrative Tools>PAS>Process Administration**. This displays the main PAS window, Figure 317.

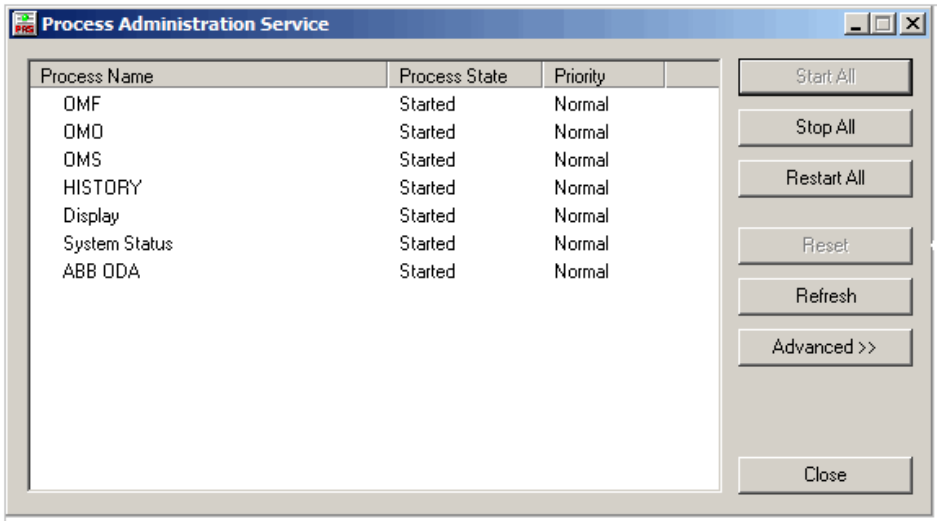


Figure 317. PAS Main Window

2. To stop all processes under PAS supervision, click **Stop All**. PAS is still active when all processes are stopped. Click **Start All** to start all processes. The **Restart All** button stops and then restarts all processes.
3. Click **Close** to exit PAS when finished.

Schedule History Backups

Use the Report Action of the Action Aspect to schedule a History backup (the scheduler always runs as an admin user).

- Creating multiple history backups in this manner will consume a significant amount of disk space. Ensure the disk where the backup files will be stored has sufficient storage capacity.
- Also, keep in mind that a new backup will be created each time the report action runs. Configure the scheduling definition aspect accordingly. Create a schedule with a reasonable interval between executions.
  - Use the **Report Template Path** to browse to and enter hsBackupApp.exe (typically C:\Program Files\ABB Industrial IT\Inform IT\History\bin).

- Select Output Options: **Report Parameters** and **Execute File**.
- Use /CreateBackup <*backup path*> as parameters to hsBackupApp by using the **Edit Parameter List** button. The hsBackupApp will accept command line arguments from the parameter list. Be sure to create a folder where the scheduled history backup files will be stored. For example, the result shown in the Edit Parameter List could be:
  path = /CreateBackup \d:\SchedHistBackups

Starting and Stopping Data Collection

Data collection is started and stopped by activating or deactivating logs. There are three basic methods:

- These two methods are applicable for history logs, but not for operator trend logs:
  - The Log List aspect is used to activate/deactivate logs on a list basis. Refer to Viewing Log Runtime Status and Configuration on page 442.
  - Log sets are used to activate and deactivate logs on a log set basis. Refer to Activate/Deactivate Logs in a Log Set on page 147.
- Individual logs (both history and trend) can be activated via the **Activate** and **Deactivate** menu items in the applicable configuration window. This is also applicable for message logs.
  - For property logs, refer to Activating/Deactivating a Property Log on page 268.

Activating/Deactivating a Property Log

Activate/deactivate all the logs in a property log as a unit, or activate/deactivate individual logs within the hierarchy. Or, activate/deactivate logs on a log set basis.

When a log is activated, it may be activated immediately, or activation may be delayed, depending upon how the log's Start Time attribute is defined. Start Time determines the earliest allowable time that the log can be activated. Start time also determines the hour, minute, and second when the first sample is collected.

To activate a property log as a unit:

1. Go to the structure where the logged object resides and click the Log Configuration aspect.
2. Use the **Enabled** check box to activate (checked) or deactivate (unchecked) all logs in the property log, Figure 318.

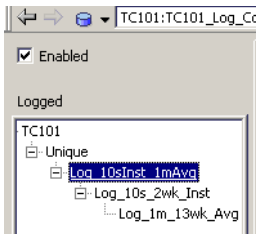


Figure 318. Activating/Deactivating a Property Log

To activate an individual log within a property log, select the log in the hierarchy, click the **IM Definition** tab, and then click **Activate** under the IM Historian Log State control, Figure 319.

To activate or deactivate property logs on a log set basis, refer to Activating Logs with Log Sets on page 441.

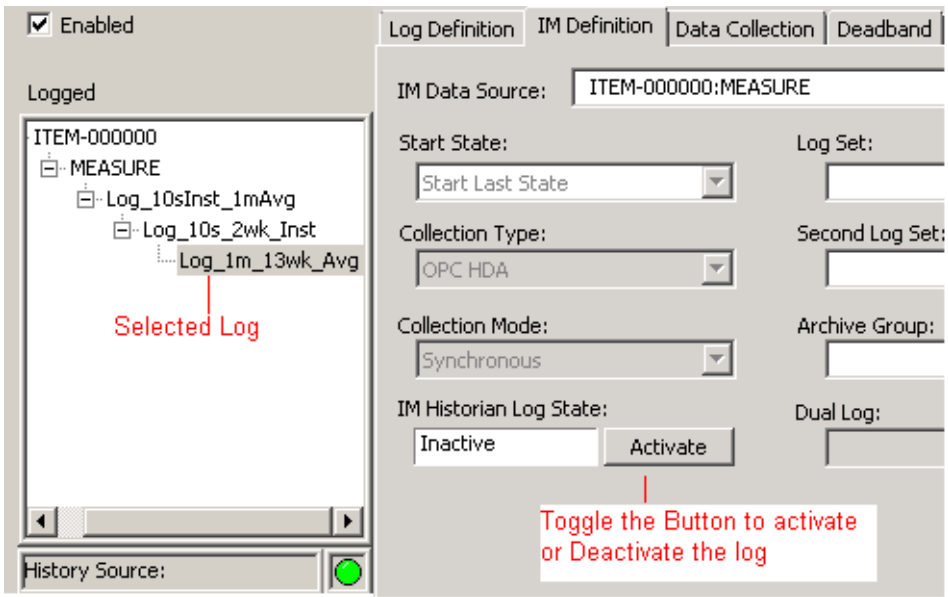


Figure 319. Activating an Individual Log in the Property Log Hierarchy

Activating Logs with Log Sets

This section describes how to activate and deactivate logs on a log set basis. Activating or deactivating a log set affects all of the logs belonging to that log set. Log sets are configured as described in Section 6, Configuring Log Sets.

This procedure is performed via the log set aspect for the log set whose logs is to be activated or deactivated. These aspects are generally located in the Node Administration structure (reference Figure 320):

1. In the Plant Explorer, select the Node Administration Structure.
2. Expand the object tree for the node where the log set is to be deleted.
3. In the object tree for the selected node, navigate to **InformIT History***_nodeName* **Service Provider > InformIT History Object>Log Sets**.
4. Select the Log Set aspect.
5. Click **Mode** and choose **Activate** or **Deactivate**.

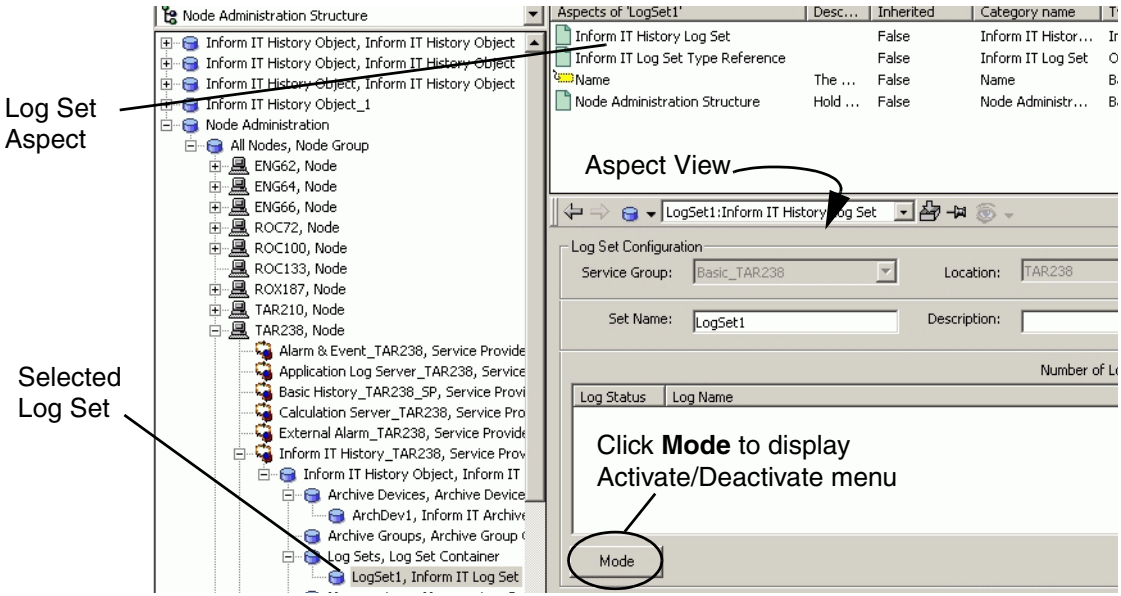


Figure 320. Accessing the Configuration View for the Log Set Aspect

Viewing Log Runtime Status and Configuration

The Log List aspect supports viewing of runtime status for property, message, and report logs. To access this view (reference Figure 321):

1. Select the Node Administration structure in the Plant Explorer.
2. Select the History service provider for the node where history is being configured (**InformIT History_Basic*Your*Service Provider)**.
3. From the History service provider, navigate to and select the **InformIT History Object**.
4. Select the **Inform IT History Log List** aspect.

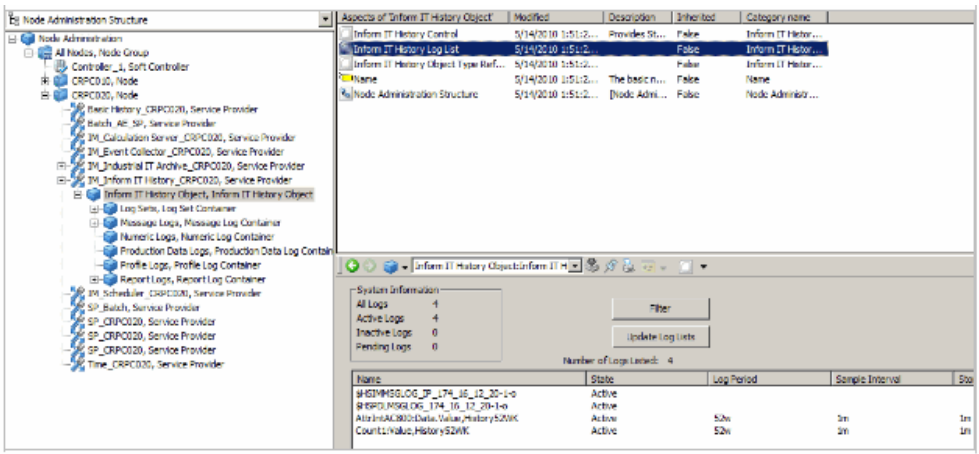


Figure 321. Accessing the Configuration View for the Log List Aspect

Click **Update Log Lists** to display the log list (Figure 322). This list provides both system-level and log-level information. The System Information section indicates the total number of logs configured for the applicable Service Group. It also indicates the number of logs whose state is Active, Inactive, or Pending.

The log-level information is displayed as columns in the log list. Select columns to show or hide. Also, specify a filter to show a specific class of the logs. The list can be filtered based on log state, log type, log name, archive group, and log set. These procedures are described in Filtering the Log List and Visible Columns on page 444.

The context menu is used to activate or deactivate logs in the list. This is described in Activating/Deactivating Logs on page 446.

Double-clicking on a specific Log Information row, the Log Configuration aspect containing the log will open as an overlap window.

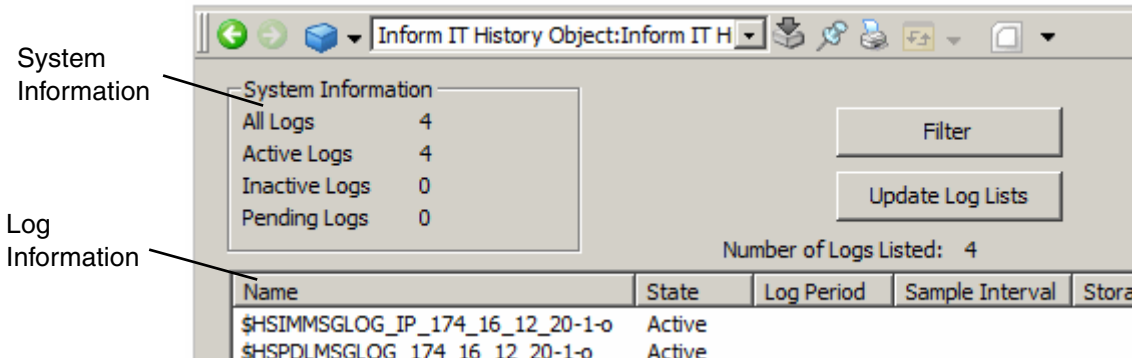


Figure 322. Log List Example

Filtering the Log List and Visible Columns

Click **Filter** to display the Filter dialog, Figure 323. Then use the Log Filter and Visible Columns sections to specify the filter. Click **Apply** to apply the filter, and then click **Update Log List**.

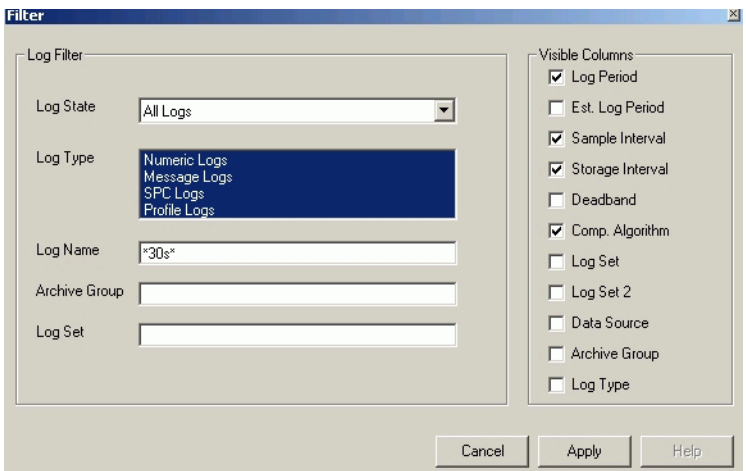


Figure 323. Filter Logs Dialog

Showing/Hiding Columns

To show/hide specific columns simply check to show or uncheck to hide the corresponding column.

Filtering the Log List

For further information regarding the log filtering criteria, refer to Table 38. An example is provided below.

Table 38. Log Filter Criteria

Field	Description
Log State	This is used to show all logs regardless of their state, or show just Active, Inactive, or Pending logs.
Log Type	This is used to select any combination of these log types: **Numeric** (property), **Message**, **Report. Profile** logs are applicable if Profiles is installed.
Log Name	Include logs whose name fits the specified text string. ***** is a wild card text string. By itself * indicates all logs.
Archive Group	Include logs assigned to an archive group whose name fits the text string specified this field. ***** is a wild card text string. By itself * indicates all archive groups.
Log Set	Include logs assigned to a log set whose name fits the text string specified this field. ***** is a wild card text string. By itself * indicates all log sets.

Filter Example

The following example show how to apply a filter. In this case the filter is based on log name. The filter will reduce the list to only those logs whose name includes the text string ***30s*** Figure 323. The filter result is shown in Figure 324.

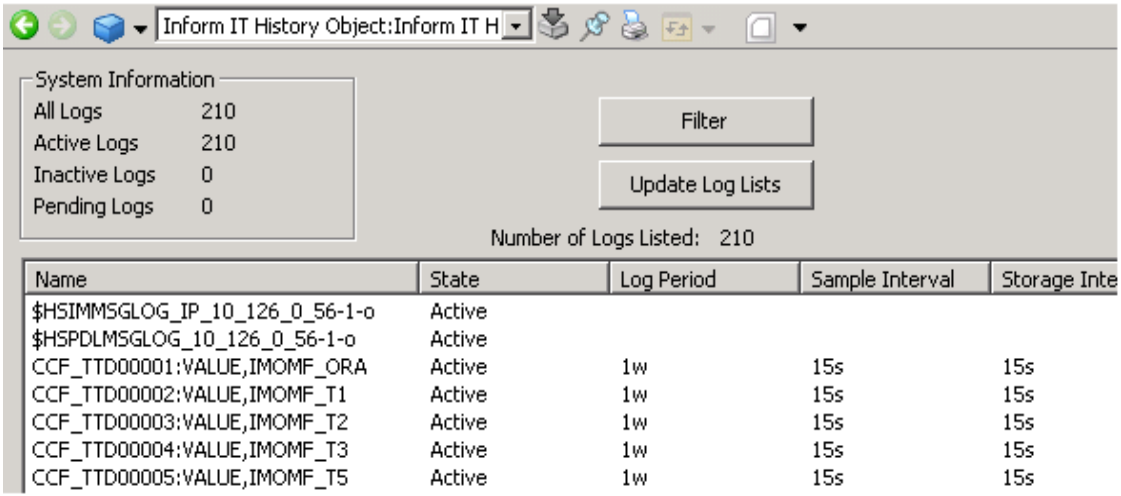


Figure 324. Example Filter

Activating/Deactivating Logs

Activate and deactivate individual logs, a subset of displayed logs, or the entire log list. To do this select the logs to be activated or deactivated, and then right-click and choose the applicable menu item from the context menu, Figure 325.

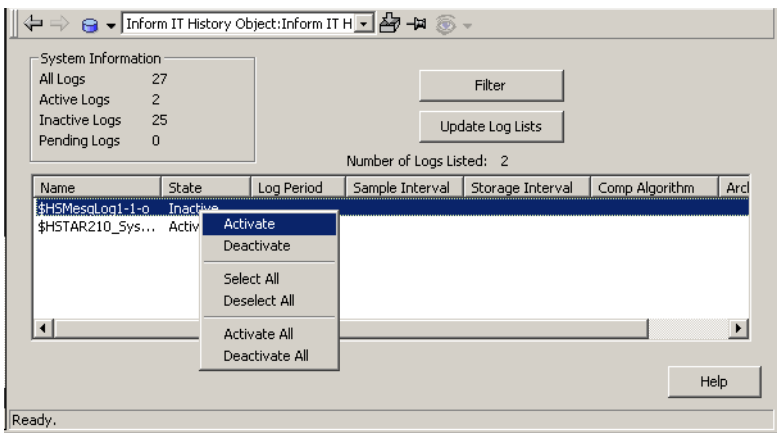


Figure 325. Activating/Deactivating Logs

After executing an **Activate All** or **Deactivate All** from the Inform IT History Log List, the State does not get updated. Use the **Update Log Lists** button to refresh the list.

Presentation and Status Functions

The log configuration aspect provides two tabs which are not available on the log template view. These are:

- Presentation for formatting history presentation for 800xA operator trends.
- Status for viewing log data directly from the log configuration aspect.

Other sources of status information are the Property Log Tab, and Status Light.

Presentation

This tab, Figure 326, is used to configure presentation attributes for the 800xA operator trend display.

The default settings for these attributes are set in one of the following ways:

- Object Type (i.e. in the Property Aspect).
- Object Instance (i.e in the Property Aspect).
- Log (in the Log Configuration Aspect).
- Trend Display.

These are listed are in order from lowest to highest precedence.

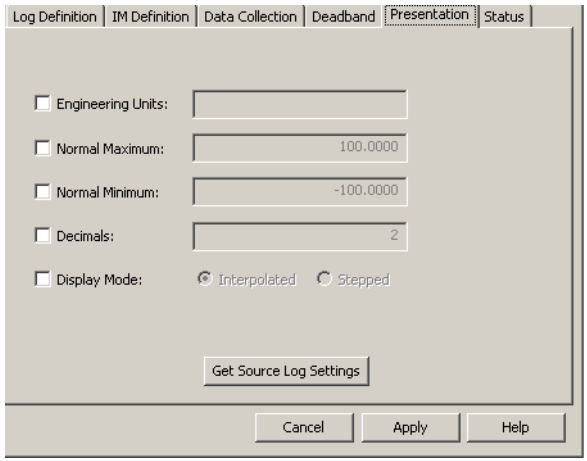


Figure 326. Presentation Tab

For example, override a value set in the Object Type by writing a value in the Object Instance. A value set in the Object Type, Object Instance, or the Log can be overridden in the Trend Display.

To override a presentation attribute the check box for the attribute must be marked. If the check box is unmarked the default value is displayed. The default value can be a property name or a value, depending on what is specified in the Control Connection Aspect.

Engineering Units are inherited from the source. It is possible to override it. Normal Maximum and Normal Minimum are scaling factors, used for the scaling of Y-axis in the Trend Display. The values are retrieved from the source but are possible to override.

The Number of Decimals are retrieved from the source but are possible to override.

The Display Mode can be either **Interpolated** or **Stepped**. Interpolated is appropriate for real values, Stepped is for binary.

Click **Apply** to put the changes into effect.

Status

This tab is used to retrieve and view data for a selected log, Figure 327. As an option, use the **Display** and **Settings** buttons to set viewing options. Refer to:

- Data Retrieval Settings on page 449.
- Display Options on page 450.

Click **Read** to retrieve the log data. The arrow buttons go to the next or previous page. Page size is configurable via the Data Retrieval Settings. The default is 1000 points per page.

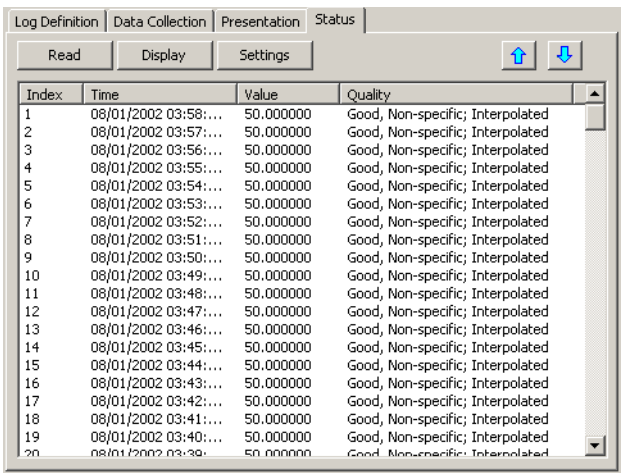


Figure 327. Status Tab

Data Retrieval Settings

The **Settings** button displays the Data Retrieval Setting dialog which is used to change the time frame and the number of data points to be retrieved, Figure 328.

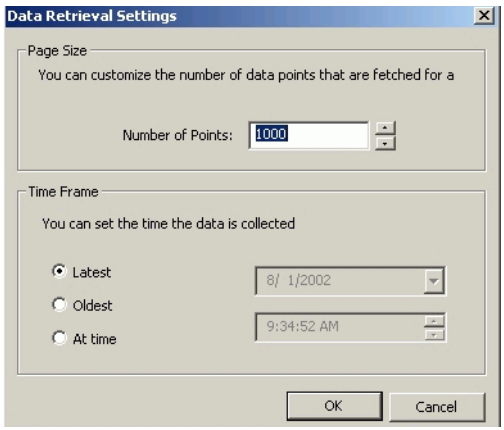


Figure 328. Data Retrieval Settings

Set the number of points per page in the Page Size area. Default is 1000 and maximum is 10,000.

Set the time range for retrieval of data in the Time Frame area.

- **Latest** retrieves one page worth of the most recent entries.
- **Oldest** retrieves one page worth of the oldest entries.
- **At time** is used to specify the time interval for which entries will be retrieved. The data is retrieved from the time before the specified date and time.

Display Options

The **Display** button displays the Display Options dialog which is used to change the presentation parameters data and time stamp, Figure 329.

Values are displayed as text by default. Unchecking the check box in the **Quality** area, displays the values as hexadecimals.

Timestamps are displayed in local time by default. Unchecking the first check box in the **Time & Date** area, changes the timestamps to UTC time.

Timestamps are displayed in millisecond resolution by default. Unchecking the second check box in the Time & Date area changes the timestamp resolution to one second.

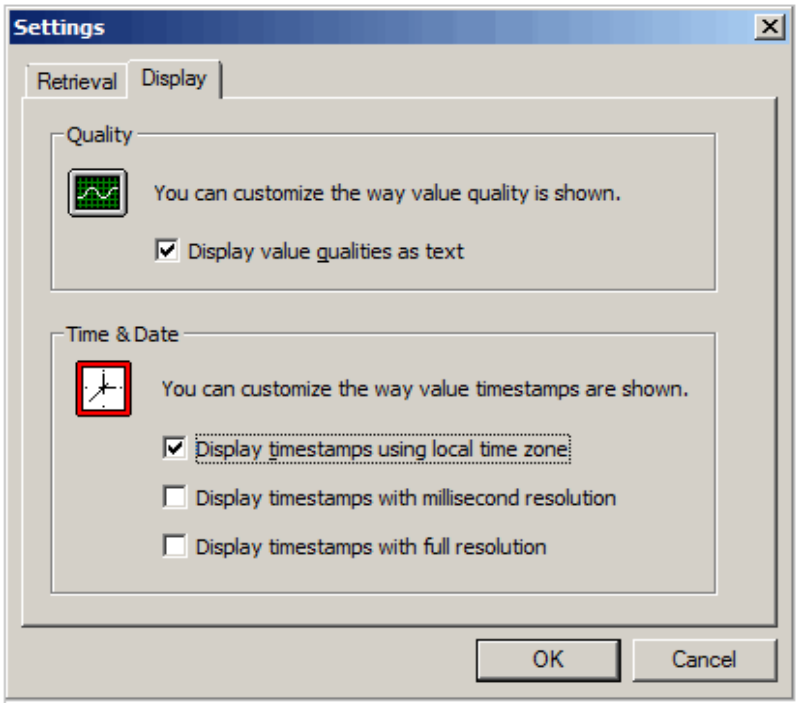


Figure 329. Display Options Dialog

Property Log Tab

The Property Log tab, Figure 330, displays log information. This tab is displayed by clicking the Log Configuration Template placeholder in the Log Configuration hierarchy.

The Data Size is the size of the Property Log on disk. It is the sum of the size of all logs. The size of each log file is the sum of the file header and the data storage size.

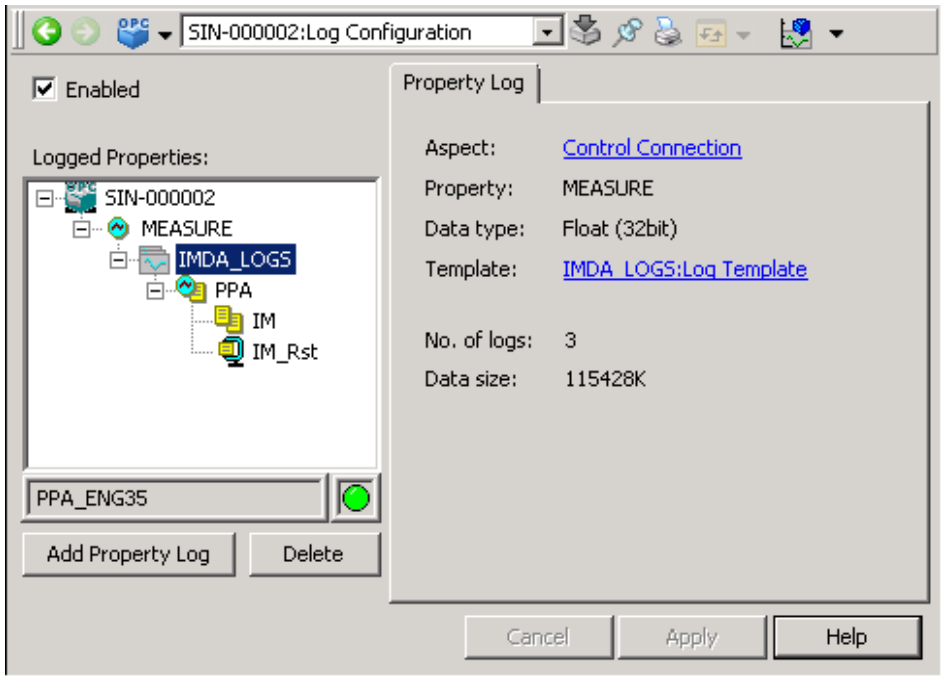


Figure 330. Property Log Tab

Status Light

The Service Status row shows status information for the configured Service Group. The name of the configured group is presented in the row. If no history sources have been defined (refer to Configuring Node Assignments for Property Logs on page 189, the text ‘Not Specified’ is displayed. Status for configured Service Group may be:

- OK: All providers in the service group are up and running.
- OK/ERROR: At least one service provider is up and running.
- ERROR: All providers in the service group are down.

History Control Aspect

The History Control aspect provides status and network information for the History Service Provider for a selected node in the Node Administration structure.

1. Select the **Inform IT History Object** under the applicable node in the Node Administration structure.
2. Select the **Inform IT History Control** aspect, Figure 331.

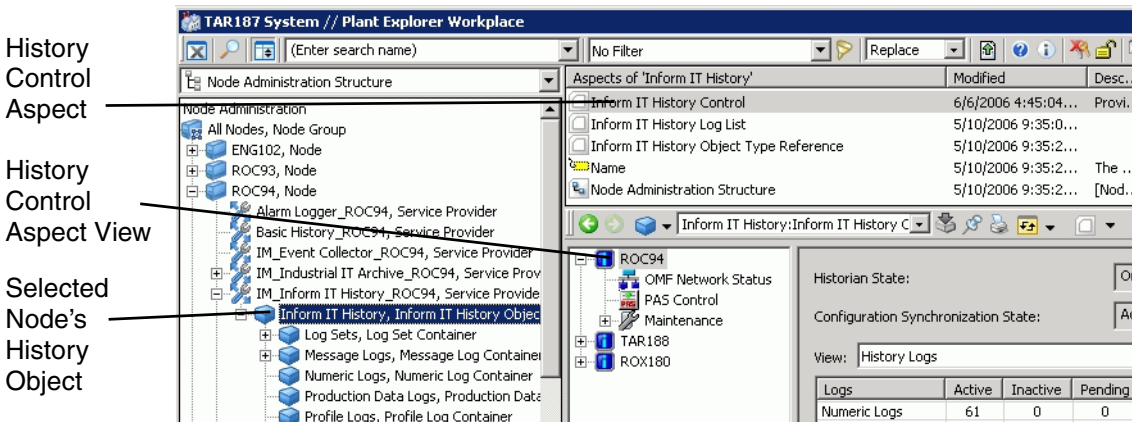


Figure 331. Accessing the Configuration View for the History Admin Aspect

Use the View selection to get information on: Hard Drive Usage, Database Usage, and History Logs for a particular node. The configuration view allows selection of History nodes from an available list. Additionally, the tree view has selections for OMF Network Status, PAS Control, and User Tag Management status.

PAS Control

Refer to Starting and Stopping History on page 437.

Synchronization

For object and log name synchronization, use the **Resynchronize Names** button. To force sysncronization of every object on the selected node with the Aspect Server, use the **Force Synchronization** button. The **Check Names** button produces a report of the object names, log names, and object locations in the aspect directory that are

different or not in the Oracle database. The **Check Synchronization** button produces a report of all the items in the aspect directory that are different or are not in the Oracle database.

User Tag Management Status

User Tag Management is described in *System 800xA Information Management Data Access and Reports (3BUF001094*)*.

OMF Network Status

The detailed view of the OMF Network Status for History Manager data is shown in Figure 332 and is described in Table 39.

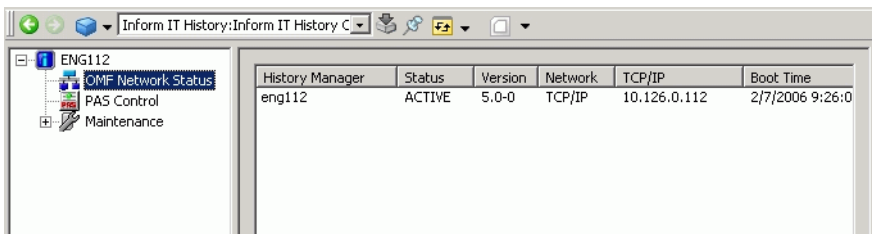


Figure 332. History Managers List

Table 39. History Manager Data and Related Information

Field	Description
History Managers	The History Managers field indicates the node name for the local History Manager and any available remote History Managers.
Status	A History Manager's status is set by the system: • ACTIVE - History Manager is ready to receive data. • INACTIVE - History is on the node, but is not running.
Version	This indicates the version of the History software for the corresponding History Manager. When working with a remote History Manager via the local History Manager, both must use the same version of History software.

Table 39. History Manager Data and Related Information

Network	This field indicates: • DCN address for nodes in systems with MOD 300 software. • MasterNet address for nodes in systems with Master software. • TCP/IP for nodes that are not connected to OCS control network via an RTA board.
TCP/IP	This field indicates the IP address, for example: 222.111.99.66
Boot Time	This field indicates the time that the node started.

Database Maintenance Functions

History Services provides several utilities for maintaining the history database. This includes functions for changing tablespace capacity for Oracle tablespaces, changing file storage capacity for file-based logs (TYPE1-TYPE5), and staggering data collection for numeric logs, as well as other various maintenance functions. Most of these functions may be performed with History running. Some functions require that History be stopped. There are three methods for using these utilities. A brief overview of each is provided below.

- The Instance Maintenance Wizard provides utilities for extending Oracle tablespaces and for managing files for file-based logs using a graphical user interface. How to access the utilities via the wizard is described in Maintaining the Oracle Instance on page 137. In addition, the wizard can be run using hsDBMaint -instance.
- The hsDBMaint menu provides access to the utilities for extending Oracle tablespaces and for managing files for file-based logs, as well as other utilities not supported by the Instance Maintenance Wizard. Guidelines for launching and using this menu are provided in Using the hsDBMaint Menu on page 456.
- Some database maintenance functions can be performed using the command line (DOS Command Prompt). Instructions are provided when applicable.

Using the hsDBMaint Menu

The hsDBMaint menu is started by invoking the hsDBMaint command which resides in the %HS_HOME%\bin directory. Invoke the hsDBMaint command from the DOS command prompt:

```
C:> "%HS_HOME%\bin\hsDBMaint.exe"
```

The list of available functions depends on whether the menu is invoked while History is online or offline. The online and offline menus are described in

- Online Database Maintenance Functions on page 456.
- Offline Database Maintenance Functions on page 458.

Online Database Maintenance Functions

If the hsDBMaint menu is launched without stopping History, the online hsDBMaint menu is displayed, Figure 333. These functions are described in Table 40. To invoke a function, enter the corresponding number, and then press ENTER. For example, for a Database Free Space Report, enter **1**. Then follow the instructions as indicated in Table 40. To exit hsDBMaint, choose **Exit** (**0**).

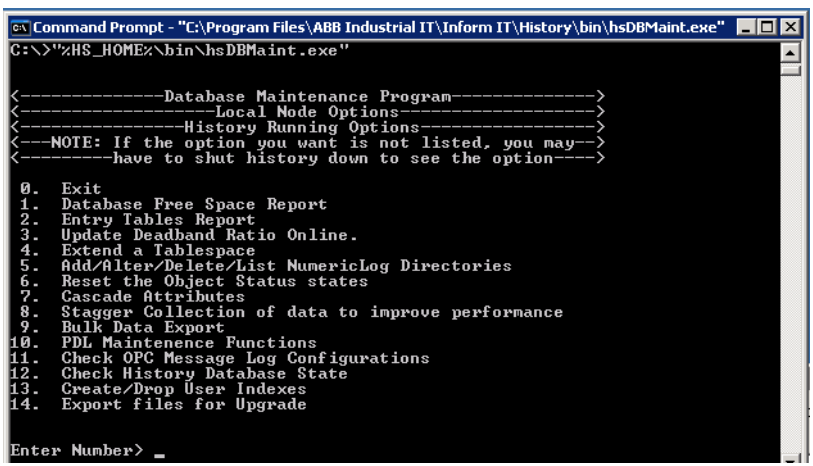


Figure 333. Database Maintenance Menu when History is Running

Table 40. Online Database Maintenance Functions

Refer to	When to Use
Database Free Space Report on page 459.	Generate a report on free space in Oracle tablespaces on the History disk.
Entry Tables Report on page 461.	Create a list of all property logs, their capacity, and other statistics.
Update Deadband Ratio Online on page 463.	Update deadband ratio and estimated log time period for logs that use deadband.
Extend a Tablespace: Extend Tablespace for Oracle-based History Files on page 464 or Extending Temporary Tablespace on page 466.	If a greater storage capacity is required than provided by default Oracle tablespace sizes. If the user gets a message indicating that there is not sufficient temporary tablespace to complete a certain task.
Add/Alter/Delete/List NumericLog Directories, refer to Directory Maintenance for File-based Logs on page 467.	Configure directories for file-based storage.
Reset Object Status States on page 470.	Determine whether the collection function will update the trend presentation attributes, or whether user-entered values will be used - Not applicable for OPC-type logs.
Cascade Attributes for Composite Logs on page 471.	This is required for logs where presentation, engineering units, and alarm attribute values have been entered manually for the primary log and must be propagated from the data source to secondary logs.
Stagger Collection and Storage on page 472.	Adjust data collection attributes for more even distribution of CPU load.
PDL Maintenance Functions on page 478.	Check the status of PDL configuration and correct error if necessary.
Check OPC Message Log Configurations.	Check tables, indexes, constraints for all OPC Message logs or for a specific OPC Message log.

Table 40. Online Database Maintenance Functions (Continued)

Refer to	When to Use
Check History Database State.	Checks state of logs (unused, empty, invalid access, and so forth) that need to be deleted, dropped or fixed.
Create/Drop User Indexes on page 478.	Create and drop user indexes to improve performance for SQL queries.
Export Files for Upgrade.	Do not use. This is to support future upgrades from 800xA 5.0.

Offline Database Maintenance Functions

Some database maintenance functions require History to be shut down. This is done via the Process Administration tool as described in Starting and Stopping History on page 437. If History is shut down before invoking the hsDBMaint command the complete (offline) menu is displayed, Figure 334. The offline functions are described in Table 41.

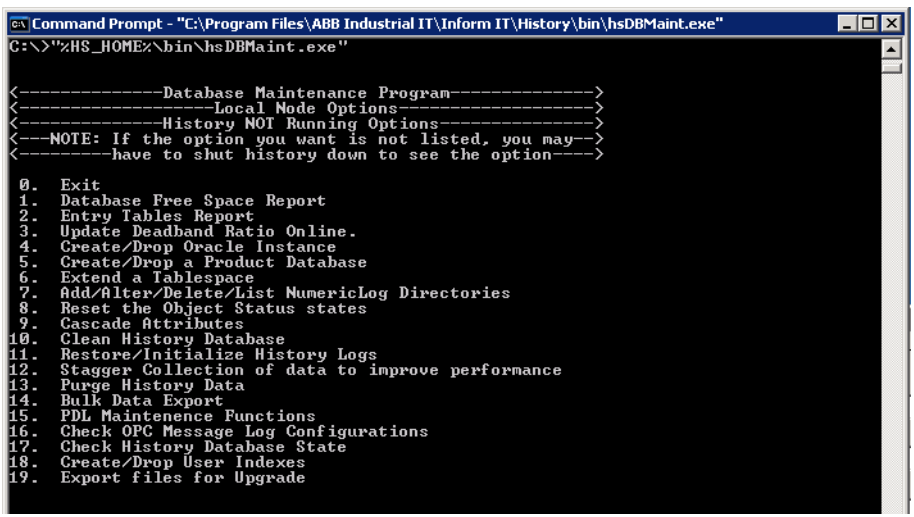


Figure 334. Database Maintenance Menu when History is offline

Table 41. Offline Database Maintenance Functions

Refer to	When to Use
Create or Drop Oracle Instance on page 480	Create a new Oracle instance or drop (delete an existing instance).
Create or Drop a Product Database on page 481	Create a new data base of type History or PDL, or to drop (delete an existing database).
Clean History Database on page 482	Deletes dangling references from the database. In the case of the dangling log name, this means that the name can be used again.
Restore or Initialize History Logs on page 482	Redefine missing tables for Oracle logs and missing files for file-based logs. Initialize all log tables and log file essentials, and erase all runtime data.
Purge History Data on page 483	Clear History data erroneously collected in the future.

Database Free Space Report

The Free Space Report indicates how much space is available in the tablespaces. To create a Free Space Report, choose **Database Free Space Report** from the hsDBMaint menu. This displays the Free Space Report Menu, Figure 335.

```
Free Space Report Menu

 0. Return to Main Menu
 1. Create free space report now
Enter Number>
```

Figure 335. Free Space Report Menu

Choose option **1** - **Create free space report now**. This creates the Free Space Report. An example is shown in Figure 336.

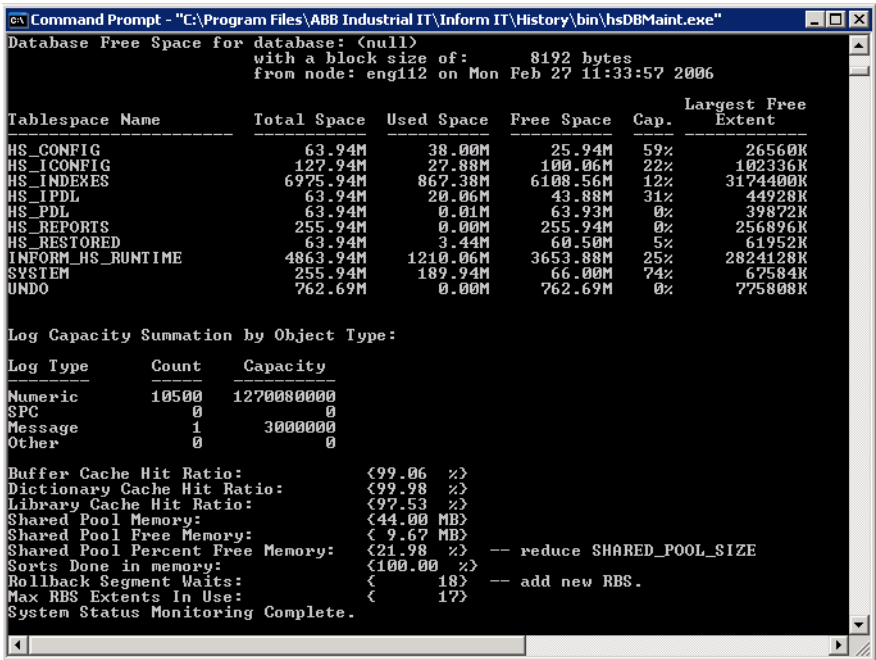


Figure 336. Example, Free Space Report

Also, view a free space report on the Tablespace/Datafiles tab of the Instance Maintenance wizard. Refer to Maintaining the Oracle Instance on page 137.

- **OPC Message log** uses INFORM_HS_RUNTIME and HS_INDEX tablespaces.
- **numeric logs** use HS_CONFIG and HS_ICONFIG tablespaces.
- **PDL application** uses Adjust the HS_PDL and HS_IPDL tablespaces.
- **Consolidation of OPC and PDL messages, and PDL data** use the INFORM_HS_RUNTIME and HS_INDEX tablespaces. If they run out of space, OPC and PDL message consolidation will fail. If the HS_PDL and HS_IPDL (index) tablespaces run out of space, PDL data consolidation will fail.

Entry Tables Report

This function generates a report of all property logs, their current capacity, and other useful statistics. It also provides good information for troubleshooting numeric logs. This report can be used to determine whether the problem is with history or another application such as the base system, or data access application. Run this function from the hsDBMaint menu, or from the DOS command prompt.

If this function is run via the hsDBMaint menu, then these report settings can be modified before the report is run:

- log ID to process. Use to generate a report for all logs, just one log, or a range of logs. The default is for ALL LOGS.
- change the line per page. The default is 24 lines per page.

If the above default settings are OK, it is best to run from the command line. This is used to direct the output to a file. Use the hsDBMaint menu to change a default setting.

This function may take a couple hours to complete for a large database, and generates a high CPU load.

To use one of these methods refer to:
- Running Entry Tables Report from the Command Prompt.
- Running Entry Tables Report from the hsDBMaint Menu.

Running Entry Tables Report from the Command Prompt

This generates a report for all logs and directs the report output to a specified text file. To run the report, from the DOS command prompt, enter:

c: > **hsDBMaint -report >** *tmp**filename***.txt**

where *filename* is the name of the file where the report output will be directed to.

or enter

hsDBMaint -report > *report.txt* **2>** *error.txt*

This creates two text files in the location from which the report was ran.

The file is given a .txt extension for viewing in a text editor. The entry table provides statistics to verify whether or not numeric logs are functioning properly. It can also be used to look up the Log ID for a specific log name. Log IDs are used in SQL queries against historical tables.

Rows marked with an asterisk (*) indicate logs with errors. A summary is provided at the end of the report. This information is useful for level-4 (development) support.

Running Entry Tables Report from the hsDBMaint Menu

To run this function from the menu, choose the **Entry Tables Report** option. This displays the Entry Tables Report Menu, Figure 337.

Figure 337. Entry Tables Report Menu

This menu shows the current settings for log ID to process and lines per page. The current settings can be changed before running the report.

If any of the settings need to be changed, choose the corresponding option from this menu and follow the dialog. When finished with the dialog, the Entry Tables Report menu is shown and the current settings will be changed according to the dialog. When the settings are correct, at the menu prompt, choose the **Create entry table report now** option. This generates the report in the Command Prompt window, Figure 338. A short summary is provided at the end of the report, Figure 339.

Adjust the height and width of the Command Prompt window to display the report properly.

S	Start Time	End Time	dbRatio	Next	Rows	GoodDt	BadDt	NoDt	lo
A	07 Jun 10 04:36:15	14 Jun 10 05:30:15	1.00:1	-1	40537	40537	0	0	
A	07 Jun 10 04:53:45	14 Jun 10 05:14:00	1.00:1	-1	40386	40386	0	0	
A	07 Jun 10 04:30:22	14 Jun 10 05:32:37	1.00:1	-1	40570	40570	0	0	
A	07 Jun 10 05:23:37	14 Jun 10 05:30:37	1.00:1	-1	40349	40349	0	0	
A	31 May 10 07:41:00	14 Jun 10 05:13:15	1.00:1	-1	40409	40409	0	0	
A	27 May 10 08:57:45	14 Jun 10 05:14:30	1.00:1	-1	40351	40351	0	0	
A	06 Jun 10 22:05:45	14 Jun 10 05:15:00	1.00:1	-1	40440	40440	0	0	
A	06 Jun 10 19:57:45	14 Jun 10 05:15:00	1.00:1	-1	40417	40417	0	0	
A	07 Jun 10 05:03:52	14 Jun 10 05:31:52	1.00:1	-1	40433	40433	0	0	
A	31 May 10 07:52:30	14 Jun 10 05:17:00	1.00:1	-1	40431	40431	0	0	
A	07 Jun 10 03:32:30	14 Jun 10 05:30:00	1.00:1	-1	40872	40872	0	0	
A	07 Jun 10 04:17:30	14 Jun 10 05:17:15	1.00:1	-1	40641	40641	0	0	
A	05 Jun 10 16:47:45	14 Jun 10 05:17:15	1.00:1	-1	40335	40335	0	0	
A	07 Jun 10 04:53:15	14 Jun 10 05:17:15	1.00:1	-1	40401	40401	0	0	

Figure 338. Sample Entry Tables Report

	62 Active Logs	0 Inactive Logs
Good Values	182624	0
badData Values	0	0
noData Values	34	0
Percent Good	99.9814	NA

Logs With Errors: 0

Figure 339. Report Summary

Update Deadband Ratio Online

Use this function to update deadband ratio and estimated log time period for logs that use deadband. Disk space is allocated to logs according to their log capacity, and not the amount of data actually stored. The log period can be adjusted on an individual log basis to keep log capacity at a reasonable size for each log. For all synchronous logs with or without data compaction, log capacity is calculated as follows: log capacity = (log period/storage interval).

When compaction is used, the log period is effectively increased so the same number of samples (as determined by the log capacity) are stored over a longer period of time. Consider the following two cases. On a very unstable process, such that every sample is stored and there is no data compaction, the effective log period will be equal to the configured log period. On a perfectly stable process (maximum data compaction, samples stored only at deadband storage interval), the effective log period is extended according to the following formula:

*effective log period = (deadband storage interval/storage interval) * configured log period*

In most cases, since the amount of process variation will not be at either extreme, the effective log period will fall somewhere in between the configured (minimum) and maximum value. The effective log period calculated according to the above formula is stored in the EST_LOG_TIME_PER attribute. This attribute is updated periodically according to a schedule which repeats as often as every five days. If necessary, use the History Utility to manually update this attribute.

EST_LOG_TIME_PER is used by the History Manager when it needs to determine which log to retrieve data from (seamless retrieval). For instance, consider an application with a composite log where the primary log stores data at 30 second intervals over a one-day time period, and the secondary log stores data at one-hour intervals over a one-week time period. The data compaction ratio is 5:1 which extends the effective log period of the primary log to five days. If a request is made to retrieve data that is three days old, the History Manager will know to get the data from the primary log rather than the secondary log, even through the configured log period of the primary log is only one day.

When EST_LOG_TIME_PER is updated, both internal History memory and the History database are updated with the new values of EST_LOG_TIME_PER and COMPACTION_RATIO. This will allow History Managers access to the information and when the History node is restarted the information will also be available.

To update deadband ratio and estimated log time period, choose **Update Deadband Ratio Online** from the Main Menu. This will generate a prompt that asks whether or not to continue. At the prompt, enter **y** (Yes) to continue, or **n** (No) to cancel and exit out of the hsDBMaint menu.

For a large database, this function may take several hours to complete.

Extend Tablespace for Oracle-based History Files

Extend the tablespace in the system when greater storage capacity is required than is provided by the default tablespace sizes. Use one of the following methods:

- Extending Oracle Tablespace Via Instance Maintenance Wizard.
- Extending Tablespace via the hsDBMaint Menu.

Extending Tablespace via the hsDBMaint Menu

First, review the current settings in the Tablespace Extension menu, Figure 340. To display this menu, choose **Extend a Tablespace** from the Main Menu.

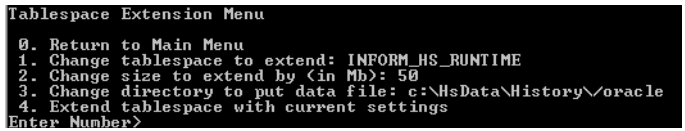


Figure 340. Tablespace Extension Menu

If these settings are acceptable, simply select option **4** - **Extend tablespace with current settings**. If one or more of these settings need to be changed, then follow the applicable procedure below.

Specify the Tablespace to Extend

The default tablespace is INFORM_HS_RUNTIME. To extend a different tablespace, select option **1** - **Change tablespace to extend**. This displays the Table Spaces available for extension menu, Figure 341.

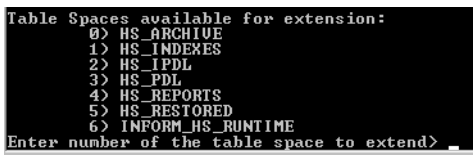


Figure 341. Table Spaces Available for Extension Menu

Enter the number corresponding to the tablespace to be extended (for example **3** for HS_REPORTS). The Tablespace Extension menu is returned with the new tablespace indicated for option 1.

Specify the Amount of Tablespace

The default amount is 50 Mb. To change this, select option **2 - Change size to extend by**. This displays a prompt to enter the size to extend table space by in Mb. Enter the size as shown in Figure 342.

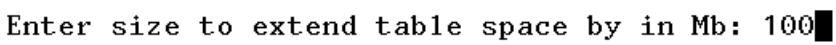


Figure 342. Entering New Tablespace Size

The Tablespace Extension menu is returned with the new value indicated for option 2.

Change the Directory

The default directory is `c:\HsData\History\oracle`. The oracle Instance Wizard requires the default directories and will not work if the directories are changed.

To specify a different directory for the extended tablespace data, select option **3 - Change directory to put data file**. This displays a prompt to enter the new data file directory. Enter the directory. The Tablespace Extension menu is displayed with the new directory indicated for option 3.

Apply the Changes

At this point, the criteria for extending tablespace should be correct, and the tablespace can actually be extended.

To extend tablespace with the current settings for size, directory, and tablespace, select option **4** - **Extend tablespace with current settings**.

This extends the specified tablespace by the specified amount, and then returns the Tablespace Extension menu.

Repeat above procedures to extend another tablespace. When finished, return to the Main Menu by selecting option **0-Return to main menu**.

Extending Temporary Tablespace

Temporary and rollback (or UNDO) tablespaces are sized based on the selections made in the Oracle instance wizard. Generally, the default tablespace is sufficient. If a message indicating that there is not sufficient temporary tablespace to complete a certain task appears, then extend the temporary tablespace. Reasons for extending temporary tablespace are:

- Complex SQL queries that use joins.
- Complex SQL queries against PDL.
- PDL deletion operations.
- Restoring a history database.

- Additional users may need added tablespaces.

For PDL operations, an indication of insufficient temporary or rollback space means the query needs adjusting or deleting the PDL data and making smaller increments. The restore will fail if it runs out of space during a restore operation and the space will need to be increased before retrying the restore.

Use the Instance Maintenance Wizard and click on the **Oracle Temp Space** link to to change the **Extend By** or **Max Size** of the TEMP_1.DBF tablespace. The Datafile may autoextend if the Autoextend column is checked. The Datafile capacity will extend by the amount indicated in the Extend By column. The Datafile may be extended in this manner up to the specified Max Size.

Managing Rollback Segments (Oracle UNDO Rollback)

Oracle provides tablespace for Rollback Segments (UNDO Rollback). This is used to temporarily hold the original data when changes are being made to the Oracle database. For example, if data is being deleted from a message log, the messages to be deleted are held until the change is actually committed. If an attempt to change the Oracle database would exceed rollback space, the change is not allowed and an error message is generated.

Use the Instance Maintenance Wizard and click on the **Oracle UNDO Rollback** link to change the **Extend By** or **Maz Size** of the UNDO_1.DBF tablespace. If the current size has reached the maximum size, then increase the maximum size. The recommended guideline is to double the current size.

The temp or rollback size cannot be decreased. Therefore, take care to increase the size by a reasonable amount that will provide the space needed. The only way to revert back to a smaller size is to make a backup of the database, drop and recreate the Oracle instance, and then restore the database.

Directory Maintenance for File-based Logs

This function is used to configure and maintain a directory structure for file-based storage of property logs. (File-based storage is applicable for property logs whose Storage Type is configured to be one of the filed-based types: TYPE1, TYPE2, TYPE4, TYPE5. For guidelines on file-based storage, refer to File Storage vs. Oracle Tables on page 255. To learn about the directory structure, refer to Directory Maintenance Overview on page 468. To get started, use one of the following methods:

- Flat File Maintenance via the Instance Maintenance Wizard on page 139.

- Using the Directory Maintenance Functions from the Menu on page 469.

Directory Maintenance Overview

Use the Directory Maintenance window to change the filed-based storage allocation or allocate space on other disks. To allocate space on a disk, specify the path for the disk, and how much memory is to be allocated for files. The system automatically creates directories on the specified disks as needed.

The example directory structure shown in Figure 343 illustrates how file-based storage works. This list can be displayed using the List property Log Directories function of hsDBMaint.

```
 3. Delete from Directory List
 4. List the Directory List
Enter Number> 4
 idx  dirID    Kbytes/KbytesQuota   file/Quta    ioUse/ppm   dirPath
-----  -----  --------------------   ---------   ----------   --------------------
[  0]   -1       758490/   1228800   2437/    0    12492.000   'c:\HsData\History\HSFF'
[  1]    0       310388/   1228800   1000/1000     2286.000   'c:\HsData\History\\HSFF\000'
[  2]    1       398850/   1228800   1000/1000     7584.000   'c:\HsData\History\\HSFF\001'
[  3]    2        49252/   1228800    437/1000     2622.000   'c:\HsData\History\\HSFF\002'
-----  -----  --------------------   ---------   ----------   --------------------
```

Figure 343. Example Directory Structure

This list shows one default directory: c:\HsData\History\HSFF. File-based logs are always stored sub-directories under a directory named HSFF. This is why the HSFF directory as a file quota of 0 (each log constitutes one file). These subdirectories are automatically created under the HSFF directory as needed.

The subdirectory names are three-digit sequential numbers starting with 000. For instance, `c:\HsData\History\HSFF\000`. Each subdirectory stores up to 1000 files. When the number of files exceeds 1000, another subdirectory (\001) is created to store the excess. Subdirectories are added as required until the quota on that disk is reached. New logs create files on the next disk (if space has been allocated). Disks are used in the order they are added.

In the example above, the default location has three subdirectories. A total of 2,437 files are stored in these directories. Directories \000 and \001 hold 1000 files each. Directory \003 holds 437 files.

Using the Directory Maintenance Functions from the Menu

To configure a specific directory structure, choose **Add/Alter/Delete/List NumericLog Directories** from the main menu. This displays the Add Directories for Fast Numeric menu, Figure 344.

```
Add Directories for Fast Numeric

 0. Return to Main Menu
 1. Add to Directory list
 2. Update Directory list
 3. Delete From Directory list
 4. List the Directory List
Enter Number>
```

Figure 344. Add Directories For Fast Numeric Menu

This menu is used to add a directory, change the quota for a directory, delete a directory, or list all directories. To return to the main menu, choose option **0** - **Return to Main Menu**.

For guidelines on using one of these functions, refer to:

- Add a Directory on page 469.
- Update Directory List on page 469.
- Delete From Directory List on page 470.
- List the Directory on page 470.

Add a Directory

To add a directory, choose option **1** - **Add to Directory List**. This initiates a dialog for configuring a new directory. Specify a quota and path for the new directory. Follow the guidelines provided by the dialog. Make sure the disk where the directory is being added has enough space for the directory. A directory with the name HSFF will be added under the path entered. For example, if the path is `\disk3`, a new directory will be created as follows: `\disk3\HSFF`

Update Directory List

This function is used to change the quota for a specified directory. The dialog is the same as for adding a directory.

Delete From Directory List

This function is used to remove a directory from the list of directories that are available for file-based storage. Delete is not allowed if there are files in the directory.

List the Directory

This function provides a list of available directories. The information provided in this list is similar to the information provided on the **Flat File Maintenance** tab of the Instance Maintenance window. Refer to Flat File Maintenance via the Instance Maintenance Wizard on page 139.

Reset Object Status States

Use this function to reset *attrsUpdated* and *userEntered* bits. These bits determine whether the collection function for a log will update the data presentation attributes for property logs, or whether user-entered values will be used. If the *attrsUpdated* bit for a property log is set (1), when the log is activated, the collection function gets the data presentation attribute values from the data source. Any user-defined values will be overwritten. When the *userEntered* bit is set, it prevents the collection function from overwriting the user-defined values.

At any given time, only one of these bits can be set for a property log. The Reset Object Status States function toggles the state for one or both bits.

To reset object status states, from the hsDBMaint menu, choose the **Reset the Object Status states** option. This displays the reset OBJECT_TABLE Status Bits menu, Figure 345.

```
Reset OBJECT_TABLE Status Bits Menu

 0. Return to Main Menu
 1. Change log name to process: All Logs
 2. Change attributes to reset: Both attrs_updated and user_entered
 3. Reset specified log(s) with current settings now
Enter Number>
```

Figure 345. Reset OBJECT_TABLE Status Bits Menu

This menu shows the current settings for logs to process, and attributes to reset. The current settings can be changed before executing the reset operation. To change any

of the settings, choose the corresponding option from this menu and follow the dialog. For log name to process, choose:

- All logs.
- All logs in a specific composite log hierarchy.
- Just one specific log.

For attributes to reset, choose:

- Both *attrsUpdated* and *userEntered* bits.
- Just *attrsUpdated.*
- Just *userEntered.*

When the settings are correct, choose the **Reset specified log(s) with current settings now** option.

To return to the database maintenance menu, choose the **Return to Main Menu** option.

Cascade Attributes for Composite Logs

This function propagates presentation, engineering units, and alarm attributes from the data source to secondary logs in the composite hierarchy. This is required for logs where attribute values have been entered manually.

To do this:

1. From the hsDBMaint main menu, choose the **Cascade Attributes** option. This displays the Cascade Attributes Log Menu, Figure 346.

```
Cascade Attributes Log Menu

 0. Return to Main Menu
 1. Change log ID to process: All Composite Logs
 2. Cascade attributes with current settings now
Enter Number>
```

Figure 346. Cascade Attributes Log menu

2. Specify the log to pass attributes to. The default is for all logs. If this is acceptable, skip this step and go directly to step 3. To specify a specific log,

choose option **1** - **Change log ID to process**. This displays the prompt: `Do you want to do all composite logs? [yn]`

Choose **n**. This displays the following prompt:

```
Enter the log ID of any log in the composite
hierarchy>
```

Enter the log name. The Cascade Attributes Log Menu is returned.

3. To cascade attributes with current setting for log, choose option **2** - **Cascade attributes with current settings now**. This passes the applicable attributes from the data source to the specified log, and then returns the Cascade Attributes Log Menu.

Stagger Collection and Storage

The following stagger functionality is recommended for all Information Management History configurations. After a history configuration is staggered, the collection of Basic History Service data from Process Portal is normalized to distributed loading of the Connectivity Servers evenly over time. History configurations are not staggered by default. If 960 logs are created with a one-second sample/storage interval, the activated logs will wake up every four minutes, which is the default for one-second logs, and request four minutes of one-second data from 960 logs. When staggered, collection will request four minutes of data from four logs every second. Over time, the Connectivity Servers will see a steady load from the Information Management Server. This improves performance of the entire 800xA System.

There are two methods for distributing CPU load for collection and storage: Stagger and Phasing. Refer to the applicable sections to determine which method to use, and then refer to How to Use the Stagger Collection Function on page 473.

Stagger

Stagger is typically applied to a number of logs with the same alignment, storage interval, and sample interval, and where sample and storage interval are equal. Stagger assigns a blocking rate greater than the storage interval. The first Activation Time is modified to distribute the load. For example, consider an application with 120 logs with the following configuration:

sample & storage interval = 30 seconds
alignment = 12:00:00
Blocking Rate = 150s

Without staggering, 120 messages with 5 entries each will be sent to storage every 2.5 minutes (150s). This will cause a spike in CPU usage. To distribute the load more evenly, use the Stagger Collection function to apply a blocking rate of 10m, and specify that 1/10 of the logs (12 messages with 20 entries each) be sent to storage every minute.

Phasing

Phasing is typically applied to a number of logs with the same alignment, storage interval, and sample interval, and the storage interval is significantly greater than the sample interval. For example, consider an application with 120 logs sampled at a 1-minute rate and an average calculated on a daily basis.

In this case, the 150s blocking rate would not prevent a CPU load spike at 00:00:00 (midnight). Phasing allows a set of logs to have slightly different blocking rates to prevent all logs from processing at the same time. For example, for a blocking rate of 30 minutes, phasing can be implemented by sub-dividing the 120 logs into groups and assigning different blocking rates to each group as follows:

40 logs w/ Blocking Rate = 29m
40 logs w/ Blocking Rate = 30m
40 logs w/ Blocking Rate = 31m

In this case a spike will only occur once every 26,979 minutes (29x30x31).

How to Use the Stagger Collection Function

This function must be run while History is running. To run this function, from the hsDBMaint main menu, choose the **Stagger Collection of data to improve performance** option. This displays a summary of the current collection/storage performance, Figure 347.

```
Stagger Summary Information: ( Sample/Storage/Blocking units are seconds )

Total(Type)              Time          Sample Storage Blocking Range AvgRate
---------- ------------------ ------ ------- -------------- ---------
   50(OPC HDA)  01 Jan 90 00:00:00      1       1     240/240        12.50
  400(OPC HDA)  01 Jan 90 00:00:00      5       5     300/300        80.00
---------- ------------------ ------ ------- -------------- ---------
Average Requests Per Minute from TTD/PHL:      0.00
Average Requests Per Minute from OPC HDA:     92.50
Average Requests Per Minute to hsStorage:     92.50
```

Figure 347. Summary of Collection/Storage Performance

The summary categorizes the logs by sample rate and storage rate. Each line of this summary provides the following information for a category of logs:

Total	Total number of logs in this category. This includes both cyclic subscriptions to OCS objects and periodic retrieval of OPC/HDA from controllers.
Prim	Number of logs in this category from cyclic subscriptions to OCS objects.
OPC/HDA	Number of logs in this category from periodic retrieval of OPC/HDA from controllers (refer to Figure 347).
Time	First Activation time for logs in this category.
Sample	Sample rate for logs in this category (in seconds).
Storage	Storage rate for logs this category (in seconds).
Blocking Range	Shortest blocking rate for any logs in this category (in seconds)/ Longest blocking rate for any logs in this category (in seconds).
AvgRate	Calculated average number of requests per minute for this category based on number of logs, and sample, storage, and blocking rates. This equates to the average number of disk access transactions per minute for the system.

The prompt following the summary is used to either exit this function (by pressing `n`), or continue with the Stagger function (by pressing `y`) .

If continuing with the Stagger function, recommendations to improve performance for logs in the first category are displayed, Figure 348.

Using the Default Values

If the defaults are used, the program will ask whether or not to use stagger now. Press `y` to apply the new stagger values, or `n` to ignore the changes made. The Information Management processes (PAS) must be restarted in order for the stagger changes to take effect.

Other Considerations

It is generally accepted that the Information Management requests per minute should be kept in the 500 to 1,000 requests per minute range. If the requests per minute are over 1,000, the performance of the Connectivity Servers can suffer. Because most 800xA sources are exception driven, a one-second log typically does not have 60 values per minute. If the rate is over 1,000 per minute, the custom selection should be used to increase the blocking rates, decreasing the average OPC/HDA request rate to the Connectivity Servers.

```
Stagger Summary Information: ( Sample/Storage/Blocking units are seconds )

Total(Type)              Time          Sample Storage Blocking Range AvgRate
------------------ ------------------ ------ ------- -------------- ---------
   60(OPC HDA)     01 Jan 90 00:00:00      1       1        1/240       74.75
   10(OPC HDA)     01 Jan 90 00:00:00      5       5      300/300        2.00
   12(Primary)     01 Jan 90 00:00:00     15      15      900/900        0.80
   12(TTD/PHL)     01 Jan 90 00:00:00     15      15     1800/1800       0.40
   36(OPC DA)      01 Jan 90 00:00:00     15      15      900/900        2.40
   36(OPC HDA)     01 Jan 90 00:00:00     15      15     1800/1800       1.20
------------------ ------------------ ------ ------- -------------- ---------
Average Requests Per Minute from TTD/PHL:      0.40
Average Requests Per Minute from OPC HDA:     77.95
Average Requests Per Minute to hsStorage:     81.55

Do you wish to continue? [yn] y
Default Summary:
   60(OPC HDA) logs with blocking   4:00( 240 points ), stagger = 240
   10(OPC HDA) logs with blocking   5:00(  60 points ), stagger =  60
   12(Primary) logs with blocking  15:00(  60 points ), stagger =  60
   12(TTD/PHL) logs with blocking  30:00( 120 points ), stagger = 120
   36(OPC DA)  logs with blocking  15:00(  60 points ), stagger =  60
   36(OPC HDA) logs with blocking  30:00( 120 points ), stagger = 120
Average requests per minute from TTD/PHL:       0.40
Average requests per minute from OPC HDA:      18.20
Average requests per minute to hsStorage:      21.80

Do you wish to use the defaults? [yn] _
```

Figure 348. Defaults for Stagger Function

Changing the Default Values

If the defaults are not used, the program will step through a procedure for changing the defaults, one value at a time. This procedure is described below.

If the values are changed, instructions for entering a new blocking rate are displayed, Figure 349.

```
Do you wish to change these recomendations? [yn] y
Enter the new Blocking rate in OmfTimeInterval format:
omfSECONDS(3)
omfMINUTES(4)
omfHOURS(5)
omfDAYS(6)
omfWEEKS(7)
Enter value( 0-4095 ):
```

Figure 349. Instructions for Changing Blocking Rate

Enter a new blocking rate (without units) at the prompt. The range is 0 to 4095. For example, enter 12.

```
Enter value( 0-4095 ): 12
```

When the prompt for units is displayed, enter the units by entering the appropriate integer code: 3 = seconds, 4 = minutes, 5 = hours, 6 = days, 7 = weeks. In the example below, the integer code for minutes (4) is selected.

```
Enter unit(3 for omfSECONDS etc.):4
```

When the prompt for stagger is displayed, Figure 350, enter an integer value within the specified range. The range is based on the blocking rate specified.

```
Stagger is number of intervals to divide the Blocking rate into
Enter stagger ( 2 - 120 ):
```

Figure 350. Prompt for Stagger

The stagger value specifies the number of divisions within the specified blocking rate that will be used to distribute the load. After entering the stagger value, a summary of the changes made is displayed, Figure 351.

```
Enter stagger ( 2 - 120 ): 24

NEW USER ENTERED DATA
---------------------
  Blocking Rate:  12:00
  Stagger:        101 PRIMARY logs staggered in 24 time slots.  Each
                  time slot will take 0:30
################################################################################

Do you wish to change these recomendations? [yn]
```

Figure 351. Summary of Changes

The prompt following this summary can be used to either accept these values (by pressing n), or change these values (by pressing y). In this case, the specified blocking rate is 12 minutes. The selected stagger divides each blocking interval into 24 30-second time slots. Thus at every 30-second time slot,

This procedure must be repeated for each category. When completed for each category, a new summary will be displayed with a prompt to choose whether or not the changes made should be applied, Figure 352.

```
Is your selection correct? [yn] y
Summary:
  101 PRIMARY (   8.42) logs with blocking  12:00( 120 points ), stagger = 24
  101 PRIMARY (  10.10) logs with blocking  10:00(  50 points ), stagger = 10
 1202 PRIMARY (  80.13) logs with blocking  15:00(  30 points ), stagger = 15
 1602 PRIMARY ( 106.80) logs with blocking  15:00(  15 points ), stagger = 15
 1001 PRIMARY (  33.37) logs with blocking  30:00(  15 points ), stagger = 15
Average requests per minute from TTD:         0.00
Average requests per minute to hsStorage:   238.82

Do you wish to stagger NOW? [yn]
```

Figure 352. Revised Summary with Prompt to Apply Changes

Press `y` to apply the new stagger values, or `n` to ignore the changes made. The Information Management processes (PAS) must be restarted in order for the stagger changes to take effect.

PDL Maintenance Functions

These functions, Figure 353, are used to check the status of the PDL configuration and correct error if necessary. The PDL configuration may become corrupted following an hsBar restore if the backup file being used was made when history was still running. The check function (2) may be done online. The correct errors function (3) requires history to be stopped via PAS.

```
Enter Number> 15
hsConnectToOra @ 1616 ¦ LOCALE set to 'ENU'
*****************************************
********* PDL Maintanence Menu **********
*****************************************
***** History Status:      INACTIVE *****
***** Information Level:          1 *****
*****************************************

 0) Exit Menu
 1) Change Information Level: 1
 2) Check status of PDL database
```

Figure 353. PDL Maintenance Menu

Create/Drop User Indexes

This is used to create and drop user indexes to improve performance for SQL queries.

To access this function, from the hsDBMaint main menu, choose the **Create/Drop User Indexes** option. This displays the Table Index Maintenance menu, Figure 354.

```
Table Index Maintenance Menu

 0. Return to Main Menu
 1. Action to perform: Create
 2. Current index name: None
 3. Name of table index belongs to (for create): None
 4. Name of column index will be based on (for create): None
 5. Name of tablespace to create index in (for create) : HS_INDEXES
 6. Size of Initial and Next extents (for create): 512K
 7. Create the index None based on the table None
    and the column None
    in the tablespace HS_INDEXES with an Initial and Next extent size of 512K
Enter Number> _
```

Figure 354. Table Index Maintenance Menu

Dropping an Index

To drop an index:

1. From the Table Index Maintenance menu, choose **Action to perform** (**1**), and then enter **d** to select the drop action.
2. Choose **Current index name** (**2**), then enter the name of the index to drop.
3. Choose **Drop the index** (**7**) to drop the selected index.

Creating an Index

To create an index:

1. From the Table Index Maintenance menu, choose **Action to perform** (**1**), and then enter **c** to select the create action.
2. Choose **Current index name** (**2**), and then enter the name for the new index.
3. Choose **Name of table index belongs to** (**3**), then enter the name of the table.
4. Choose **Name of column index will be based on** (**4**), then enter the column names. Separate each name with a comma, and DO NOT enter any white space between names. Press **<Enter>** to display a list of column names.
5. Choose **Name of tablespace to create index in** (**5**), then enter the tablespace name. Press **<Enter>** to display a list of tablespace names.

6. Choose **Size of Initial and Next extent(s)** (**6**), then enter the size in kilobytes.
7. Choose **Create the index...** (**7**) to create the new index based on the entries made for steps 3-6.

Creating indexes requires additional disk space. Be sure to extend the PDL tablespace or PDL Index tablespace (if it exists) before creating a new index. Also, indexes make retrieval faster, but slow down storage and deletion operations. This should be considered when adding indexes to tables.

Create or Drop Oracle Instance

Use this offline function to create or drop (delete) an Oracle instance. History MUST be shut down prior to creating or dropping a database. Use PAS to stop History Services as described in Starting and Stopping History on page 437.

Choosing **Create/Drop Oracle Instance** displays a usage file describing the hsDBMaint -instance options (refer to below) and steps. The command prompt is then returned.

The History database configuration and all Oracle-based Historical data will be lost when the database instance is dropped.

USAGE: hsDBMaint -instance [-c [-h][-r <resumeStep>] | -d | -m | -q] [-a][-f <configFileName>][-v][-t <traceFileName>]

-c: Create an Oracle instance.

-h: Create the History database after creating the Oracle instance.
Must be used in conjunction with -c.

-f: Use <configFileName> instead of the one stored in "%HS_DATA%".

-r: Start the creation process at <resumeStep>.
Must be used in conjunction with -c.

-d: Drop the entire Oracle instance and any associated files.

-m: Modify the table spaces and flat files to match the config file.

-q: Query the table space and flat files and save to the config file.

-v: Run in verbose mode.

-t: Write all output to <traceFileName>.

-a: Auto adjust table spaces and flat files to fit the History configuration. Must be used in conjunction with -c or -m.

Create or Drop a Product Database

Use this offline function to create or drop (delete) a database. History MUST be shut down prior to creating or dropping a database. Use PAS to stop History Services as described in Starting and Stopping History on page 437.

When dropping a database, all configuration data and stored log data is deleted. Use this function only if it is determined that it would be better to delete any work up to this point and start from scratch.

If dropping a database after History has been running (database has live data), be sure to archive any data that needs to be saved.

To create or drop a product database, from the hsDBMaint main menu.choose **Create/Drop Product Database**. This displays the Create/Drop Product Database menu, Figure 355.

```
Create/Drop Product Database Menu

 0. Return to Main Menu
 1. Change type of product database to create/drop: History
 2. Change operation on product database: CREATE
 3. Create/drop product database with current settings now
Enter Number> 
```

Figure 355. Create/Drop Product Database Menu

This menu shows the current settings for type of database to create/drop, and the operation to perform (create or drop). Change the current settings before executing the create or drop operation.

To change any of the settings, choose the corresponding option from this menu and follow the dialog. For type of database choose **History** or **PDL**.

For operation to perform, choose **Create** or **Drop**.

Creating History will automatically create PDL if the option is installed.

When finished with the dialog, the Create/Drop Product Database menu will return and the current settings will be changed according to the dialog. When the settings

are correct, choose the **Create/drop product database with current settings now** option. When the operation is finished, the Main Menu will be displayed again.

To return to the database maintenance main menu, choose the **Return to Main Menu** option (0).

Clean History Database

Use this function to clean the History database. History MUST be shut down prior to cleaning the History database. The menu item will not be available in the hsDBMaint menu if the menu is opened while History is started. Use PAS to stop History Services as described in Starting and Stopping History on page 437.

Once, History has been stopped, from the hsDBMaint main menu choose **Clean History Database**. This displays the Clean History Database Menu, Figure 356.

```
Clean History Database Menu

 0. Return to Main Menu
 1. Change version of History database: 1.2-0
 2. Clean History database now
Enter Number> 
```

Figure 356. Clean History Database Menu

Choose option **2** - **Clean History database now**. This cleans up the database, and then returns the Clean History Database menu. To return to the Main Menu, choose option **0** - **Return to main menu**.

Restore or Initialize History Logs

Use this function to:

- Redefine missing tables for Oracle logs (storage type Oracle) and missing files for file-based logs storage type (TYPE1, TYPE2, TYPE4, TYPE5).
- Initialize all log tables and log file essentials, and erasing all runtime data.

History MUST be shut down prior to initializing History logs. The menu item will not be available in the hsDBMaint menu if the menu is opened while History is started. Use PAS to stop History Services as described in Starting and Stopping History on page 437.

To restore or initialize History logs, from the hsDBMaint main menu, choose **Restore/Initialize History Logs**.

Purge History Data

If, for some reason, the system time is set ahead this will cause History to collect data "in the future". For instance, this would occur if the time was set wrong. If this occurs, even if the time is fixed, History will not collect any more data because it will not collect data that is older than the most current data collected. In this case use the Purge History Data function to delete all data with future time stamps.

History MUST be shut down prior to purging History data. The menu item will not be available in the hsDBMaint menu if the menu is opened while History is started. Use PAS to stop History Services as described in Starting and Stopping History on page 437.

Use either the -purge command line option, or use hsDBMaint.

To use hsDBMaint, from the hsDBMaint main menu, choose **Purge History Data**.

Invoking this function prompts to actually purge the data or just check which logs have future data, Figure 357. The default selection is to just check, and not purge.

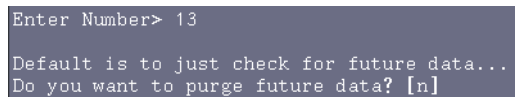


Figure 357. Prompt to Purge or Check

After specifying whether or not to purge, a prompt for a time to compare data time stamps with is displayed, Figure 358. The default is the current time.

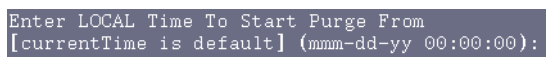


Figure 358. Prompt for Time

When the function is finished, a table is displayed that indicates the number of logs checked, and the number of logs found to have future data, Figure 359. When History detects a time change of greater than 75 minutes (this is configurable via the environment variable HS_MAX_TIME_CHANGE), History will log an operator message stating that fact.

```
                    ORACLE        TYPE 1        TYPE 2
LogType           cnt    err    cnt    err    cnt    err
---------------- -----  -----  -----  -----  -----  -----
hsNUMERIC_LOG(1)    0      0      8      0      0      0
hsMESSAGE_LOG(2)    0      0      0      0      0      0
hsSPC_LOG(6)        0      0      0      0      0      0
---------------- -----  -----  -----  -----  -----  -----
Total Logs Checked:                8
Total Logs With Future Dates:      0
OK to restart History
```

Figure 359. Purge Future Data Results

History Resource Considerations

History places a load on the CPU, disk, and process memory resources in the History node. Data collection and storage place a fixed load on History node resources, based on the History database configuration. This load can only be adjusted by changing the database configuration.

For instance, sample rates, deadband, blocking rates and so on can be adjusted. Data retrieval applications such as Reports and trend displays, place a variable load that depends on the extent and frequency of those applications.

For a detailed breakdown of these resources, refer to Resource Requirements Related to History Configuration on page 485 and Resource Requirements for User Applications on page 486.

It is important to determine whether or not the History configuration and user applications will overload the History node resources. To analyze resource usage by the History configuration and user applications, refer to Determining Resource Usage for Your Application on page 488. This analysis will show where the configuration or user applications may overload resources. Based on this analysis, any one or a combination of the following steps to make any necessary adjustments can be made:

- Expand/upgrade the hardware for the History node (more disks, more memory, faster processor), or install a second node.
- Modify the History configuration.

- Modify user applications.

Resource Requirements Related to History Configuration

CPU

CPU performance is related to capacity for History collection and storage functions. This can be adjusted by configuring sample rates, deadbands, and blocking rates.

Disk Space

Disk space resources are used by the History filesets that are loaded on disk when History is installed, by Oracle tablespace for logs and archive database entries, and by file-based logs. Disk space allocation for Oracle and file-based logs is calculated and set by the Database Instance Configuration wizard based on the selections made in the wizard at post-installation or later if required, refer to Maintaining the Oracle Instance on page 137.

- The disk space required for the History filesets is fixed at 30 meg.
- Oracle tablespace is used by message logs, report logs, PDLs, asynchronous property logs, synchronous property logs with Oracle storage type, log configuration data, and archive database entries. Extend Oracle tablespaces for runtime based on the per/entry requirements in Considerations for Oracle or File-based Storage on page 188.
- File space for file-based property logs is automatically allocated. Additional disk space can be allocated. Memory requirements per file are described in Considerations for Oracle or File-based Storage on page 188.

Random Access Memory (RAM)

RAM resources are required for:

- Operating System resources. This includes:
  - Six Shared Memory IDs for:

    Runtime Shared Memory Size = (600+ total # of all logs) * 64 + (500 + # of property & message logs) * 184 Bytes

Other resources = 3.76 Meg.

- OMF.
  - Object names = 500 * objects defined (50 maximum) Bytes.
  - Advant OCS Object Data Sources = 1Kbyte per primary log minimum.
- hsServer process = total # of History objects * 100 Bytes where History objects are Log Groups, Log Sets, Composite Log Typicals, Log Typicals, Composite Logs, and so on.
- hsCollection process = total # of all logs * 376 Bytes.
- Overhead.
  - History executables = 11 Meg.
  - Oracle Shadow Processes = 24 Meg. + (6 Meg. * # of archive devices).

Resource Requirements for User Applications

Load on resources for User applications comes from Reports, Display Services, and trend displays. These applications place a load on local and remote History Managers:

- Requests from local History Manager to local hsServer = 40Kbytes OMF shared memory per request for property log (500 property values maximum).
- Requests from remote History Manager to local hsServer = 20Kbytes OMF shared memory per request for property log (500 property values maximum).

Expanding dB Block Buffer Space

History uses Oracle for everything except synchronous file-based logs. Performance for PDL, Report, and Message logs can be improved with additional shared memory. The default size for shared memory is based on the configuration parameters selected when the database was configured, and the maximum memory

in the server. The minimum value is 512M. The maximum value depends on the operating system architecture.

For x86 operating systems, the maximum value for these parameters should be 1792 M, 1.75GB. If more memory is configured, the Oracle database runs out of resources and generates errors.

For x64 operating systems the maximum values used when IM configures Oracle is 3840 M, 3.75GB. Larger amounts of memory can be used if the hardware has the additional memory.

The following procedure can be used to adjust the Oracle shared memory. It is recommended that the value is modified and the server is rebooted. After reboot, the new values will be used.

Here is an example that changes the value to 600 M.

The procedure to type in the command line is as follows (commands to be typed are shown in bold and the responses are shown in normal text):

```
C:\Users\800xaInstall> sqlplus /

SQL*Plus: Release 11.2.0.2.0- Production on Tue Jun 8
16:41:58 2010

Copyright (c) 1982, 2008, Oracle. All rights reserved.

Connected to:

Oracle Database 11g Release 11.2.0.2.0- Production

SQL> show parameter memory

NAME                      TYPE                    VALUE

-----------------------------------------------------
hi_shared_memory_address  integer                  0

memory_max_target         big integer             532M

memory_target             big integer            532M

shared_memory_address     integer                 0


SQL> alter system set memory_max_target=600M
scope=spfile;

System altered.
```

```
SQL> alter system set memory_target=600M scope=spfile;
System altered.
SQL>exit
```

Then restart the PC to have the new values take effect.

The memory in use for a given server should always be less than the memory installed. If a system has 700 M for oracle shared memory, has 3GB of memory installed, and the current memory usage is 2.5GB, it would be safe to add 500 M or less to the oracle shared memory. If the value is changed to a higher value and the memory in use is consistently higher than the installed memory, then the value should be reduced to keep the memory equal or less than the memory installed.

Determining Resource Usage for Your Application

Use these guidelines to determine how the History database configuration and user applications will load resources. The affected resource is indicated in parenthesis:

- What type of Operator Station Graphics and Trending, Reports, and other user applications are expected:
  - How many operator stations will this History node service? (User Applications, CPU and Memory).
  - The frequency and type of applications using NUMLOGVAL? (User Applications, CPU and Memory).
  - Will Display Services be Installed? (User Applications, CPU and Memory).
  - Are any reports expecting certain types of data? (1 hour report returning 60 values, 1 for each minute in the hour) (History configuration, CPU, memory).
- Will PDL be used? If YES, consider the following:
  - Adjustment of HS_PDL tablespace to hold max number of PDLs.(Disk).
  - Log associations to property logs. The maximum log period should be greater than the expected duration of the batch to allow archival of all data and reporting to complete before the property log wraps.(History configuration, CPU memory).

- Will message logs be used? If YES, consider the following:
  - ORACLE tablespace is used for message logs.(Disk).
  - Capacities over 250,000 yield poor performance.(History configuration, CPU).
  - Will archiving be used for message logs (CPU, will need archive group).
- Will report logs be used? If YES, consider the following:
  - ORACLE tablespace (HS_REPORTS) is used to store reports in history. (Disk).
  - Will archiving be used for report logs (CPU, will need archive group).
- Will property logs be used? If YES, determine:
  - the number of source tags and their names. (Real Time Accelerator Board, CPU).
  - the rates to collect from tags. (100 logs at 30s, 200 logs at 1m, 1000 logs at 2m)(Real Time Accelerator Board, CPU).

    In MOD 300, sample rates for tags should be based on the loop processing rate configured for the tags in CCF (interaction of processing rate, processing phase, and base rate). A sample rate for a tag should never be faster than the tag's loop processing rate.

    In fact, in many cases, the sample rate can be slower than the loop processing rate. If sampling is done at a rate that is faster than the rate at which the process value changes, most of the sample data will be redundant, and will use CPU resources which could otherwise be used for additional logs. If a report requires 60 values for an hourly report, a 1m storage interval would make sense.
  - Can deadband be used. If YES, what is the expected compaction ratio?(5:1?) (History configuration, CPU, memory).
  - Will archive of data be necessary?
  - For all clients of history (numlogval, OS, reports, PDL), What log periods are required? (1 w, 2 w, 4w, etc.) (Disk).

Log Period should be determined by the clients who view log data. The OS is the biggest concern here. Do the operators need 1 week or 4 weeks of online data.Batch applications are another consideration. The log associations in PDL require the log to have a log period at least 2 times the longest batch time to guarantee all the property data can be archived with the PDL. If all functionality for history data is clearly defined, a database satisfying 95% of the planned functionality can be created on the first try.

In addition, if care is taken to only define what is needed (1 w logs instead of 1 year logs), overall system performance will be much better.

– Are hierarchies needed?

- Will DUAL history be used? If YES, consider the following:
  – DUAL history allows history configurations to be distributed on multiple history nodes. A composite log with a primary log on two different history nodes is a DUAL history log. With a DUAL history log, each primary log collects data from the same data source independently. This means the data and timestamps will be similar but not equal for the two primary logs.
  – DUAL history allows clients of history (OS, REPORTS) to get data for logs when one of the two nodes is down.
  – DUAL history creates twice the load on controllers and the DCN.

  Having two nodes configured with DUAL history logs makes loading an issue. With DUAL history, both nodes use the same amount of RTA/OMF resources to collect data. However, the retrieval of data can be directed to one node more than the second node.

  In addition to the seamless algorithm applied for retrieval requests, the following criteria have been added to better handle selection of a log for a request by access name:

  – Uptime of node where log exists.
  – local is chosen over remote (local should always be faster than remote).
  – sequence number of log. (all other conditions being equal, the lowest sequence numbered log will be used. Sequence number is the ‘-1-o’ or ‘-2-o’ attached to the generated log name.

  These additional sorting parameters will allow configurations to:

- Evenly distribute retrieval load on two History nodes (on average) by creating half the logs with the first log created on Node A and the other half with the first log on Node B.
- Favor one History node over the other by creating all the logs with the first log on NodeA. (One history node may be faster and have more memory than the other.)

Temporary File Disk Space for History

History uses disk space for temporary files during runtime. This includes:

- Archive - 16 meg. per volume (M/O drive = one volume).
- Inter-process communication - 5 meg.
- Collection snapshot files:
  - For Numeric, Message, and Report Logs: Minimum = # of Start Active Logs * 136, Maximum = 4 * (# of Start Active Logs * 136).

Start Active logs are logs whose Log Start-up state is configured to be Active.

- Storage shutdown file - 2 to 5 Meg.

To assure that there is disk space to accommodate these temporary files, leave free space in whichever volume group %HS_DATA uses, using the above numbers as guidelines. This free space is in addition to any other free space requirements that your History application may call for.

Environment Variables

Environment variables are created when installing Information Management software, and are used by Information Management to specify locations for data, configuration, and executable files. The location of these files may vary from one installation to the next. The environment variables provide a uniform and consistent method for specifying the location of these files for all Information Management installations.

Most environment variables are used internally by Information Management applications during installation and start-up. Some environment variables can be

used while performing system administration functions, for example to run the hsBAR.exe file when backing up History data (Running hsBAR on page 432).

The environment variables that are most likely to be used are described in Table 42.

These variables are generally used in commands entered via the Windows Command Prompt. The variables may also be used in Windows dialogs that require a path specification (for example, Open File dialogs).

Table 42. Environment Variables

Variable	Points To(1)
%ABB_ROOT%	The *ABB Industrial IT* directory where Information Management System Services is installed. Default path: C:\Program Files\ABB Industrial IT\ This is the location for most directories and files related to Information Management software.
%HS_HOME%	The *History* directory where History Services software is installed. The default path is C:\Program Files\ABB Industrial IT\History\ This is the location for executable files including system administration utilities such as hsDBMaint (database maintenance), hsBar (History backup), and dbConfig (History configuration).
%HS_DATA%	The *HsData\History* directory on the History data drive specified during History Services installation. The default path is C:\HsData\History\ This is the location for Oracle and History runtime data files.
%HS_CONFIG%	The *HsData\History* directory under the History data drive specified during History Services installation. The default path is C:\HsData\History\ This is the location for History configuration data including the History domain list, and the OPC server configuration file.
%HS_LOG%	The *History\Log* directory where History Services software is installed. The default path is C:\Program Files\ABB Industrial IT\History\Log\ This is the location for log files. ABB support may ask for the contents of these files for troubleshooting.

Table 42. Environment Variables (Continued)

Variable	Points To(1)
%USERAPI_HOME%	The *UserAPI* directory where Information Management System Services are installed. The default path is C:\Program Files\ABB Industrial IT\SystemServices\UserAPI This is the location for directories and files related to User API. The examples folder in this directory contains example User API programs.
%RTA_HOME%	The default path is C:\ABB\RTA
%ABB_DATA%	Where all the data that cannot be stored in ABB_ROOT anymore is stored. The path is c:\programdata\ABB\

(1) To look up the actual definition for an environment variable, use the Set command in the Windows Command Prompt. Refer to Figure 360.

Enter the **set** command at the prompt to list all environment variables

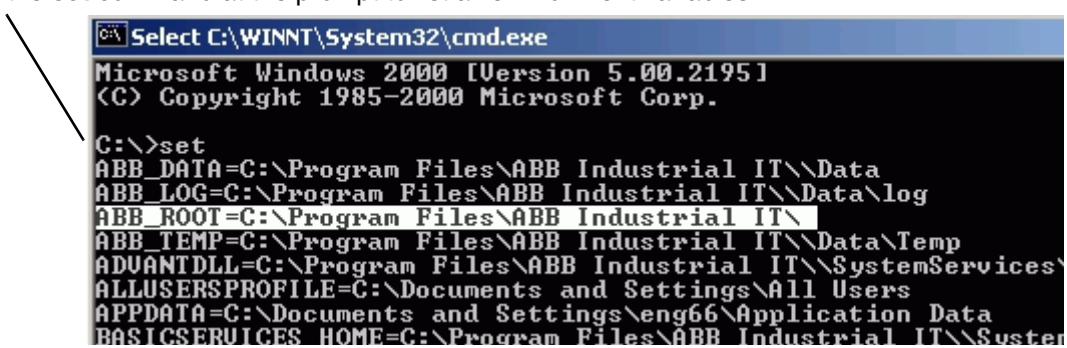


Figure 360. Looking Up Environment Variable Definitions

Linking Information Management History with the 800xA System Trend Function

The Information Management history logs collect from 800xA System operator trend logs which collect directly from OPC data sources. To support this functionality, three links are added to the Basic Platform Service Group for the node where Information Management History Server runs. These are IM History Log, IM

Archive Link, and IM Importer Link. The importer link supports importing log configurations from remote history servers. This linkage is required to consolidate historical process data from Information Management history servers that reside in other Aspect Systems.

These components are automatically added to the Basic Platform Service Group when the Information Management Process Administration Supervision (PAS) is started during post-installation.

Confirm that these components have been added as follows (reference Figure 361):

1. Go to the Service structure in the Plant Explorer workplace.
2. Select the Basic History Service Group for the Information Management node.
3. Click the **Service Group Definition** aspect.
4. Go to the **Special Configuration** tab.
5. Check for the presence of IM History Log, IM Archive Link, and IM Importer **link** in the Supported Links (right) list.
6. If one or more of these required items are not in the Supported Links list, select the item from the Available Links list and click the right arrow button. This puts the item in the supported links list.
7. Click **Apply**.

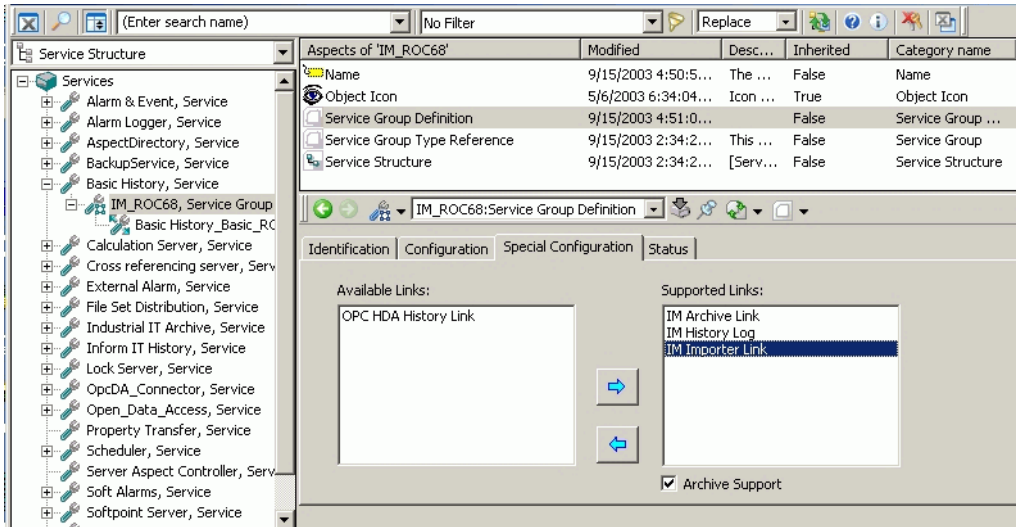


Figure 361. Checking IM History Links for Basic History Service Group

Integrating System Message and History Servers

Certain message-generating applications in the 800xA system, including History Services, SoftPoint Services, AC 800M/C, and AC 400 write their messages to the 800xA System Message Server via application-specific OPC alarm servers. These application-specific OPC alarm servers must be added to the Alarm and Event Service Group for each connectivity server where the applications run. These Service Groups reside under the Alarm and Event Services category in the Service structure.

This configuration should have already been done as part of the Information Management post-installation. Guidelines for verifying this configuration are provided below.

To check (reference Figure 362):

This example shows how to check for the OPC alarm server for history. The OPC alarm server for SoftPoints will reside in a different Service Group (for example, tar101SoftPoint Service Group in Figure 362).

1. Go to the Service structure in the Plant Explorer, find the **Event Collector, Service**, and then select the Service Group for a node where History Services runs (for example, Figure 362).
2. Click on the **Service Group Definition** aspect to display the configuration view, and select the **Special Configuration** tab.

   ABB Hs OPC Alarm Event Server should be present in the Collector Program IDs list. The applicable OPC alarm servers are added when the History Service is started under PAS during the post installation set up for History.

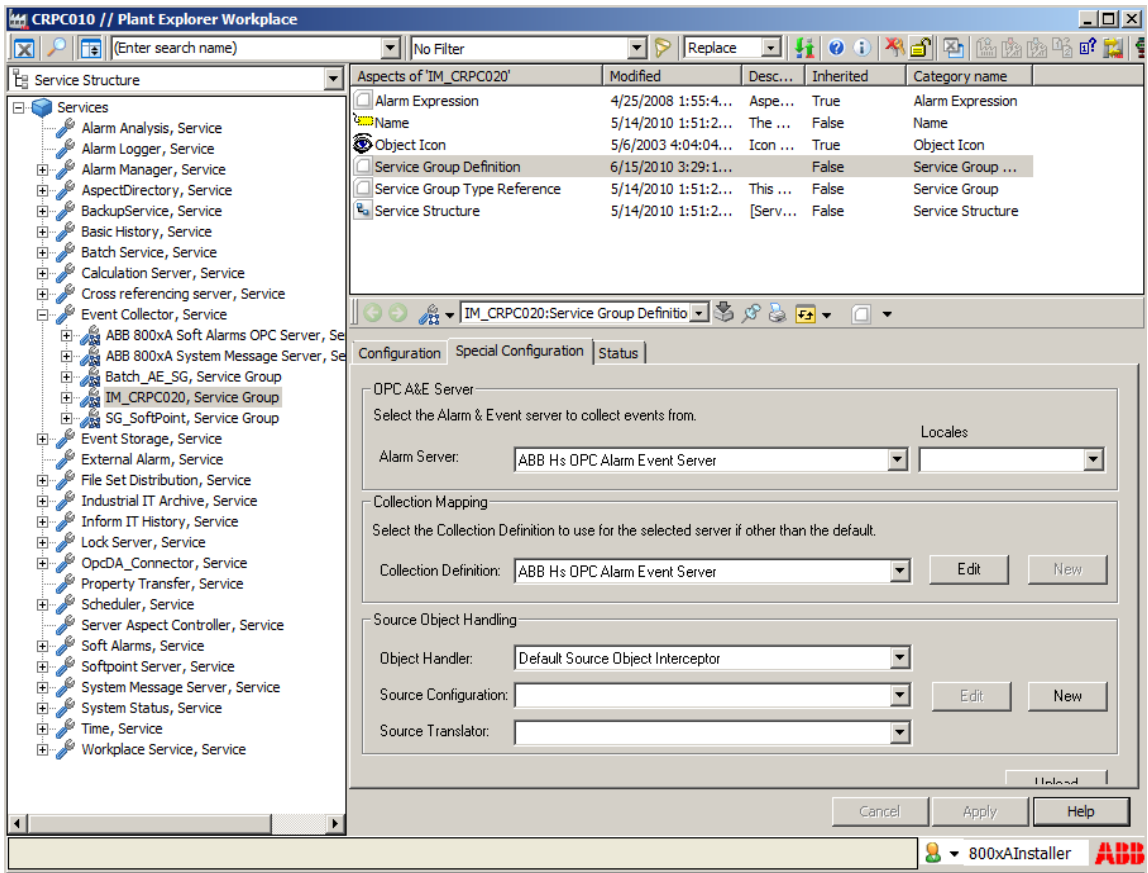


Figure 362. Checking for OPC Alarm Server

Section 14 Open Data Access

This section provides instructions for setting up real-time and historical data access for third-party applications via the Open Data Access (ODA) Server. This server supports client applications that use ODBC data source, for example: Crystal Reports or Microsoft Excel (without DataDirect add-in).

Virtual database tables in the ODA Server map 800xA system data types to ODBC data type. When the client application submits an SQL query toward a virtual table, the ODA Server parses the query, and returns the requested data to the client.

There is one predefined table named **numericlog** to support access to historical process data.

For real-time data access, configure custom tables to expose aspect object properties that need access. These tables are then combined to form one or more virtual real-time databases. The custom-built real-time databases support read and write access. The custom views are used to impose restrictions on data access for certain users.

One predefined table named **generic_da** is provided for real-time data access. This is a read-only table that exposes all properties for all real-time objects.

Client applications which access data via ODA may run locally on the Information Management server where the ODA Server is installed, or on a remote computer client. Remote computer clients required the Information Management Client Toolkit. To install this software, refer to the *Information Management* section of *System 800xA Installation (3BSE034678*)*.

The architecture for Open Data Access is illustrated in Figure 363, and described in ODA Architecture on page 501. When ready to begin configuring ODA access, refer to What to Do on page 502 for guidelines.

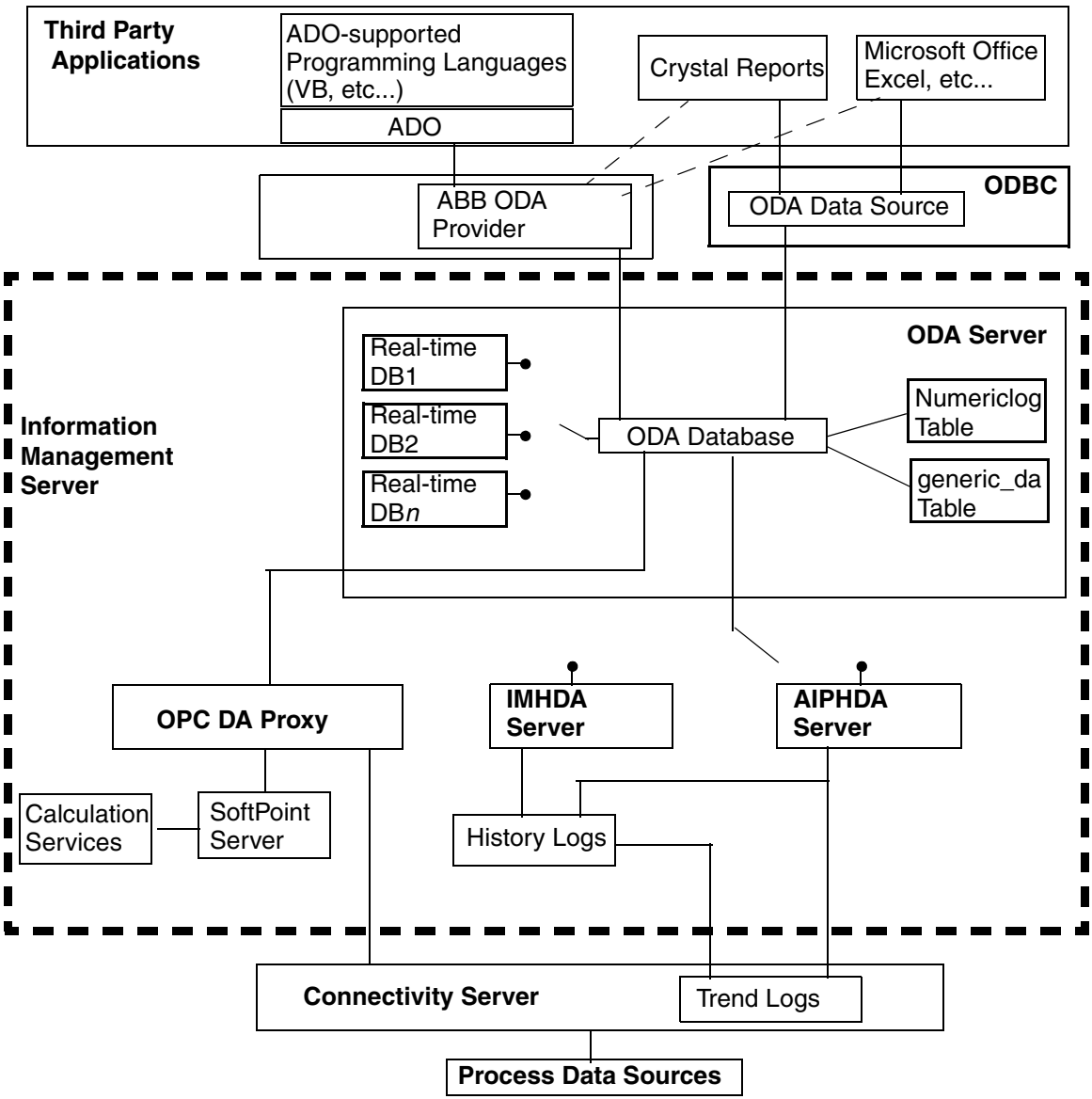


Figure 363. Historical and Real-time Data Access Via ODA Server

ODA Architecture

Client applications that use Open Data Access must connect to a specified ODA database. This is a virtual database which merges the predefined numericlog and generic_da tables with one user-configured real-time database. The real-time database is configured to have one or more custom table definitions which expose selected object properties.

For example, Figure 364 shows two sets of user configured table definitions (motor1, pump1, valve1; and motor2, pump2, valve2). These table definitions are combined to form two real-time database definitions: DatabaseX and DatabaseZ. These two database definitions are then used to create two ODA databases: Database1 and Database2, each of which includes the two predefined tables (numericlog and generic_da), and one custom real-time database definition. The client application may connect to one ODA database at a time.

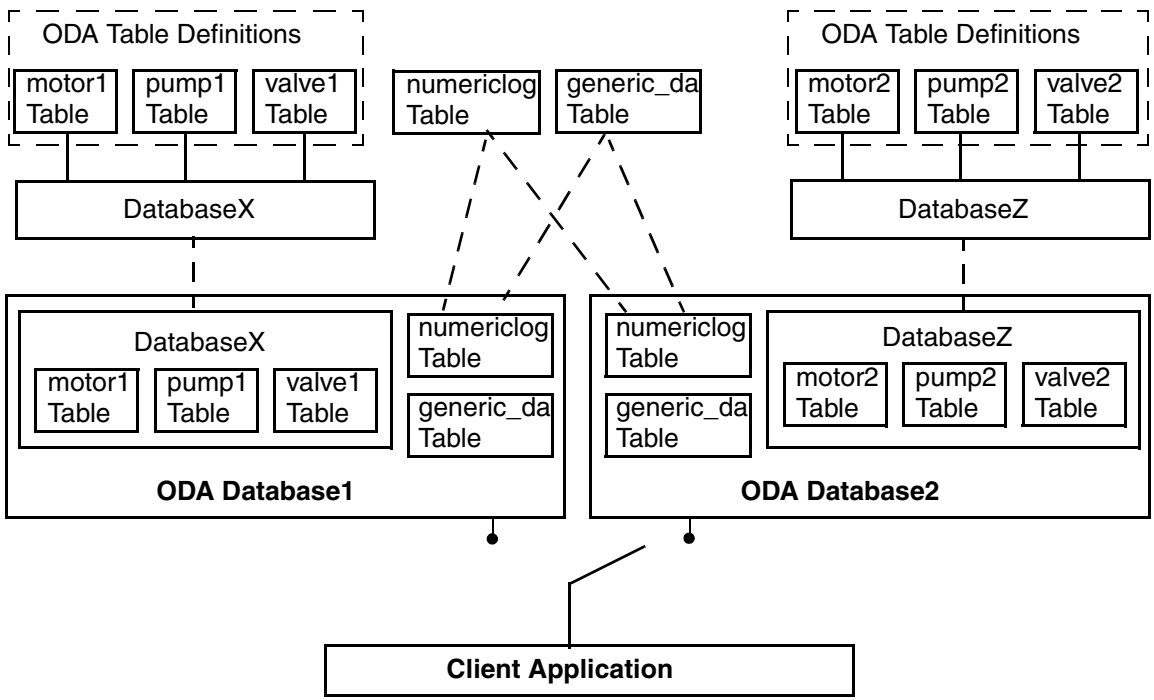


Figure 364. ODA Database Architecture

The contents and operating parameters for the ODA database are specified in a configuration file that was set up using the ODBC Data Source Administrator. This file is used by the ODBC data source. This configuration file specifies:

- Which user-configured real-time database to use for real-time data access. Clients connect to one database at one time.
- Which OPC HDA server to use for historical data access: the 800xA OPC HDA server, or the History Server (IM) OPC HDA server.

  The 800xA OPC HDA server is the default, and is recommended because it supports seamless access to both operator trend logs and history logs. This server also supports the ability to access log attributes.

  The IM OPC HDA server can only access history logs. This server does not support access to log attributes. It is provided primarily to support earlier data access applications configured to use this server.
- For remote client connections, this file also specifies the server's IP address.

Default Set-up

The default connection is to an ODA database named DATABASE1. Without requiring any configuration, this database supports read access to all system objects via the generic_da table, and read/write access to history data via the numericlog table and 800xA OPC HDA server.

The default set up can be changed to use the IM OPC HDA server, and/or specify a different real-time database table which includes user-configured ODA table definitions. Further, additional ODA databases can be created where each one specifies a different real-time database. This is used to connect the client application to a different ODA database, depending on the particular data access requirements.

What to Do

ODA can be used with the default set up (DATABASE1). This supports access via the predefined numericlog and generic_da tables. To use the functionality supported by the custom-built real-time database tables (read/write access, and restricting access on a user-basis), then use the following procedures to configure ODA to meet the data access requirements:

- Configure the virtual database tables for real-time data access. Then combine these tables to form one or more real-time databases. This is described in Configuring Real-time Data Access Via the ODA Server on page 503.

Access to historical process data stored in property logs is supported by a predefined mapping table that does not require configuration by the end-user.

- Set up an ODA database per the data access requirements and specify:
  - Which user-configured real-time database the client will be able to access. Only one can be connected at one time.
  - Whether to use the 800xA OPC HDA server, or IM OPC HDA server for accessing history data.
  - For remote client connections, specify the server's IP address.

  As an option, create additional ODA databases to provide different views to real-time data. Since a client application can only connect to one ODA database at a time, only one real-time database view is available at a time. This set up is performed via the ODBC Data Source Administrator as described in Setting Up the ODA Database on page 523.

Configuring Real-time Data Access Via the ODA Server

The mapping required to support real-time access via the ODA server is implemented in two parts: **ODA Table Definition aspects** and **Database objects**.

ODA Table Definition aspects expose selected object properties as columns in a database table. These aspects also specify whether each column (property) is writable, and whether there will be additional columns for each property's OPC data quality and timestamp.

ODA Table Definition aspects may be added to object types in the Object Type structure, or to instantiated objects in any other structure. Some object types, for example the AC 800M Control Program, do not permit ODA Table Definition aspects to be added. To expose object properties under these circumstances, be sure to add the ODA Table Definition aspects to the instantiated objects.

When adding ODA Table Definition aspects to an object type, each aspect represents one table through which data from all objects of the containing type will be made accessible. The table may be configured to also expose properties for

children of the object type (as defined by a formal instance list). The table will contain a row for each instantiated object created from the object type. An example is shown in Figure 365.

Object Instance	Property 1	Property 2	Property 3	Property *n*
Instance 1				
Instance 2				
Instance *n*				

Figure 365. Example Table for Object Type

Adding ODA Table Definition aspects directly to instantiated objects is used to expose object properties that are unique to a specific instance. When adding the aspect to an object instance, the resulting table will contain a single row, and only those properties that belong to that object may be exposed, Figure 366.

Instance 1	Property 1 Value	Property 2 Value	Property 3 Value	Property 4 Value

Figure 366. Example Table for Object Instance

If properties for children of an object instance need to be exposed, add an ODA Table Definition aspect to those children.

Any number of tables can be defined for the same object or object type to be used in the same or different databases. Once the ODA table definitions are created, assign the table definition aspects to one or more **Database objects** to create real-time databases. This is done in the Library structure. Each database will consist of one or more tables defined by the ODA table definition aspects described above. Tables may be reused in more than one database.

The relationship between database objects and table definitions is illustrated in Figure 367. Two databases are defined. DB1 exposes properties for three object types. DB2 exposes one object type, and also exposes an instantiated object.

The scope of each database is restricted to a specified object root. Typically the object root is specified at or near the top of a structure, for example Control structure, in the Plant Explorer. When adding table definitions in the database for instantiated objects, the instantiated objects must reside under the object root

defined for the database. When adding table definitions that point to an object type, those table definitions will provide access to objects of that type that are instantiated under the object root. To expose objects in more than one structure, configure a dedicated database for each structure.

For example, since DB2 includes a table definition for an object which is instantiated in the Control structure, its object root must be defined within the Control structure (probably at the top). In that case the AI table definition in DB2 will be limited to AI objects in the Control structure. Assuming the object root for DB1 is specified at the top of the Location structure, the tables in DB1 will be limited to PID, DI, and AI objects instantiated in the Location structure.

Multiple databases can be configured to expose different sets of properties for different users. Client applications can only connect to one real-time database at a time.

The database objects must be inserted under the Databases group in the Library structure. Configure as many database objects as required to provide different views of the real-time database.

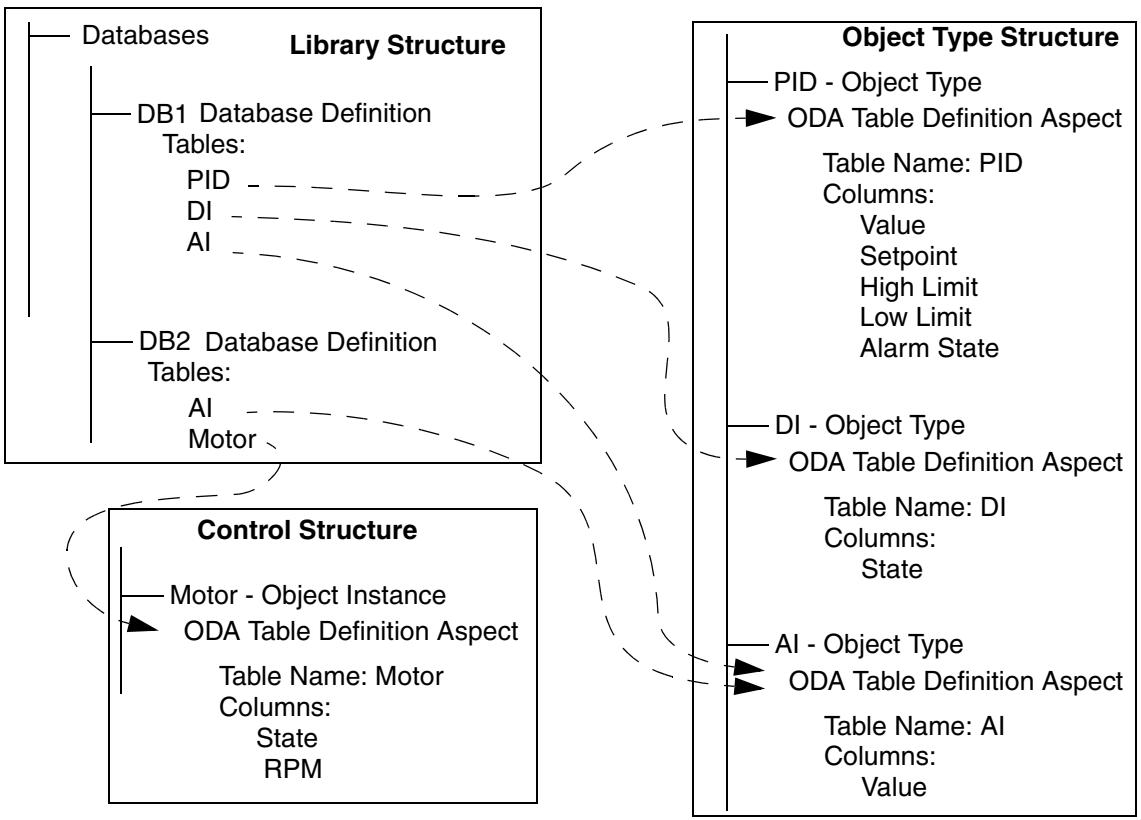


Figure 367. Object Structure for Databases and Database Table Definitions

To implement real-time data access via the ODA server:

- Create ODA table definition aspects for object types and objects whose properties need to be accessed. Refer to Creating ODA Table Definitions on page 507.
- Then create one or more database objects in the Library structure, and add the ODA table definitions to them. Refer to Creating a Database Object on page 518.

Creating ODA Table Definitions

The following procedure show how to add an ODA Table Definition aspect to an object type in the Object Type structure. The same procedure can be used to add the aspect to an object instance in any other structure. Exceptions are noted where applicable.

To add an ODA table definition aspect:

1. Go to the Object Type structure in the Plant Explorer and select the object type where the ODA Table Definition aspect is to be added. Then right-click and choose **New Aspect** from the context menu.
2. In the New Aspect dialog find and select the ODA Table Definition aspect, Figure 368.

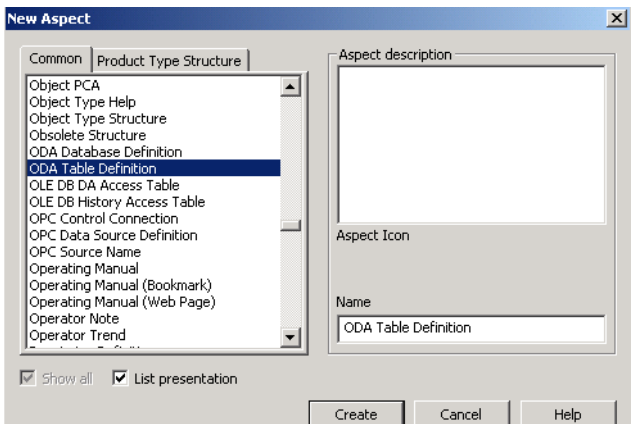


Figure 368. Selecting the Table Definition Aspect

Use either the default aspect name, or enter a different name. The table represented by this aspect will be named from within the Table Definition aspect as described later. Specify a unique Aspect Description to help identify the table. This is particularly important when creating multiple tables with the same table name.

For example, in Figure 368 the Aspect Description is specified as *Engineer's View*. This description will be used as the Table Description when adding tables to the database. If the aspect has no description, the object type description (if

any) will be used. Change the Aspect Description later as described in Using the Description Property on page 516.

3. Click **Create** when finished. This adds the aspect in the object type's aspect list.
4. To show the configuration view for this aspect, select the object type and then click on the aspect, Figure 369.

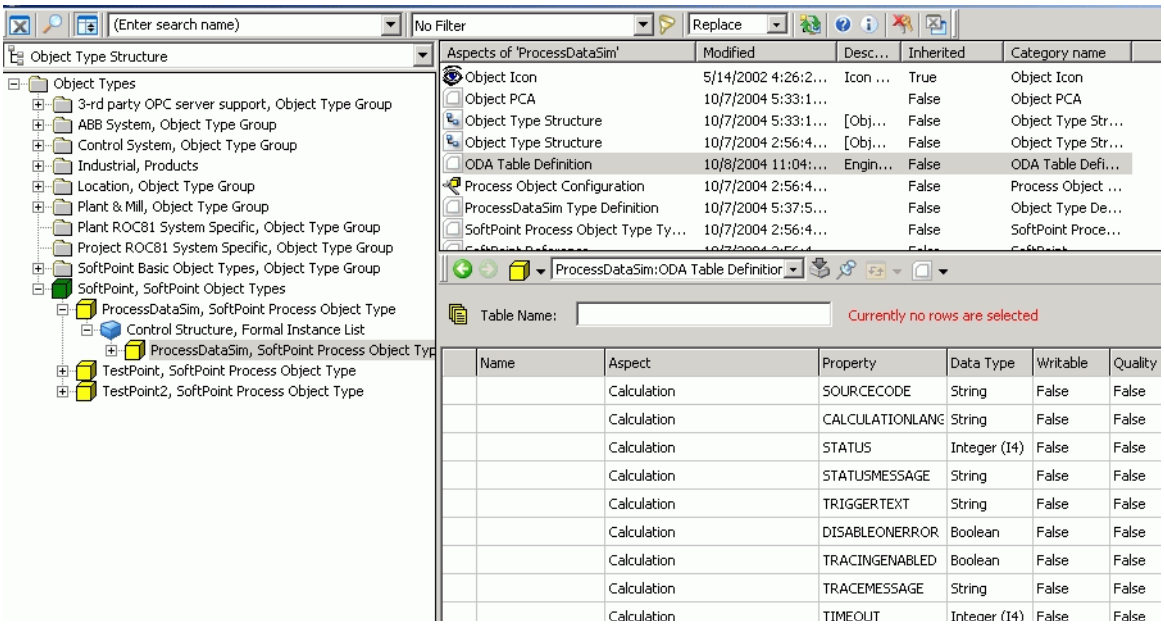


Figure 369. Table Definition Aspect

5. Configure the table definition as described in ODA Table Definition View.

ODA Table Definition View

To use the ODA Table Definition View:

1. Enter a table name.
2. Filter the property list to make it easier to find the properties needed.
3. Select properties to include in the table.

4. Configure selected properties.

For details on these basic steps, refer to:
Table Name on page 509
Filtering the Property List on page 509
Selecting Properties on page 513
Configuring Selected Properties on page 514

Table Name

Specify the table name as shown in Figure 370. This name is referenced in the ODA Database Definition when adding tables to the database (Creating a Database Object on page 518). This is also the name used in SQL queries. More than one table with the same name can be created; however, table names must be unique within the database where they are used.

- If the same table name for more than one table needs to be used, use the aspect's Description property to differentiate tables. If a unique Aspect Description is not already specified, do it later as described in Using the Description Property on page 516.
- If table names that have leading numeric characters are specified (for example *01Motor*, or are completely numeric (*01*), when using those names in queries, the names will have to be entered in double quotation marks ("01Motor")

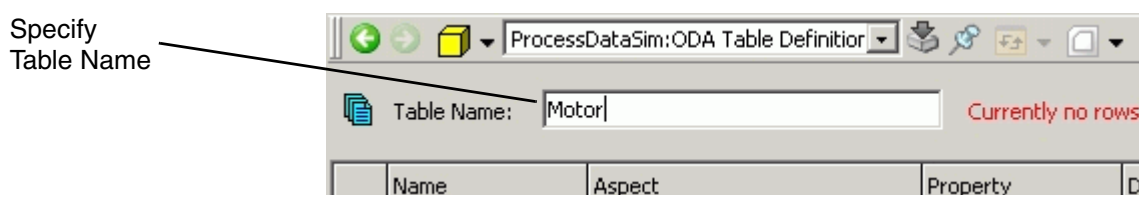


Figure 370. Specifying the Table Name

Filtering the Property List

Use the filtering options to reduce the number of properties presented in the property list to a manageable quantity. Properties are filtered on a group basis. as shown in Figure 371 and as described below:

- **Special Properties**:

– **Object Attributes** - These are not actually properties, but rather attributes of the object such as the object name, hierarchical name, parent, and so on.

Always include object attributes. It is strongly recommended that to always include the :NAME object attribute in the table. This is the most efficient way of referencing an object.

– **Name Categories** - An object can have a number of different names. Each name is assigned as an aspect. The main name is the name associated with the object in the Plant Explorer. This is generally the name used for absolute object references. It is accessed as an object attribute as described above. Other names may be accessed as Name categories, but this is seldom necessary.

- **Aspect Properties** - These are properties related to aspects of the object (or object type). Properties can be filtered on an aspect-by-aspect basis. For example, in Figure 371 the ProcessDataSim object has four aspects, but only one (Calculation) is selected.
- **Child Object Properties** - For object types, the properties of children objects are grouped on an object-by-object basis in a manner similar to the aspect properties. This is not applicable for object instances. For example, in Figure 371, the ProcessDataSim object has two child objects: RealSim and upDown. Each of these objects has aspects and special properties.

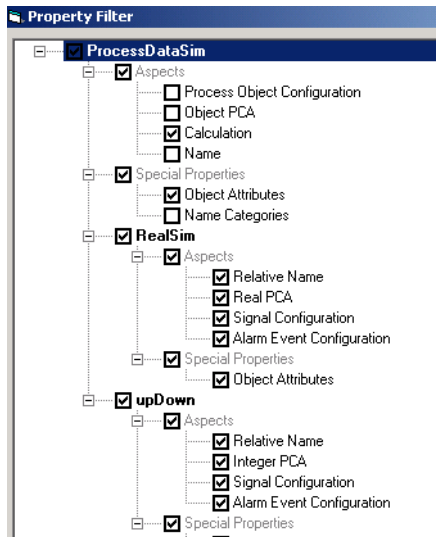


Figure 371. ODA Table Definition - Property Filter

The Aspect column in the property list indicates to which group each property belongs, Figure 372. For aspect properties, the aspect name is shown. Object attributes and name categories are indicated by these respective text strings in the Aspect column: (Object Attributes) or (Name Categories). For properties of child objects, the object relative name is indicated in the Aspect column, in the form: **.***relName***:***aspectName*.

	Name	Aspect	Property	Data Type	Writable	Quality
		Calculation	ENABLED	Boolean	False	False
		Calculation	EXECUTIONTIME	Integer (I4)	False	False
		Calculation	RESULT	Variant	False	False
•	realsim_value	.RealSim:Real PCA	VALUE	Real (R4)	False	False
		.RealSim:Real PCA	ISCONTROLDISABLED	Boolean	False	False
		.RealSim:Real PCA	ISFORCED	Boolean	False	False

Figure 372. Aspect Column

To remove a property group, click the **Filter View** button in the ODA Table Definition aspect, Figure 374. This displays the Property Filter dialog.

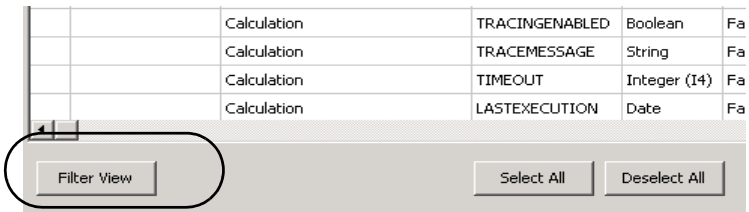


Figure 373. Filter View Button

Property groups that are visible in the list are marked with a check. To remove a group, uncheck the corresponding check box. As this is done, the applicable properties are removed from the property list. Replace any group and corresponding properties that have been removed by reselecting the check box. Figure 374 shows the following groups selected:

- The RealSim object (child of the ProcessDataSim object).
- The Real PCA and Relative Name Aspects of the RealSim object.
- The Object Attributes for the RealSim object.

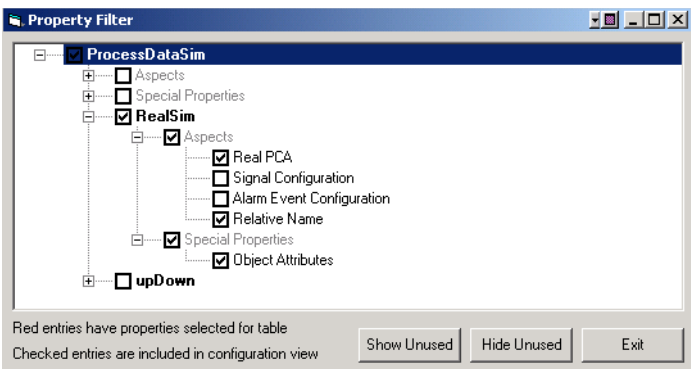


Figure 374. Launching the Filter Dialog

A group that already has properties selected cannot be removed. To remove such a group, unselect the properties first.

Unchecking groups whose properties will not be used significantly reduces the property list. This makes it easier to select the properties to be included in the table.

DO NOT click the Hide Unused button at this point. This hides unselected properties. Since no properties have been selected yet, this will hide all properties.

To proceed, click **Exit** to close the Filter dialog. Then follow the guidelines for selecting properties in Selecting Properties on page 513.

Selecting Properties

To select a property, click the far left column. Selected properties are indicated by black dots, Figure 375. Clicking on a selected property will un-select it.

Remember, it is recommended that the NAME object attribute be selected as an efficient means of referencing the object in data queries. Do not confuse this NAME attribute with other name properties in the property list.

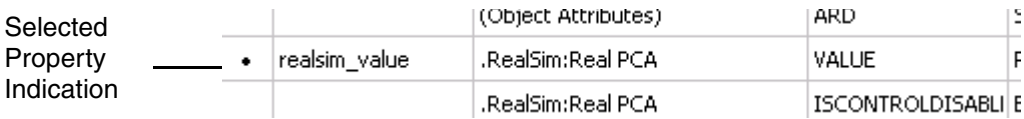


Figure 375. Selecting Object Properties to Expose

When finished selecting properties, simplify the view by showing just selected properties. To do this, right click on the list and choose **Show Selected** from the context menu, Figure 376. The result is shown in Figure 377.

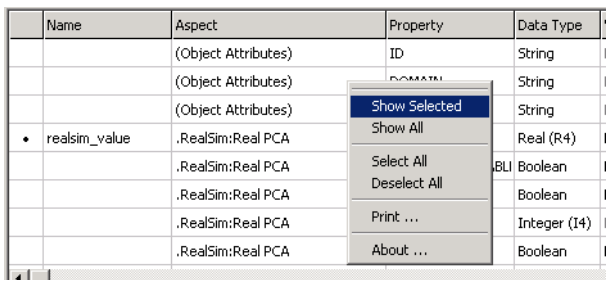


Figure 376. Showing the Selected Properties

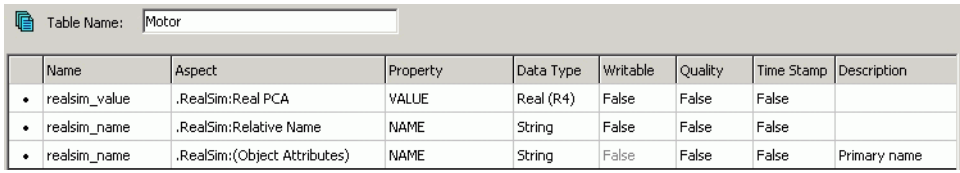


Table Name: Motor

	Name	Aspect	Property	Data Type	Writable	Quality	Time Stamp	Description
•	realsim_value	.RealSim:Real PCA	VALUE	Real (R4)	False	False	False	
•	realsim_name	.RealSim:Relative Name	NAME	String	False	False	False	
•	realsim_name	.RealSim:(Object Attributes)	NAME	String	False	False	False	Primary name

Figure 377. Showing Selected Properties Result

To bring back the hidden unselected properties, click **Show All**. If any mistakes are made selecting the wrong properties, start over by clicking **Deselect All**. To select all properties, click **Select All**. The **Deselect All** and **Select All** functions operate on the properties that are currently visible on the grid. Hidden properties will not be selected or deselected.

Configuring Selected Properties

The columns for selected properties are configurable. This is used to specify the column name for each property in the database table (default is property name), whether a property may be updated, and whether additional columns for the OPC Data Quality and OPC Time Stamp will be included in the table.

These columns are described in Table 43. Refer to Navigation and Configuration Tips on page 516 for guidelines on navigating the property grid.

- The text string in the Name column in this grid will be used as the column name in the database table. The default is to use the Property name. This may result in duplicate column names which is NOT permitted.

  For example, in Figure 377, two properties named *NAME* are selected. When this occurs, change one or all duplicate names. Figure 378 shows the column name changed for one of the NAME properties. In this case, it is recommended the Object Attribute NAME keep the default name.
- If the column name includes a dot (.) character, the column name text string must be delimited with double quotation marks when used in a query, for example:
  SELECT “object1.value” FROM ai

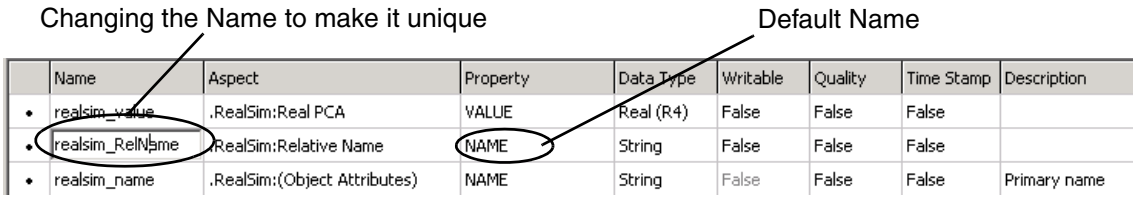


Figure 378. Duplicate Names Changed

Table 43. Database Table Definition Columns

Item	Description
Name	This is the property's column name in the database table. This defaults to the Property name. Column names cannot be duplicated within the table. If duplicate names occur, they must be changed.
Aspect	This identifies the aspect or other category to which this property belongs. Refer to Filtering the Property List on page 509. For aspect properties, the aspect name is shown. Object attributes and name categories are indicated by these respective text strings in the Aspect column: (Object Attributes) or (Name Categories). For properties of children objects, the object relative name is indicated in the Aspect column along with the aspect.
Property	This is the name of the property, object attribute, or Name category.
Data Type	Data Type
Writable	This indicates whether or not the column is writable. Properties that cannot be configured as writable are dimmed. **False** (default) - is not writable **True** - is writable
Quality	This indicates whether to create a column for data quality. The column name in the table will be the data column's name with **_qual** appended. **False** (default) - no column for quality **True** - include column for quality

Table 43. Database Table Definition Columns (Continued)

Item	Description
Timestamp	This indicates whether to create a column for the time stamp. The column name in the table will be the data column's name with **_time** appended. **False** (default) - no column for timestamp **True** - include column for timestamp
Description	This is used to specify an optional description for the column. When the system provides a description, it will be the default.

Navigation and Configuration Tips

The easiest way to navigate the grid is by mouse-clicking the targeted cell, or by the arrow keys to move from one cell to the next. The arrow keys can only be used if the current cell is not selected for updating. A cell is opened for updating by the presence of a depressed text box or combo box.

Once data is entered (or selected) into an updateable cell it will be committed to the current work area by any subsequent operation. To not commit the update, use **Esc** to return to the previous data as long as the cell remains open for updates.

Cells in the Name column must be filled in with unique names for all selected properties. Whenever a property row is selected, if the selected property row does not have an associated Name, the field will be defaulted to the property name.

The Name can be set to the default by selecting the Name cell and double clicking. When all changes to the local work area are completed, commit these changes by clicking **Apply**. At this time, the data is checked to insure that a Table Name is present, and that all selected properties have been assigned unique names.

Using the Description Property

Multiple tables having the same name can be created. For example, two different tables for a motor object type can be created where each table has different property lists. If the same name is to be used for both tables, for example *Motor*, differentiate the tables using the ODA Table Definition aspect's Description property. This will help when selecting the appropriate table when adding tables to a database (Creating a Database Object on page 518).

This does NOT permit adding multiple tables having the same name to the same database. It is merely intended to help select the correct table when adding tables to the database.

To specify the Description property for a Database Table Definition aspect:

1. Select the **ODA Table Definition** aspect, right-click and choose **Properties** from the context menu.
2. Then use the Description field in the Properties dialog to specify a meaningful description that will help identify the table, Figure 379.

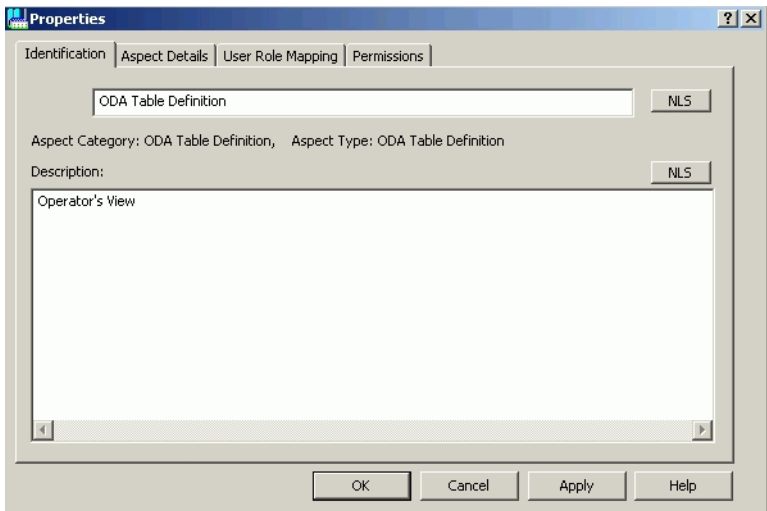


Figure 379. Specifying a Meaningful Description

3. Click **Apply**. The description will be indicated in the aspect list, Figure 380.

Aspects of 'motor'	Description	Inherited	Category name	Type name
ODA Table Definition	Operator's View	False	ODA Table Defi...	ODA Table Definition
ODA Table Definition	Engineer's View	False	ODA Table Defi...	ODA Table Definition
Aspect Category Definition	The base Aspect Cat...	False	Aspect Categor...	Aspect Category
Basic Object Name Hook	Put on the Object Ty...	False	Basic Object Na...	Basic Property Object N...
General Properties		False	General Propert...	Basic Property Properties
motor Type Definition		False	Object Type De...	Object Factory

Figure 380. New Description Defined

Creating a Database Object

This section describes how to create a Database object and add one or more table definition aspects to the Database object.

As an option, rather than creating a new object, use the default database named Database1. This database is empty initially. One advantage to using this database is that the default ODBC data source (ODA Database) is set up to point to Database1. Using this database saves from having to edit the ODBC data source to point to a database.

Database objects must be added under the ODA Databases group in the Library structure. To create a database object:

1. In the Plant Explorer, select the **Library Structure**. Then right-click on **ODA Databases** and choose **New Object** from the context menu.
2. Add the Database object as an **ODA Database** type (selected by default in the New Object dialog). Assign the object a logical name (Ctrl1 for example). This is the name to use when specifying the database in the ODBC configuration file as described in Setting Up the ODA Database on page 523.
3. Click **Create**. This creates a new database under the ODA Databases group, Figure 381.

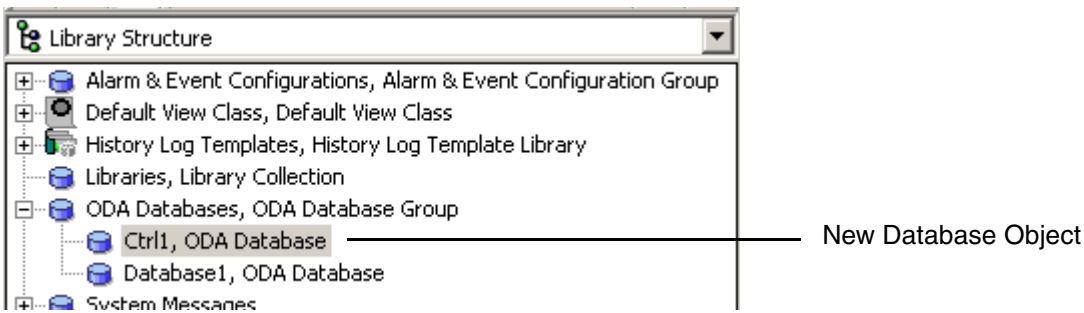


Figure 381. Database Object Added

To show the configuration view for the database, click on the **ODA Database Definition** aspect, Figure 382.

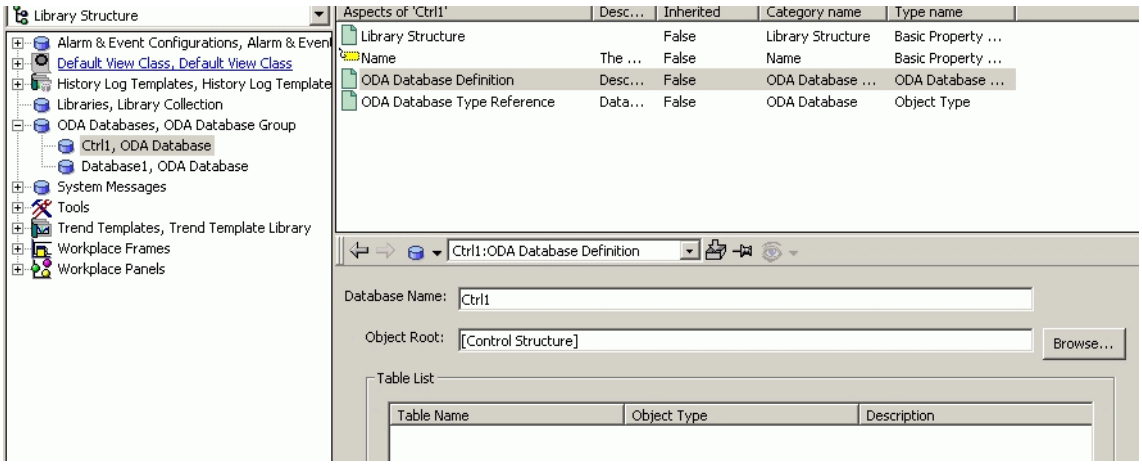


Figure 382. Database Definition View

4. Configure the database as described in Database Definition View on page 519.

Database Definition View

Use this view to add table definitions to the database, and to specify the location of the objects whose data will be accessible from this table. This is used to restrict the database to objects within a specific structure, or object root within a structure.

To add table definitions to the database:

1. Click **Add Tables**. This displays a dialog which lists all existing Database Table Definition aspects, Figure 383. If the list contains any tables that have already been added to this database, those tables will be marked with a dot (**.**). Also tables created for object instances rather than object types are indicated by the **(Instance)** label in the Object Type column.

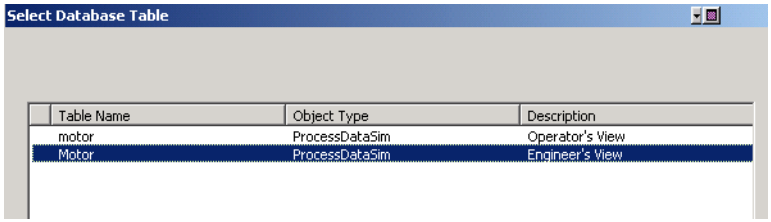


Figure 383. Adding a Table

The bottom portion of this dialog has a check box and buttons to adjust the list of tables, Figure 384. The **Exclude instance tables** check box is used to limit the list to tables defined in the Object Type structure. The **Search Under Root** button is used to limit the list to instance tables that exist under the root object specified in the Object Root field on the ODA Database Definition aspect.

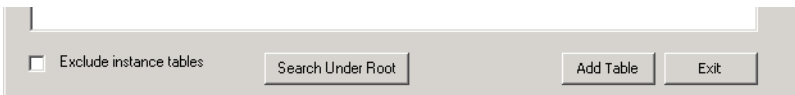


Figure 384. Select Database Table Check box and Buttons

2. Select a table, then click **Add Table**. The added table is listed immediately in the Database Definition view, Figure 385. The dialog remains open so more tables can be added.

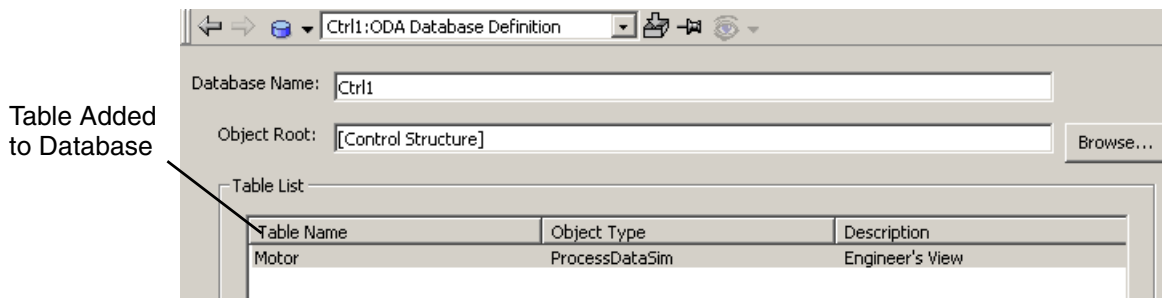


Figure 385. Table Added

3. Repeat step 2 to add as many tables as required.

The database may NOT have more than one table with the same name. Duplicate table names will be indicated in red text. In this case either delete tables with duplicate names, or change table names to make them unique within the ODA database. Refer to Fixing Databases with Duplicate Table Names on page 522.

Tables that were created for object instances rather than object types are indicated by the **(Instance)** label in the Object Type column.

4. Click **Exit** when finished with this dialog.

The scope of the database is determined by the Object Root specification. Only objects under the specified object root will be accessible via this database. This defaults to the Control structure. A different structure can be specified if necessary. Further, the scope can be made more restrictive by selecting an object within a structure. For example, to restrict the database scope to a specific controller within the Control structure. To change the object root specification:

1. Click the **Browse** button for the Object Root field, Figure 386. This displays a browser similar to the Plant Explorer.

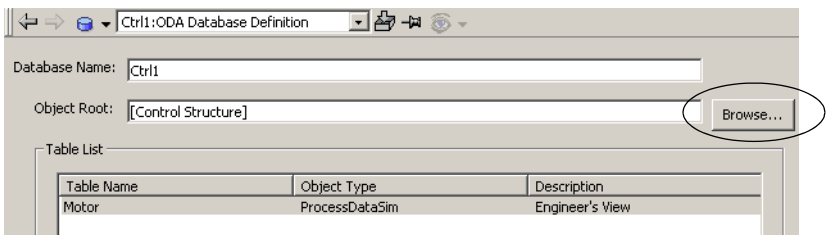


Figure 386. Starting the Browser

2. Use this browser to select the structure (and folder if necessary) where the objects that will populate this table reside, Figure 387.

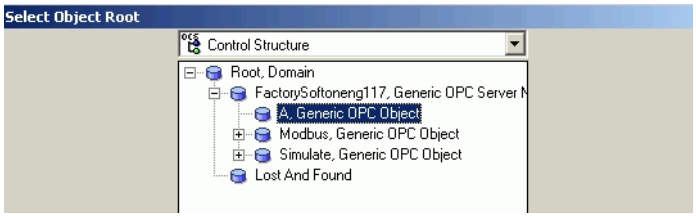


Figure 387. Selecting the Object Root

3. Click **OK** and then **Exit** when finished.
4. Click **Apply** in the Database Definition view to save the definition.

Fixing Databases with Duplicate Table Names

Do not add multiple tables having the same name; however, duplicate table names may occur if a table name is changed after it has been added to the database. This will prevent client applications from connecting to the database, and the database definition must be fixed either by deleting duplicate tables, or by renaming the tables via the Database Table Definition aspect

The condition will be annunciated the next time the Database Definition is opened. Tables having the same name will be indicated in red text, Figure 387.

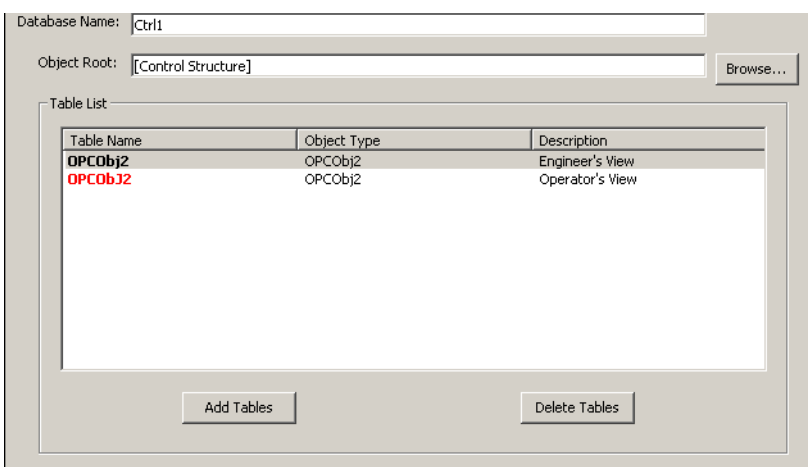


Figure 388. Duplicate Table Names Highlighted in Red

If all tables having duplicate names are required in the database, rename the tables so that each has a unique name. Do this via their respective Database Table Definition aspects as described in Creating ODA Table Definitions on page 507).

If one or more tables are not required, select those tables and then click **Delete Tables**. If a mistake is made, cancel the delete command by clicking **Cancel** BEFORE applying the change. This restores all deleted tables.

Setting Up the ODA Database

Client applications that use the ODA server must be connected to a specific ODA database. This determines the scope of data access for the client application. The ODA database specifies:

- which user-configured real-time database the client will be able to access. Only one can be connected at one time.
- which OPC HDA server to use for historical data access: the 800xA OPC HDA server, or IM OPC HDA server.
- for remote client connections, the server's IP address. A remote client must also change the OpenRDA ODBC 32 setup parameter, and edit the openrda.ini text file.

One ODA database named DATABASE1 is provided as standard. By default this ODA database uses the 800xA OPC HDA server, and connects to a real-time database named Database1, which by default has no assigned table definition aspects. Add table definitions to Database1 as described in Creating a Database Object on page 518.

The default set up supports access via the predefined numericlog and generic_da tables, and to table definitions added to Database1 (if any). To change the default set up, use the IM OPC HDA server, and/or specify a different real-time database.

Further, additional ODA databases can be created where each one specifies a different real-time database. This is used to connect the client application to a different ODA database, depending on the data access requirements.

After successfully creating the object, go to the IM node on which the ODA server is running. Using Windows Explorer, navigate down to the configuration directory for ODA. Generally the path would be:

C:\Program Files\ABB Industrial IT\Inform IT\ODA Provider\Server\cfg

In this directory, make a backup copy of the file oadm.ini. This is the ODA configuration file for the server. After making a copy, edit the original file using a text editor.

Database sources are kept in this file and have the following form:

```
[DataSource_1]

Type=6

DataSourceWorkArounds2=8192

DataSourceLogonMethod=Anonymous

DataSourceIPType=DAMIP

DataSourceIPSchemaPath=C:\Program Files\ABB Industrial
IT\Inform IT\ODA Provider\Server\ip\schema

ServiceName=ABB_ODA

DataSourceName=DATABASE1
```

This is the definition of the default database for ODA. To include the new database that was added in Process Portal, copy the above text, paste it directly after the original, and edit the DataSource_1 and DATABASE1 entries.

The DataSource_ entry is a numerical progression of how many data sources you have. So, to add a second datasource, then the entry should be for DataSource_2. The DataSourceName then needs to point to the new database name, so DATABASE1 would be changed to Ctrl1. So what we should see is now:

```
[DataSource_1]

Type=6

DataSourceWorkArounds2=8192

DataSourceLogonMethod=Anonymous

DataSourceIPType=DAMIP

DataSourceIPSchemaPath=C:\Program Files\ABB Industrial
IT\Inform IT\ODA Provider\Server\ip\schema
```

```
ServiceName=ABB_ODA
DataSourceName=DATABASE1
[DataSource_2]
Type=6
DataSourceWorkArounds2=8192
DataSourceLogonMethod=Anonymous
DataSourceIPType=DAMIP
DataSourceIPSchemaPath=C:\Program Files\ABB Industrial
IT\Inform IT\ODA Provider\Server\ip\schema
ServiceName=ABB_ODA
DataSourceName=Ctrl1;
```

Save the file, and reboot the machine. If further databases are required, the same procedure can be used, keeping in mind that the first modification to the DataSource_x number must increment upwards with each new database.

Remote Client Set-up

Client applications which access data via ODA may run locally on the server where the ODA server is installed, or on a remote computer client. Remote clients require ODA client software, and a remote ODA database must be set up on the remote client computer. A connection for a remote computer client can also be configured which is outside the 800xA system domain. Special setup requirements for remote clients are described in this procedure as applicable.

To set up (or add) an ODA database:

1. Open the Administrative Tools on the Windows Control Panel, then launch the ODBC Data Source Administrator, Figure 389.

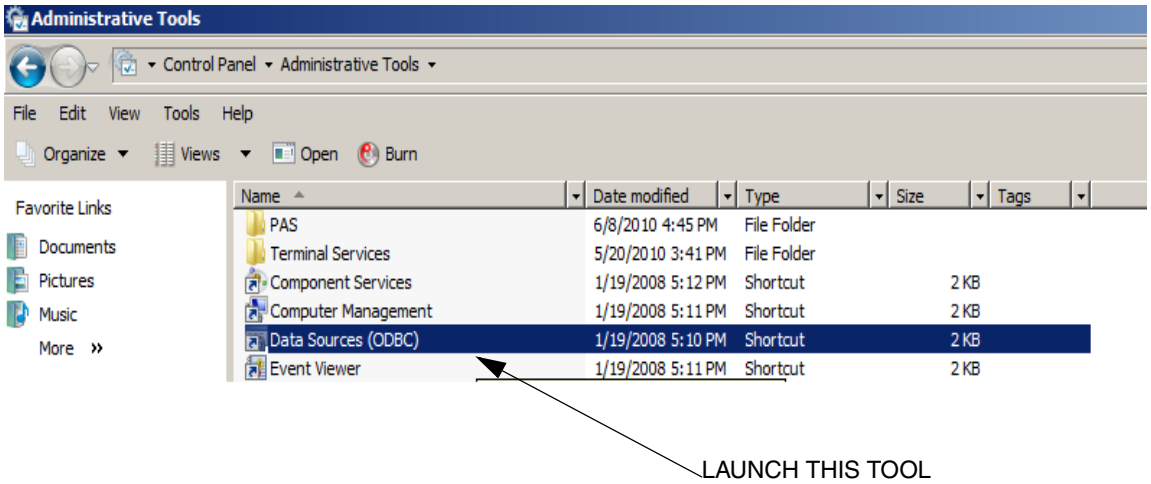


Figure 389. Launching ODBC Data Source Administrator

Go to the **System DSN** tab on the ODBC Data Source Administrator dialog and select the data source named **ABBODA**, Figure 390.

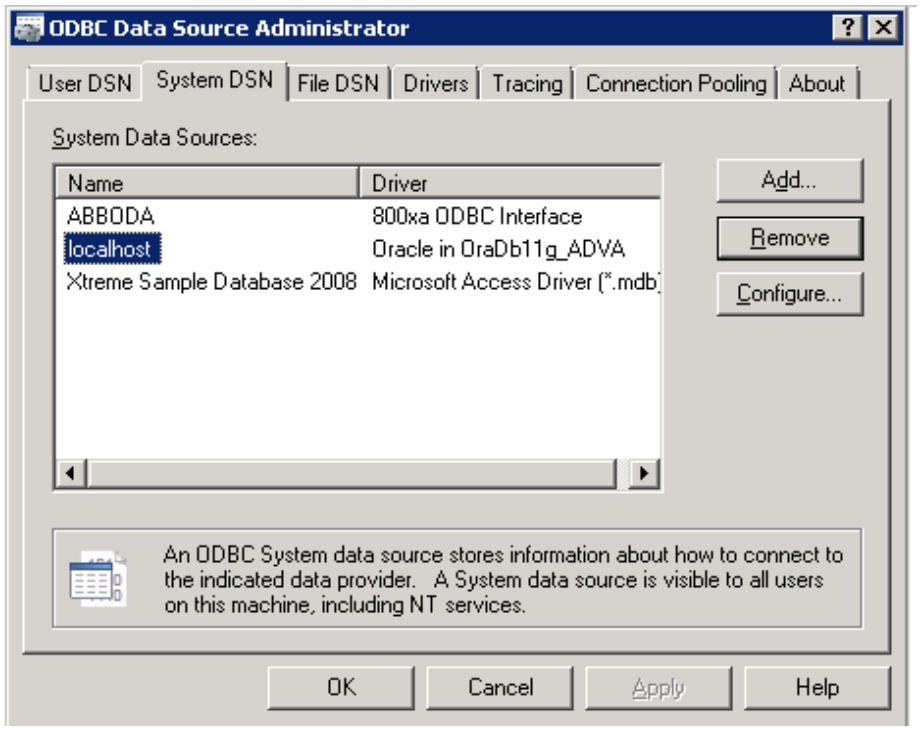


Figure 390. ODBC Data Source Administrator

2. Click **Configure**. This displays the ODBC Data Source Setup dialog. The Database name defaults to ABBODA. Typically, the default is used unless additional databases are configured.

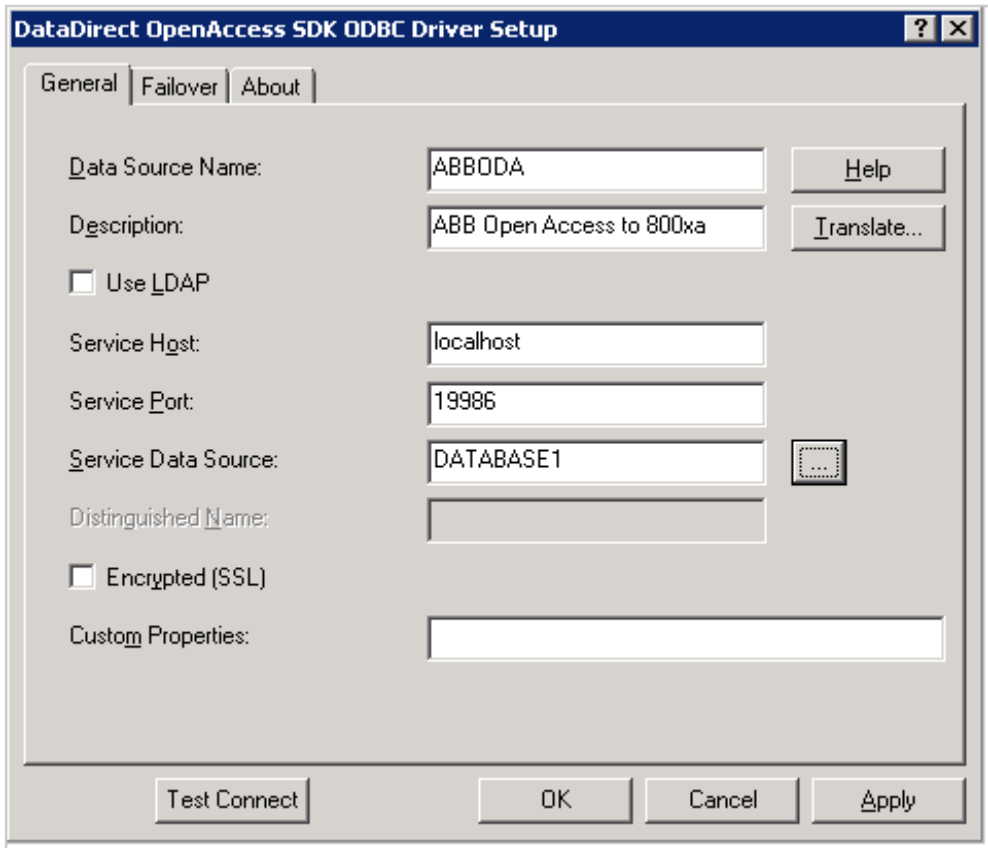


Figure 391. ODBC Data Source Set Up

3. Select the Service Host.
4. Select the Service Data Source. The data source, DATABASE1, that is created is taken as the default. But, the user can configure more as described in Setting Up the ODA Database on page 523.

To get a list of databases, including the default database, click on the selection box. View the example in Figure 392 that lists the default and other configured databases.

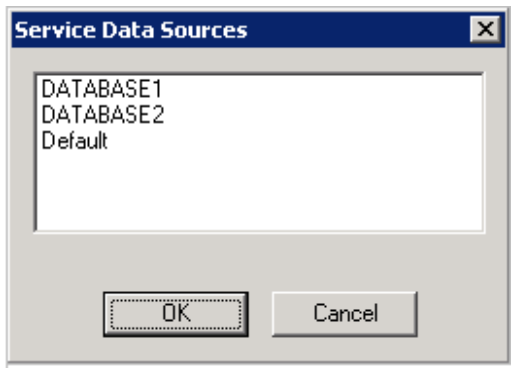


Figure 392. List of Databases

The parameters are described in Table 44.

5. Click **Apply** and then **OK** when finished.

If unable to connect to the selected database, it may be because the database definition is invalid (more than one database table having the same name). In this case, fix the database definition. Refer to Fixing Databases with Duplicate Table Names on page 522.

Table 44. ODA Setup Parameters

Field	Description
Data Source Name	The default data source name.
Service Host	The machine to connect to. By default, it is localhost for this machine. But, this would be changed for the client machines to the name of the machine that ODA is running on.

Table 44. ODA Setup Parameters (Continued)

Field	Description
Service Port	The Service Port should stay as 19986.
Service Data Source	The name of the database to be connected to in the ODA.

By default, the tool mentioned launches the 64 bit version of the ODBC configuration. The client tools run in 32 bit mode, hence the user has to launch the 32 bit version of this tool. The tool is defined by the file, as shown in Figure 393.

Figure 393. Tool that launches the 64 bit version of the ODBC Configuration.

For 64bit machine, you must access the *WOW64* version. All the other configuration remains the same.

Logging Status Information for Numericlog Queries

The ODA Server has a logging function to facilitate debugging. Status information for each query against the Numericlog table is recorded in a log file. Logging is disabled by default. Leave logging disabled unless the system is having problems that need to be provided to ABB support with a record of query activity.

This log does not record status information for queries against the real-time database tables.

To enable (or disable) logging, use the EnableLogging tool and specify a new log file name to maintain individual log files for each client connection.

The contents of the log file is deleted each time a new client connects to the ODA Server. This minimizes the likelihood that the log file will grow too large. To preserve the contents of a log file for a particular client connection, specify a unique file name when logging is enabled.

1. Open the Enable/Disable logging dialog using the **EnableLogging.exe** application file located in the ODA Provider folder. The default path is:

   C:\Program Files\ABB Industrial IT\Inform IT\ODA Provider\bin.

2. To enable or disable logging, Figure 394, click the corresponding radio button. The Log File Name is a default file name. If new logging activity cannot overwrite the existing log, enter a new name (with a .txt extension).

3. Click **OK** to register the changes.

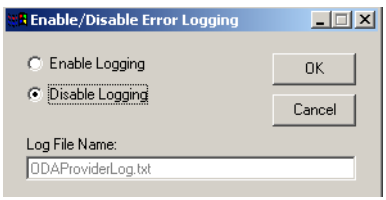


Figure 394. Enable/Disable Logging Dialog

The log file is associated with a specific data provider. The default file name for ABBDataAccess Provider is **ODAProviderLog.txt**. This file is located in the ODA Provider/Log folder. The default path is as follows:

C:\ProgramData\ABB\IM\ODA

To open the log file in the default text editor, double-click, or choose **Open** from the context menu. An example log file is shown in Figure 395.

For each query the log file indicates the data source (table name), and echoes the query. The log file also provides information regarding the status of internal ODA Server components. This information is intended for ABB technical support personnel for debugging.

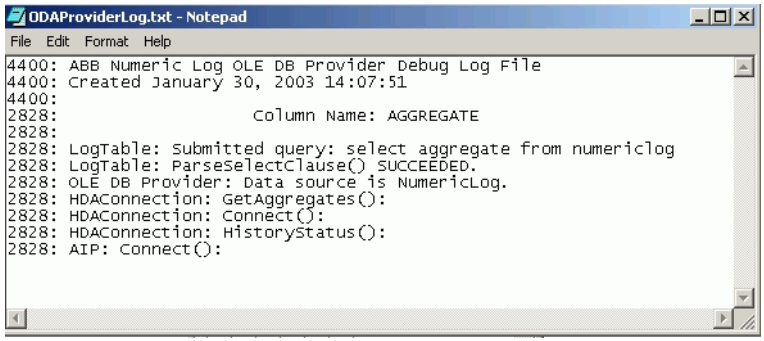


Figure 395. Example Log File

Section 15 Configuring Data Providers

This section describes the configuration of Data Providers.

Data Provider Applications

This section describes how to configure data providers to support access to 800xA system data by Display Services, Desktop Trends, and certain add-in tools in DataDirect (Excel Data Access). The default data provider installation supports most data access requirements. Table 45 provides a quick reference for common data provider applications that require site-specific configuration. Data provider architecture is illustrated in Figure 396. To learn how data providers work, refer to the Overview of Data Provider Functionality on page 537. General procedures such as adding a new data provider, editing an existing data provider, or starting/stopping data providers are covered in General Procedures on page 546.

Third party applications and SQL-based dialogs and scripts for Display and Client Services, for example the SQL Query dialog in DataDirect, use Open Data Access (ODA) rather than these data providers. How to set up ODA to support these applications is described in Section 14, Open Data Access.

Table 45. Data Provider Applications

Application	Type	Description
Access to Oracle data by: • PDL Extractor. • DataDirect and Display Services.	ADO	The ADO data provider named DBA supports access to message logs, PDLs, and Oracle-based numeric logs. **Note:** This data provider requires some site-specific configuration for Oracle access which is done during post installation.
Enabling write access to process, OPC, and numeric log objects.	DCS,OPC, LOG, IMHDA	Lets DataDirect and display clients write to process, OPC or numeric log objects. Refer to Enabling Write Access on page 544.
SQL access to numeric log data OR Improving performance on queries to DataDirect Batch_Trend view	ADO	SQL queries for numeric log data require an ADO data provider that provides Oracle access via the Open Data Access (ODA) database named Database1. The easiest way to create this data provider is to copy the existing ADO data provider, and make the following two changes to the data provider argument list: • Change the -name argument, for example: from DBA to **DBA1**. • Set the -dbname argument to **Database1**. This same data provider may be used to improve the performance of the DataDirect Batch_Trend view. For further details on how to copy and configure data providers, refer Copying an Existing Data Provider on page 546.

Table 45. Data Provider Applications (Continued)

Application	Type	Description
Disallowing write access to the ADO data provider. **Note**: Doing this will disable PDL updates via DataDirect.	ADO	If needed, add the disallow_sql_write argument to the argument list to NOT allow any SQL statements other than those that begin with the SELECT keyword. Refer to Editing Data Provider Configuration Files on page 553.
Aspect System has properties being historized whose name contains the forward slash (/) character, or whose ancestor objects contain that character. One application for this is with Harmony Connectivity Servers where the default separator cannot be used.	OPCHDA	The OPCHDA browser uses the forward slash character as a separator by default, and will not parse objects and properties correctly if they use this character. In this case, the OPCHDA browser must use a different separator character. This is done by adding an optional argument named **-Browser_Separator** to the AIPHDA data provider argument list. For systems where the / character is not used in the property names nor in ancestor object names, no change in configuration is necessary. For further details regarding configuration of this argument, refer to the OPCHDA data provider argument list in Table 54.

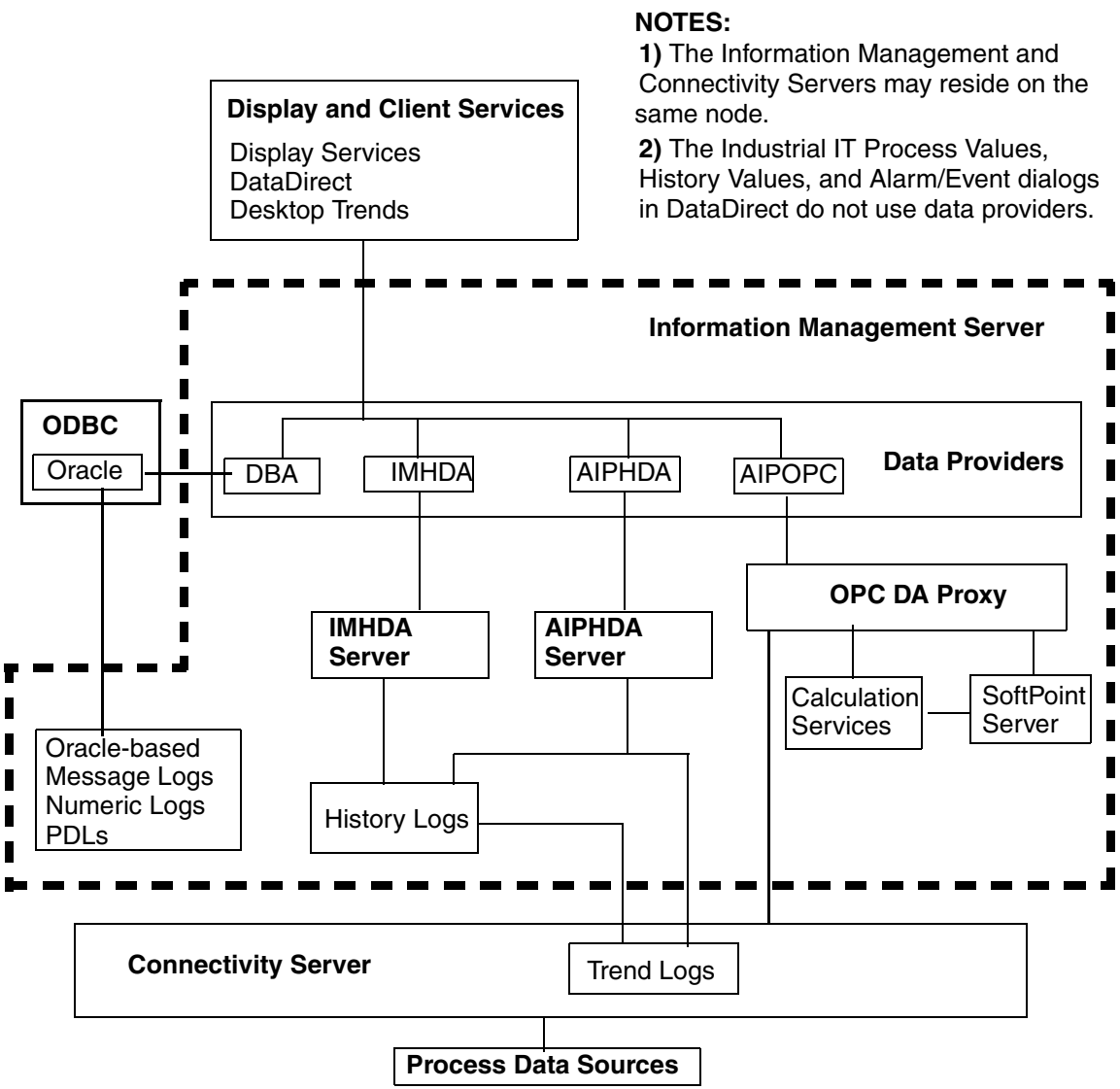


Figure 396. Data Provider Data Access

Overview of Data Provider Functionality

Data access for Display Services, DataDirect, Desktop Trends, and Batch Management PDL PFC display are supported by the ABB Data Service. This service is composed of a service provider and data providers which are installed with Display Services on the Information Management server. The default data providers and supporting services are described in Table 46.

Table 46. Default Data Providers and Supporting Services

ADSS Label(1)	Default Name(2)	Type	Description
AIPHDA IMHDA	AIPHDA IMHDA	OPC HDA	Provides access to historical data from an OPC HDA server. Returns up to 65,534 values per request. Two OPC HDA Data providers are provided by default: **AIPHDA** connects to the 800xA OPC HDA server. This server supports seamless access to trend logs and history logs. It also supports access to log attributes. **IMHDA** connects to the History Server OPC HDA server. This server can only access history logs and is provided primarily to support earlier data access applications configured to use this server. It does not support access to log attributes.
AIPOPC	AIPOPC	OPC DA	Provides access to realtime data from object properties. This data provider connects to the default Aspect System.
ADO	DBA	ADO	Provides access to Oracle data for Display Services, DataDirect, and Batch Management PDL Browser. Site-specific configuration is required during post-installation.
DCSOBJ	DCS	ABB OCS Process Data	Used on Enterprise Historian 3.2/x or earlier for access to realtime process data from Advant OCS objects, for example CCF_CONTIN_LOOP in systems with MOD 300 software, and PIDCON for systems with Master software.
DCSLOG	LOG	ABB OCS History Data	Used on Enterprise Historian 3.2/x or earlier for access to historical process (numeric) data. Returns up to 3200 values per request.
OPC	OPC	OPC	Provides access to realtime data from third party DA 1.0 & 2.0 data access server. Site-specific configuration is required.

Table 46. Default Data Providers and Supporting Services (Continued)

ADSS Label(1)	Default Name(2)	Type	Description
ADO-DEMO	DEMO	ADO	Supports Oracle data access for the Display Services demonstration displays.
COM	COM	Service Provider	Connects data providers with client applications.
DDR	DDR	Display Manager	Manages display requests and command execution on the server. These displays are described in *Information Management Configuration* for *Display Services*.

(1) Used to uniquely identify data provider in the ADSS Configuration tool.
(2) Used by data access applications to reference a specific data provider when there is more than one of the same type.

Typically the default data providers are sufficient to support most data access applications. Certain data access applications require small site-specific adjustments. These are summarized in Table 45.

Additional data providers can be configured as required by the application. For example, there may be a need to create an additional ADO data provider to support SQL access to numeric logs, or access to another third-party database.

How the Service Provider - Data Provider Architecture Works

Desktop clients connect to the service provider on a server which is specified when logging into the client application (for example, DataDirect). The client can use all data providers connected to the service provider. This is illustrated in Figure 397.

By default, only local data providers are connected to the service provider. In this case the client can only access data from the server it is connected to (local server). This functionality is implemented in Node #1.

As an option, data providers can be configured on remote nodes and then connected to the local service provider. This is used to combine data from multiple sources (nodes) on the same screen. This functionality is implemented in Node #2.

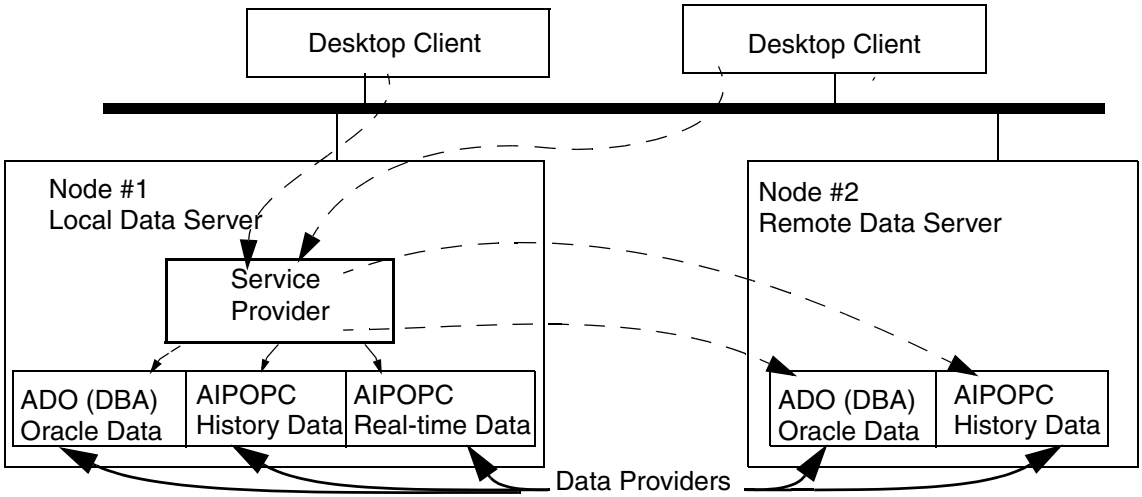


Figure 397. Architecture for Local/Remote Data Providers

To install remote data providers for Oracle Data Access on a platform that is external to an Industrial IT system, consult ABB for additional guidelines.

Uniquely Identifying and Referencing Data Providers

If multiple data providers of the same type need to be configured on a server, ensure that each data provider has a unique identification. The service provider will not allow more than one data provider of the same type to connect unless the data providers are uniquely identified. The unique ID is established by three data provider attributes: type, channel number, and name.

Earlier Enterprise Historian platforms used channel number and type exclusively. Since it is possible for an 800xA system application to have data providers and service providers running different software versions, always consider software version when configuring service and data providers. Table 47 describes the relationship between type, channel number and name for different versions of the service provider and data providers.

If more than one data provider of a given type is not used, the data access applications are not required to reference a specific data provider. Data requests will

be routed via the correct data provider based on the type of data being requested. When multiple data providers of the same type are connected, if a specific data provider is not referenced, the data request will default to the data provider with channel 0. If a different data provider needs to be used, it must be explicitly referenced. Each of the client applications use a different method for referencing data providers.

Display Services uses channel number for all data provider references EXCEPT *data* statements. The default channel is zero (0). When using *data* statements in Display scripts, and a data provider reference is required, use the -name argument.

DataDirect is used to choose whether to use channel number or -name. The default setup is to use -name. This is recommended because channel number does not support some data access functions, including process value update, history update, history retrieval of raw data, and history bulk data retrieval. If channel number is used, the same channel number applies to all data providers. The default channel is zero (0). Select a different channel if necessary.

Batch Management PDL PFC display is hardcoded to use an ADO data provider with the name **DBA**. This is the default name for the ADO data provider. Be sure that the ADO data provider supporting those applications is named DBA, and that no other active data providers use DBA.

Table 47. Operation of Channel Number, Type, and Name Based on Software Version

Versions	Description
Both Service Provider and Data Provider are Version 3.2 or higher	The data provider connects to the service provider with name, type and channel number. When connecting only one data provider of a given type to a service provider, the default channel number and name can be used. When more than one data provider of the same type connects to a service provider, each data provider MUST have a unique name. The service provider does not allow multiple data providers of the same type to connect unless their names are unique. **NOTE:** Assign a unique channel number to each data provider when connecting multiple data providers of the same type to a service provider. This is not required to connect to the service provider; however, channel number is used by many Display scripting functions and DataDirect. Assigning a new channel number also assigns the data provider a unique default name. For example: DSCOBJ -channel 0 is assigned the name **DCS** DCSOBJ -channel 8 is assigned the name **DCS8** Therefore, if a new channel number is assigned, a new name does not have to be assigned. A name other than the assigned default can be used.
Service Provider = 3.2 or higher Data Provider = 3.0 or 3.1	The Data Provider connects to the Service Provider with type and channel number. The Service Provider then uses these attributes to generate a unique name as described above.
Service Provider = 3.0 or 3.1 Data Provider = 3.2 or higher	The Data Provider connects to the Service Provider with type and channel number. The version 3.0 Service Provider ignores Data Provider names.

Data Service Supervision Configuration Dialog

Configuration and management of services providers and data providers is via the ABB Data Service Supervision (ADSS) Config dialog. To display the ADSS Config

dialog, from the Windows task bar, choose **Start>Settings>Control Panel** and then double-click on the ADSS Config Icon, Figure 398.

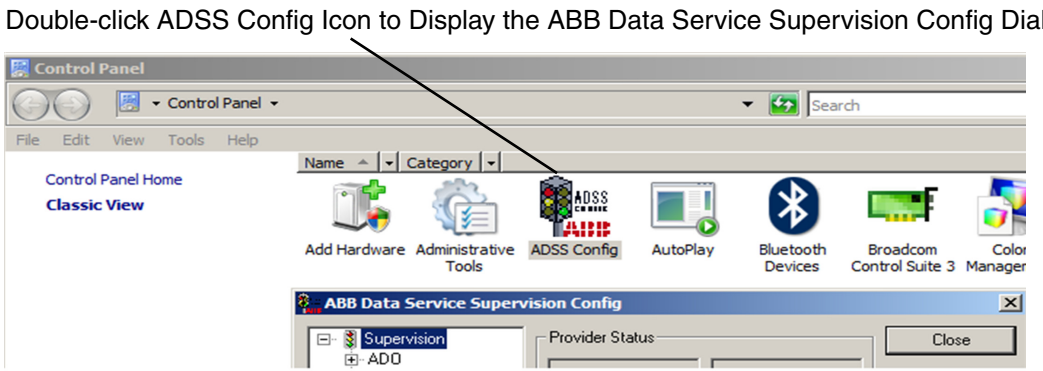


Figure 398. Opening the ABB Data Services Supervision Configuration Dialog

The left side of this dialog is a browser that is used to select the data providers and their respective attributes, Figure 399. Use the +/- button to show/hide data provider's and attributes.

The Supervision process for ADSS supervises all service and data providers. ADSS is automatically started as a Windows process. To monitor and control ADSS, use the Services icon in the Windows control panel. ADSS starts the service provider and data providers according to their respective configurations. The service provider (COM) is always started first. For details on configuring a data provider's attributes, refer to Editing Data Provider Configuration Files on page 553.

When a provider is selected in the browser, the following information is displayed for the selected provider:

Provider, Status	Indicates the selected provider's label and status. This label is only used in this dialog. This is NOT the name that desktop client applications use to reference the data provider.
Argument Value	Indicates the current value for the selected attribute. This field can be edited to modify the service's configuration. These attributes are described in Table 49.

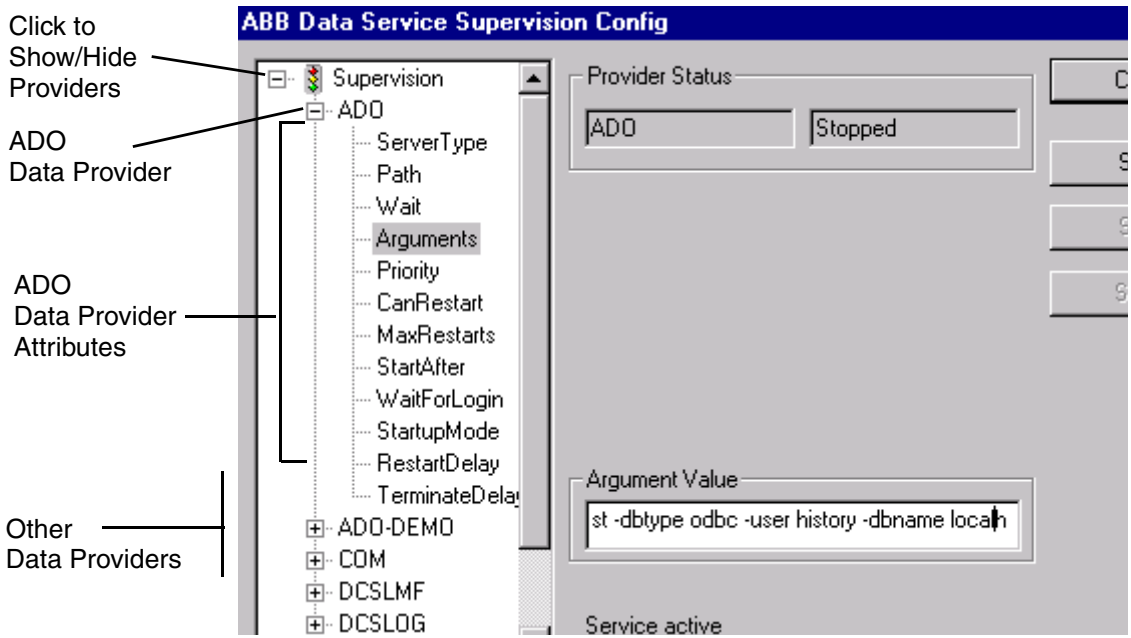


Figure 399. Showing the Data Services Under Supervision

The context (right-click) menu, Figure 400, provides access to the following provider management functions: Adding a Data Provider, Starting and Stopping Providers, Deleting a Data Provider, and Adding Arguments to a Data Provider.

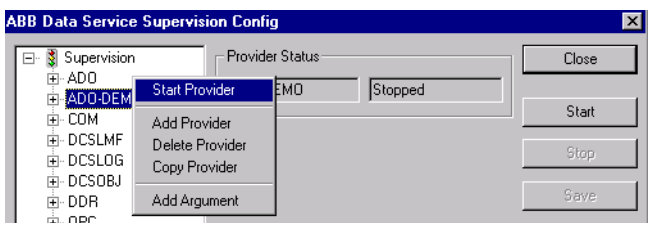


Figure 400. Context Menu for Data Services

Some functions are not available while the selected provider is running. In this case, stop the provider to see the complete menu.

Enabling Write Access

Write access lets Display and DataDirect users write to history logs, process objects, and OPC objects. This is also required to support Bulk History Data Export.

Write access must be enabled on two levels:

- For history logs:
  - **at the system level** - the corresponding (LOG or IMHDA) data provider's argument list must include the -Allow_Log_Write argument. This argument is present by default.
  - **at the user level** - the LOGWRITE user preference must be set to **True** (enabled). This user preference is disabled by default. To enable it, refer to Configuring User Preferences for Write Access on page 544.
- For process objects:
  - **at the system level** - the data provider's argument list must include the -Allow_Object_Write argument. This argument is present by default.
  - **at the user level** - the OBJECTWRITE user preference must be set to **True** (enabled). This user preference is disabled by default. To enable it, refer to Configuring User Preferences for Write Access on page 544.
- For OPC objects:
  - **at the system level** - the data provider's argument list must include the -Allow_Object_Write argument. This argument is present by default.
  - **at the user level** - the OBJECTWRITE user preference must be set to **True** (enabled). This user preference is disabled by default. To enable it, refer to Configuring User Preferences for Write Access on page 544.

Configuring User Preferences for Write Access

To do this:

1. Start the display client: **Start>Programs>ABB Industrial IT 800xA>Information Mgmt>Display Services>Client>Client**.
2. From the menu bar, choose **User > Preferences,** Figure 401.

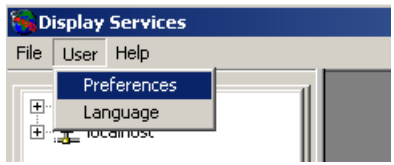


Figure 401. Accessing User Preferences

3. From the User Preferences dialog, Figure 402, Click + to expand the RIGHTS section, select OBJECTWRITE or LOGWRITE, or both, and check the check box to change the preference value to **True**.

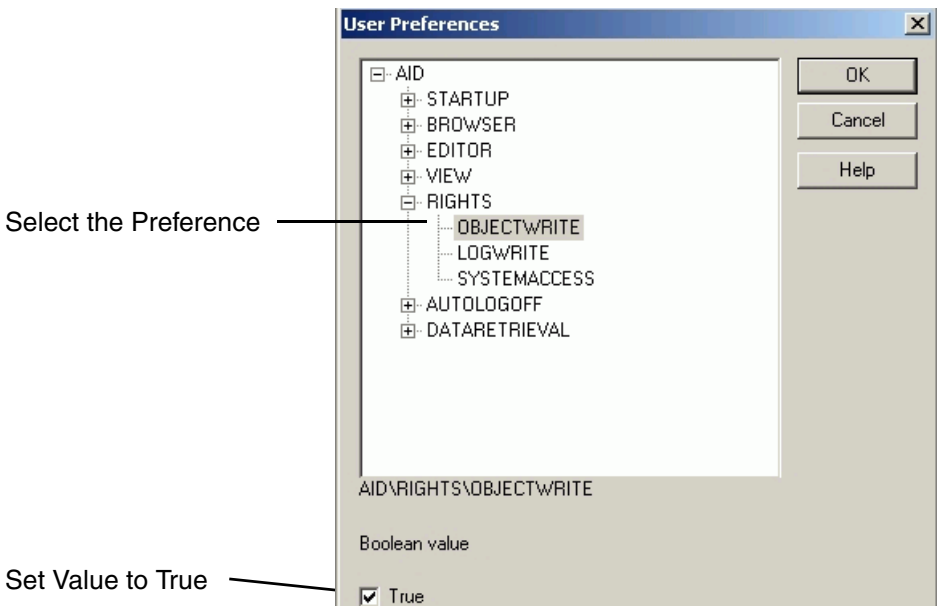


Figure 402. Enabling the OBJECTWRITE User Preference

When the LOGWRITE or OBJECTWRITE user preference is enabled for a user, make sure that user has a secure password. Change the password if the default password is still valid.

4. Reboot the computer to register the changes. Any changes made to user preferences will not take affect until the computer is restarted.

General Procedures

This section describes the following procedures:

- Adding a Data Provider on page 546.
- Editing Data Provider Configuration Files on page 553.
- Starting and Stopping Providers on page 566.
- Deleting a Data Provider on page 567.
- Adding Arguments to a Data Provider on page 567.

These procedures are supported by the Data Service Supervision Configuration Dialog.

Adding a Data Provider

There are various ways to add a data provider:

- The easiest way to add a new data provider is to copy an existing one. This procedure is described in Copying an Existing Data Provider on page 546.
- Considerations for adding a remote data provider are described in Adding a Remote Data Provider on page 548.
- A data provider can also be added from scratch. This procedure is described in Adding a New Data Provider from Scratch on page 550.

To edit a data provider, refer to Editing Data Provider Configuration Files on page 553.

The ADO data provider supports connection to third-party databases: MDB, ODBC, or GENERIC. When adding an ADO data provider for connection to a third party database, ensure the database is properly installed and configured. For details, refer to the applicable OEM user's guide.

Copying an Existing Data Provider

To copy an existing data provider:

1. Start the ADSS Config function from the Windows Control Panel.
2. Use the browser to select the data provider to be copied.

If the data provider to be copied is currently running (Status = Started), then stop it. Right-click and choose **Stop Provider** from the context menu.

3. After the data provider to be copied is stopped, right-click again and choose **Copy Provider** from the context menu. This displays a dialog for specifying the unique name for the new data provider, Figure 403.

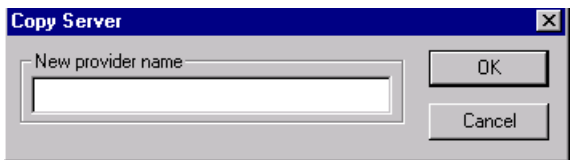


Figure 403. Copy Server Dialog

4. Enter a new name, then click **OK**. This adds the new data provider in the browser under Supervision. This name labels the data provider in the browser. This name DOES NOT uniquely identify the data provider for data access by display elements.

5. Configure any data provider attributes as required.

   If both the original data provider and the copy are being connected to the same service provider, then either the **-name** or **-channel** arguments, or both MUST be edited. The Service Provider will not allow connection of more than one Data Provider of the same type, unless they have different names.

   Assigning a new channel number also assigns a new name. Therefore, if a new channel number is assigned, then a name is not required. The assigned default name does not have to be used.

   To access the **-channel** or **-name** argument, click **Arguments**. Then enter a new channel number or name in the Argument Value field, Figure 404.

If the **-channel** or **-name** argument is not included in the argument list, then add the argument to the list.

Typically, defaults can be used for all other attributes. For details regarding attributes, refer to Table 49 in *Editing Data Provider Configuration Files* on page 553.

6. Click **Save** to save the new data provider.

7. Click **Start** to start the new data provider.
8. To verify proper start-up and review any status errors, refer to Checking Display Server Status on page 567.

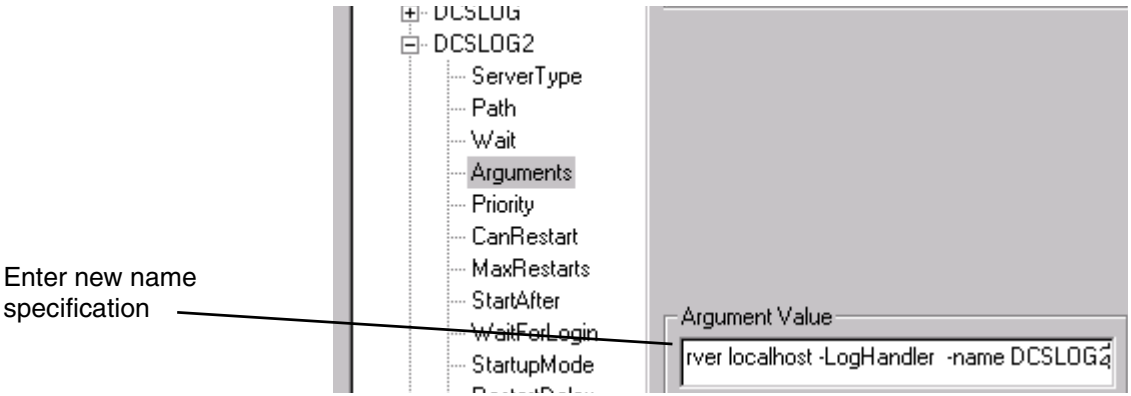


Figure 404. Configuring Data Provider Attributes

Adding a Remote Data Provider

This procedure is basically the same as Adding a Data Provider. The only difference is that data providers are installed and configured on the remote computer. Also, there are special considerations for the **-server** and **-name** arguments. Refer to the guidelines below:

- The **-server** argument must be set to the node name of the computer where the corresponding service provider resides.
- If a local Data Provider of the type being added remotely is already connected to the Service Provider, then be sure that the **-name** argument assigned to the remote Data Provider is unique.

  Assigning a new **-channel** number also assigns a new name. Therefore, if a new channel number is assigned, then a name is not required. The assigned default name does not have to be used.

The following two examples illustrate the procedure for installing and configuring a remote data provider. In both cases a remote data provider is required to connect a computer client to a third party (Oracle) database on a remote node. In example 1, Display Services software is installed on both the local display server, and the

remote computer where the Oracle database is installed. In example 2, the computer where the Oracle database is installed does not have Display Services software.

Example 1 - Remote Computer Has Display Services

Two computers, *node01* and *node02,* both have Display Services installed. The display clients connected to node01 need to access data from an Oracle database on node02.

On node02, add an ADO-type data provider and edit the new data provider's configuration file, Figure 405. The **-server** argument is set to node01 and the **channel** argument is set to a unique channel number as related to node01 (for example, channel 1).

ADSS must be stopped on node02. ADSS will be restarted when the computer is restarted. The new ADO data provider will be initiated and establish a connection to the service provider on node01.

To verify proper operation, from node01 check the server status. The newly configured data provider should be visible as a connected data provider to the service provider on node01.

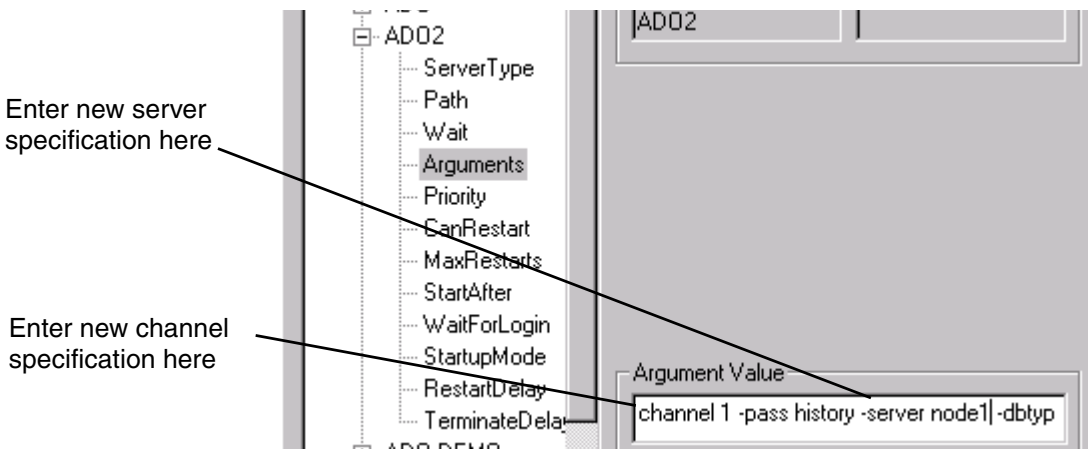


Figure 405. New Entry to Configuration File on node02

Example 2 - Remote Computer DOES NOT Have Display Services

This scenario is essentially the same as described for example 1, except the remote computer does not have Display Services software installed. Be sure to load the Display Server software on the remote computer before configuring data providers on that computer. The Client software does not need to be loaded.

In each case the license for the additional data providers is located with the service provider.

Since the computer is being used just as a data provider, only the data providers are required for the local ADSS. These services must be removed:

- COM.
- DDR.
- And any other data provider that is not required.

Then modify or add the data providers and channels required. ADSS will be started when the computer is re-started. The data providers are then started by ADSS. The data provider will establish communication to the service provider configured.

To verify configuration, on the node with the service provider, check the server status to verify that the data providers are established.

If an application client specifies an invalid channel, the application defaults to channel 0. To prevent defaulting to channel 0 in the event of an invalid channel specification, be sure that no data providers are assigned channel number 0. To do this, change the channel number of the default data providers, and do not use channel number 0 for any additional data providers.

Adding a New Data Provider from Scratch

To add a new data provider:

1. Invoke the ADSS Config function from the Windows Control Panel. To do this, choose **Start>Settings>Control Panel**. Then double-click on the ADSS Config Icon. This displays the Data Service Supervision Configuration dialog.
2. Use the browser to right-click on any item (for instance Supervision), and choose **Add Server** from the context menu, Figure 406.

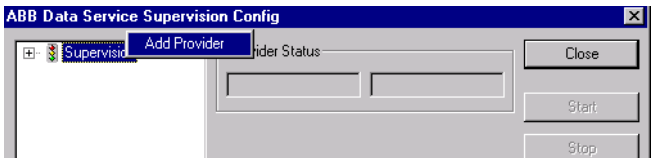


Figure 406. Adding a Data Provider

This displays the New Server dialog, Figure 407.

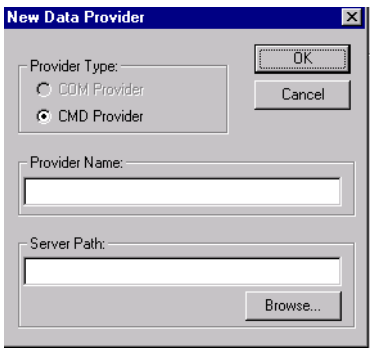


Figure 407. New Server Dialog

Server Type in this dialog does NOT refer to the type of data provider, but rather the method by which Windows handles this process (COM or Command Line). Currently, only command line processes are supported. Thus the Server Type is fixed as **CMD** and cannot be changed.

3. Enter a unique name for the data provider in the Server Name field. This is the name that the data provider will be identified by in the browser under Supervision. This name DOES NOT uniquely identify the data provider for data access by display elements.

4. Enter the path to the executable (.exe) file for the data provider. Enter the path directly in the Server Path field, or use the **Browse** button. The executable files reside in: c:\Program Files\ABB Industrial IT\ Inform IT\Display Services\Server\bin.

   The file names correspond to the process names as indicated in Table 48.

Table 48. Default Data Providers

Data Provider Type	Description	Process Name	Default Name
Historical data in 800xA system	Historical process data (numeric log data).	ADSdpOPCHDA.EXE	AIPHDA
OPC Data	Provides access to realtime data from either DA 1.0 & 2.0 data access server.	ADSdpOPC.EXE	OPC
Third-party Database	Third-party database on specified host.	ADSdpADO.EXE	DBA
ABB OCS Realtime Process Data	Realtime process data from Advant OCS objects, for example CCF_CONTIN_LOOP in systems with MOD 300 software, and PIDCON for systems with Master software.	ADSdpDCS.EXE **Note:** Same executable is used for ABB OCS real time and historical data.	DCS
ABB OCS History Data	Historical process data (numeric log data).	ADSdpDCS.EXE	LOG

5. Click **OK** when finished. This adds the new data provider under Supervision in the browser.
6. Configure any data provider attributes as required.

   If a data provider of the type being added is already connected to the service provider, then edit either the **-name** or **-channel** arguments, or both. These arguments uniquely identify the data provider. The name is NOT automatically updated when a new data provider name in the New Server dialog (step 2) is specified. The Service Provider will not allow connection of more than one Data Provider of the same type, unless they have different names.

   Assigning a new channel number also assigns a new name. Therefore, it is not necessary to edit the name unless something other than the assigned default name is preferred.

   Refer to the procedure for *Copying an Existing Data Provider* to learn how to access the **-name** and **-channel** arguments.

Editing Data Provider Configuration Files

The configurable parameters for each service are listed under the service name in the browser. Show/hide the configurable attributes for a specific service by clicking the corresponding +/- symbol.

The attribute values can be read while the service is still running; however, the service must be stopped before one or more of its attributes can be changed. To stop a service, refer to Starting and Stopping Providers on page 566.

Table 49 describes the configuration parameters for the display, management, and data access services. Common arguments for all data providers are described in Table 50. Specific arguments based on service type are described in Table 51 - ADSspCOM, Table 52 - ADSdpDDR, Table 53 - ADSdpADO, and Table 55 - ADSdpDCS.

When changing arguments to a data provider, add them at the beginning of the line or only have a single parameter at the end of a command line. Changes to data provider arguments cannot be saved when an additional argument is added at the end of the line.

Table 49. Common Configuration Parameters for Windows-based Server

Parameter	Description
Server Type	This does NOT refer to the type of data provider, but rather the method by which Windows handles this process (COM or Command Line). Currently, only command line processes are supported. Thus the Server Type is fixed as CMD and cannot be changed.
Path	This is the path to the executable (.exe) file for the data provider. The path is initially defined when the data provider was added as described in Adding a Data Provider on page 546. The executable files reside in: c:\Program Files\ABB Industrial IT\Inform IT\Display Services\Server\bin. The file names correspond to the process names as indicated in Table 46.
Wait	This is the time to wait on the initial attempt to start up. The value is in milliseconds. The default is 100.

Table 49. Common Configuration Parameters for Windows-based Server (Continued)

Parameter	Description
Arguments	Arguments depend on the data service type. For arguments common to all data providers, refer to Table 50. For specific arguments for each type of service, refer to: Table 51 for ADSspCOM.EXE Table 52 for ADSdpDDR.EXE Table 53 for ADSdpADO.EXE Table 54 for ADSdpOPCHDA.EXE Table 55 for ADSdpDCS.EXE Table 56 for ADSdpDCS.EXE Table 57 for ADSdpOPC.EXE
Priority	The default is -1.
CanRestart	This specifies whether or not the service is restarted by supervision if the service goes offline. **TRUE**- Data Service can be restarted. This is the default value. **FALSE** - Data Service can not be restarted.
MaxRestarts	This is the number of times the data provider should attempt to re-start if its server has stopped. The default is 3.
StartAfter	This specifies the place in the starting order for a Data Service. COM is always the first data service to be started, followed by DDR. After DDR, all other services are given equal precedence.
WaitForLogin	This specifies whether or not the Service will wait for login before it is started. **TRUE**- Do not start until client is logged in. Data Service can be restarted. **FALSE** - Start immediately. Do not wait for log in. This is the default value.
StartupMode	This specifies whether or not the service is started automatically by supervision, or must be started manually via the ABB Data Service Supervision Config dialog. **AUTOMATIC** - Start automatically by supervision. This is the default value. **MANUAL** - Must be started manually via the dialog.

Table 49. Common Configuration Parameters for Windows-based Server (Continued)

Parameter	Description
RestartDelay	This is an optional parameter to specify the interval between re-start attempts (in milliseconds). The default is 100.
TerminateDelay	This specifies the time to delay (in milliseconds) when the service is asked to stop. The default is 0.

Table 50. Arguments Common to All Data Providers

Argument[1]	Description
-name	This name is used by certain data access applications to identify a specific data provider when there are more than one data provider of that type. Specifically, the -name argument is used by DataDirect when the Use Channel Numbers option is disabled, and by display scripts that use the Data statement. When more than one Data Provider of the same type is connected, each Data Provider MUST have a unique name. Assigning a new channel number also assigns a new name. Therefore, if a new channel number is assigned, the name does not have to be edited unless the assigned default name is not preferred.
-channel	This also uniquely identifies a specific data provider when there are more than one data provider of that type. The -channel argument is used by DataDirect when the Use Channel Numbers option is enabled, and by display scripts that do not use the Data statement. Default data providers are assigned CHANNEL 0. If an application client specifies an invalid channel, the application defaults to channel 0. To prevent defaulting to channel 0 in the event of an invalid channel specification, be sure that no data providers are assigned channel number 0. To do this, change the channel number of the default data providers, and do not use channel number 0 for any additional data providers.

Table 50. Arguments Common to All Data Providers (Continued)

Argument[1]	Description
-server	This is the name or IP address of the service provider. For a local node installation, this should be the name of the local node. To not have the data provider start automatically when the corresponding Display server is started, set the value for Server in *Arguments* to NoNode.

NOTE:

1. For specific arguments for each type of service, see Table 51 for ADSspCOM.EXE, Table 52 for ADSdpDDR.EXE, Table 53 for ADSdpADO.EXE, Table 54 for ADSdpOPCHDA.EXE, Table 55 for ADSdpDCS.EXE, Table 56 for ADSdpDCS.EXE, Table 57 for ADSdpOPC.EXE.

Table 51 describes the arguments that are unique for the *ADSspCOM.EXE* data provider. This data provider also has arguments common to all data providers as described in Table 50.

The no reverse DNS setting on the COM process is set to -UseReverseDNS 0 by default so DNS lookup does not prevent data provider communication. COM arguments are not reset upon upgrade of a system and defaults may have to be reset.

Table 51. ADSspCOM.EXE Arguments

Argument	Description
-server	The host name or TCP/IP address where the service provider runs. For a local node installation, this should be the name of the local node. To not have the service provider start automatically when the corresponding ADSS is started, set the value for Server to NoNode.
-port	The TCP/IP socket port number. Three sockets are used, starting from the one specified. Default: 19014 Range: 1000<= n <= 65000

Table 51. ADSspCOM.EXE Arguments (Continued)

Argument	Description
-AliveTimeOut	Used to disconnect clients if they are quiet for the time specified. It can also be used on data providers to indicate how long to wait to disconnect a data provider from the main service provider. Default: 60 Range: 0 <= x <= ... Unit: Seconds Remote data providers to an Information Management node may not recognize that the Information Management node has been restarted causing the data provider to wait for the period defined by this argument before reconnecting. Normally, the Data Provider on the remote node detects that the Com Port has been closed and kills itself so that it can reconnect after the node restarts. In some cases, the Data Provider doesn't detect the Com Port closing and hangs until the -AliveTimeout argument expires causing the Data Provider to wait before killing itself so that it can reconnect. Set this argument to a reasonable period (about 300 seconds) so the node has time to reboot. This is especially important for the AIPHDA and IMHDA data providers if their values retain an old default value of 6000 seconds.
-license_key & -license_text	These are the license key and text as entered when the display server is installed, or when the license was updated. The key is entered in quotes.
-UseReverse DNS 0	ADSspCOM is configured to avoid DNS lookup (default for a new system). This is used when there are DNS server problems. For example, ADSspCOM may time out accessing the DNS server and clients trying to connect will time out as their time out is likely to be less than the DNS server time out.

Table 52 describes the arguments that are unique for the *ADSdpDDR.EXE* data provider. This data provider also has arguments common to all data providers as described in Table 50.

Table 52. ADSdpDDR.EXE Arguments

Argument	Description
-server	The host name or TCP/IP address where the service provider runs. For a local node installation, this should be the name of the local node. To not have the data provider start automatically when the corresponding ADSS is started, set the value for Server to NoNode.
-port	The TCP/IP socket port number. Three sockets are used, starting from the one specified. Default: 19014 Range: 1000<= n <= 65000
-datapath	This is the path to the directory where the display files reside. The default is: c:\Program Files\ABB Industrial IT\ Inform IT\Display Services\Server\Data
-Allow_System_Access	This argument is required to support bulk data export for client applications such as DataDirect. This argument is provided by default. In addition to this argument, the SYSTEMACCESS user preference must be enabled. This is the default setting for this preference.

Table 53 describes the arguments that are unique for the *ADSdpADO.EXE* data provider. This data provider also has arguments common to all data providers as described in Table 50.

Table 53. ADSdpADO.EXE Arguments

Argument	Description
-server	The host name or TCP/IP address where the service provider runs. For a local node installation, this should be the name of the local node. To not have the data provider start automatically when the corresponding ADSS is started, set the value for Server to NoNode.
-port	The TCP/IP socket port number. Three sockets are used, starting from the one specified. Default: 19014 Range: 1000<= n <= 65000

Table 53. ADSdpADO.EXE Arguments (Continued)

Argument	Description
-dbtype	This is the type of database to which this data provider connects. The choices are: **MDB**, **ODBC**, and **GENERIC**. For MDB and ODBC databases, be sure to also configure the corresponding username and password. For GENERIC databases, be sure to specify the connection string (-constr argument).
-dbname	This is the name or full path to the database. For dbtype **MDB**, use the full path. For dbtype **ODBC**, use the ODBC name.
-user	This is the user name for logging into the database. This is required when the dbtype is **ODBC** or **MDB**.
-password	This is the password for logging into the database. This is required when the dbtype is **ODBC** or **MDB**.
disallow_sql_write	Add this argument to the argument list to NOT allow any SQL statements other than those that begin with the SELECT keyword.
-constr	This is the full connect string. This is only required when the dbtype is **GENERIC**.
-CmdTMO <n>	This is the command timeout in seconds (*n* = number of seconds). The default is 30 seconds
ReConnINT <n>	This is the reconnect interval in seconds (*n* = number of seconds). The default is 0. When n = 0, reconnect is disabled, otherwise the provider will attempt connecting the database at the specified interval, until successful connection or the reconnect timeout period (if specified) is exceeded.
-ReConnTMO <n>	This is the Reconnect timeout period in seconds (*n* = number of seconds). The default is 0. When n = 0 there's no timeout and the provider will keep on trying to connect to the database. If the reconnect timeout period is exceeded, the provider will terminate.

Table 53. ADSdpADO.EXE Arguments (Continued)

Argument	Description
FatalErrors "<n1>;<n2>;....;<nX>"	This is a listing of error codes which should be treated as Fatal (for example, missing connection to the database). If such an error occurs, the data provider will terminate. The format to specify error codes is a list of numbers separated by semicolons (;) and enclosed in quotation marks. For example: `-FatalErrors "11; 567;-26"` Entering an asterisk in place of error codes will cause the provider to terminate at any error. For example: `-FatalErrors "*"`
-ReturnAllErrors "TRUE|FALSE"	Refer to *ReturnAllErrors Argument* below.

ReturnAllErrors Argument

Normally, on errors the ADO provider returns an array containing:

ADS Error Header	DP Error Header	DP Error Text
ADS specific Error code	Last DP specific Error code	Last DP specific Error Text

Some data base providers are capable of generating several errors as a result of one request. If 'ReturnAllErrors' is FALSE (default), only the last error code/text is returned. If 'ReturnAllErrors' is TRUE, the returned array will contain all the error codes/texts encountered:

ADS Error Header	DP Error Header	DP Error Text
ADS specific Error code	1st DP specific Error code	1st DP specific Error Text
ADS specific Error code	2nd DP specific Error code	2nd DP specific Error Text

Example:

The connection to the SQL server fails. If ReturnAllErrors FALSE (default):

ADS Error	ADO Error	Text
-106	11	[Microsoft][ODBC SQL Server Driver][TCP/IP Sockets] General network error. Check the network documentation.

If ReturnAllErrors TRUE:

ADS Error	ADO Error	Text
-106	10054	[Microsoft][ODBC SQL Server Driver][TCP/IP Sockets] ConnectionWrite (send()).
-106	11	[Microsoft][ODBC SQL Server Driver][TCP/IP Sockets] General network error. Check the network documentation.

Table 54 describes the arguments that are unique for the *ADSdpOPCHDA.EXE* data provider. This data provider also has arguments common to all data providers as described in Table 50.

Table 54. ADSdpOPCHDA.EXE Arguments

Argument	Description
-OPCHDAServer <xxx>	ProgID for OPCHDA server. If the AIPOPCHDA Server is being used, the Server ProgID is **ABB.AfwOpcDaSurrogate**. If the IMOPCHDA Server is being used, the Server ProgID is **HsHDAServer.HDAServer**.
-OPCHDAServerHost <xxx> H	Host name of the computer where the OPCHDA server resides. This is only required if OPCHDA server is on a remote host.
-ALLOW_LOG_WRITE	If this argument is not specified, clients cannot write to history logs. In addition to this argument, the LOGWRITE user preference must be enabled. The default setting for this preference is disabled. For details on how to set this user preference, refer to Configuring User Preferences on page 601.

Table 54. ADSdpOPCHDA.EXE Arguments

Argument	Description
-Browser_Separator	This applies to the AIPHDA data provider and is used when an Aspect System has properties being historized whose name contains the forward slash (/) character, or whose ancestor objects contain that character. One application for this is the Harmony Connectivity Server where the default separator cannot be used. The OPCHDA browser uses the forward slash character as a separator by default, and will not parse objects and properties correctly if they use this character. In this case, the OPCHDA browser must use a different separator character. The supported separator characters are "\","-" , "," and ".". As an example, to declare the backslash as the separator, add the argument as follows: **-Browser_Separator ** For Aspect Systems where the / character is not used in the property names nor in ancestor object names, no change in configuration is necessary. **NOTE:** A data provider will support access to a data source that uses a special browser separator, but it will not support the other data source. For example, the data provider will not support both a special browser separator configured for 800xA for Harmony process opbjects and data sources that use the default browser separator such as AC 800M process objects. Creating multiple data providers, one for each different required browser separator, and selecting the applicable data provider as required in the client application will solve this problem.

Table 55 describes the arguments that are unique for the *ADSdpDCS.EXE* data provider. This data provider also has arguments common to all data providers as described in Table 50.

Table 55. ADSdpDCS.EXE Arguments

Argument	Description
server	The host name or TCP/IP address where the service provider runs. For a local node installation, this should be the name of the local node. To not have the data provider start automatically when the corresponding Display server is started, set the value for Server to NoNode.
-port	The TCP/IP socket port number. Three sockets are used, starting from the one specified. Default: 19014 Range: 1000<= n <= 65000
-name	This is the assigned name for the data provider.
-Allow_Object_Write	If this argument is specified, clients can write to process objects. If this argument is not specified, write transactions are not allowed. In addition to this argument, the OBJECTWRITE user preference must be enabled. The default setting for this preference is disabled. For details on how to set this user preference, refer to the section on configuring user preferences in Configuring User Preferences on page 601.

Table 56 describes the arguments that are unique for the *ADSdpLOG.EXE* data provider. This data provider also has arguments common to all data providers as described in Table 50.

Table 56. ADSdpLOG.EXE Arguments

Argument	Description
server	The host name or TCP/IP address where the service provider runs. For a local node installation, this should be the name of the local node. To not have the data provider start automatically when the corresponding Display server is started, set the value for Server to NoNode.
-port	The TCP/IP socket port number. Three sockets are used, starting from the one specified. Default: 19014 Range: 1000<= n <= 65000
-name	This is the assigned name for the data provider.
LogHandler	
-Allow_Log_Write	If this argument is specified, clients can modify existing log entries, and add new log entries. If this argument is not specified, write transactions are not allowed. In addition to this argument, the LOGWRITE user preference must be enabled. The default setting for this preference is disabled. For details on how to set this user preference, refer to the section on configuring user preferences in Configuring User Preferences on page 601.

Table 57 describes the arguments that are unique for the *ADSdpOPC.EXE* data provider. This data provider also has arguments common to all data providers as described in Table 50.

Table 57. ADSdpOPC.EXE Arguments

Argument	Description
-OPCprogID <xxx>	Logical name for OPC server. For remote OPC servers such as Symphony, Freelance, or SattLine, use the Data Provider Configuration Wizard to get this information.
-OPChost <xxx>	Host name of the computer where the OPC server resides. This is only required if OPC server is on a remote host. Do not specify if OPC server and data provider are on the same host. For remote OPC servers such as Symphony, Freelance, or SattLine, use the Data Provider Configuration Wizard to get this information.
-Allow_Object_Write	If this argument is specified, clients can write to OPC objects. If this argument is not specified, write transactions are not allowed. In addition to this argument, the OBJECTWRITE user preference must be enabled. The default setting for this preference is disabled. For details on how to set this user preference, refer to the section on configuring user preferences in Configuring User Preferences on page 601.
-EventUpdateRate <nnn>	There is no event subscription on OPC, but a low update rate can simulate this. Default: 250, Unit: milliseconds
-EventDeadBand <nnn>	Percent of difference between low and high engineering units range of analog values. (EU values are specified on the OPC server). Default: 0.00, Unit: Percent
-CacheWrites	If true, the OPC initialization and setup for write commands will never be removed, resulting in better performance for following writes to the same object. This uses more memory. Default: Not specified not cached). Range: Specified | Not specified

ADO Data Provider for Oracle Access

Table 58 indicates the ADO data provider configuration to support Oracle access. If another ADO data provider is being added to support SQL access for Oracle-based numeric logs, copy the existing data provider with the configuration shown in Table 58, and change the -name argument (for example: DBA2).

Table 58. ADO Data Provider Set-up for Oracle Access

Argument	Value
-port	**19014**
-channel	**0**
-pass	**<Password for the History account>** **Note**:The password for the history user should be created by the user while creating the database instance. The History account does not have default password.
-server	**localhost**
-dbtype	**ODBC**
-ReconnINT	**10** - Retry interval (in seconds) for reconnecting to the Oracle database if the data provider is disconnected from the Oracle database.
-user	**history** - Username for the history user.
-dbname	Defaults to **localhost**. DO NOT change this specification
-name	**DBA**
-FatalErrors	**"03114"** (quotation marks required) - Indicates that oracle error code "03114" for disconnect will be considered fatal.

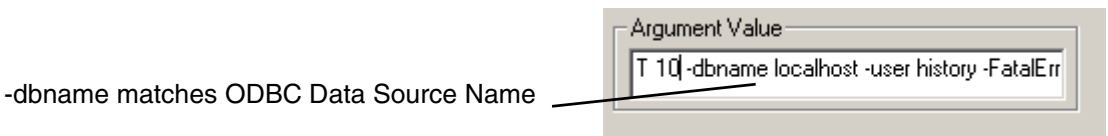


Figure 408. Detailed View of ADO Arguments

Starting and Stopping Providers

To manually start or stop a data provider, select the data provider, right-click, and then choose **Start Provider** or **Stop Provider** from the context menu.

Deleting a Data Provider

To delete a data provider, use the browser to select the data provider, right-click, and then choose **Delete Provider** from the context menu.

The data provider will be deleted immediately (after the specified TerminateDelay time). There is NO prompt to confirm or cancel.

Adding Arguments to a Data Provider

For some data providers, arguments can be added which are then displayed in the configuration dialog. To add arguments, first stop the provider. Then right-click, and choose **Add arguments** from the context menu. This displays a dialog for selecting the argument to add. Figure 409 show the add arguments dialog for an ADO data provider.

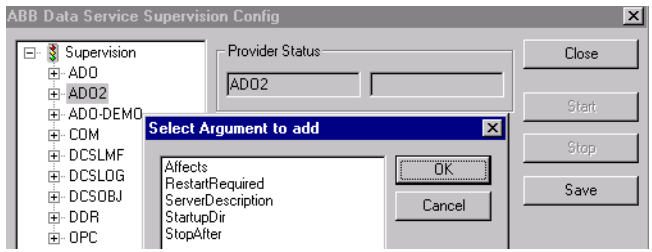


Figure 409. Example Add Arguments Dialog

Checking Display Server Status

To check server status, from the Windows task bar,

1. Choose **Start>Programs>ABB Industrial IT 800xA>Information Mgmt>Display Services>Server Status.**

   This displays a dialog for specifying the server hostname, Figure 410.

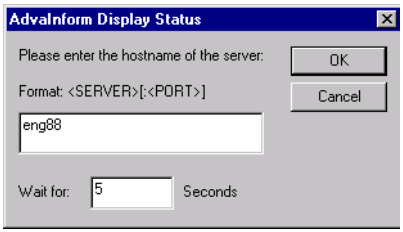


Figure 410. Specifying Server Hostname

2. Enter the hostname for the data server where the data providers are located. Leaving the hostname blank defaults to **localhost**. As an option, specify the maximum time to wait for the server to respond.

3. Click **OK**. This displays the server status window, Figure 411.

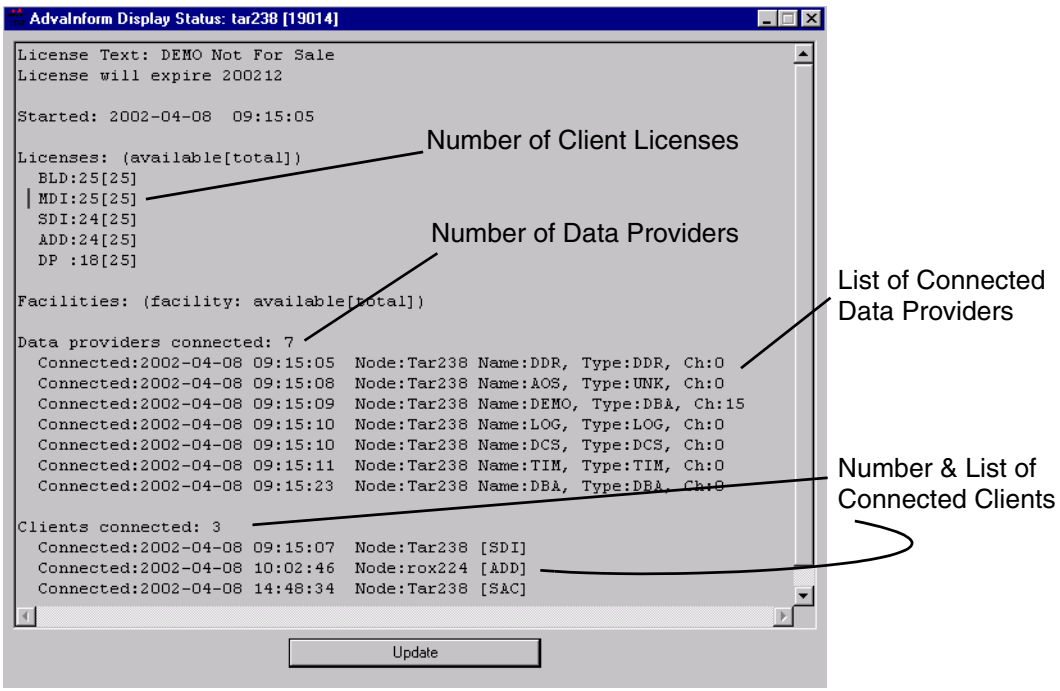


Figure 411. Display Server:COMM Window

This window provides the following information related to the Display Server:

License Text	This field indicates the license text.
License Will Expire	This indicates when the current license is scheduled to expire. After entering the permanent license, this should indicate that the permanent license will never expire.
Started	This field indicates when the server was started.
Facilities	This is not applicable at this time.
Licenses	These fields show the total number of purchased licenses, and the available licenses not currently being used: • Build - When logging in with Build access Display Services can be used in both the Build mode and the Run mode. • MDI - Multiple Document (Display) Interface. When logging with MDI Run access, multiple displays can be run at the same time. • SDI - Single Document (Display) Interface. When logging with SDI Run access, only one display at a time can be run. • ADD - DataDirect. • DP - This is the number of data providers connected.
Data Providers Connected	This shows the number of data providers connected. Display Services must have one data provider connected.
Clients Connected	This shows how many clients are connected to this server. The information provided includes the node name, type (MDI or SDI), and date and time when the client connected.

Section 16 Authentication

This section consists of the following topics:

Usage within Information Management

Authentication can be configured for certain operations related to Information Management to help meet FDA 21CFR part 11 requirements. Authentication determines whether or not approval will be required before a user will be allowed to perform a certain function, for example: activating logs in a log set.

Authentication may be set to one of three levels:

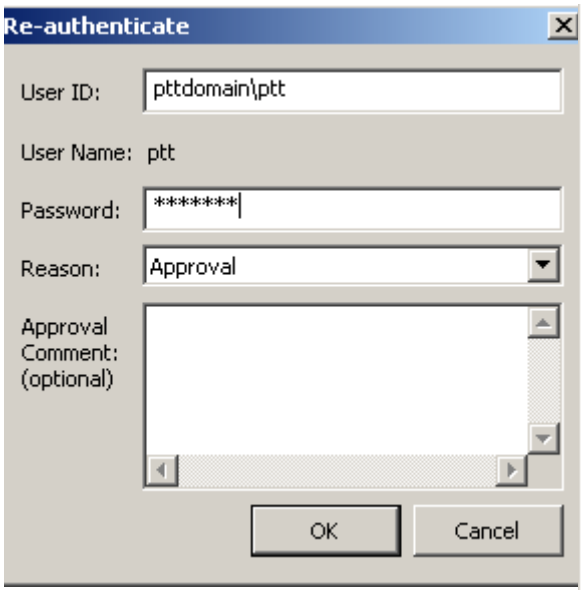
- None,
- Reauthentication requires one user to be approved (by entering user name and password), and
- Double authentication requires two users to be approved.

If authentication for an operation is configured, a dialog is displayed when the operation is attempted, Figure 412, and the user must enter the proper credentials before the operation can proceed.

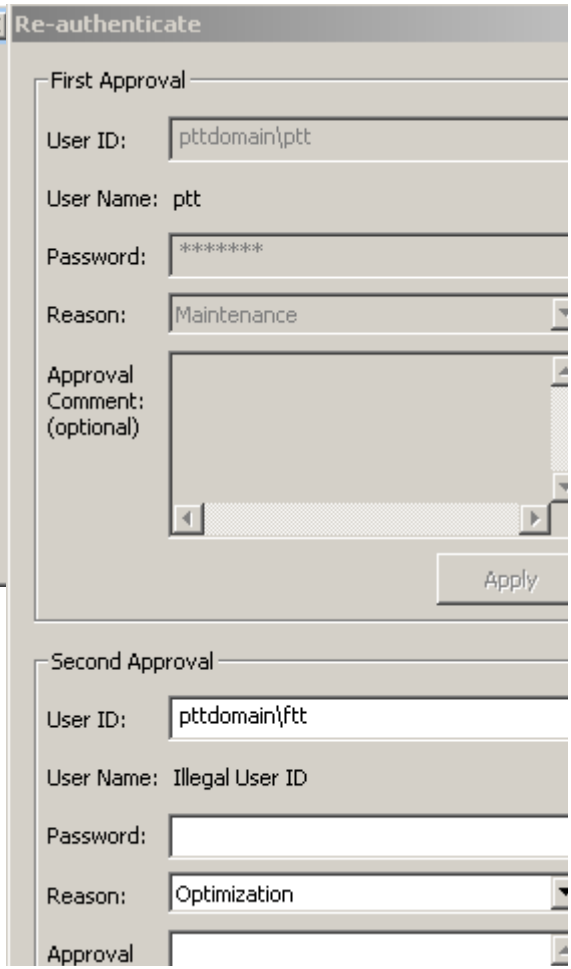
The standard authentication mechanism used by other 800xA system functions is NOT used by Information Management. This section describes how to use authentication specifically for Information Management.

Authentication for runtime operations related to SoftPoint signals is configured via the signal configuration aspect. This is described in Configuring Authentication for Signal Types on page 49.

For aspects related SoftPoint configuration, refer to Configuring Authentication for Aspects Related to SoftPoint Configuration on page 577.



Single Authentication Dialog



Double Authentication Dialog

Figure 412. Authentication Dialogs

Configuring Authentication

Authentication may be configured on an individual basis for each operation associated with aspect categories (except SoftPoints). Table 59 lists the operations for which authentication may be configured. The operations are listed by aspect category, which are grouped by aspect system.

Authentication is configured via an aspect category's Inform IT Authentication Category aspect. Typically, this aspect already exists in the aspect category's aspect list, and all operations will be preset to **None**. It this case, the authentication level may be changed for an aspect category operation by displaying the aspect's configuration view, selecting the operation, selecting the authentication level, and then clicking **Apply**. This is illustrated in Figure 413.

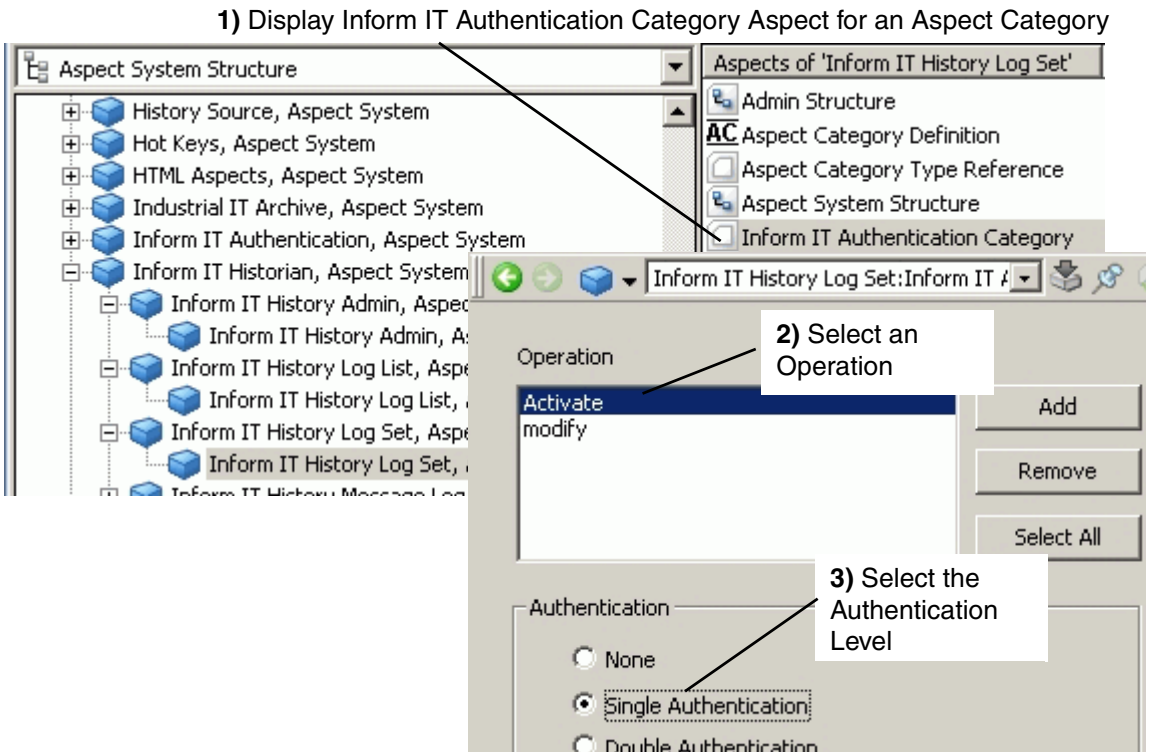


Figure 413. Configuring Authentication

If the Inform IT Authentication Category aspect does not yet exist, then add it to the aspect category's aspect list. If the list of operations for the aspect category is not complete, then any missing operations may be added. To do this, click **Add**, then enter the operation name and click **OK**, Figure 414.

Add the Inform IT Authentication Category aspect to the History Configuration, Aspect System/Log Configuration, Aspect Type/Log Configuration, Aspect Category to allow authentication of numeric logs.

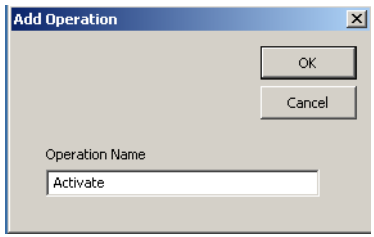


Figure 414. Adding an Operation

Table 59. Operations for which Authentication may be Configured

Aspect System	Aspect Category	Operations
Inform IT Authentication	Inform IT Authentication	Modify
History Configuration	Log Configuration	Modify Activate
Inform IT Historian	Log List	Activate
	Log Set	Modify Activate
	Message Log	Modify Activate
	Report Log	Modify Activate
	View Logs	Activate
	View PDL	Modify Delete PDL

Table 59. Operations for which Authentication may be Configured (Continued)

Aspect System	Aspect Category	Operations
Industrial IT Archive	Archive Device	Modify Publish Unpublish Initialize Remount Override Backup
	Archive Group	Modify Rescan Reset Last Archive Time Manual Archive
	Archive Volume	Modify Restore Initialize Publish Unpublish Copy Volume
Scheduler	Action Aspect	Modify Read
	Job Description	Modify Read Manage
	Scheduling Definition	Modify Read Start
Calculation	Calculation	Enable Modify Execute
	Calculation Status View	Enable Modify Execute

Table 59. Operations for which Authentication may be Configured (Continued)

Aspect System	Aspect Category	Operations
OLE DB Data Access Tables	OLE DB DA Tables	Modify
	OLE DB HIstory Access Tables	Modify

Configuring Authentication for Aspects Related to SoftPoint Configuration

This section describes how to configure authentication for aspects related to SoftPoints. This procedure is different than the one used to configure authentication for aspects related to other Information Management functions.

Authentication for runtime operations for SoftPoint signals is configured as described in Configuring Authentication for Signal Types on page 49.

Authentication can be configured for the following aspect types related to SoftPoint configuration:

- In the PCA Aspect System:

  Binary
  Integer
  Object
  Real
  String

- In the SoftPoint System Aspect System:

  Alarm Event Configuration
  Alarm Event Settings
  Collection
  Deploy
  Generic Control Network
  Process Object Configuration
  Signal Configuration

For these aspect types, when authentication is configured, that configuration applies to all operations which may be performed on those aspects (unlike the other

Information Management aspects where authentication can be applied on an operation-by-operation basis.

To configure authentication for one or more of the above aspect types:

1. Set the Advanced Access Control to **True**. This is done via the System Settings aspect for the Domain object in the Admin structure, Figure 415.

This will enable authentication for all aspect categories which have been configured to be enabled. If the Advanced Access Control was set to false to disable authentication on those aspect categories, they will now be enabled.

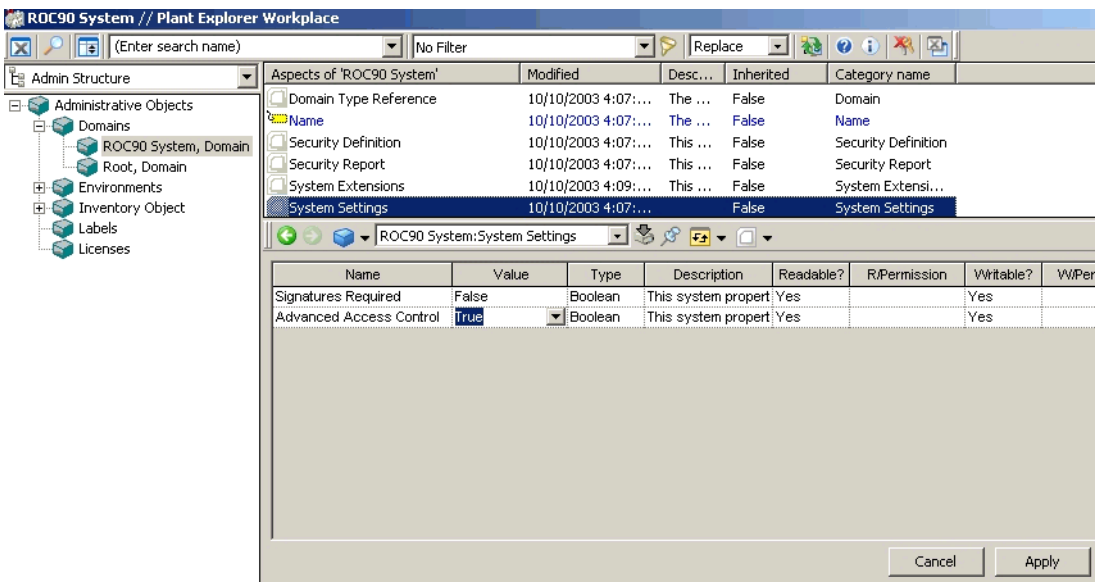


Figure 415. Setting Advanced Access Control to True

2. Then (reference Figure 416):
   a. Go to the Aspect System structure.
   b. Select the applicable aspect type under the PCA or SoftPoint aspect system.
   c. Select the Aspect category Definition aspect.

d. Check the check box for the authentication level desired: **Re-authentication Required** or **Double Authentication Required**.

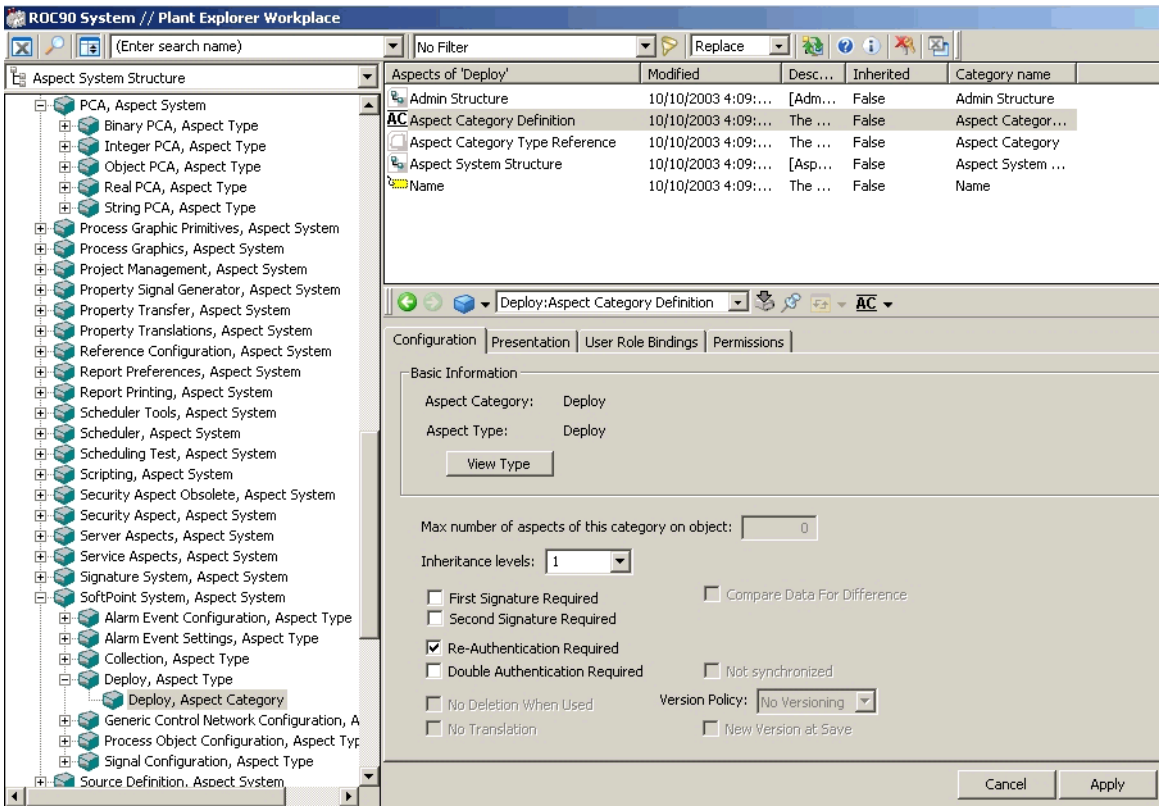


Figure 416. Configuring Authentication

Section 17 System Administration

This section describes administrative procedures for Information Management applications in the 800xA system. Access to most of these functions is via the Administrative Tools utility on the Windows Control Panel, Figure 417. To access the PAS administrative tools, from the Windows task bar choose: **Start>Settings>Control Panel>Administrative Tools>PAS**.

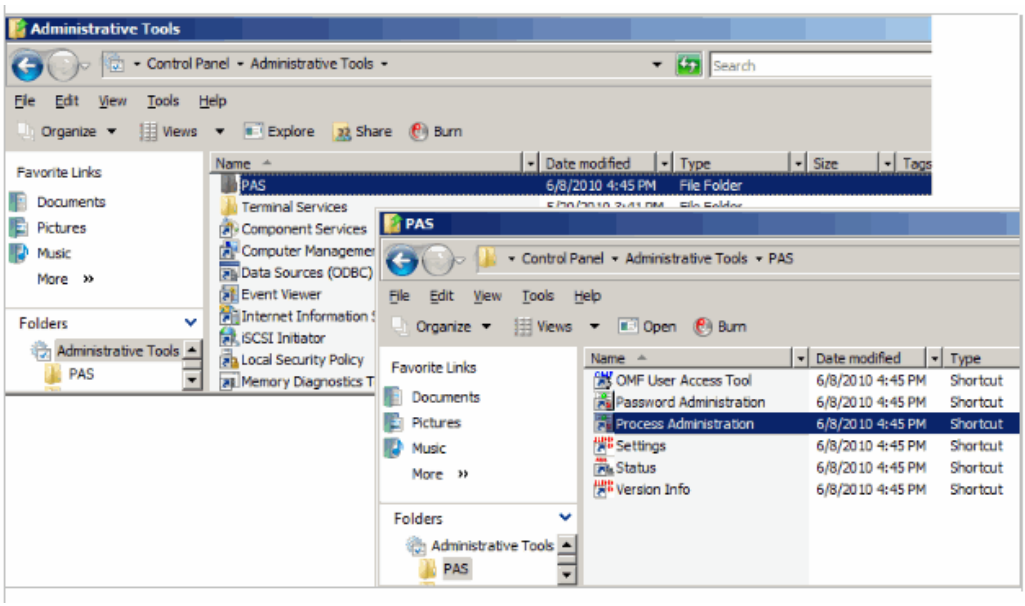


Figure 417. Accessing Information Management System Administration Functions

To use these system administration functions, Domain Administrator privileges is required in the Windows system. The user name used must have Domain Administrator rights.
For consolidation nodes, a domain user with domain administrator privileges may be used, or a locally-defined user with administrator privileges, depending on whether or not the consolidation node resides in a domain.

Configuring OMF Shared Memory

OMF Shared memory size defines the size of shared memory for processes that use the OMF software. If the shared memory is too small, processes may be aborted without any error messages (the Process Administration Services will discover that the process has been aborted). The default shared memory size is 8,192 Kbytes. To adjust OMF shared memory size, refer to OMF Shared Memory Size on page 629.

Start-up and Shutdown

Information Management software processes are started, shut down, and supervised by the Process Administration Services (PAS) window, Figure 418, independent of Windows. Refer to Stopping Oracle on page 587 to stop Oracle processes.

Stop PAS first before restarting or shutting down the computer once History Services has been started and is collecting data. PAS performs an orderly shutdown of the ABB software and any user processes under its supervision. Failure to stop PAS before restarting or shutting down the computer will result in loss of History data, and may corrupt History data files.

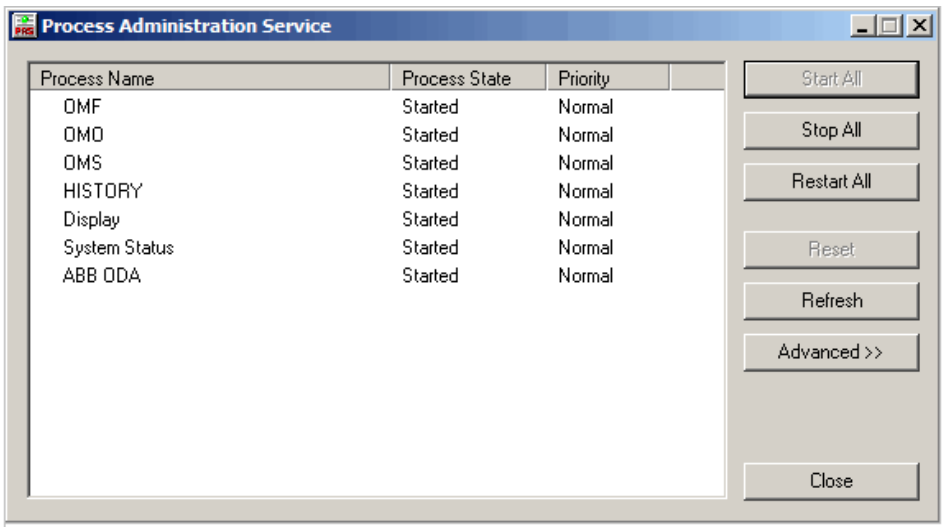


Figure 418. Process Administration Service Window

PAS Window

To access the PAS window, log in as an Domain Administrator-level user or a user in the PAS operator list. Open the PAS window from the Windows task bar: **Start>Settings>Control Panel>Administrative Tools >PAS>Process Administration**.

This window is for starting and stopping all processes under PAS supervision. It also provides access to advanced functions for debugging.

The process list shows all processes specified in the Windows registry under HKEY_LOCAL_MACHINE\SOFTWARE\ABB\PAS\Startup. The information provided in this list is described in Table 60. PAS Window controls are described in Table 61.

Table 60. PAS Process List

Field	Description
Process Name	Name of the processes.
Supervision Enabled or Disabled	When a process is removed from PAS supervision (Advanced Functions on page 585), an X icon is placed to the left of the process name.
Process State	State of the supervised process, normally Started or Stopped.
Priority	Order in which processes run. When the processor is busy, this determines which processes will be run at all.

Table 61. PAS Window Controls

Button	Description
Start All/Stop All	Start or stop all processes. Refer to Starting and Stopping All Processes.
Restart All	Stop and then restart all processes.
Reset	Resets *failed* processes to the *Stopped* state.
Refresh	Clears the process list and queries PAS for the current process list. This may be used if the list gets out of sync with the PAS internal state.
Advanced>>	Expands the window to show controls for advanced functions. Refer to Advanced Functions on page 585.
Connect	This button is only visible when the PAS window is disconnected from PAS. If this occurs, use this button to reconnect the PAS window to PAS

Starting and Stopping All Processes

To stop all PAS processes without shutting down the computer, open the PAS window, from the Windows task bar.

1. Choose **Start>Settings>Control Panel> Administrative Tools>PAS>Process Administration**.
2. Click **Stop All**. PAS is still active when all processes are stopped.
3. To restart all stopped processes, click **Start All**.

Processes start in the order specified by their respective WhenToStart parameters. This order corresponds to the order in which the processes are listed in the PAS window. The processes are stopped in the reverse order from which they are started.

Advanced Functions

Click **Advanced>>** in the PAS Window to show the Advanced functions, Figure 419.

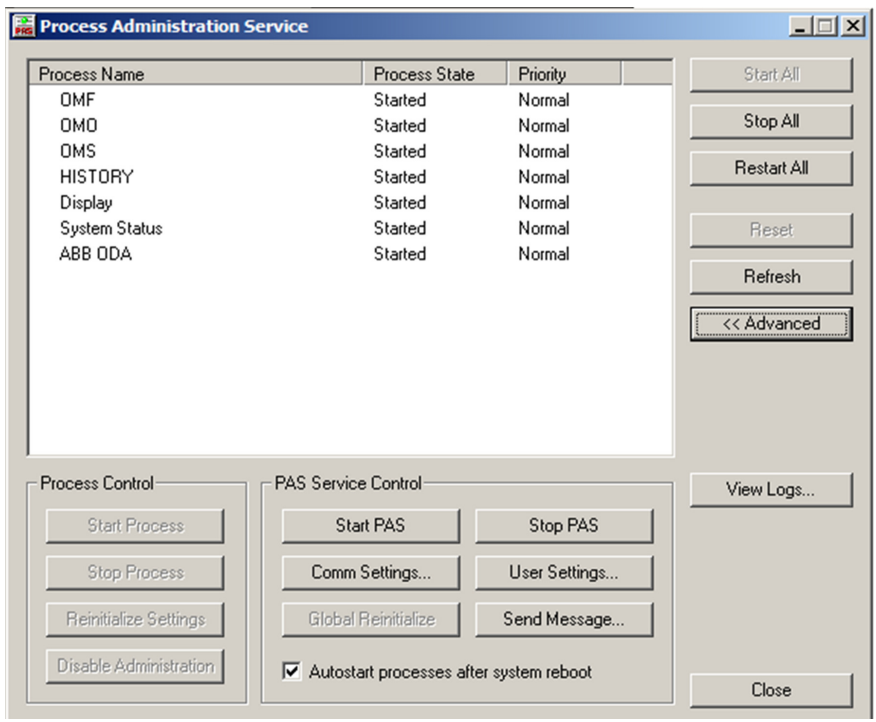


Figure 419. Advanced Dialog

The advanced functions are organized on three levels:

- Process Control - Table 62.
- PAS Service Control - Table 63.

Table 62. Advanced Functions for Individual Processes

Button	Description
Start Process/Stop Process	Start or stop the selected process. **CAUTION**: PAS does not perform dependency checks before starting/stopping an individual process. Therefore, as a rule, stop or start all processes rather than individual processes.
Disable/Enable Administration	This button alternately disables and re-enables PAS supervision for a process. For example, if a process fails to start for some reason, consider removing it from PAS supervision so as not to affect the start-up of other processes. This is generally used for troubleshooting.
Reinitialize Settings	This is for technical support personnel. It is used when the registry information for a process has been changed. When this occurs, the process must be reinitialized in order for the changes to take effect.

Table 63. Advanced Functions for PAS Service

Button	Description
Stop PAS/Start PAS	Start PAS starts PAS. If Autostart flag in the registry is set or is not specified, PAS will begin the *Start All* sequence as soon as it starts. Stop PAS stops PAS. Before PAS service stops, it will shutdown all processes including the ones that have disabled administration. **NOTE:** Only Domain Administrator users are permitted to use start/stop. Users in the PAS Operator list are not permitted to start/stop PAS.
Global Reinitialize	This command can only be issued when all processes are stopped. It tells the PAS service to completely erase all its data structures and reinitialize them with current registry settings. Any changes in the registry including PAS global setting, Node Type, and individual process settings will take effect when this command is issued.
Send Message	This displays a dialog for sending messages as an alternative method for interacting with the PAS service, Figure 420. The messages that have been sent to a process can also be read. This functionality is generally reserved for debugging by technical support personnel.
View Logs	Displays execution log for PAS service.

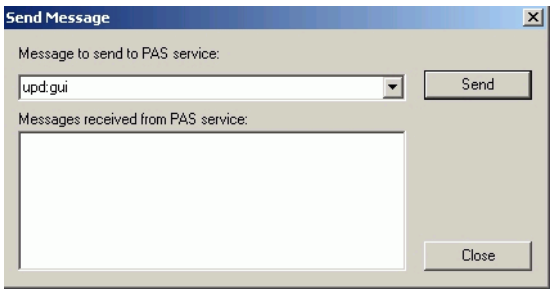


Figure 420. Send Message Dialog

Stopping Oracle

To stop Oracle, first stop the Windows services for the Oracle database instance and Listener. This is done via the Services utility in the Administrative Tools on the Windows Control Panel. To do this:

1. Launch the Services utility as shown in Figure 421.
2. Scroll to the `OracleServiceADVA` service, right-click and choose **Stop** from the context menu. Figure 422.
3. Scroll to the service named OracleTNSListener, right-click and choose **Stop** from the context menu.

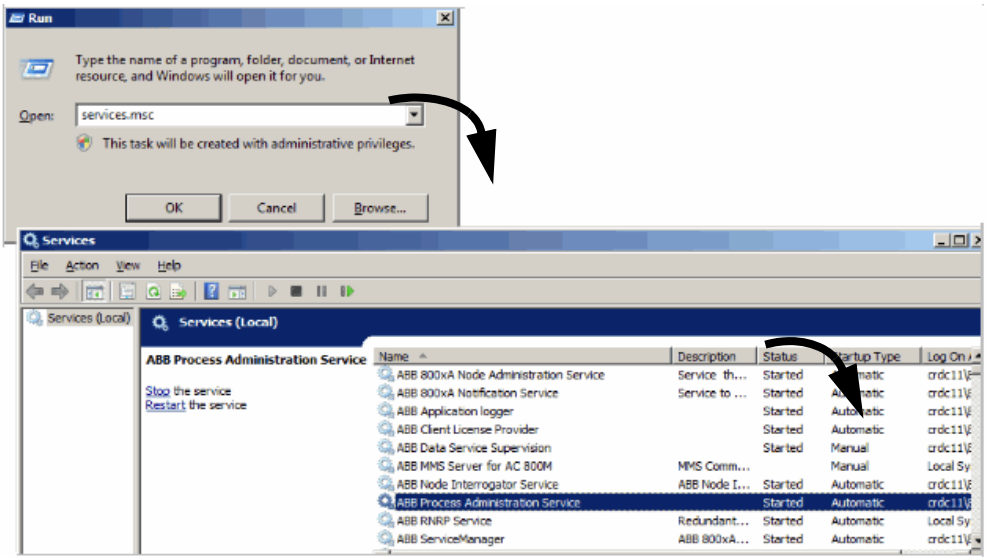


Figure 421. Accessing the Windows Services Utility

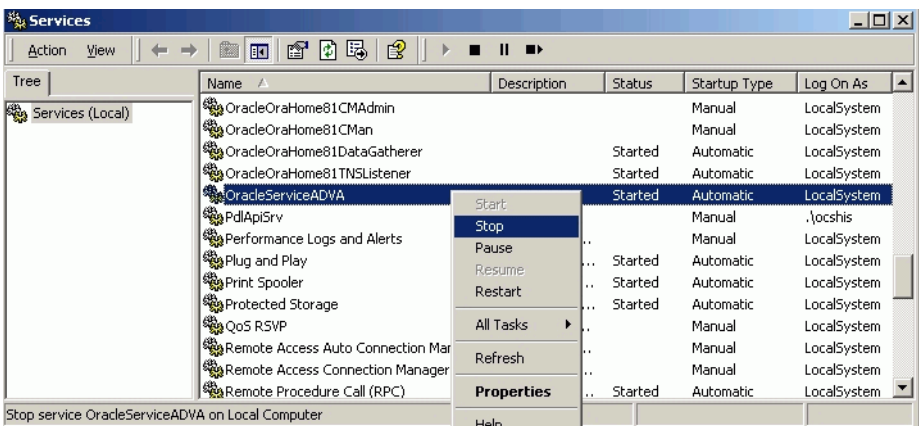


Figure 422. Stopping OracleServiceADVA

Managing Users

The Information Management installation creates default user accounts and user groups. To add, remove, or otherwise edit any user accounts, use this section to understand how to manage user access for the Information Management server. User access is handled on four levels:

- **Windows User** - The computer where the server software is installed requires you to log on with a Windows username and password. Windows users are created and managed via the User Manager on the Windows control panel. A default user configuration is created by the Windows installation. Other Windows users may be created by the installing user.

  The Information Management and Oracle software installations create additional users and groups. These are described in Windows Users and Groups for Information Management on page 590.

- **Oracle Access** - Oracle user authority is required by certain Information Management processes for access to the Oracle database. A default user configuration is created by the Information Management and Oracle software installations. These users are described in Oracle Users on page 592.

- **Display Client Users** - Standard user files are provided for the desktop applications - DataDirect, Desktop Trends, and Display Client. Additional users can be created by copying and renaming an existing user file. New passwords, configure user preferences, and customize language translations can also be specified. Refer to Managing Users for Display Services on page 596.

Change the default passwords for certain Windows users and Oracle users immediately after installing the Information Management software. This helps protect the system from unauthorized access. Guidelines are provided in Securing the System on page 593.

To support Information Management applications such as historical database maintenance, and other administrative procedures one or more Industrial IT users must be added to the HistoryAdmin Group.

Windows Users and Groups for Information Management

The default user configuration following installation of Information Management and Oracle software is illustrated in Figure 423, and described below.

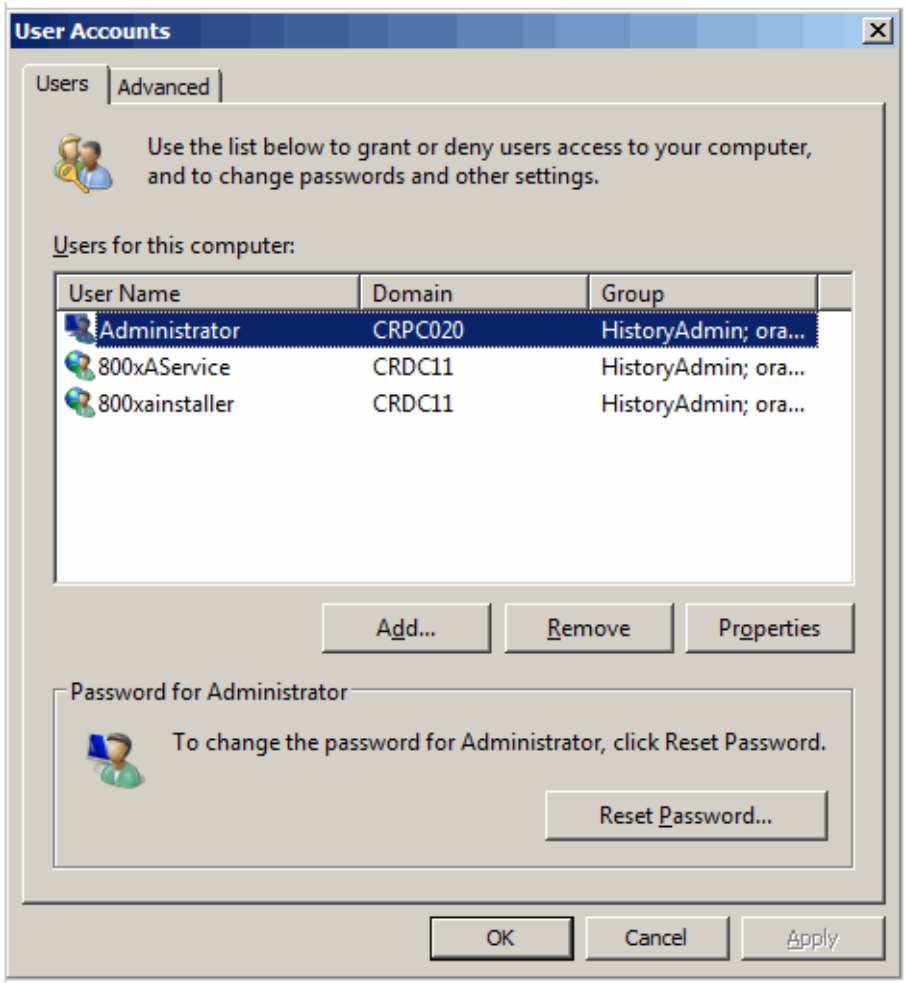


Figure 423. Default Windows Users

This default user configuration is sufficient to support all Information Management operation, configuration, and administration activities. Create additional users and modify user accounts as required via the Windows User manager.

HistoryAdmin Group

This group is created by the Information Management software installation. Users in this group have access to History configuration and History database maintenance functions. These users can access History database maintenance functions and other Information Management administrative procedures seamlessly without having to change users, or enter Oracle passwords.

This group is included in the PAS OperatorList by default. This grants all users in the HistoryAdmin group access to PAS, even if these users are not Domain Administrator-level users. This enables HistoryAdmin users to start and stop PAS services as required by certain History database maintenance functions.

To grant HistoryAdmin users access to History database configuration, but deny access to PAS, remove this group from the PAS OperatorList. Instructions for editing the PAS OperatorList are provided in PAS OperatorList on page 595.

User for 800xA System Service Account

When the 800xA base system software is installed, an 800xA service account is configured. The user for this account is assigned to the ORA_DBA and HistoryAdmin user groups and all History services will run as this user. To support this, this user must also be specified for the log-in accounts for these services. This is done when the History database instance is configured. The IM History Service Account dialog used to set up this account may also be launched from the Windows task bar **Start>Settings>Control Panel>Administrative Tools>PAS> Password Administration**). This method for launching the IM History Service Account dialog should only be used for certain upgrade procedures. Instructions for using the dialog will be provided in the applicable upgrade instructions.

ORA_DBA Group

This group is created by the Oracle software installation. Users in this group have Oracle access for database maintenance functions. Such access is generally reserved for technical support personnel.

Oracle Users

The default Oracle user configuration created by the Oracle and Information Management software installations is described in Table 64.

Table 64. Oracle Users

User(1)	Description
SYS	Created by Oracle.
SYSTEM	Created by Oracle.
HISTORY	This user is created when the Oracle database instance is created. This user has read access to Oracle tables and views.
OPS$OCSHIS	This is the oracle account for the IM history database.
OPS$_____	This is an Oracle user account created for the Windows user that installed the Information Management software. This user account is classified as EXTERNAL, meaning it does not have an Oracle password. To log into Oracle when you are already logged in as the Windows user that installed the Information Management software, just enter a slash(/) following the *sqlplus* command. For example, sqlplus /.

(1) Since Information Management users are not licensed for write access to Oracle tables, the HISTORY user account is the only Oracle user that operators should use.

Any new Oracle instance will force the user to select passwords for the administrative Oracle users (sys, system and ops$ocshis) and optionally the history user. Users can change these passwords at any time with the IM Oracle instance wizard. Users whose password is indicated as EXTERNAL are authenticated by the Windows operating system. The manual procedure is described in Securing the System on page 593

There is no default password for Oracle accounts. The password should be set while creating the instance.

Securing the System

The user will be prompted to select passwords for administrative users (SYS, SYSTEM, and ops$ocshis) and the read only account (history) when they create the History Database. All other Oracle accounts are disabled.

Changing Passwords for Oracle Users

The default Oracle users are described in Oracle Users on page 592. The current Oracle user information resides in the Oracle table named dba_users. Use sqlplus to access the contents of this table as shown in Figure 424.

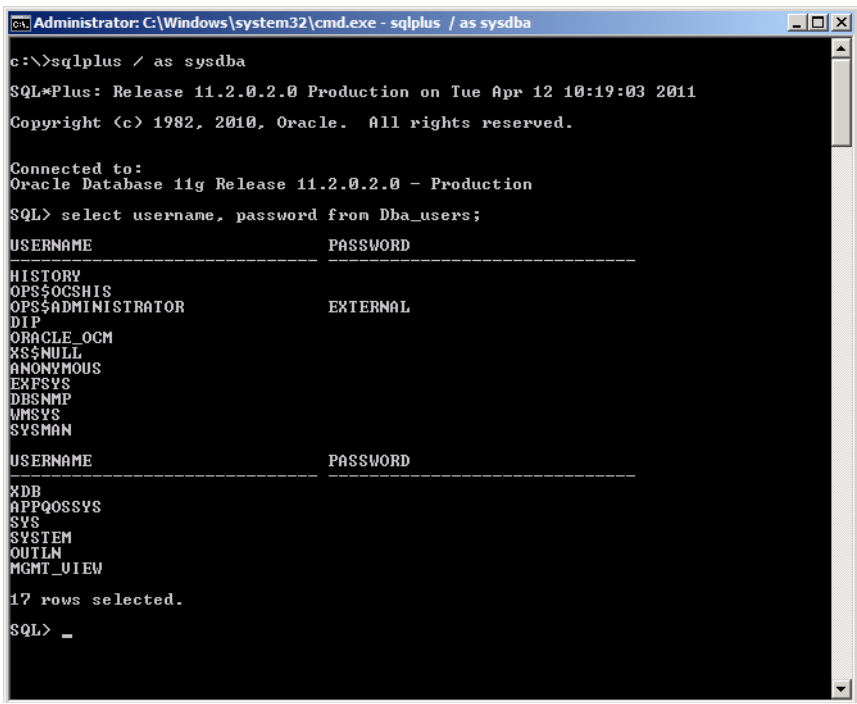


Figure 424. Displaying Oracle Users and Changing Passwords

The Oracle Instance wizard can be used at any time to change the password for these accounts. The runtime software does not depend on these accounts. See Figure 425.

If any reports or other Oracle access to these accounts use a hard coded password, those scripts will have to be modified to use any new password.

All History Services Oracle access use Windows authentication to access the Oracle database. Runtime access is independent of the passwords for these accounts.

Figure 425. Change Oracle Passwords

PAS OperatorList

OperatorList is a configuration parameter for the PAS Service in the Windows Registry. This parameter specifies a list of groups and users other than Domain Administrator-level users who can use the PAS window to start and stop processes under PAS supervision. Groups and users in this list have complete access to all functions in the PAS window, except starting and stopping the PAS service itself. Only Domain Administrator users can start and stop PAS. This list can be edited to grant/deny users and groups access to PAS.

The HistoryAdmin Group is included in the PAS OperatorList by default. To grant HistoryAdmin users access to History database configuration, but deny access to PAS, remove this group from the PAS OperatorList.

To edit the PAS OperatorList:

1. Open the Registry Editor. From the Windows task bar, choose **Start>Run**. Then enter **regedit** in the Run dialog.
2. Navigate to the location in the registry where the processes under PAS supervision are specified - HKEY_LOCAL_MACHINE\SOFTWARE\ABB\PAS, Figure 426.

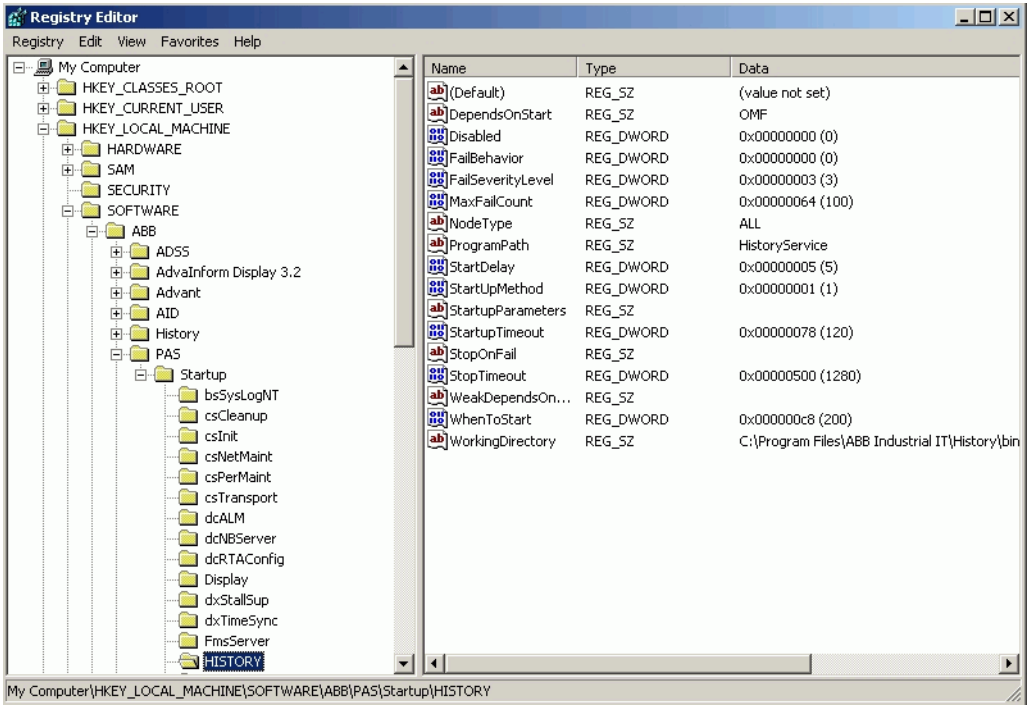


Figure 426. Locations of PAS Processes

3. Double-click the OperatorList parameter.

   The list is specified as names separated by pipe(|) character with no spaces between the name and the pipe. Example: HistoryAdmin|User1|User2

Managing Users for Display Services

The operator must create its own client users.These users are used when logging into Displays Services, a display client, DataDirect (Excel Data Access), and Desktop Trends.

User configurations are maintained in a separate `Preferences.svd` file for each user. These files are located in `.svg` folders in:

C: \Program Files\ABB Industrial IT\Inform IT\Display Services\Server\Data

The user name is the second word in the name (following the underscore), Figure 427.

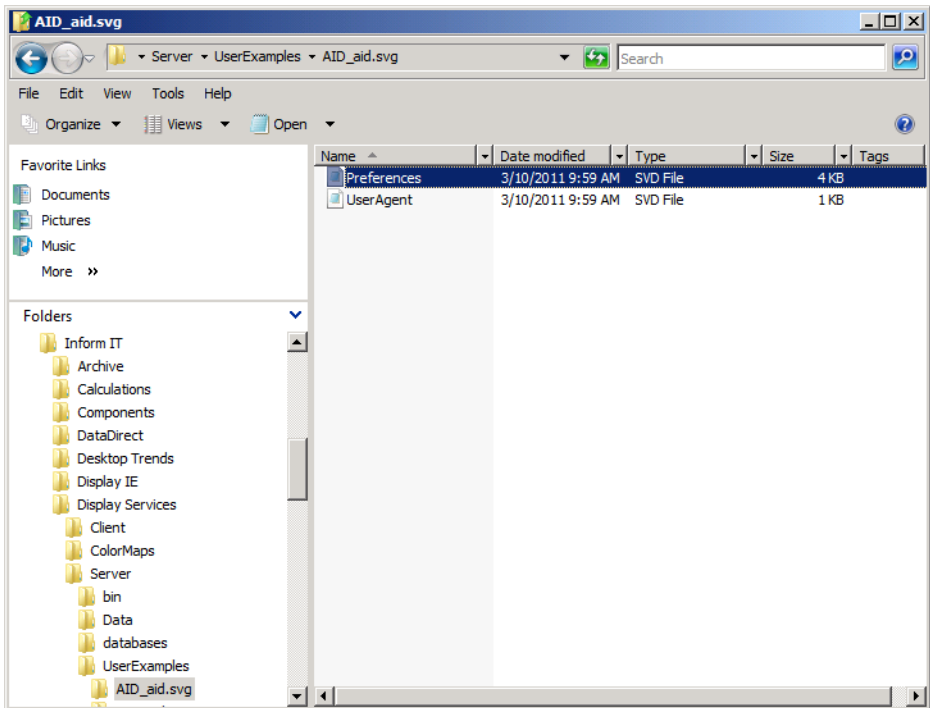


Figure 427. User Configuration Data

The following user management procedures are covered here:

- Creating Users.
- Creating User Passwords.
- Configuring User Preferences.
- Customizing Language Translations.

Creating Users

Three users templates have been provided to simplify the user configuration. The templates are located in:

C:\Program Files\ABB Industrial IT\Inform IT\DisplayServices \Server\UserExamples

The following users exist:

- **aid**: The user having all rights (build/MDI/SDI). No startupdisplay or basedisplay is defined for this user.
- **mdi**: The user runtime rights only (MDI/SDI). The startupdisplay is [Displaydemo/Demostart] and the basedisplay is defined to [Displaydemo/AlarmBase].
- **sdi**: A user with SDI runtime rights only. The startupdisplay is [Displaydemo/Demostart] and no basedisplay is defined. The browser and the status bar are not visible to provide a fullscreen display.

Select one of these users as the base user, change the name and create a password for it.

For instance, navigate to the AID_aid.svg folder as shown in Figure 428, in the following location:

C:\Program Files\ABB Industrial IT\Inform IT\Display Services\Server\UserExamples

Then right click the folder and use the context menu to copy and paste as shown in Figure 427, in the following location:

C: \Program Files\ABB Industrial IT\Inform IT\Display Services\Server\Data

Then, rename the folder.

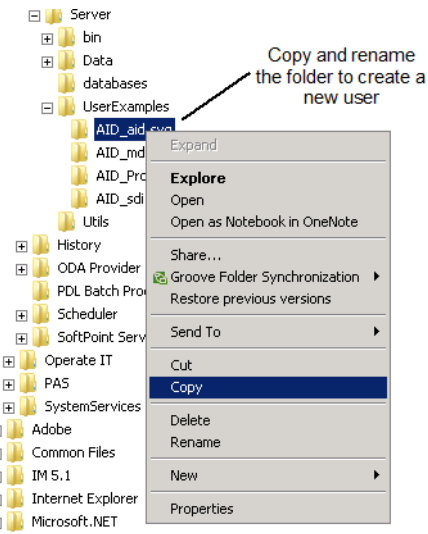


Figure 428. Copying and Renaming an User

When a folder is copied and renamed, the user name is the second word in the name (following the underscore, for example in AID_aid.svg, the user name is aid).

When a new user is created, then create a unique password for that user. Refer to Creating User Passwords on page 600.

Also, configure user preferences and/or customize language translations for that user. Refer to Configuring User Preferences on page 601, and Customizing Language Translations on page 607.

Creating User Passwords

This describes how to change the password for an existing user, or assign a unique password for a new user. This is a two-step process. First run the Create User Password utility to get an encrypted password key. Then, associate the new password with a specific user by entering the password key in the `Preferences.svd` file for the user.

Create User Password

To run the Create User Password utility,

1. From the Windows task bar, choose:
   Start>Programs>ABB Industrial IT 800xA>Information Mgmt> Display Services>Server>Create Password.

   This displays the Create Password dialog.

2. Enter the new password in the **Typed Password** field. The encrypted password key is displayed in the **AdvaInform Display Password** field, Figure 429.

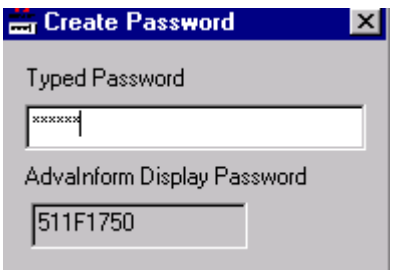


Figure 429. Password Key

Associate New Password with a Specific User

To associate a new password with a specific user, record the password key (from AdvaInform Display field in Create Password dialog), and enter it in the user's `Preferences.svd` file.

Navigate to the applicable user folder (directory) as described in Managing Users for Display Services on page 596. Then open the applicable folder, for example,

AID_AID.svg. This folder contains `Preferences.svd` for the AID user, Figure 427.

To enter the new password, open the `Preferences.svd` file with a text editor, and then edit the file. An example is shown in Figure 430.

Copy the value from the AdvaInform Display Password field and then paste it in the Preferences.svd file, or simply enter the value directly.

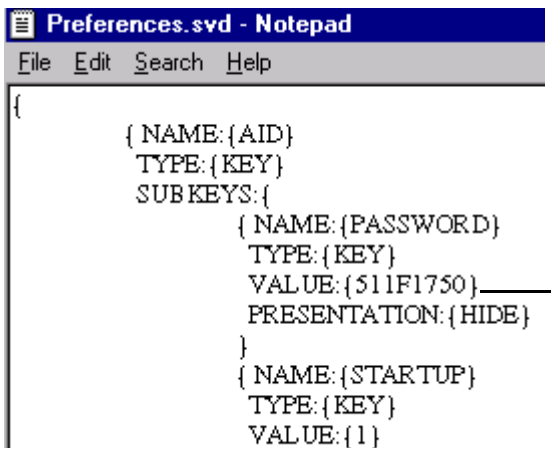


Figure 430. Editing the Preferences.svd File

Configuring User Preferences

By configuring user preferences, Table 65, the user interface can be customized according to the specific user that logs in to Display Services (or one of the desktop applications).

To allow users to change preferences, the corresponding preferences file must have read/write access, Figure 431. To find how to access a preferences file, refer to Managing Users for Display Services on page 596.

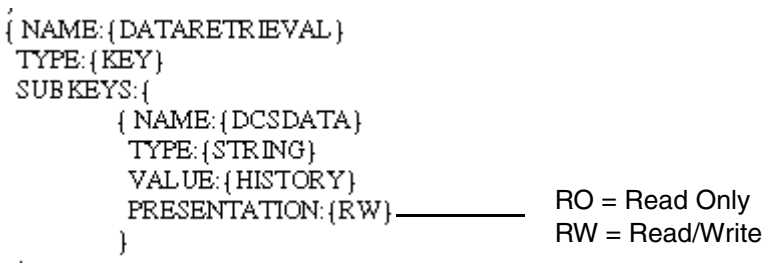


Figure 431. Configuring a Preference for Read-Only or Read/Write Access

To configure a user preference:

1. Start the Windows client.
2. From the menu bar, choose **User > Preferences**. This displays the User Preferences dialog.
3. Use the navigation tool to find the user preference to be configured.
4. Click on the preference. When a selection is made, the dialog displays interactive widgets as required to configure the selected preference. For instance, if TEXTCOLOR under EDITOR is selected, the dialog displays the current color selection, and a **Change** button. Clicking this button displays the color palette so the color can be changed, Figure 432.

Any changes made to user preferences will not take affect until the computer is restarted.

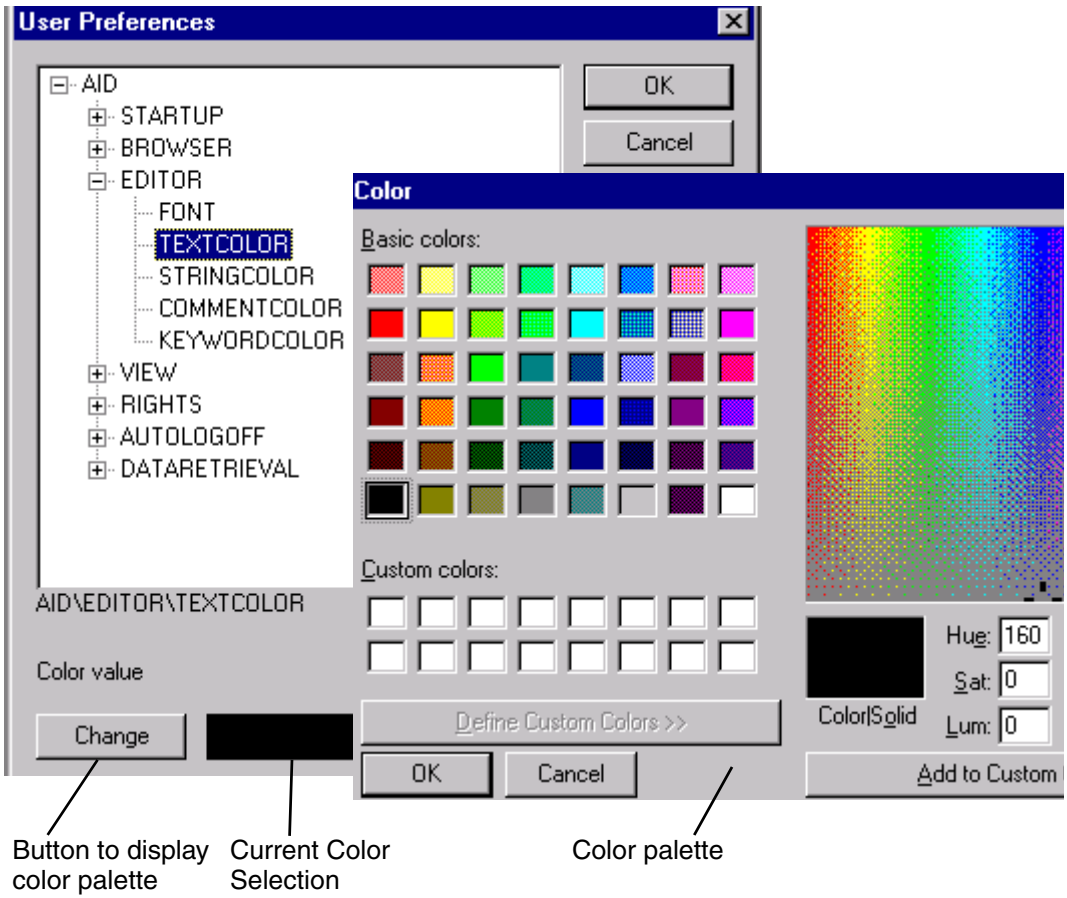


Figure 432. Configuring TEXTCOLOR

Table 65. User Preferences

Name	Default	Description
AID-PASSWORD	47D2D5F8	Encrypted password, not accessible via the User Preferences dialog. Refer to Creating User Passwords on page 600.
AID-STARTUP-DISPLAY-GROUP		Group name of the display presented when the application is started. By default, no startup display is specified. If a startup display is specified, it is generally from the Batch group. Display groups are listed under the host root in the Object Browser.
AID-STARTUP-DISPLAY-DISPLAY		Name of the display presented when the application is started. By default, no startup display is specified. If a startup display is specified, it is generally from the Batch group.
AID-STARTUP-BASE_DISPLAY-GROUP		Group name of the base display presented when the application is started.The base display is shown in a third frame at the bottom of the main window. By default, no base display is specified. If a base display is specified, it is generally from the Batch group.
AID-STARTUP-BASE_DISPLAY-DISPLAY		Name of the base display presented when the application is started.The base display is shown in a third frame at the bottom of the main window. By default, no base display is specified. A command application is to specify the MessageDisplay as the base display.
AID-BROWSER-ROOTASTAB	False	This toggles between the normal browser view (default) and the tabbed view where roots on the tree are displayed as tabs.
AID-BROWSER-SCROLLSPEED	5	This adjusts the scroll speed for the browser.
AID-BROWSER-SPRINGLOADDELAY	2	This adjusts the delay (in seconds) for expanding a folder in the browser when using drag-and-drop.

Table 65. User Preferences (Continued)

Name	Default	Description
AID-EDITOR-FONT	49	Font used in Display Services. This is only applicable with a build license for Display Services.
AID-EDITOR-TEXTCOLOR	0	Color for text in Display Services. This is only applicable with a build license for Display Services.
AID-EDITOR-STRINGCOLOR	255	Color for strings in Display Services. This is only applicable with a build license for Display Services.
AID-EDITOR-COMMENTCOLOR	35328	Color for comments in Display Services. This is only applicable with a build license for Display Services.
AID-EDITOR-WORDCOLOR	16000000	Color for keywords in Display Services. This is only applicable with a build license for Display Services.
AID-VIEW-TOOLBAR	True	Show or hide the toolbar.
AID-VIEW-ELEMENTBAR	True	Show or hide the element bar. This is only applicable with a build license for Display Services.
AID-VIEW-USERBAR	True	Show or hide the user bar.This is only applicable with a build license for Display Services.
AID-VIEW-STATUSBAR	True	Show or hide the status bar.
AID-VIEW-BROWSER	True	Show or hide the browser.
AID-VIEW-OLDSTYLEFRAME	True	Use the old-style or new-style frame.This is only applicable with a build license for Display Services.
AID-VIEW-DISPLAYBOARDERCOLOR	1	Color of display boarder.
AID-VIEW-INACTIVEAREACOLOR	1	Color of the inactive area.
AID-AUTOLOGOFF-AFTERMINUTES	15 Minutes	This is used to configure a user to be automatically logged off after a period of inactivity. The autologoff function must be enabled for this setting to take affect.

Table 65. User Preferences (Continued)

Name	Default	Description
AID-AUTOLOGOFF-ENABLED	False (0)	This is used to enable or disable the AUTOLOGOFF function. False(0) = Disabled True (1) = Enabled
AID-DATARETRIEVAL-DCSDATA	NORMAL	This is used to specify whether numeric display elements will query process objects directly for real-time data, or whether the numeric elements will query the process object's associated History log. The default is to get real-time data directly from process objects (NORMAL). By setting this preference to HISTORY, the query returns the last History value that was logged for the process object. The numeric element is configured the same way, whether the query is for real-time data, or Historical data.
AID-RIGHTS-OBJECTWRITE	False (0)	This is used to enable or disable write access to process and/or OPC objects. False(0) = Disabled - no write access True (1) = Enabled - write access allowed
AID-RIGHTS-LOGWRITE	False (0)	This is used to enable or disable write access to numeric history logs (modify or add entries). False(0) = Disabled - no write access True (1) = Enabled - write access allowed
AID-RIGHTS-SYSTEMACCESS	True (1)	This is used to enable or disable the System function in a Display script. False(0) = Disabled True (1) = Enabled

Customizing Language Translations

The language for the user interface is determined by the language file selected when logging in to a client application. An ENGLISH language file is provided as standard. Create additional language files by copying, renaming, and then editing this standard file.

As shown in Figure 433, the language files reside in: `C:\ProgramFiles\ABB Industrial IT\Inform IT\Display Services\Client\Lang`

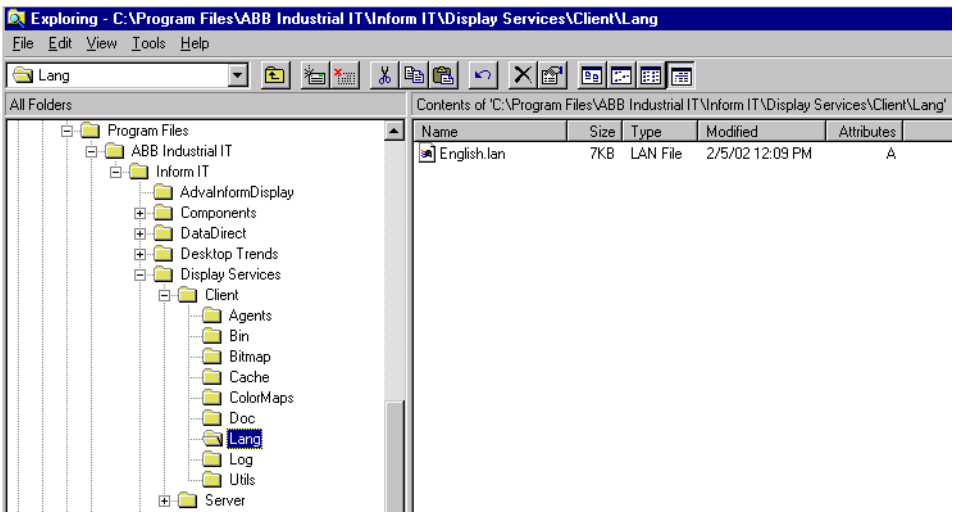


Figure 433. Navigating to Language Files

The first time you log in using a custom language, a prompt to define any unknown text strings is made, Figure 434. Either define the strings at this time, skip some strings on an individual basis, or skip all definitions at this time.

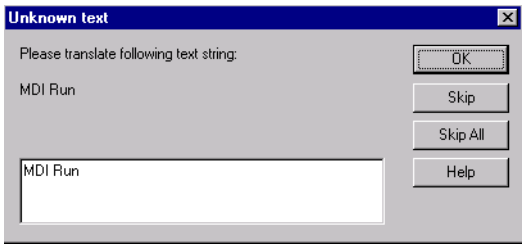


Figure 434. Prompt to Define Unknown Text

To customize the language, choose **User > Language** from the menu bar and then use the Edit Language dialog, Figure 435.

The Texts list displays the English version of all text used in the user interface. Selecting a text line from the list displays the translation for that text according to the language chosen for this session. English text is the default. Edit the translation in the Translation field, and then click **Apply**.

Some texts have special characters. DO NOT remove these special characters.

- & is used in menu texts to indicate that the next character is a mnemonic key.
- % is used by the system to fill in additional text such as the name of a display.

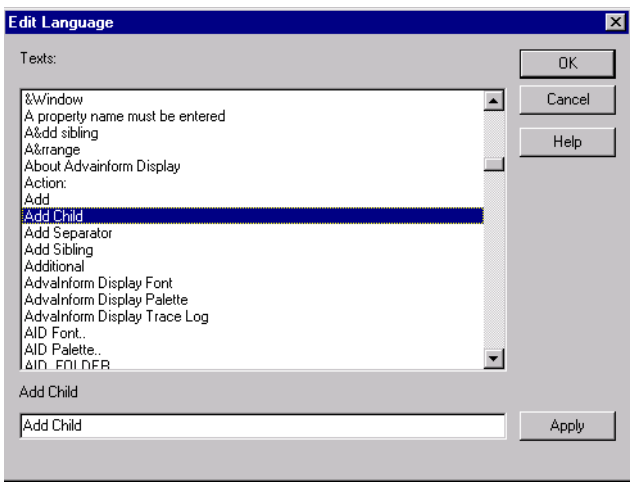


Figure 435. Edit Language Dialog

Checking Display Server Status

To check server status, from the Windows task bar choose **Start>Programs>ABB Industrial IT 800xA>Information Mgmt>Display Services>Server Status**. This displays a dialog for specifying the server hostname, Figure 436.

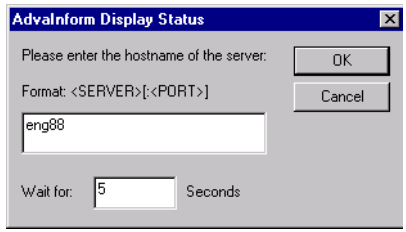


Figure 436. Specifying Server Hostname

Enter the hostname, then click **OK**. As an option, specify the maximum time to wait for the server to respond. This displays the server status window, Figure 437.

Leaving the hostname blank defaults to **localhost**.

This window provides the following information related to the Server:

License Text	This field indicates the license text.
License Will Expire	This indicates when the current license is scheduled to expire. After entering the permanent license, this should indicate the permanent license will never expire.
Started	This field indicates when the server was started.
Facilities	This is not applicable at this time.
Data Providers Connected	This shows the number of data providers connected. Display Services must have one data provider connected.
Clients Connected	This shows how many clients are connected to this server. The information provided includes the node name, type (MDI or SDI as described in Licenses), and date and time when the client connected.

Licenses

These fields show the total number of purchased licenses, and the available licenses not currently being used:

- Build - Build access allows use of Display Services both in the Build mode and the Run mode.
- MDI - Multiple Document (Display) Interface. MDI Run access allows multiple displays to be run at the same time.
- SDI - Single Document (Display) Interface. SDI Run access allows one display to be run at a time.
- ADD - DataDirect.
- DP - Number of data providers (refer to Data Providers Connected). Information regarding data providers is in Section 15, Configuring Data Providers.

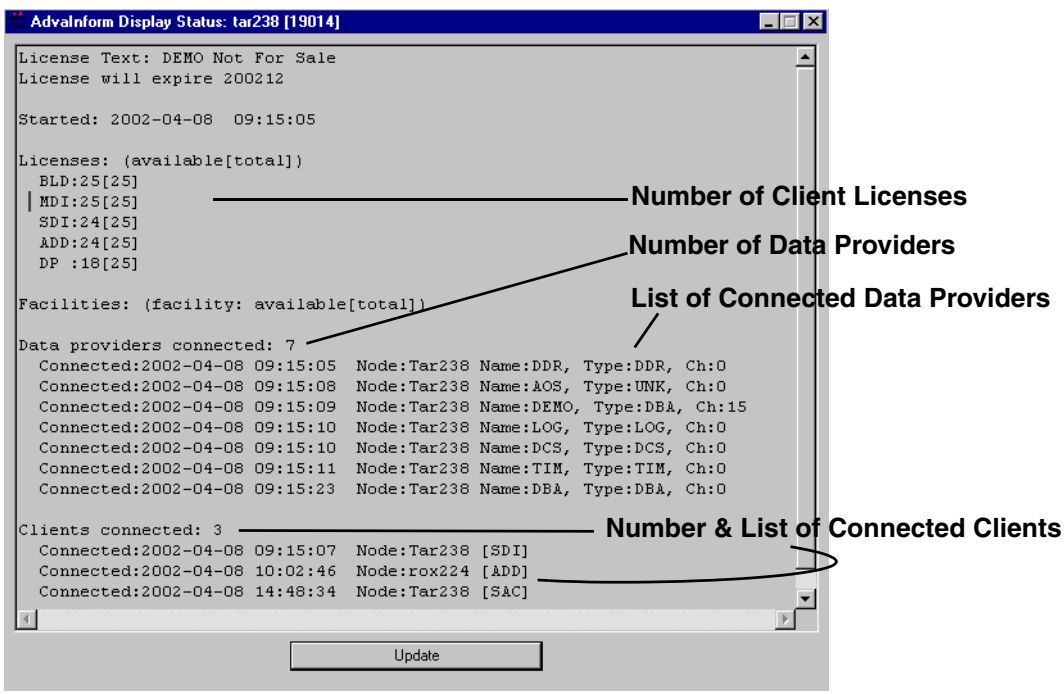


Figure 437. Display Server:COMM Window

Software Version Information

To check software version information for Information Management products and components, from the Windows task bar choose: **Start>Programs>ABB Industrial IT 800xA>Information Mgmt>Version Info**. This displays the Version Information window, Figure 438. To save the version information to a text file, choose **File>Save As** and specify the text file path.

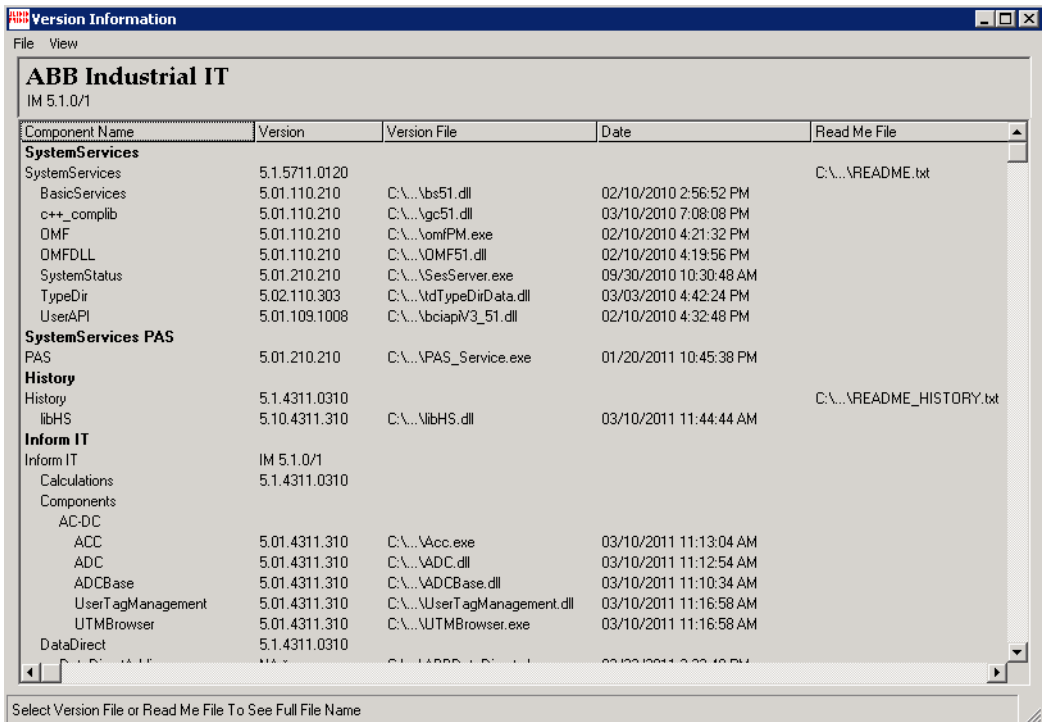


Figure 438. Software Version Information Tool

Disk Maintenance - Defragmenting

Information Management configuration procedures that involve the creating, deleting, and then recreating of a large quantity of objects may cause the associated disks to become fragmented. This, in turn, may impair the response time and general performance of the system. Procedures which cause fragmented files

include the configuration of the History database using the Bulk Import tool and deleting and then recreating an Aspect System. Check the system for fragmented files after any such procedure and defragment disks as required.

History configuration impacts not only the Information Management disk(s), but also the disk(s) on any Connectivity Servers where the operator trend function runs. Therefore, check the disks on those Connectivity Servers also.

Appendix A Extending OMF Domain to TCP/IP

Certain Information Management functions require the OMF domain to be extended to the TCP/IP network and all other ABB nodes that exist on the TCP/IP network. Some of these functions are:

- Consolidating history data from different Information Management servers.
- Using one Information Management server to schedule event-triggered data collection for logs that reside on a different node.
- Using a Display client to view History trend data for any Information Management node within the OMF domain.

Example Applications

The OMF TCP/IP domain can be defined by Multicast communication, point-to-point communication, or a combination of Multicast and point-to-point communication. Example applications are described in:

- OMF TCP/IP Domain with Multicast Communication.
- OMF TCP/IP Domain with Point-to-Point and Multicast.
- OMF TCP/IP Domain with Point-to-Point Exclusively.

OMF TCP/IP Domain with Multicast Communication

There may be many Information Management nodes on a large company's intranet. To keep separate logical areas of the system from conflicting with each other, OMF uses Ethernet's Multicasting capabilities to establish Domains on the TCP/IP network. This is done by configuring OMF on Information Management nodes that are to be in the same domain with the same Multicast address. This prevents OMF from communicating with any other OMF on the TCP/IP network having a different Multicast address.

Multicast addresses look very similar to IP addresses. In fact, Multicast addresses occupy a reserved section of the IP address range: 224.0.0.2 to 239.225.225.225. The default Multicast address used when multicasting is first enabled is 226.1.2.3.

The default address is okay for systems that have their own TCP/IP network (for small plants with no company intranet). For large companies with complex intranets connecting multiple sites, the default address is NOT recommended.

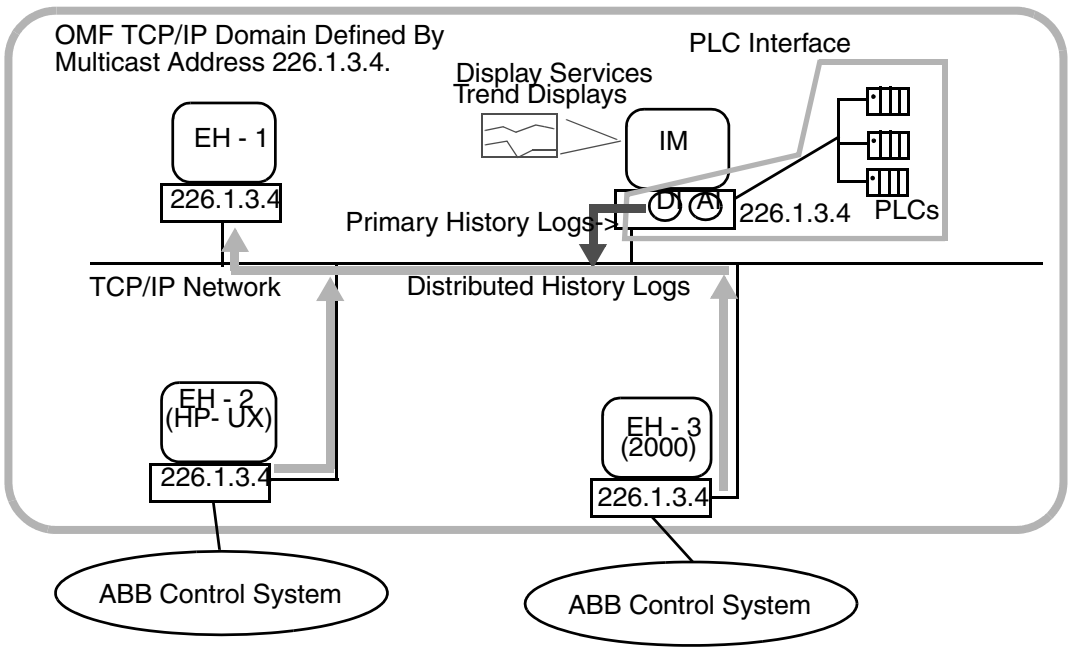


Figure 439. Example System Configuration Using Multicast Only to Establish Domain

Any valid address may be selected and assigned to each of the Information Management nodes that are required to be in the same Domain. Some companies have the network administrator maintain control over usage of Multicast addresses. This helps prevent crossing of Multicast defined Domains, and the problems that may result.

Once the OMF TCP/IP Domain where the Information Management node will reside is defined, use the Communication Settings dialog to enable the OMF TCP/IP socket, and assign the Multicast address. Use the following three fields in the lower left corner of this dialog: TCP/IP Multicast enabled, Multicast address, MulticastTTL. This procedure is described in Configuring OMF for TCP/IP on page 621.

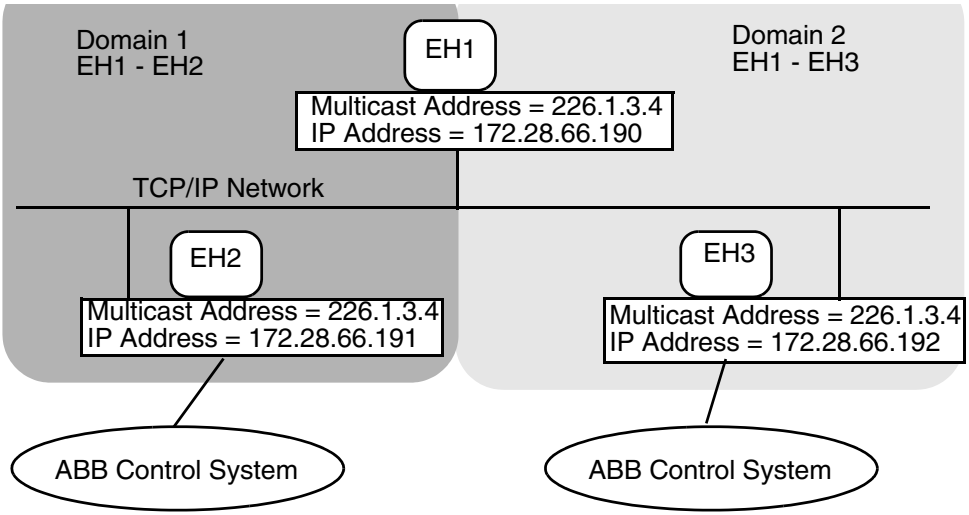
TCP/IP protocol must be configured before configuring the Multicast address, and enable the OMF TCP/IP socket. This is described in Configuring TCP/IP Protocol on page 621.

OMF TCP/IP Domain with Point-to-Point and Multicast

Point-to-point may be used in combination with Multicast to restrict access between certain nodes in the Multicast OMF TCP/IP domain. This is illustrated in Figure 440. EH1 must have OMF access to both EH2 and EH3, but at the same time OMF access must be prohibited between EH2 and EH3.

Again, use the Communication Settings dialog to configure TCP/IP Multicast enabled, Multicast address, MulticastTTL. This procedure is described in Configuring OMF for TCP/IP on page 621.

For point-to-point communication, modify the Socket Configuration List to match the table shown in Figure 440. This is described in Configuring OMF Socket Communication Parameters on page 624.



Socket Configuration Lists for EH1, EH2 & EH3

NODE	IP CONFIG	SEND/RECEIVE
EH1	172.28.66.191	RECEIVE
	172.28.66.192	RECEIVE
	multicast	SEND
EH2	172.28.66.190	SEND
	multicast	RECEIVE
EH3	172.28.66.190	SEND
	multicast	RECEIVE

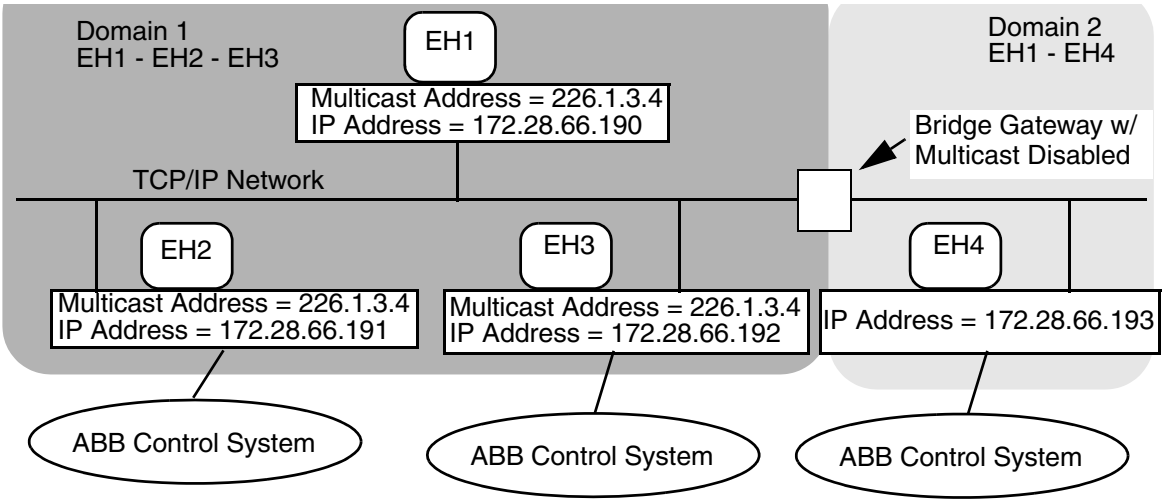
Figure 440. Example 1 - Point-to-Point Communication w/Multicast

Another application where point-to-point is used in combination with Multicast is where one node is located in an area where the network hardware does not allow Multicast. This is illustrated in Figure 441. EH1, EH2 and EH3 reside in an OMF/TCP/IP domain defined by multicast communication. EH4 is isolated from

this domain by network hardware that does not support multicast. Therefore a second domain is established for EH1 and EH4 for point-to-point communication.

In this case use the Communication Settings dialog to configure TCP/IP Multicast enabled, Multicast address, and MulticastTTL for EH1, EH2, and EH3. This procedure is described in Configuring OMF for TCP/IP on page 621.

The Socket Configuration Lists on EH1 and EH4 must be configured for point-to-point communication. For EH4, because all OMF communication to this node should be via point-to-point, Multicast should be disabled to prevent unwanted OMF communication on the default Multicast address. Do this by deleting Multicast from the Socket Configuration List. This is described in Configuring OMF Socket Communication Parameters on page 624. For EH2 and EH3, leave the Socket Configurations at the default values.



NOTE: By specifying one TCP/IP address as RECEIVE, all other addresses must also be specified as receive. This includes all multicast senders.

For example, the socket configuration list for EH1 has a multicast SEND/RECEIVE specification, but must also include a RECEIVE specification for EH4 since EH4 cannot use multicast. Therefore EH1 must also have RECEIVE specifications for all other nodes it is receiving from (EH2 and EH3).

Socket Configuration Lists for EH1, EH2 & EH3

NODE	IP CONFIG	SEND/RECEIVE
EH1	multicast	SEND/RECEIVE
	172.28.66.193	RECEIVE
	172.28.66.193	SEND
	172.28.66.191	RECEIVE
	172.28.66.192	RECEIVE
EH2	multicast	SEND/RECEIVE
EH3	multicast	SEND/RECEIVE
EH4	172.28.66.190	RECEIVE

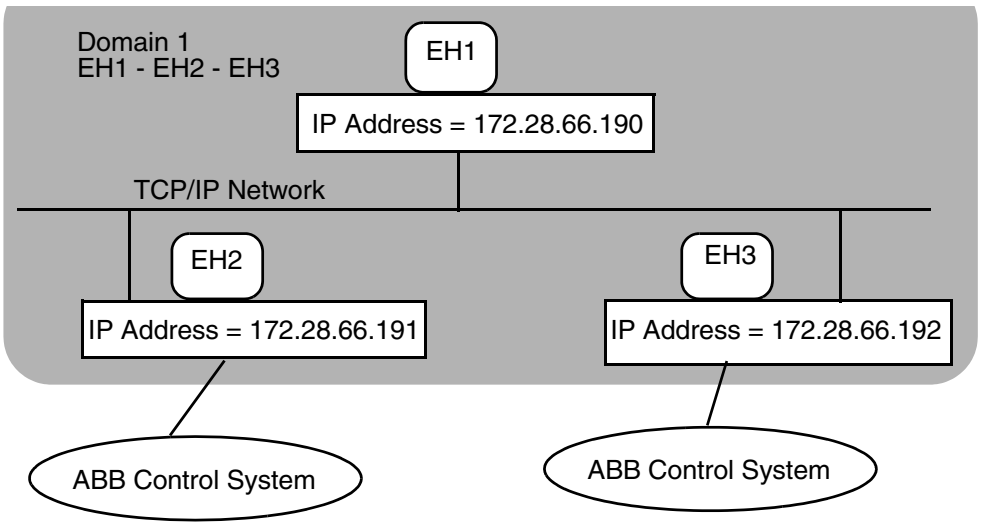
Figure 441. Example 2 - Point-to-Point Communication w/Multicast

OMF TCP/IP Domain with Point-to-Point Exclusively

This is required when the network hardware does not support, or is configured to not allow Multicast communication. Exclusive point-to-point communication is illustrated in Figure 442.

EH1, EH2 and EH3 reside in an OMF/TCP/IP domain defined by point-to-point communication. In this case, use the Communication Settings dialog to configure the Socket Configuration Lists in EH1, EH2, and EH3 for point-to-point communication. This is described in Configuring OMF Socket Communication Parameters on page 624.

All OMF communication to these nodes should be via point-to-point. Therefore, Multicast should be disabled to prevent unwanted OMF communication on the default Multicast addresses. Do this by deleting Multicast from the Socket Configuration Lists on each node. This is described in Configuring OMF Socket Communication Parameters on page 624.



Socket Configuration Lists for EH1, EH2 & EH3

NODE	IP CONFIG	SEND/RECEIVE
EH1	172.28.66.191	RECEIVE
	172.28.66.191	SEND
	172.28.66.192	RECEIVE
	172.28.66.192	SEND
EH2	172.28.66.190	RECEIVE
	172.28.66.190	SEND
	172.28.66.192	RECEIVE
	172.28.66.192	SEND
EH3	172.28.66.191	RECEIVE
	172.28.66.191	SEND
	172.28.66.190	RECEIVE
	172.28.66.190	SEND

Figure 442. Example - Exclusive Point-to-point Communication

Configuring TCP/IP Protocol

TCP/IP protocol must be configured before configuring the Multicast address, and enable the OMF TCP/IP socket. Typically, TCP/IP protocol is configured by a network administrator. Consult the network administrator to obtain the proper setting for the TCP/IP protocol, then check the TCPIP configuration for IP Address, Subnet Mask, and Default Gateway.

The TCPIP configuration properties are accessible via the Network Connections Tool in the Windows Control Panel.

Do not enable the TCP/IP socket unless this functionality is being used. If the TCP/IP socket is enabled and the TCP/IP connection is not operational, or a default gateway is not configured for routing, OMF will not start.

From the Windows Control Panel:

1. Select **Network Connections**.
2. In the network and Dialup Connections list, select **Local Area Connection**.
3. In the Local Area Connection Status dialog, click **Properties**.
4. In the LAN properties dialog, select **Internet Protocol (TCPIP)**, then click **Properties**. This displays the TCPIP properties dialog.

Configuring OMF for TCP/IP

The OMF parameters related to extending the OMF domain to TCP/IP are the Multicast address and TTL. In addition, OMF on TCP/IP must be enabled. This is done via the Communication Settings dialog, Figure 443. To launch this tool, from the Windows task bar, choose:
Start>Settings>Control Panel>Administrative Tools>PAS>Settings.

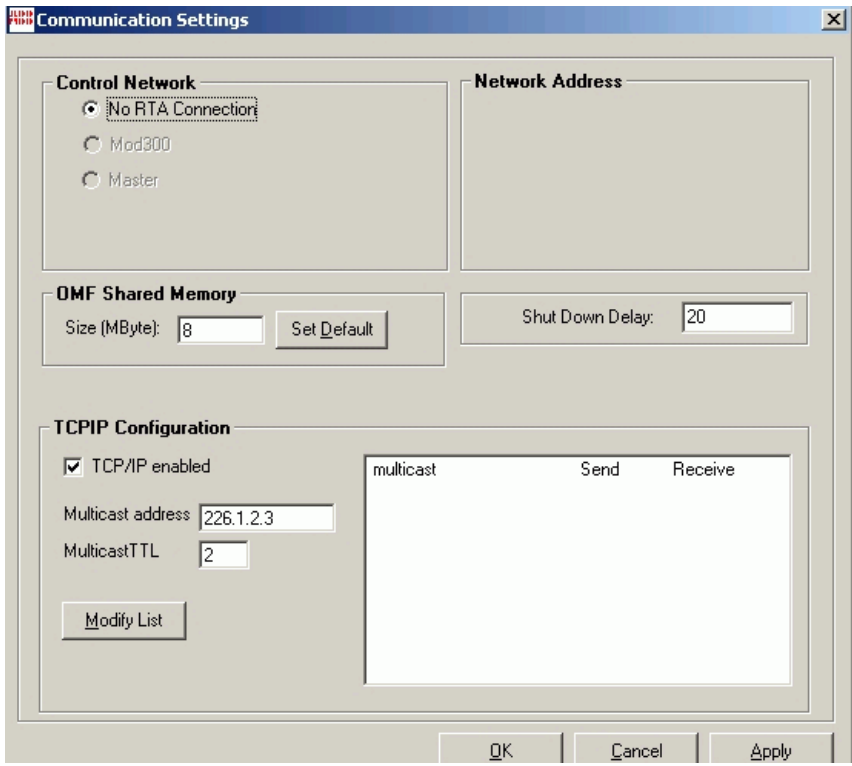


Figure 443. Communication Settings Dialog

Use the TCP/IP Configuration section to configure the required parameters and enable OMF on TCP/IP. Refer to Table 66 for details.

All processes under PAS supervision must be stopped and then restarted for any changes to take affect. Refer to Starting and Stopping History on page 437.

Table 66. Communication Settings Dialog

Field	Description
TCP/IP Multicast enabled	Enable or disable OMF for TCP/IP network via this check box (check indicates enabled). **Do not enable this parameter, unless this functionality is going to be used**. If distributed History logs are being used, then check TCP/IP Multicast enabled, and increase the OMF shared memory: • for History nodes that send History data to a consolidation node, add 5 meg to the OMF shared memory requirement. • for a consolidation node that collects from one or more History nodes, add 5 meg to the OMF shared memory requirement for each node from which it collects History data. For example, if a consolidation node is receiving from eight History nodes, the consolidation node will require 8*5 = 40 meg additional OMF shared memory.
Multicast address	This is the Multicast address used by OMF (omfNetworkExt and omfNameProc). To extend the OMF domain to TCP/IP, the Multicast Address must be the same in all nodes on a network, otherwise the nodes will not be able to see each other. A valid multicast group address that enables routing of multicast messages must be in the range: 224.0.0.2 to 239.225.225.225. The default multicast address is 226.1.2.3. This is a valid multicast address and can be used by OMF; however, it is recommended that this default address NOT be used. Not using the default minimizes the possibility of conflicts with other unknown nodes on the TCP/IP network. Contact the network administrator to obtain an appropriate multicast address to ensure secure communication between intended nodes.

Table 66. Communication Settings Dialog (Continued)

Field	Description
MulticastTTL	This is the *time-to-live* value which indicates the number of router hops to do before a message is discarded (not sent any more). This prevents endless message loops that may occur as a result of unexpected or unusual network partitioning. 0 = multicast messages are not sent at all 1 = multicast messages are sent only on the local subnet >1 = multicast messages are forwarded to one or more hops To extend the OMF domain to TCP/IP, this must be set >= 1; otherwise, nodes will not hear each other.
Socket Configuration List	The socket configuration specifies this node's scope of OMF access to other nodes in the domain. The default is **Multicast Send Receive**. This means this node can see and be seen by all other nodes in its multicast domain. Restrict OMF access between certain nodes in a multicast domain by configuring their respective socket configurations. For details on how to do this, refer to Configuring OMF Socket Communication Parameters on page 624.

Configuring OMF Socket Communication Parameters

Within an OMF domain, nodes establish OMF access with each other by periodically sending and receiving network maintenance messages that inform the respective nodes of each other's existence on the TCP/IP network. Each node must be capable of both sending messages to, and receiving messages from any node with which it wants to have OMF access. When the OMF communication socket is enabled, these OMF network maintenance messages are sent/received by either Multicast protocol (one message to many nodes), or point-to-point protocol (one message from one node to another).

By default, all nodes in a Multicast domain have access to each other. Change OMF access between nodes, or change the communication protocol for OMF maintenance messages in an OMF TCP/IP domain by configuring their respective socket configurations. To do this, click the **Modify List** button below the MulticastTTL field. This displays the Socket Configuration dialog, Figure 444.

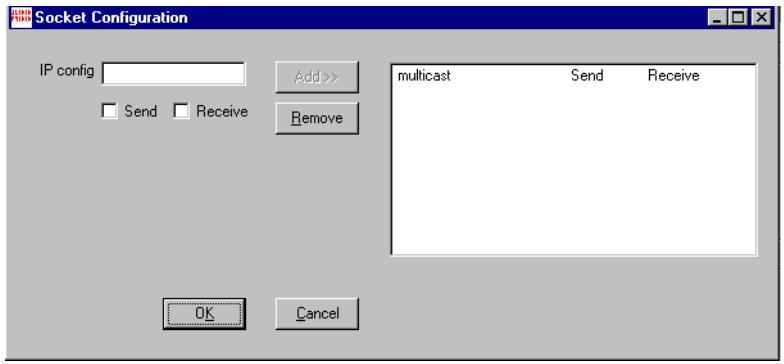


Figure 444. Socket Configuration Dialog

The default setting is **multicast Send Receive**. This means this node can see (receive the signal from) and be seen by (send the signal to) all other nodes in its domain.

To change access within the domain, edit the socket configuration list for all nodes in the domain. A summary of the possible entries is provided in Table 67.

Table 67. Socket Configuration Options

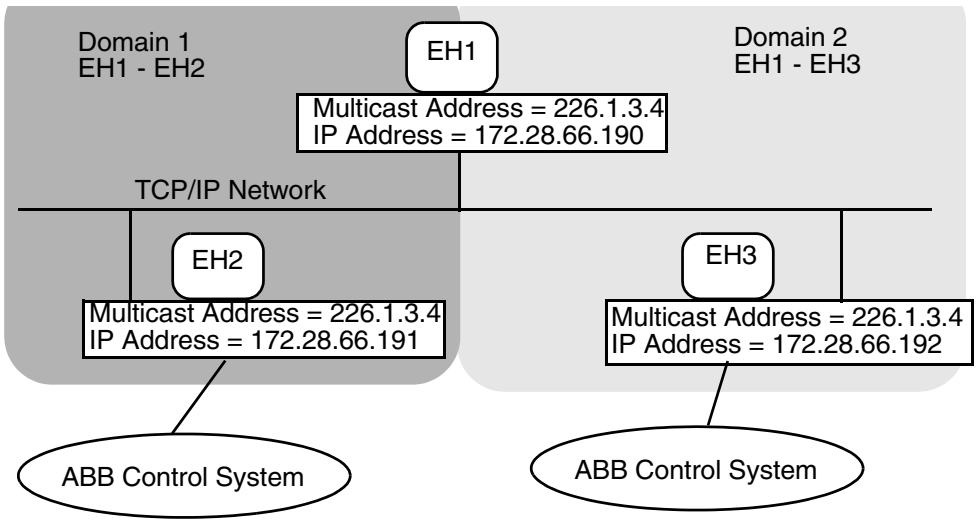
multicast SEND	This node can be seen by other nodes in the domain that are configured to RECEIVE network maintenance messages from this node. To do this, enter **multicast** in the IPconfig field, click the **Send** check box, then click **Add**.
multicast RECEIVE	This node can see other nodes in the domain that are configured to SEND network maintenance messages to this node. To do this, enter **multicast** in the IPconfig field, click the **Receive** check box, then click **Add**.

Table 67. Socket Configuration Options (Continued)

multicast SEND	This node can be seen by other nodes in the domain that are configured to RECEIVE network maintenance messages from this node. To do this, enter **multicast** in the IPconfig field, click the **Send** check box, then click **Add**.
point-to-point SEND	This node can only be seen by (send network maintenance messages to) the node whose IP address is specified. To do this, enter the node's IP address, click the **Send** check box, then click **Add**.
point-to-point RECEIVE	This node can only see (receive network maintenance messages from) the node whose IP address is specified. To do this, enter the node's IP address, click the **Receive** check box, then click **Add**.

Example Changing the Socket Configuration

In Figure 445, EH1 must have OMF access to both EH2 and EH3, but at the same time OMF access must be prohibited between EH2 and EH3. To implement OMF access in this way, each node's Socket Configuration List must be configured as described in the table in Figure 445. How to do this is described in the procedure following Figure 445.



Socket Configuration Lists for EH1, EH2 & EH3

NODE	IP CONFIG	SEND/RECEIVE
EH1	172.28.66.191	RECEIVE
	172.28.66.192	RECEIVE
	multicast	SEND
EH2	172.28.66.190	SEND
	multicast	RECEIVE
EH3	172.28.66.190	SEND
	multicast	RECEIVE

Figure 445. Example - Changing the Socket Configuration

To edit the Socket Configuration List for EH1:

1. Modify the default entry. To do this:

a. Select the entry in the list. This puts the value for each of the configuration parameters (IP config, Send, and Receive) in their respective fields.

b. Remove the check from the **Receive** box, Figure 446, then click **Add**.

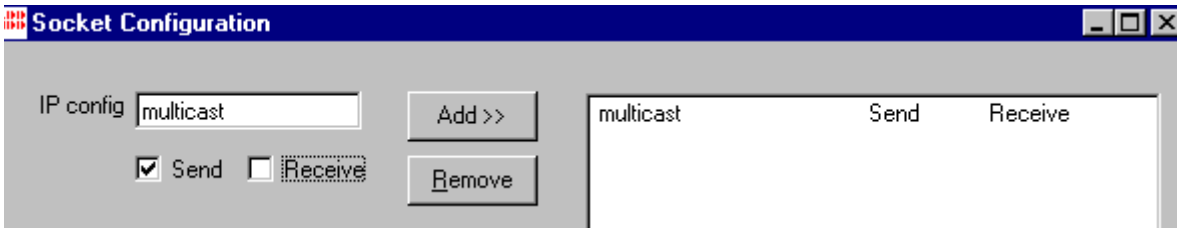


Figure 446. Modifying the Default Entry

This changes the entry to **multicast Send**. EH1 can now be seen by other nodes in the domain configured to receive from EH1; however, EH1 can only see nodes from which it is specifically configured to **Receive**.

2. Add an entry for EH2 to specify it as a node from which EH1 can receive:

   a. Enter the IP address for EH2, **172.28.66.191**, in the IP config field.

   b. Enter a check in the Receive check box, and make sure the Send box is unchecked. This configuration is shown in Figure 447.

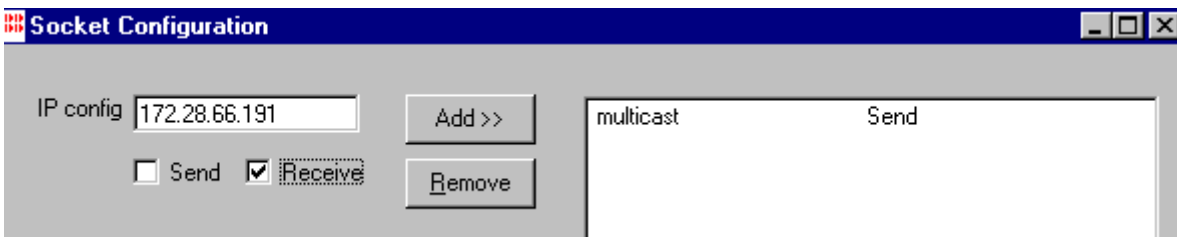


Figure 447. Adding an Entry for EH2

   c. Click **Add**. This adds a new entry: **172.28.66.191 Receive**.

3. Repeat step 2 for EH3, substituting the correct IP address (**172.28.66.192**) in the IP config field. The finished configuration for EH1 is shown in Figure 448.

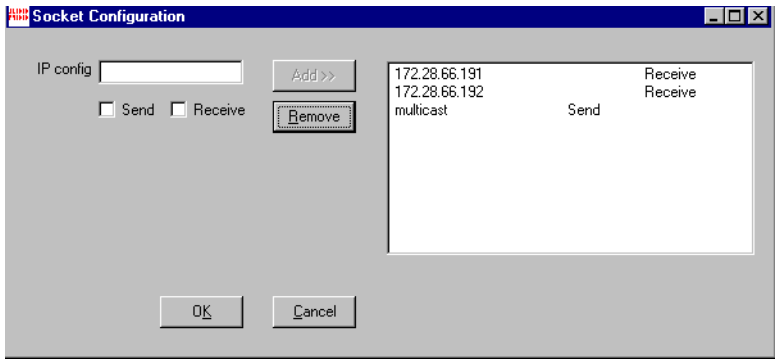


Figure 448. Socket Configuration for EH1

Repeat the above procedure to modify the socket configurations for EH2 and EH3. Refer to Figure 445 for the respective socket configurations.

To remove an entry from the Socket Configuration List, select the entry then click **Remove**.

OMF Shared Memory Size

The OMF shared memory size defines the size of shared memory for processes that use the OMF software. If the shared memory is too small, processes may be aborted without any error messages (PAS will discover that the process has been aborted).

The default size for OMF shared memory is 8,192 Kbytes (8 MBytes). Use the Communications Settings dialog to adjust the size. To launch this tool, from the Windows task bar, choose:
Start>Settings>Control Panel>Administrative Tools>PASS>Settings.

The OMF Shared Memory section on this tool displays the current size of the OMF shared memory in Megabytes. Enter a new value (in Mbytes) if necessary. (For details regarding this dialog, refer to Configuring OMF for TCP/IP on page 621.)

If distributed History logs are being implemented (this requires the TCP/IP Enabled option to be checked in the Communication Settings dialog), then increase the OMF shared memory:

- for History nodes that send History data to a consolidation node, add 5 meg to the OMF shared memory requirement.
- for a consolidation node that collects from History nodes, add 5 meg to the OMF shared memory requirement for each node from which it collects data. For example, if a consolidation node receives from eight History nodes, the consolidation node will require 8*5 = 40 meg additional OMF shared memory.

To change the OMF Shared Memory size and then set it back to the default, click **Set Default**. When finished, click **OK**.

Processes under PAS must be stopped and restarted for changes to take affect. Refer to Starting and Stopping History on page 437.

Shutdown Delay

This delays the Windows shutdown to let PAS shut down processes for History Services. The default is 20 seconds. The delay may need to be increased depending on the size of the History database. Use the Communication Settings dialog. To launch this tool, from the Windows task bar, choose:
Start>Settings>Control Panel>Administrative Tools>PAS>Settings.

Appendix B General Log Property Limits

Table 68 lists limits on general log properties. Refer to the section related to the specific log type for more detailed information.

Table 68. General Log Property Limits

Type	Property	Limitation
LogSet	Set Name	Length: 32 characters (if length limitation exceeded it will be truncated to size).
		Character
	Description	Length: 40 characters
		Character: none
Message Log	Access Name	Length: 64 characters (if length limitation exceeded it will be truncated to size).
		Character: no spaces
	Description	Length: 40 characters
		Character: none
	Capacity	Numeric: numbers ≤ 9999999
		Character: numeric only

Table 68. General Log Property Limits (Continued)

Type	Property	Limitation
Report Log	Access Name	Length: 64 characters (if length limitation exceeded it will be truncated to size).
		Character: no spaces
	Description	Length: 40 characters
		Character: none
	Capacity	Numeric: Numbers < 9999
		Character: numeric only
Numeric Log	Log Name	Length: 1,000 characters
		Character: no spaces. All characters except \ / : ? " < > | * ,

Appendix C Open Source Code and Copyright Information

This appendix contains information on the following open source code and third party software:

- cygwin
- mkisofs
- gzip
- Accusoft Corporation

cygwin

The System 800xA Information Manager includes the software **cygwin.exe** which is licensed to ABB Inc. pursuant to the GNU General Public License. The copy of that license can be obtained at:

http://www.gnu.org/copyleft/gpl.html

The source code for the **cygwin.exe** software is installed with System 800xA Information Manager in the following folder:

```
\ABB Industrial IT\Inform IT\History\cygwin
```

This folder is placed in `C:\Program Files` by default.

mkisofs

The System 800xA Information Manager includes **mkisofs.exe**. The source code for the **mkisofs.exe** software is installed with System 800xA Information Manager in the following folder:

```
\ABB Industrial IT\Inform IT\History\cdrtools
```

This folder is placed in `C:\Program Files` by default.

The source code is made available under the terms of the Common Development and Distribution License (CDDL) which can be obtained at:

http://www.opensource.org/licenses/cddl1.php

Any warranty or support offered by ABB for the **mkisofs.exe** software is offered only by ABB and not by the entity that first makes the software available under the CDDL and any terms in any ABB license for the software that differ from those in the CDDL are offered only by ABB.

gzip

The System 800xA Information Manager includes the software **gzip.exe** which is licensed to ABB Inc. pursuant to the GNU General Public License. The copy of that license can be obtained at:

http://www.gnu.org/copyleft/gpl.html

As the possessor of System 800xA Information Management, for the greater of three years after the possession of System 800xA Information Management or for as long as ABB Inc. offers spare parts or customer support for System 800xA Information Management, the user can send a request in writing that ABB Inc. should provide the source code for the **gzip.exe** software used in System 800xA Information Management. The written request should be sent to one of the addresses listed at the back of this User Manual. ABB Inc. will charge its reasonable cost for physically performing the actions necessary to send the source code and that cost may depend on the ship to address.

Accusoft Corporation

Information Manager contains portions of imaging code owned and copyrighted by Accusoft Corporation, Tampa, FL. ALL RIGHTS RESERVED.

Index

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

Revision History

Introduction

This section provides information on the revision history of this User Manual.

The revision index of this User Manual is not related to the 800xA 5.1 System Revision.

Revision History

The following table lists the revision history of this User Manual.

Revision Index	Description	Date
-	First version published for 800xA 5.1	June 2010
A	Updated for 800xA 5.1 Rev A	May 2011
B	Updated for 800xA 5.1 Rev A 64 Bits Updated the subsection **Expanding dB BlockBuffer Space**	August 2011 September 2011
C	Updated for 5.1 Rev B release	March 2012
D	Updated for 5.1 Rev E release Updated the **Appendix C**	July 2015

Updates in Revision Index A

The following table shows the updates made in this User Manual for System 800xA 5.1 Rev A.

Updated Section/Sub-section	Description of Update
Section 12, Consolidating Historical Data	Add a Note: This Delete operation deletes all the existing history data in the selected log.
Section 3, Adding a New SoftPoint Object Type	After step 2, a new step has been added as: In the New Object dialog select **SoftPoint Process Object Type** under Common tab.
Section 3, How to Delete a Signal Type	Step 2 has to be modified as: Press the **Delete** key on the keyboard or right-click the signal type and select **Delete** from the context menu.
Section 7, Table 19, Message Log Attributes	Column 1 needs to be modified as: Service Provider - When adding the Message Log object in the Node Administration structure, the Service Provider defaults to the Basic History Service Group for the selected node. This specification cannot be changed.
Section 9, Installing Add-ins in Microsoft Excel	Step7 has been modified as: Click **Browse**, then browse to **C:\Program Files\ABB Industrial IT\Inform IT\History\bin**. Select the file named **Inform IT Bulk Import.xla**.
Section 10, Table 30, Main Tab	From the table, the last two attributes *Calculation Type* and *Calculation Mode* has been modified as: *Collection Type* and *Collection Mode* respectively.
Section 11, Launching the Reel Report Configuration Tool	The path has been updated as: Start>Programs>ABB Industrial IT 800xA>Information Mgmt>Profiles>Reel Report Configuration.

Updated Section/Sub-section	Description of Update
Section 10, Reading the Remote Log Configurations	A step has been added between step 4 and step 5 as mentioned below: Enter the password of history account for the remote history server in the Password field or leave the Password field as <default>.
Appendix C, Open Source Code Information	A new topic *gzip* has been added.
Section 5, Configuring History Database	A new topic *Oracle User Password Guidelines* has been added.

Updates in Revision IndexB

The following table shows the updates made in this User Manual for System 800xA 5.1 Rev A1, 64 bits.

Updated Section/Sub-section	Description of Update
Section 13, Expanding dB Block Buffer Space	Updated Oracle version from 11.1.0.7.0 to 11.2.0.2.0 Added note for x86 and x64 operating systems.

Updates in Revision Index C

The following table shows the updates made in this User Manual for System 800xA 5.1 Rev B.

Updated Section/Sub-section	Description of Update
Section 12, Consolidating Historical Data, Using the Plug-in for IM Consolidation	Replaced image, *Consolidate Logs Aspect View.* Changed Step 3 **From**: Leave the **Unix Node ?** checkbox cleared, as it is not supported. **To**: The password is used to connect to Oracle on the remote IM server. If the source IM node is in the same domain and has the same service account as the destination IM, the field can remain <default>. If the source IM is in another domain, the Oracle Administration password for the remote IM must be entered. If the source IM password is updated, the consolidation action must be updated to reflect the change.
Section 7, Alarm/Event Message Logging, Creating an Inform IT Event Filter	Replaced the subsection *Creating an Inform IT Event Filter* completely with new content and images.
Section 7, Alarm/Event Message Logging, Creating an Inform IT Event Filter	Changed step 1 **From**: Select the Inform IT History Object in the Node Admin/Inform IT Historystructure. **To**: Select the Inform IT History Object in the Node Administration structure.

Updates in Revision Index D

The following table shows the updates made in this User Manual for System 800xA 5.1 Rev E.

Updated Section/Sub-section	Description of Update
Appendix C Open Source Code and Copyright Information	1. The name of Appendix C changed to "Open Source Code and Copyright Information" 2. We have added the following list items In the Introduction paragraph: • gzip • Accusoft Corporation 3. At the end of Appendix C the following paragraph has been added: **Accusoft Corporation** Information Manager contains portions of imaging code owned and copyrighted by Accusoft Corporation, Tampa, FL. ALL RIGHTS RESERVED.

Contact us

www.abb.com/800xA
www.abb.com/controlsystems

3BUF001092-510 D

Power and productivity
for a better world™