**LISTEN.
THINK.
SOLVE.**℠

FactoryTalk® Historian SE

INTRODUCTION TO PI SYSTEM MANAGEMENT

ALLEN-BRADLEY • ROCKWELL SOFTWARE

**Rockwell
Automation**

PREFACE – USING THIS GUIDE

About this Guide

This guide provides a starting place for new FactoryTalk Historian System Managers on Windows-based FactoryTalk Historian Systems. This guide:

- ❑ Introduces the FactoryTalk Historian System, PI Server and Interface Nodes.

- ❑ Explains PI system components, architecture, data flow, utilities and tools.

- ❑ Provides instruction for managing points, archives, backups, interfaces, security and trusts, and performance.

- ❑ Includes a glossary and resource guide.

This guide does not cover systems with PI Servers on VMS. For information on managing an OpenVMS-based PI Server System, see the VMS PI Server page on the Technical Support Web site (http://techsupport.osisoft.com/).

The PI Server Documentation Set

The PI Server Documentation Set includes seven user guides, described below.

Tip: Updated user guides, which provide the most up-to-date information, are available for download from the OSIsoft Technical Support Web site (http://techsupport.osisoft.com).

Title	Subject Matter
Introduction to FactoryTalk Historian System Management	A guide to the PI Server for new users and administrators. It explains PI system components, architecture, data flow, utilities and tools. It provides instruction for managing points, archives, backups, interfaces, security and trusts, and performance. It includes a glossary and resource guide.
PI Server Installation and Upgrade Guide	A guide for installing, upgrading and removing PI Servers on Windows and UNIX platforms, including cluster and silent installations.
PI Server System Management Guide	An in-depth administration guide for the PI Server, including starting and stopping systems, managing the Snapshot, Event Queue and Data Archive, monitoring system health, managing backups, interfaces, security, and moving and merging servers. Includes comprehensive instructions for using command-line tools: PIConfig, PIDiag, and PIArtool, and in-depth troubleshooting and repair information.
PI Server Reference Guide	A comprehensive reference guide for the system administrator and advanced management tasks, including: databases; data flow; PI Point classes and attributes, class edit and type edit; exception reporting; compression testing; security; SQL subsystem; PI time format; and overviews of the PI API, and PI-SDK System Management Tool (SMT).
Auditing the PI Server	An administration guide that explains the Audit Database, which provides a secure audit trail of changes to FactoryTalk Historian System configuration, security settings, and Archive Data. It includes administration procedures to enable auditing, to set subsystem auditing mode, to create and archive database files, and to export audit records.
PI Server Applications User Guide	A guide to key add-on PI Server Applications: Performance Equations (PE), Totalizer, Recalculator, Batch, Alarm, and Real-Time SQC (Statistical Quality Control). Includes a reference guide for Performance Equations, and Steam calculation functions.
PINet and PIonPINet User Guide	A systems administration guide, including installation, upgrade and operations, for PINet for OpenVMS and PIonPINet, which support migration and interoperability between PI2 and PI3 Systems.

Conventions Used in this Guide

This guide uses the following formatting and typographic conventions.

Format	Use	Examples
Title Case	▪ PI Client Tools ▪ FactoryTalk Historian System Elements ▪ PI Server Subsystems	▪ Use the client tool, FactoryTalk Historian ProcessBook, to verify that all data has been recovered. ▪ All incoming data is queued in the Event Queue by the Snapshot Subsystem.
Italic text	▪ Files, Directories, Paths ▪ Emphasis ▪ New Terms ▪ Fields ▪ References to a chapter or section	▪ The backup script is located in the *\PI\adm* directory. ▪ Archive files can be either *fixed* or *dynamic*. The archive receiving current data is called the *Primary Archive*. ▪ See Section 4.2, *Create a New Primary Archive.*
Bold Italic text	▪ References to a publication	▪ See the ***PI Server Reference Guide***.

Bold text	▪ System and Application components:
	  ▪ Subsystems
	  ▪ Tools / Utilities
	  ▪ Processes / Scripts / Variables
	  ▪ Arguments / Switches / Options
	  ▪ Parameters / Attributes / Values
	  ▪ Properties / Methods / Events / Functions
		▪ The Archive Subsystem, **piarchss**, manages data archives. **Piarchss** must be restarted for changes to take effect.
		▪ On UNIX, invoke site-specific startup script, **pisitestart.sh**, and on Windows, invoke **pisrvsitestart.bat**.
		▪ Three Point Database attributes affect compression: **CompDev**, **CompMin**, and **CompMax**. These are known as the *compression specifications*.

	▪ Procedures and Key Commands	▪ On the **Tools** menu, click **Advanced Options**. ▪ Press **CTRL+ALT+DELETE** to reboot

	▪ Interface components
	  ▪ Menus / Menu Items
	  ▪ Icons / Buttons / Tabs
	  ▪ Dialog box titles and options
		▪ Click **Tools** > **Tag Search** to open the **Tag Search** tool.
		▪ Click the **Advanced Search** tab.
		▪ Use the search parameters **PImean Value = 1**.

`Monospace type:` **"Consolas"** `font`	`Consolas monospace` is used for:
	▪ Code examples
	▪ Commands to be typed on the command line (optionally with arguments or switches)
	▪ System input or output such as excerpts from log files and other data displayed in ASCII text
	▪ **`Bold consolas`** is used in the context of a paragraph
		To list current Snapshot information every 5 seconds, use the **`piartool -ss`** command. For example:
		
		``
		$ piartool -ss
		   Counters for 7-Aug-05 14:35:56
		               Point Count: 10033 0
		           Snapshot Events: 1228011 0
		  Out of Order Snapshot Events: 130 0
		        Snapshot Event Reads: 392 0
		        Events Sent to Queue: 771952 0
		   Primary Capacity Remaining: 2540349 0
		``

[Light Blue - Underlined](#)	Links to URL / Web sites, and email addresses	http://www.osisoft.com/procedures.aspx support@osisoft.com

Related Documentation

OSIsoft provides a full range of documentation to help you understand and use the PI Server, PI Server Interfaces, and PI Client Tools. Each Interface has its own manual, and each Client application has its own online help and/or user guide.

The *UniInt End User Manual* describes the **OSIsoft Universal Interface (UniInt)***,* which is recommended reading for PI Server system managers. Many PI Interfaces are based upon **UniInt**, and this guide provides a deeper understanding of principals of Interface design.

Using PI Server Tools

The PI Server provides two sets of powerful tools that allow system administrators and users to perform system administration tasks and data queries.

❑ The **PI Server** includes many command-line tools, such as **pidiag** and **piartool**. The PI Server Documentation Set provides extensive instruction for performing PI Server administrative tasks using command-line tools.

❑ The **FactoryTalk Historian System Management Tools** (**SMT**) is an easy-to-use application that hosts a variety of different plug-ins, which provide all the basic tools you need to manage a FactoryTalk Historian System. You access this set of tools through a single host application. This host application is sometimes referred to as the SMT Host, but it is more commonly called **System Management Tools** or **SMT**.

You can download the latest version of **SMT** from the Technical Support Web site:

http://techsupport.osisoft.com

In addition to extensive online help that explains how to use all of the features in the **SMT**, the **SMT** includes the *Introduction to FactoryTalk Historian System Management* user guide.

QUICK TABLE OF CONTENTS

TABLE OF CONTENTS

TABLES AND FIGURES

Tables

Figures

Chapter 1. INTRODUCTION TO FACTORYTALK HISTORIAN SYSTEM MANAGEMENT

1.1 About this Book

This guide provides a starting place for new FactoryTalk Historian System Managers on Windows-based FactoryTalk Historian Systems. It introduces the FactoryTalk Historian System, PI Server and Interface Nodes, and gives you the FactoryTalk Historian System Management basics, including system backups, archive management, and security. This guide contains the following topics:

This guide does not cover systems with PI Servers on VMS. If you need information on managing an OpenVMS-based PI Server System, a good starting point is the VMS PI Server page on the Technical Support Web site (http://techsupport.osisoft.com/).

Chapter 2. SYSTEM MANAGER CHECKLIST

2.1   System Manager Checklist

This section provides a quick-reference table for checks that you need to perform regularly, to make sure that your FactoryTalk Historian System is working properly. This is sometimes called a "Daily Health Check" because we recommend you perform each of these checks each day.

The checklist is organized into functional areas, with a list of things to check in each area. This table does not provide detailed instructions for checking each item. To learn more about how to check a particular item, go to the section listed in the column on the right. You perform most of the checks using the FactoryTalk Historian System Management Tools (SMT). To learn how to get these tools, see *FactoryTalk Historian System Administration Tools* on page 5.

FactoryTalk Historian System Health Checklist

Area	Check to see that:	How to check it:
PI Server Subsystems	✓ Core PI systems and interfaces are running	SMT Server Process Manager (see *Checking whether the PI Server is Running* on page 12)
Archives	✓ All archives are loaded ✓ There are no gaps between archives ✓ There is an empty archive available for the next shift ✓ There is enough disk space for the new archives (for automatically created archives) ✓ The Archive data for a reference tag looks normal	SMT Archive Manager (see *Managing Archives* on page 25) Update Manager
Backups	✓ FactoryTalk Historian System backups have been run ✓ There is enough disk space for future backups ✓ The backup files are copied to the backup media or device	Check the files and disk (see Managing Backups on page 35)

Area	Check to see that:	How to check it:
Event Queue	✓ The Archive data flow is normal ✓ The Snapshot data flow is normal ✓ The Event Queue flow is normal ✓ There is enough disk space available for the Event Queue(s) ✓ There are no unexpected out-of-order events	PI Performance Monitor points (see *Monitoring FactoryTalk Historian System Performance* on page 57)
Message Log	✓ There are no errors or unusual events in the message logs	SMT Message Log viewer
Connections	✓ No unusual connection losses/reconnections ✓ No stale connections are accumulating ✓ There are no network errors	SMT Network Manager Statistics
Data Sources	✓ The I/O rate tag trends look good ✓ There are no error messages in the *pipc.log file*	Check I/O rate tags (see *Monitoring Interface Performance* on page 44)
Points (Tags)	✓ There are no stale or bad tags	SMT Stale and Bad Tags (see *Finding Malfunctioning Points* on page 23)
Technical Support Site	✓ There are no new bulletins ✓ There are no software upgrades or patch releases you should install	Technical Support Web Site (http://techsupport.osisoft.com/)

Chapter 3. FACTORYTALK HISTORIAN SYSTEM ADMINISTRATION TOOLS

3.1 Getting and Using the Tools You Need

OSIsoft provides two tools that make it much easier to manage a FactoryTalk Historian System. If you haven't already installed these two tools, do it now:

❑ *The FactoryTalk Historian System Management Tools (SMT)* on page 5
❑ *The PI Interface Configuration Utility (ICU)* on page 6

Make a habit of checking the Technical Support Web Site (http://techsupport.osisoft.com) regularly for updates to these tools.

OSIsoft also provides some other helpful tools that you might want to look at, if you have the time:

❑ *Using Other PI Tools* on page 7

3.1.1 The FactoryTalk Historian System Management Tools (SMT)

The FactoryTalk Historian System Management Tools (SMT) is set of easy-to-use plug-ins, that provide all the basic tools you need to manage a Windows-based FactoryTalk Historian System. You access this set of tools through a single host application, called the SMT Host. The SMT Host is more commonly called the System Management Tools or simply, **SMT**.

How to Install SMT

You can get the latest version of SMT on the Technical Support Web Site (http://techsupport.osisoft.com):

1. On the **Products** menu, point to **System Management** and click on **PI SMT 3**.
2. On the right side of the page, under **Docs and Downloads**, left-click on **Install Kits**. At this point, you might need to log into the Web site, if you haven't already.
3. Click the **Download** button for the System Management Tools install kit.
4. After downloading the install kit, double-click the **Setup** executable to install SMT.

How to Run SMT

To run SMT, on the Windows **Start** menu, point to **Programs**, point to **FactoryTalk Historian System**, and then click **FactoryTalk Historian System Management Tools**.

How to Select a Server in SMT

SMT includes a PI Servers pane, which lists all the available Servers. Most tasks are easier in SMT, if you select a single PI Server on which you want to perform that task. To select a PI Server in SMT:

1. Open the **SMT Console** (How to Run SMT on page 5).

2. From the **PI Servers** pane, click the box next to the PI Server that you want to work with.
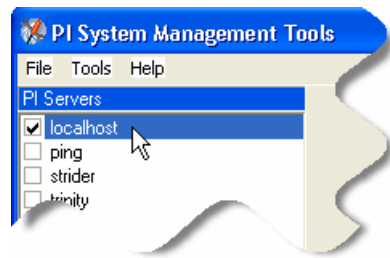
You can select more than one server at a time.

3.1.2 The PI Interface Configuration Utility (ICU)

The PI Interface Configuration Utility (ICU) is a tool that makes it easy to configure and manage your PI interfaces. Install the ICU on each of your Interface Nodes.

How to Install the ICU

You can get the latest version of the ICU on the Technical Support Web Site (http://techsupport.osisoft.com):

1. On the **Products** menu, point to **Interfaces** and click on **PI Interface Configuration Utility**.

2. On the right side of the page, under Docs and Downloads, left-click on **Install Kits**. At this point, you might need to log into the Web site, if you haven't already.

3. Click the **Download** button for the PI Interface Configuration Utility (ICU) install kit.

4. After downloading the install kit, double-click the **Setup** executable to install the **ICU**.

How to Run the ICU

The ICU is a point-and-click tool for configuring interfaces. You can only run it directly on the Interface Node where the interface is installed.

To run the ICU, go to the Interface Node and, on the Windows Start menu, point to Programs, point to FactoryTalk Historian System, and then click PI Interface Configuration Utility.

The first time you run the ICU on an Interface Node, you need to register each interface. Also make sure that the interface supports the ICU utility. See *Configuring Interfaces* on page 43.

3.1.3 Using Other PI Tools

In addition to SMT and the ICU, OSIsoft provides many other tools that are useful to System Managers. The most current tools are always available on the Technical Support Web Site (http://techsupport.osisoft.com). Here are a few that are good to know about:

- ❑ **PI TagConfigurator:** An Excel plug-in that allows you to create new tags and modify the attributes of existing tags from a spreadsheet.

- ❑ **PI Module Database Builder:** An Excel plug-in that allows you to view and modify items from the Module Database in an Excel spreadsheet.

- ❑ **PI SQC Alarm Manager:** Used to manage the Real Time SQC Alarms on PI Servers.

3.1.4 Using the Windows Command Interpreter

As a FactoryTalk Historian System Manager, you sometimes need to use one of PI's command line utilities. To do this, you need to open a Windows command prompt window:

1. On the Windows **Start** menu, click **Run**. The Windows Run dialog box appears.

2. In the **Open** text field, type:

   `cmd`

3. Click **OK**. A Windows command prompt window appears.

4. Use the **cd** command to change to the directory that contains the PI utility. For example to change to the *PI\adm* directory on the D drive, you would type:

   `cd /D D:\PI\adm`

Chapter 4. INTRODUCTION TO THE FACTORYTALK HISTORIAN SYSTEM

The FactoryTalk Historian System collects, stores and manages data from your plant or process. You connect your data sources to one or more PI Interface Nodes. The Interface Nodes get the data from your data sources and send it to the PI Server. Users get data from the PI Server and display it with client tools (for example, ProcessBook, DataLink or RtWebParts).
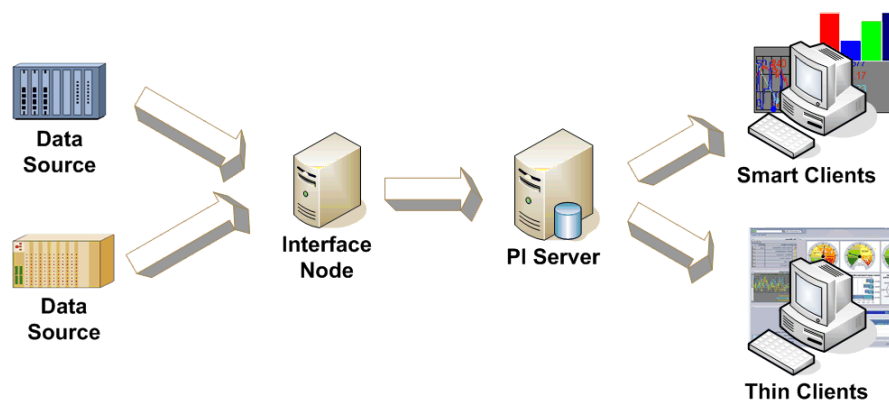
Overview of FactoryTalk Historian System Data Flow

- ❑ **Data Sources:** Your data sources are the instruments that generate your data. They can be almost anything and they can connect to the Interface Nodes in a variety of different ways. PI's Performance Equations, ACE, and Totalizer are also all data sources. See *Managing Data Source Equipment* on page 71 for more information about Data Sources.

- ❑ **Interface Nodes:** Interface Nodes run PI interfaces. PI interfaces get the data from the data sources and send it to the PI Server. Each different data source needs a PI interface that can interpret it. OSIsoft has over 300 different interfaces. For more information, see *About PI Interface Nodes* on page 10 and *Managing* Interfaces on page 41.

- ❑ **PI Server:** The PI Server stores the PI data and acts as a data server for Microsoft Windows-based client applications. You can also use the PI Server to interact with data that is not stored in PI (external systems). For more information, see *About the PI Server* on page 11.

❑ **Clients:** Operators, engineers, managers and other plant personnel use a variety of client applications to connect to the PI Server to view plant data. For more information, see *About Client Applications* on page 16.

4.1   About PI Interface Nodes

OSIsoft provides specialized interface programs (interfaces) for each data source. These interfaces typically run on a dedicated system, called an Interface Node, which connects both to the data sources and to the PI Server. For historical reasons, Interface Nodes are also sometimes referred to as API Nodes or Data Source Nodes.

Interface Nodes can run multiple interfaces to multiple PI Servers. The Interface Node might be a machine that is a part of the foreign data system, or a stand-alone dedicated interface machine, or even a PI Server itself (PI to PI).

❑ *Data Flow on the Interface Nodes*

❑ *About Buffering*

Data Flow on the Interface Nodes

The PI Server stores data in the form of events. Each event has a value and a timestamp that tells you what time the value was collected. The interfaces collect data from the data sources and typically use exception reporting, meaning that they pass significant events on to the PI Server and discard the rest. If the buffering service *(Managing Buffering* on page 63) is configured on the interface node, then the events go through the buffering service. If the Interface Node cannot connect to the PI Server, the buffering service holds the data until the Server connection is restored.

What is Exception Reporting?

The point of exception reporting is for the interface to send you the data you are interested in, rather than taxing the network connection by sending a lot of data that is not meaningful.

Exception reporting uses a simple dead band algorithm to determine whether to send events to the PI Server. For each point, you can set exception reporting specifications that create the dead band. The interface ignores values that fall inside the dead band.
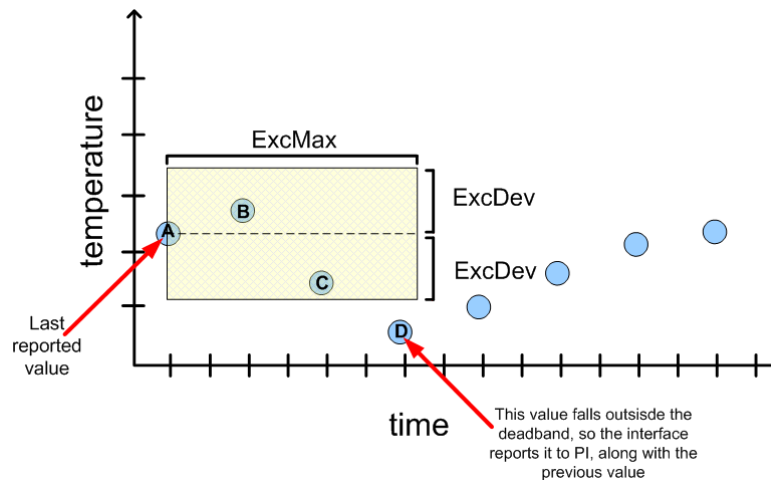
In the preceding illustration, values A, D and C are reported to the PI Server. Value A is the last reported value, values B and C fall within the exception dead band, but value D falls outside the deadband, so the interface reports value D and the previous value, in this case, value C.

The interface uses the point's **ExcDev**, **ExcMin** and **ExcMax** attributes to decide whether to report the new value to PI:

❑ The **ExcDev** (or **ExcDevPercent**) attributes determine how much a point's value needs to change before the interface sends it to the Server. For example, a 12 bit A/D converter can never be more precise than 1 part in 4096.

❑ The **ExcMax** attribute sets a limit on how long the interface can go without reporting a value to PI. After the **ExcMax** time period, the interface sends the next new value to PI, regardless of whether the new value is different from the last reported value.

❑ The **ExcMin** attribute sets a limit on how frequently the interface can report values. For example, if you want the interface to wait a full ten minutes before reporting a new value to the PI Server, then you would set the **ExcMin** attribute to ten minutes.

For details on setting exception reporting attributes, see *Exception Specifications* on page 20. Some interfaces do not support exception reporting. See the documentation for your interface to determine whether it supports this capability.

4.2 About the PI Server

The PI Server is the heart of your FactoryTalk Historian System. It gets the data and routes it in real time throughout the FactoryTalk Historian System and your entire information infrastructure, making it possible for everyone to work from a common set of real data. Operators, engineers, managers, and other plant personnel can connect to the PI Server and view manufacturing data from PI Data Storage or from external data storage systems.

❑ *What's in the PI Directory?*

❑ *File System Dos and Don'ts*

❑ *Checking whether the PI Server is Running*

❑ *Data Flow in the PI Server*

4.2.1   What's in the PI Directory?

By default, the PI Server installs its files in a folder called PI on the disk with the most available space, but you can choose a different location for PI during installation. Within the PI directory, the PI Server installs the subdirectories listed in the table below.

Directory	Contents
adm	Contains administrative tools.
bin	Contains subsystem or PI service executables.
dat	Contains databases such as points and digital states. This is also the default directory for archives.
log	Contains log files.
setup	Contains files for install and uninstall.

4.2.2   File System Dos and Don'ts

The two most important tips on the file system are:

❑ **Disable virus scanning on the archive folder.** Virus scanning may corrupt archive files. The problem with virus scanning is that, because the data is random, it might have a bit pattern that matches a known virus signature. The virus scanning software then locks and quarantines the primary archive.

❑ **Don't use the Windows File System Compression feature on the PI Server.** When you use compressed files, you slow down PI's access to archive files. The compression might save disk space, but it requires more CPU resources.

4.2.3   Checking whether the PI Server is Running

The PI Server consists of several modules, including a set of core subsystems. To check whether the PI Server is running, you simply check that the core subsystems are running.

Core Subsystem	What It Does
PI Archive subsystem	The PI Archive subsystem stores and serves the data after it comes out of the Snapshot subsystem.
PI Backup subsystem	The PI Backup subsystem controls the backup of the PI Server.
PI Base subsystem	The PI Base subsystem maintains the point configuration data. This subsystem also hosts the PI Module Database.
PI License Manager	The PI License Manager maintains licensing information for the PI Server and all connected applications

Core Subsystem	What It Does
PI Message subsystem	The PI Message subsystem records status and error messages for the PI Server in a log file.
PI Network Manager	The PI Network Manager provides the connection between all the subsystems in the Server. This subsystem also manages network connections between the FactoryTalk Historian System and client applications.
PI Shutdown subsystem	Determines when the PI Server was stopped and writes shutdown events to points configured to receive these events (runs only at startup and then stops on non-Clustered PI Servers)
PI Snapshot subsystem	The PI Snapshot subsystem stores the most recent event for each point. It applies compression, sends data to the Event Queue, and serves Snapshot events to the client applications.
PI SQL Subsystem	The PI SQL Subsystem processes SQL statements, including those submitted by the PI ODBC Driver.
PI Update subsystem	The PI Update subsystem sends notifications of changes in values or point attributes to any interface or client application that is signed up for notification.

To check whether the core subsystems are running, you can use the Server Process Manager plug-in in the FactoryTalk Historian System Management Tools (SMT):

1. Open the FactoryTalk Historian System Management Tools and select the Server you want to check (see *The FactoryTalk Historian System Management Tools (SMT)* on page 5).

2. In the System Management plug-ins list, under **Operation**, choose **Server Process Manager**. The list of processes appears.

3. If the core subsystems (except Shutdown on non-Clustered PI Servers) are running, then the PI Server is running. The Shutdown subsystem runs only when the PI Server is starting up, so Shutdown is listed as **Stopped**.
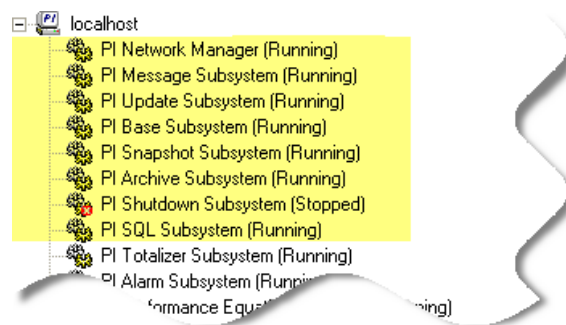
In addition to the core subsystems, the Server Process Manager lists the status of optional subsystems, such as Batch and Performance Equation Scheduler. These optional subsystems do not need to be running in order for the PI Server to be running.

4.2.4　Data Flow in the PI Server

When the PI Server gets a new event from an interface or manual input program, it sends the event to the Snapshot Subsystem. The Snapshot Subsystem holds a single value for each PI point in memory. When a new value comes in, the PI Server sends the old value to the Archive Subsystem. The Archive Subsystem performs compression testing on the value and either discards it or sends it on to the Event Queue, depending on the result of the test.

4.2.5　What is the Snapshot?

The Snapshot Subsystem gets the new data from the Interface Node and holds the most recent value for each point. This most recent value is called the Snapshot for that point.
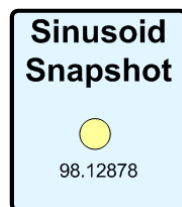
When a new event comes in, it becomes the Snapshot for that point. The PI Server evaluates the previous Snapshot according to the compression specifications and either sends the new value to the Event Queue or discards it.
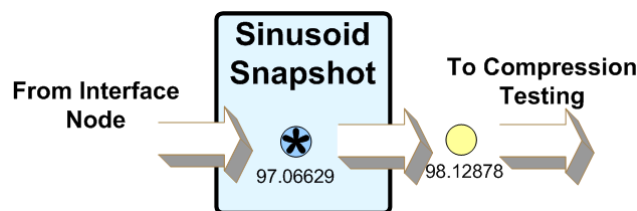
These values in the Snapshot Subsystem are called Snapshot events or, sometimes, just Snapshots. The collection of all the Snapshot values for all the points is the Snapshot database.

4.2.6　What are Out of Order Events?

An out of order event is an event that enters the Snapshot Subsystem with a timestamp that is older than the current Snapshot value. The PI Server sends out of order events directly to the Event Queue for archiving, without compression testing.

4.2.7　What is Compression Testing?

The Archive Subsystem uses compression testing to determine what events need to be saved in the Archive. The point of compression testing is to store just enough data to accurately reproduce the original signal.

For example, in the following illustration, all the events fall on the same straight line. In a simple case like this, you don't actually need to store all the points on the line. If you store just two points, you can exactly recreate the point value for any other time.
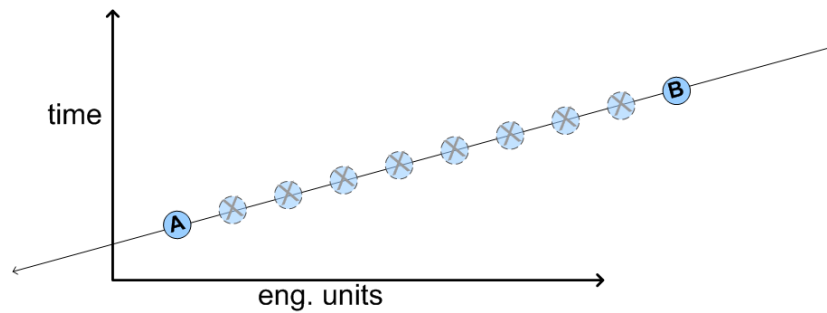
The same principle applies to compressing real-world data. The PI Server uses a sophisticated compression algorithm to determine which events it needs to keep in order to provide an accurate data history. The **CompDev**, **CompMin** and **CompMax** attributes allow you to control the "granularity" of the compression algorithm.

- ❑ The **CompMin** and **CompMax** attributes give you some control over how often the PI Server should save a new value for a particular point.

- ❑ **CompDev** and **CompDevPercent** allow you to decide how much a point's value needs to change in order for PI to save it.

For details on setting compression testing attributes, see *Compression Specifications* on page 21.

4.2.8 What is the Event Queue?

The PI Event Queue serves as a memory buffer between the Snapshot and Archive Subsystems. The Snapshot Subsystem adds data to this queue while the Archive Subsystem removes data from the queue.
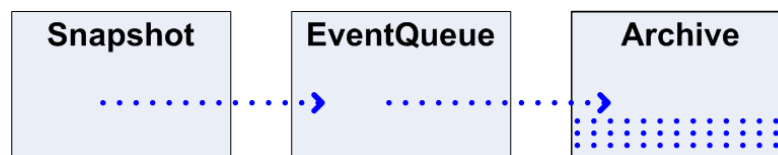
Normally the Event Queue passes events through to the Archive as quickly as they arrive, but in some circumstances the Archive Subsystem might be too busy or an archive file might be unavailable. When this happens, the Event Queue holds the data, filling until the Archive is again available. This is called Archive Queuing.
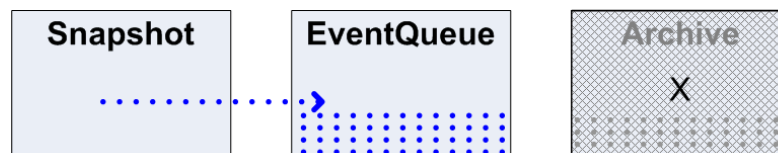
The most common causes of Archive Queuing are:

- ❑ The archives are unavailable because Archive Shift or archive backups are occurring.

❑ The Archive Subsystem is busy because incoming events are out of order.

4.3   About Client Applications

The PI Server works with a wide range of client applications, from ProcessBook and DataLink to RtWebParts. Some of the client software packages available from OSIsoft are listed in the table below.

Software	Description
FactoryTalk Historian ProcessBook	An easy-to-use graphics package that allows users to create dynamic, interactive graphical displays featuring real-time PI data.
PI ProfileView	Creates a comprehensive display of surface data for monitoring sheet products.
FactoryTalk Historian DataLink	Allows PI to access and deliver data to and from spreadsheet programs and create easy-to-read reports.
PI Control Monitor	Oversees plant control systems, ensuring accuracy, and keeps a historical system record.
PI BatchView	Displays PI Batch data on Windows desktop computers.
PI Manual Logger	Used to organize and manually enter data from handheld loggers, computer terminals, scanners, and other input devices into the FactoryTalk Historian System.
PI AlarmView	Summarizes PI Alarm server information and displays those data in a hierarchical tree structure to any number of clients on- or off-site.
PI ActiveView	Seamlessly renders existing FactoryTalk Historian ProcessBook displays for the Web.
RtWebParts	Web parts that work with RtBaseline Services to display PI and other data in various ways.

Chapter 5. MANAGING PI POINTS

This section gives you a brief introduction to PI points and point attributes and then covers the basic point-related tasks that a System Manager needs to know how to do:

- ❑ *About Points*
- ❑ *About Point Attributes*
- ❑ *Creating New Points*
- ❑ *Finding Malfunctioning Points*
- ❑ *Decommissioning Points*
- ❑ *Deleting Points*

This material is an introduction to PI points. For comprehensive documentation on PI points, see the ***PI Server Reference Guide.***

5.1 About Points

Points, sometimes also called tags, are the basic building blocks of a PI system, because they are how you track the events that comprise your data history. When the System Manager or OSIsoft Field Services engineer installs a PI Server, he creates a PI Point for every source of data that the FactoryTalk Historian System needs to track.

Each point has more than 50 attributes (*About Point Attributes* on page 17) that define exactly how the data should be collected for that point. These attributes determine how frequently the point gets new values, the data type of the point values (whether integer or string, for example), who is allowed to view and/or edit the point, and so on. The PI Base Subsystem stores points and their attributes in the Point Database.

> **Note:** Some PI interfaces are compatible with PI Auto Point Sync (PI APS), which tracks changes in foreign data systems and automatically updates the PI Points Configuration to reflect those changes.

5.2 About Point Attributes

Point attributes are where you configure how and when PI should collect data from a particular data source. Point attributes specify the data source location, how often PI should

get new values from the data source, which values PI can ignore and which represent valid data, and much more.

Point attributes are fully documented in the **_PI Server Reference Guide_**. This section gives you a brief overview of a few key attributes:

❑ _Point Name: Tag Attribute_

❑ _Class of Point: PtClass Attribute_

❑ _Data Type of Point: PointType Attribute_

❑ _Data Source: PointSource Attribute_

❑ _Interface ID Number: Location1 Attribute_

❑ _Setting Scan Class: Location4 Attribute_

❑ _Exception Specifications_

❑ _Compression Specifications_

❑ _Point Value Range: Zero, Span and Typical Value_

❑ _Configuring Shutdown Events: Shutdown_

❑ _Point Security: PtOwner, PtGroup, PtAccess, DataOwner, DataGroup, DataAccess_

There are more than 50 different point attributes that you can specify for each point. The exact list of attributes that configures a point depends on the class of the point (see _Class of Point: PtClass Attribute_ on page 18).

5.2.1   Point Name: Tag Attribute

The **Tag** attribute specifies the name of the point. Many PI users use the terms tag and point interchangeably, which is fine. Technically though, the tag is actually just the name of the point. Follow these rules for naming PI points:

❑ The name must be unique on the PI Server

❑ The first character must be alphanumeric, the underscore (_), or the percent sign (%)

❑ No control characters are allowed; such as linefeeds or tabs

❑ The following characters are not allowed:
    `* ' ? ; { } [ ] | \ ` ' "`

5.2.2   Class of Point: PtClass Attribute

The attributes that you need to configure for a particular point depend on what the point is for. PI provides several different classes of points, each of which provides a slightly different set of attributes to work with. You can also build your own point classes.

Points that represent data from a PI interface are always in the Classic point class. The list of available PI point classes is as follows:

Classic: The Classic point class includes attributes used by interfaces.

Base: The Base class is a common set of attributes that all point classes include. The Base class includes both system-assigned and user-assigned attributes. This is the minimum set of attributes that a PI point needs in order to function.

Alarm: The Alarm class is used for alarm points. See the *PI Server Applications Guide* for more information on Alarm points.

Totalizer: The Totalizer class is for a type of point that represents a running total of data. There are many different kinds of Totalizer points. For more information on Totalizer points, see the *PI Server Applications Guide* and the SMT help topic for the Totalizer Editor.

SQC_Alarm: The SQC_Alarm class is for SQC_Alarm points. See the PI Server Applications Guide for more information on SQC_Alarm points.

5.2.3 Data Type of Point: PointType Attribute

Use the **Type** attribute to specify the data type of the point values.

Point Type	When to Use It
Digital	Use the Digital point type for points whose value can only be one of several discrete states, such as ON/OFF or Red/Green/Yellow.
Int16	Use the Int16 point type for points whose values are integers between 0 and 32767 (15-bit unsigned integers).
Int32	Use the Int32 point type for points whose values are integers between -2147450880 and 2147483647 (32-bit signed integers).
Float16	Use the Float16 point type for floating point values, scaled. The accuracy is one part in 32767.
Float32	Use the Float32 point type for single-precision floating-point values, not scaled (IEEE floating points).
Float64	Use the Float64 point type for double-precision floating-point values, not scaled (IEEE floating points).
String	Use the String point type for strings of up to 976 characters.
Blob	Blob stands for Binary Large Object. Use the Blob point type to store any type of binary data up to 976 bytes.
Timestamp	Use the Timestamp point type for any time/date in the range 1-jan-1970 to 1-Jan-2038 Universal Time (UTC).

5.2.4 Data Source: PointSource Attribute

The **PointSource** attribute specifies which interface is the data source for this point. Set the **PointSource** attribute to match the point source character for the interface (see *What's a Point Source?* on page 41).

The default point source is **L**, which stands for "Lab". Depending on your installation, the default point source is either **L** or **Lab.** Use **L** or **Lab** for points that are not associated with any interface to specify lab-input points.

5.2.5 Interface ID Number: Location1 Attribute

The **Location1** attribute is only for interface points, meaning points that get their data from a PI interface, as opposed to some other source. Most interfaces use the Location1 attribute to specify the interface ID number (described in *What's an Interface ID Number?* on page 42).

5.2.6 Setting Scan Class: Location4 Attribute

The **Location4** attribute is only for interface points, meaning points that get their data from a PI interface, as opposed to some other source. Each PI interface has one or more scan classes for scheduling data collection (See *What's a Scan Class?* on page 42). You set the Location4 attribute for a point to specify which of the interface's scan classes you want to use.

> **Note:** Most interfaces require you to use the **Location4** attribute to set the scan class, however there are exceptions, particularly among older interfaces. Also, some interfaces get data on command, rather than scanning. Always check the documentation for the interface.

5.2.7 Exception Specifications

Exception reporting specifications determine which events the interface sends to PI and which it discards. To learn more about exception reporting, refer to *What is Exception Reporting* on page 10.

Each point can set the following three attributes to configure the exception reporting specifications:

Specification	Attribute	How and When to Use it
Exception Deviation	**ExcDev**	Use this attribute to specify how much a point value must change before the interface reports the new value to PI. Use ExcDev to specify the exception deviation in the point's engineering units. As a general rule, you should set the exception slightly smaller than the precision of the instrument system.
	ExcDevPercent	You can use ExcDevPercent instead of ExcDev. ExcDevPercent sets the exception deviation as a percentage of the Span attribute, but be careful. If your Span attribute is not set correctly, your exception reporting will be wrong too. A typical exception deviation value is about 1% of Span.
Exception Minimum	**ExcMin**	Use ExcMin to limit how often (in seconds) the interface reports a new event to PI. For example, if you set ExcMin to five, then the interface discards any values collected within five seconds of the last reported value. ExcMin is typically set to zero.
Exception Maximum	**ExcMax**	Set ExcMax to the maximum length of time (in seconds) you want the interface to go without reporting a new event to PI. After this time, the interface will report the new event to PI without applying the exception deviation test.

To learn more about exception reporting, see the ***PI Server Reference Guide***.

> **Note:** For Digital, Blob, or String points, only the exception maximum and minimum times are important. PI ignores the exception deviation specification for them.

5.2.8   Compression Specifications

PI uses the compression specifications to filter the data passed from the Snapshot to the Archive. The goal is to store just enough data to accurately reproduce the original signal. By filtering out data that you don't need, you get more efficient Archive storage and the Archive can serve the data to the clients more efficiently.

Where exception reporting uses a simple dead band method for filtering data, PI's compression testing uses a more complex method that follows the slope of the data (the swinging door compression algorithm). PI's compression testing algorithm is explained in detail in the ***PI Server Reference Guide***.

The compression specifications include a flag that allows you to turn compression on or off. We recommend you turn compression on for all real-time points in the system. You usually turn compression off for points with manually-entered data, production targets, control limits, and so on.

For each point, you can set four attributes to configure the compression specifications.

Specification	Attribute	What it Does
Compression Flag	**Compressing**	Turns compression on or off (set to 1 to turn compression on or 0 to turn compression off).
Compression deviation	**CompDev or**	As a rule of thumb, set CompDev to the precision of the instrument. Set it a little "loose" to err on the side of collecting, rather than losing data. After collecting data for a while, go back and check the data for your most important tags and adjust CompDev if necessary. Use CompDev to specify the compression deviation in the point's engineering units.
	CompDevPercent	Use CompDevPercent to specify the compression deviation as a percent of the point's Span attribute.
Compression minimum time	**CompMin**	Sets a minimum limit on the time between events in the Archive. Set the CompMin attribute to zero for any point coming from an interface that does exception reporting. You typically use CompMin to prevent an extremely noisy point from using a large amount of archive space
Compression maximum	**CompMax**	CompMax sets a maximum limit on the time between events in the Archive. If the time since the last recorded event is greater than or equal to CompMax, then PI automatically stores the next value in the Archive, regardless of the CompDev setting. You typically set CompMax to the same value for all points in the system. It's common practice to choose a CompMax setting of one work shift (for example, 8 hours).

To learn more about how PI calculates compression, see the ***PI Server Reference Guide***.

> **Note:** For Digital, Blob, or String points, only the compression maximum and minimum times are important. PI ignores the compression deviation specification for them.

5.2.9   Point Value Range: Zero, Span and Typical Value

The **Zero**, **Span**, and **Typical Value** attributes specify the range of values for a point.

Zero	Indicates a point's lowest possible value. Zero does not have to be the same as the instrument zero, but that is usually a logical choice. This attribute is required for all numeric data type points and is critically important for float16 points.
Span	The difference between the top of the range and the bottom of the range. This attribute is required for all numeric data type points.
Typical Value	Documents an example of a reasonable value for this point. For a numeric tag, it must be greater than or equal to the zero, and less than or equal to the zero plus the span.

5.2.10 Configuring Shutdown Events: Shutdown

The **Shutdown** attribute has two possible values: **1 (On)** and **0 (Off)**. If the PI Server shuts down, PI writes a shutdown event to all points that have the shutdown flag set to **1 (On)**. Set **Shutdown** to **Off** for points on interfaces that are buffered (See *Managing Buffering* on page 63). The buffering service restores the data for these tags as soon as it connects to the PI Server again.

5.2.11 Point Security: PtOwner, PtGroup, PtAccess, DataOwner, DataGroup, DataAccess

Each point has different, configurable, access privileges for its data and its point configuration. To control who has access to what, you assign an owner and a group for each point's data and attributes, respectively. Then you set owner, group and world privileges. Read *What are PI Access Categories?* on page 48 to learn how this works.

PI point security is divided into two separate pieces, Data Access and Point Access.

Data Access	Specifies who has access to a point's data values (Snapshot and Archive data).
Point Access	Specifies who has access to the point's attributes (Zero, Span, Descriptor, and so on).

You can have different owners and different group access for a point's attributes than for the point's data. So, for example, one user might be allowed to edit the data for a point, but not be allowed to edit the attributes of that point.

> **Note:** If users don't have permission to view a point's attributes, they won't be able to see that point's data either, in most cases. This is because client applications need access to the point attributes in order to get the data.

To edit the owner, group and permissions for a point, select the point in SMT's Point Builder and click the **Security** tab.

5.3   Creating New Points

As a FactoryTalk Historian System Manager, you might need to create a new PI Point. The easiest way to create a new tag is to copy an existing tag that is very similar to the tag you want to create—then you can just edit the **Tag** attribute and any other attributes that you want to change. SMT's Point Builder provides an easy interface for editing and creating PI points. If you're very familiar with Excel, you will probably find the Excel plug-in, TagConfigurator, a better tool to use. If you want to create more than one point at the same time, use TagConfigurator.

5.4   Finding Malfunctioning Points

At least once a month, you should use SMT's PI Stale and Bad Tags plug-in to search for stale and/or bad points. This plug-in identifies points that have not received data for a long time or that have current values representing error conditions such as "I/O timeout", "Pt Created", "bad input" or, in many cases "Shutdown". Also check any "flat-lined" or "stuck" points.

When you find points that are no longer useful (points that represent data from obsolete equipment, for example), you should decommission them (see *Decommissioning Points* on page 24).

5.5 Decommissioning Points

Typically, to decommission a point, you set the **Scan** attribute to **0 (off)**:

1. Open **SMT** and select the **PI Server** for that point (see *How to Run SMT* on page 5).

2. In the **System Management** plug-ins list, under Points, choose **Point Builder**.

3. **Search** for the point.

4. Click the **Archive** tab.

5. Under **Scan**, click **Off**.

Some interfaces don't use the scan bit to turn off points. If you want to decommission a point for such an interface, change the point source attribute for that point to a value that you use only for decommissioned points.

5.6 Deleting Points

When you delete a point, you lose all data for that point, so you break any client displays that use the point. Further, once you delete a point, you can't get it back. If you are unsure about the purpose of a point's existence or about the need for any historical data associated with it, it's safer to decommission the point (see *Decommissioning Points* on page 24) rather than deleting it.

Chapter 6. MANAGING ARCHIVES

You need to pay close attention to the PI archives, because this is where PI stores your data (*About Archives* on page 25). As the data accumulates, you need new archives to hold it. Otherwise, PI overwrites data in existing archives. You can do all or most of your archive management with the SMT Archive Manager plug-in.

- ❑ *About Archives*
- ❑ *Finding the Archive Files*
- ❑ *Making Sure PI Doesn't Overwrite Your Archives*
- ❑ *Creating an Archive*
- ❑ *Registering an Archive*
- ❑ *Unregistering an Archive*
- ❑ *Moving an Archive*
- ❑ *Fixing Archive Gaps*
- ❑ *Automating Archive File Creation*

6.1 About Archives

The FactoryTalk Historian System stores your data in archives. Typically, archives are files of a fixed size that can hold PI data. Fixed archives allocate the full amount of space upfront, meaning that an empty archive and a full archive take the same amount of disk space.

The archive receiving current data is called the *Primary Archive*. When the Primary Archive becomes full, an *Archive Shift* occurs and the next available archive becomes the new Primary Archive.

Note: PI actually performs the archive shift before the Primary Archive is completely full so that older data can be added later, if necessary.
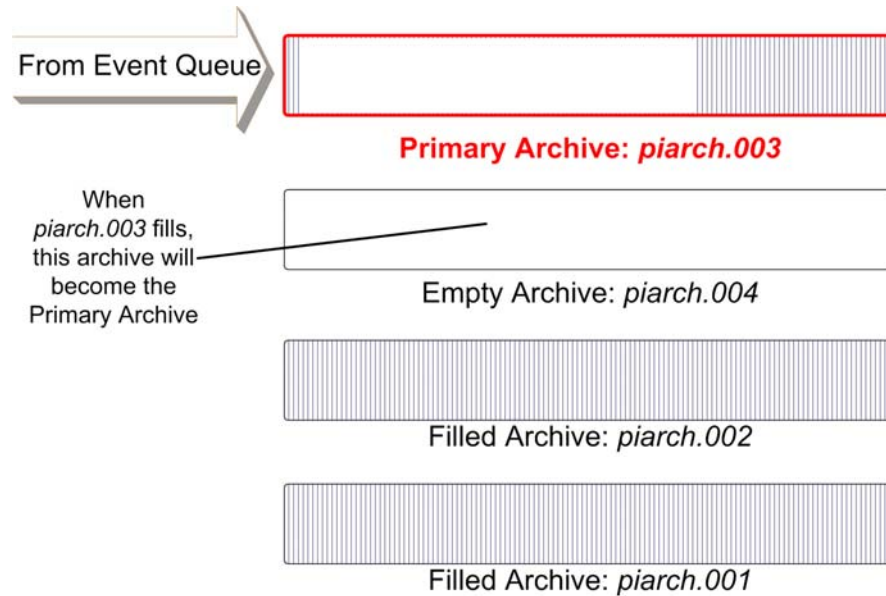
For an archive file to be eligible to be the new Primary Archive, it must be registered (*Registering an Archive* on page 29), writeable, shiftable, and large enough to handle the current size of the Point Database.

If no eligible archives are available for an Archive Shift, PI uses the oldest available filled archive as the new Primary Archive, overwriting the data in the old archive. For example in the preceding illustration, after the shift from *piarch.003* to *piarch.004*, no empty registered archives are left. If no new archives are created, then *piarch.001* becomes the next Primary Archive.

It takes PI a few minutes to complete an Archive Shift. During that time, you are not allowed to add, edit, or delete points. PI stores incoming data in the Event Queue until the shift is complete and then writes the queued events into the new Primary Archive.

Note: Archives that can grow in size to accept variable amounts of data are called dynamic archives. Dynamic archives are not discussed in this document. See the *PI Server System Management Guide* for details.

6.2   Finding the Archive Files

By default the archives are placed in the *PI\dat* directory, but you can put them anywhere you like. The Archive Manager lists the location of each registered archive file:

1. Run SMT and select the PI Server on which you want to view the archives (*How to Run SMT* on page 5).

2. Click to expand the **Operation** item in the System Management Plug-ins pane.
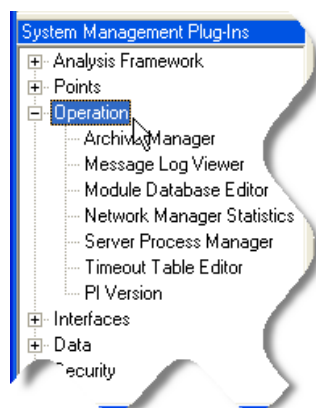
3. From the list of **Operation** plug-ins, double-click on **Archive Manager**. The Archive Manager plug-in appears in the Active Plug-in pane. The Archive File column lists all the archives registered on the selected server and displays the full path for each. The Primary Archive is first on the list.

Note: Don't use anti-virus software to scan the directories containing PI Server database and archive files on systems collecting production data.

6.3   Making Sure PI Doesn't Overwrite Your Archives

If you don't have an empty, writable, shiftable, archive available when the archive shift occurs, PI overwrites the oldest available full archive—unless your Server is set up to create archives automatically. If your FactoryTalk Historian System does not create new archives automatically, you can set it up to do that, if you like *(Automating Archive File Creation* on page 32).

Here's what you need to do:

1. Find out where the PI archive files are located (*Finding the Archive Files* on page 26).

2. If you don't want PI to create new archives automatically, then you need to figure out how often your archives fill and you need to create new archives (*Creating an Archive* on page 27) as needed so that PI doesn't run out of space and start overwriting data.

3. If you do use automatic archive creation, then you need to make sure you don't run out of disk space on that machine.

6.4   Creating an Archive

The SMT Archive Manager plug-in provides an easy interface for creating, editing, and monitoring your PI archives. To create a new PI archive, follow these steps:

1. Run SMT and select the **PI Server** on which you want to view the archives (*How to Run SMT* on page 5).

2. Click to expand the Operation item in the System Management Plug-ins pane.
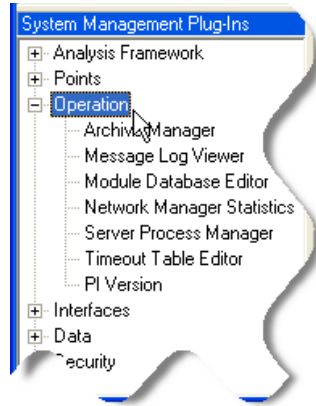
3. From the list of Operation plug-ins, double-click on **Archive Manager**. The Archive Manager plug-in appears in the Active Plug-in pane. It lists all the archives registered on the selected server. The Primary Archive is first on the list.

4. To create a new archive, click the **Create a New Archive** icon.
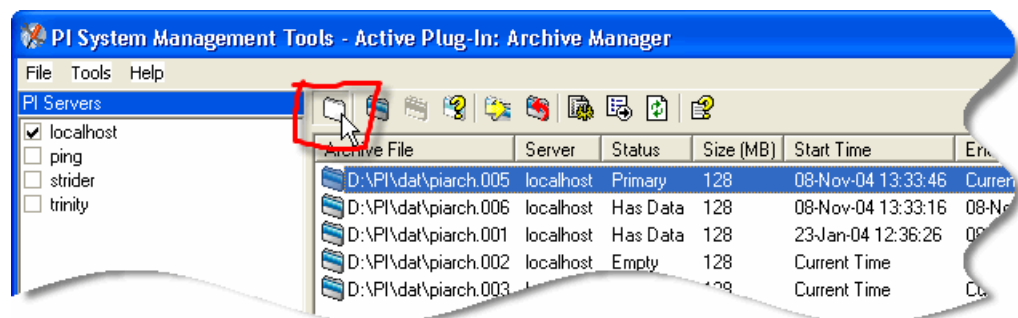
5. The **Create New Archive** dialog box appears. Name the new archive file and choose the **Clone the primary archive fixed size** radio button.

6. If you want to choose a different size for the archive, you need to understand the issues in archive sizing. Read the chapter on managing archives in the *PI Server System Management Guide*.

7. Click the **OK** button. The **Archive Manager** plug-in creates and registers the archive.

6.5   Registering an Archive

In order for the PI Server to recognize a file as an archive file you must register it. By registering an archive file, you tell the PI Server that the file exists and that is an archive file. The PI Server cannot access data in unregistered archives, nor can the PI client applications.

The SMT Archive Manager plug-in provides an easy interface for registering archives. To register an archive, follow these steps:

1. Run SMT and select the **PI Server** on which you want to view the archives (*How to Run SMT* on page 5).

2. Click to expand the **Operation** item in the System Management Plug-ins pane.
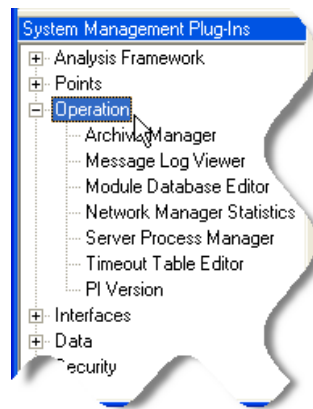
3. From the list of Operation plug-ins, double-click on **Archive Manager**. The Archive Manager plug-in appears in the Active Plug-in pane. It lists all the archives registered on the selected server. Unregistered archive files do not appear in the list.

4. Click the **Register an Archive** icon to register the archive.
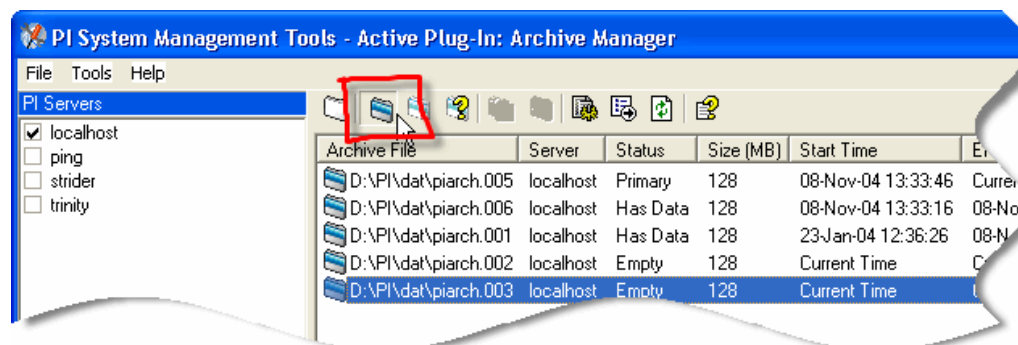
5. A file-browsing window appears. Double-click on the **archive file** you want to register. The Archive Manager plug-in registers the file and it appears in the list of registered archive files.

6.6   Unregistering an Archive

If you want to move or reprocess an archive file, you need to unregister it, make your changes, and then re-register it. You cannot unregister the primary archive.

The SMT Archive Manager plug-in provides an easy interface for unregistering archives. To register an archive, follow these steps:

1. Run SMT and select the **PI Server** on which you want to view the archives (*How to Run SMT* on page 5).

2. Click to expand the **Operation** item in the System Management Plug-ins pane.
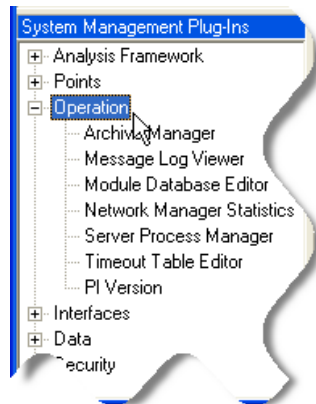
3. From the list of Operation plug-ins, double-click on **Archive Manager**. The Archive Manager plug-in appears in the Active Plug-in pane. It lists all the archives registered on the selected server. Unregistered archive files do not appear in the list.

4. To unregister an archive file, click the **Unregister selected archive** icon to unregister the archive. The archive disappears from the list.
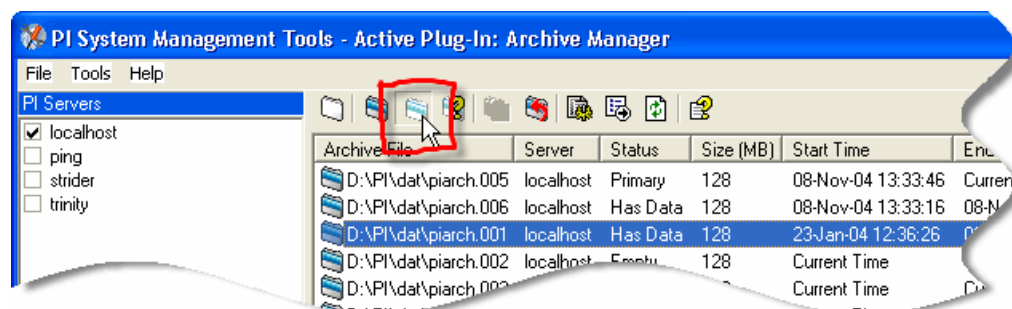
6.7   Moving an Archive

If you want to move an archive file, you unregister it (*Unregistering an Archive* on page 30), move it to the new location, and then register it again (*Registering an Archive* on page 29).

When you move the archive file, be sure to move the associated annotation file as well. The annotation file has the same name as the archive file, except that it ends in *.ann*. For example, if the archive is named *piarch.003*, then the associated annotation file would be called *piarch.003.ann*.
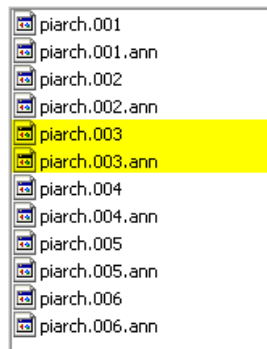
6.8   Fixing Archive Gaps

PI archive files meet chronologically end-to-end, accounting for all of history with no gaps and no overlaps. If a gap does occur, it's important to identify and fix it as soon as possible. To check for (and fix) archive gaps, follow these steps:

1. Run SMT and select the **PI Server** on which you want to view the archives (*How to Run SMT* on page 5).

2. Click to expand the *Operation* item in the System Management Plug-ins pane.
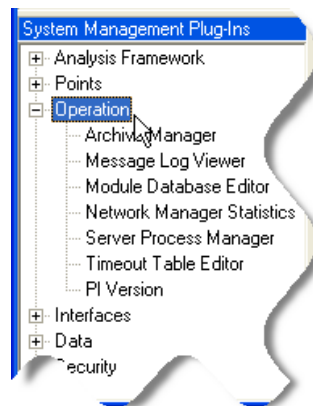
3. From the list of Operation plug-ins, double-click on **Archive Manager.** The Archive Manager plug-in appears in the Active Plug-in pane. It lists all the archives registered on the selected server. Any archive gaps are clearly labeled and highlighted in red.

4. To fix an archive gap, right-click on the line displaying the archive gap.

5. Select **Create New** from the resulting pop-up menu. The **Create New Archive** dialog box appears. The dialog box is already filled in for you, with the right start and end times to fill the archive gap.

6. Click **OK**. The Archive Manager plug-in creates and registers the new archive and an archive gap no longer appears in the archive list.

6.9   Automating Archive File Creation

If your PI Server is running PI 3.4 or later, you can set it up to create archive files automatically, as needed. This is very convenient, but you need to keep a close eye on available disk space for the archives.

To automate archive file creation, follow these steps:

1. Make sure you have a fixed-size Primary Archive (automatic archive generation won't work if you have a dynamic Primary Archive). The automatically-generated archives will be the same size as the Primary Archive.

2. Make sure you have a valid, shiftable target archive available. This gives you a backup in case the file creation fails for some reason. This target archive can be dynamic or fixed.

3. Run SMT (*How to Run SMT* on page 5).

4. Select the Server on which you want to automate archive file creation (*How to Select a Server in SMT* on page 6).

5. From the list of Operation plug-ins, double-click on **Timeout Table Editor**. The Timeout Table Editor appears in the Active Plug-in pane.

6. Click the **Archive tab** and select any item in the list.

7. Click the **New Timeout Setting** button.

A dialog box appears.

8. In the top field, which is the **Name** field, type in:

   `Archive_AutoArchiveFileRoot`

9. In the **Value** field, type in the path to the directory where you want PI to create the archives, along with the archive file prefix. For example, if you type:

   `D:\PI\arc\piarch`

   then PI creates new archive files in the D:*PI*\arc\ and names them *piarch.001*, *piarch.002,* and so on.
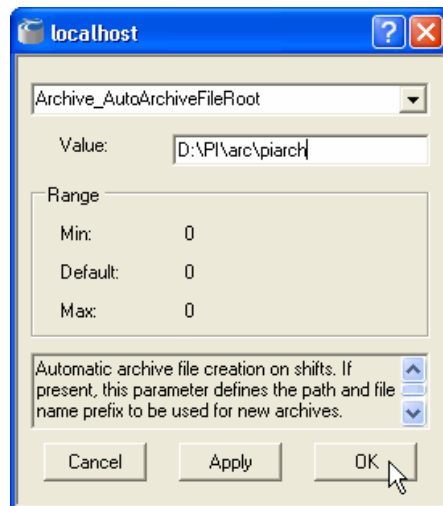
10. Click **OK**.

If there is not enough disk space to create the new archives, PI overwrites data in the old archives. You can find more information about automatic archive file creation in the *PI Server System Management Guide*.

Chapter 7. MANAGING BACKUPS

It's important to back up the PI Server daily, so that you don't lose data and configuration information. On a PI Server that is already correctly configured to back itself up automatically, all you need to do is check your backups periodically and check that you have enough disk space available for upcoming scheduled backups.

- ❑ About PI Server Backups
- ❑ Choosing a Backup Strategy
- ❑ Checking Whether Backups are Scheduled
- ❑ Setting up Automatic Backups

7.1 About PI Server Backups

All backups of PI that are done while the FactoryTalk Historian System is running are managed by the PI Backup Subsystem (*PI\bin\pibackup.exe*). Typically, the backups are launched via the *PI\adm\pibackup.bat* backup script. However, if PI is installed on Windows 2003 Server, you can backup PI with any third party backup application that supports Volume Shadow Copy Services (VSS). This chapter only discusses backups with the backup script.

> **Note:** If you don't have buffering turned on for all interfaces, you might lose data during backups. See *Checking whether the Buffering Service Is Running* on page 64. (PINet buffers data for interfaces running on PINet nodes.)

7.2 Choosing a Backup Strategy

The easiest backup strategy is to set up PI to automatically run the backup scripts every day as described in *Setting up Automatic Backups* on page 38 and in more detail in the PI Server System Management Guide.

You can also run the scripts manually. The backup script initiates backups via **NTBackup** (**NTBackup.exe**) on platforms that support VSS, and/or with the **piartool** –backup command on platforms that do not support VSS. It is highly recommended that you run PI on a platform that supports VSS because VSS backups cause minimal disruption to the operation of PI.

7.3   Checking Your Backup Files

It's important to check your backup files each day:

❑ Check the backup log daily (*Checking the Message Logs* on page 37). Make sure that the last backup completed successfully.

❑ Periodically test your backups by simulating a disaster recovery.

❑ Check to make sure that the backup files are where they're supposed to be, and that the file size looks right. These files should be about the same size each time—if the backup file is suddenly substantially smaller, then the backup might not have successfully completed.

❑ Make sure that you are not running out of space on the disk where PI creates the backups.

7.4   Checking Whether Backups are Scheduled

Open the "Scheduled Task" control panel.
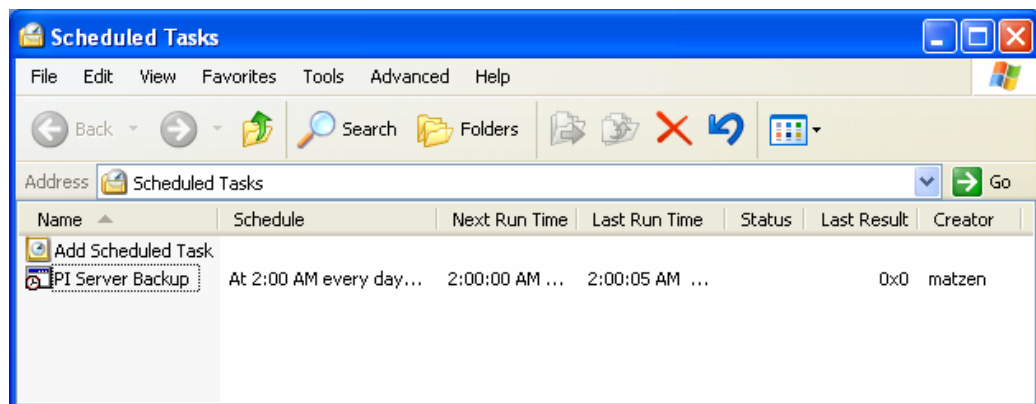
The name of the backup task that is installed with the *pibackup.bat* file is called "PI Server Backup". If backups have been running successfully, the "Last Run Time" should not be blank. The next run time should be about one day apart from the Last Run Time. Details of the scheduled task can be viewed by double clicking on the task.
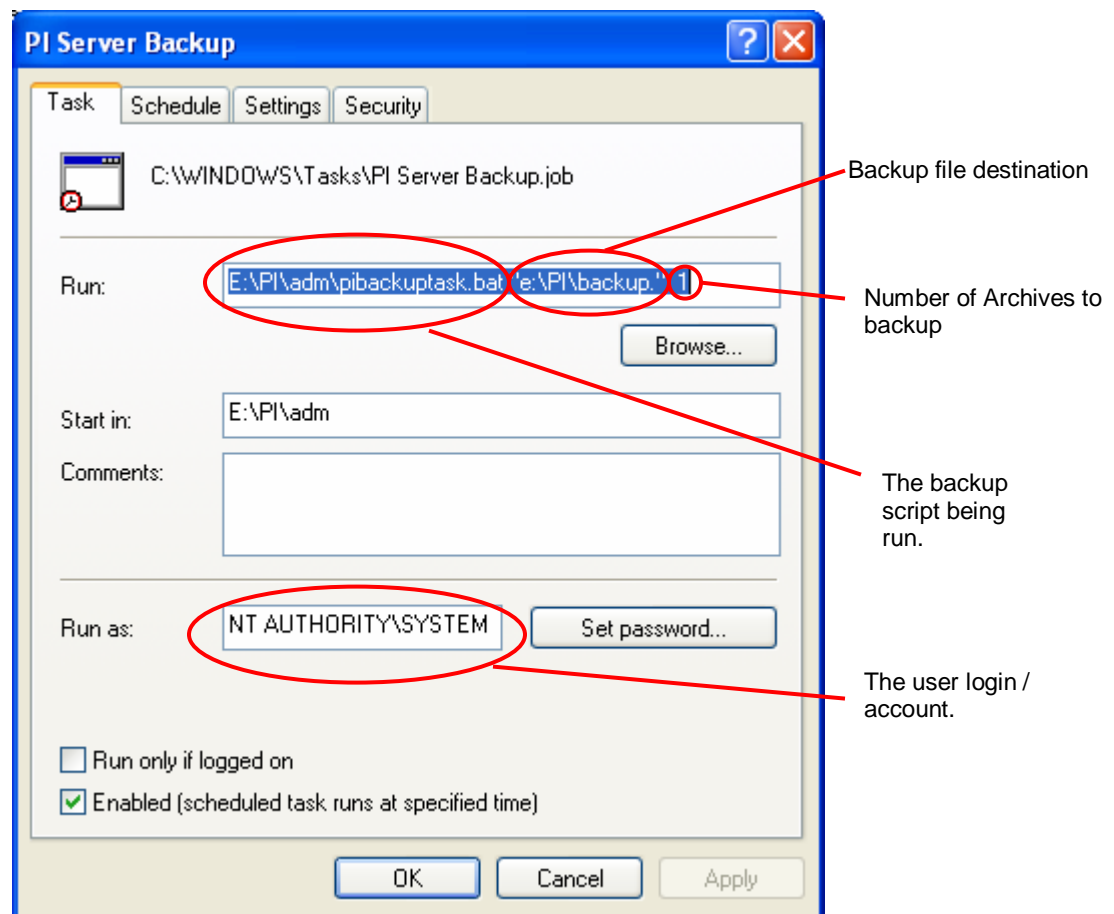
From the above picture, you can tell the following.

- ❑ The destination of the files for the backup operation is e:*PI*\backup.

- ❑ The number of Archives to be backed up is 1. That is, only the primary archive will be backed up.

- ❑ The scheduled task will run the e:*PI\adm\pibackuptask.bat* command file. The *pibackuptask.bat* in turn launches the e:*PI\adm\pibackup.bat* command file.

- ❑ The backup task will be run under the System account.

7.5   Checking the Message Logs

There are several places to look for messages related to backup.

1. You can search the PI Message Log for messages related to backup with the following command.

   ```
   \pi\adm\pigetmsg -st t -et * -pn pibacku*
   ```

2. The above command will search for backup-related messages from midnight until current time.

3. The output of the *pibackup.bat* script is written to a log file. The destination of this log file is the same as the destination of the backup, and the name of the log file is of the form *pibackup_dd-mmm-yy_hh.mm.ss.txt.*

4. The **NTBackup** log file (VSS Backups only). If there was a problem creating a VSS shadow copy during a backup, the reason for the failure will be logged at the beginning of the **NTBackup** log file.

5. If the "Run As" user for the "PI Server Backup" scheduled task is the same as your account, then you will be able to view the **NTBackup** log from the "Tools | Report…" menu of **NTBackup**. Launch **NTBackup** from a DOS command prompt and choose to run in advanced mode.
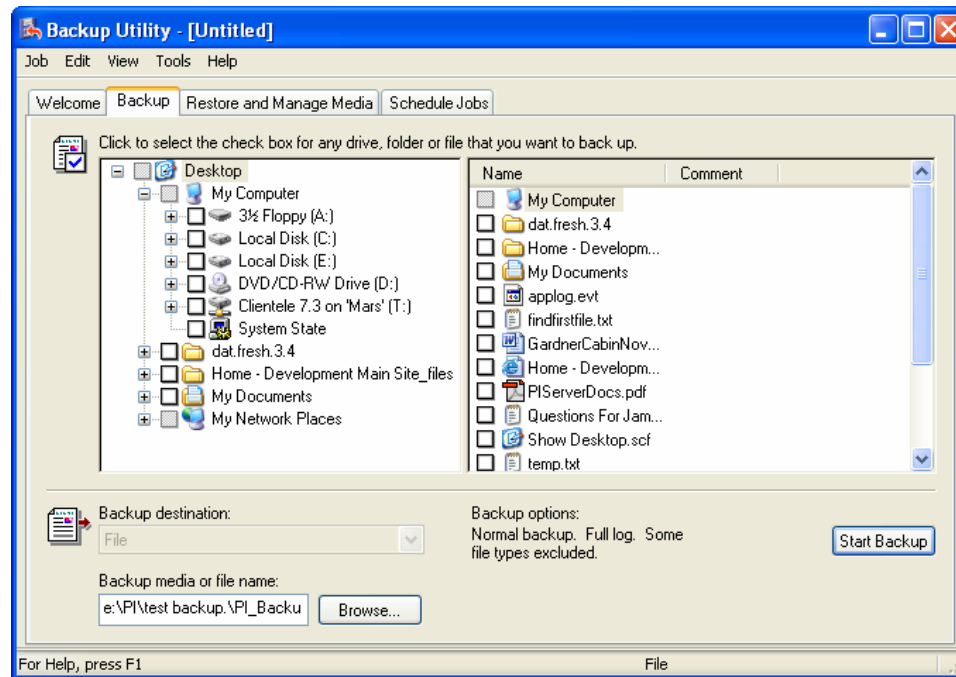
7.6 Setting up Automatic Backups

An automated backup task can be installed with the *PI\adm\pibackup.bat* backup script. You must install the automated task because the installation of the PI Server does not install a backup task. The syntax for using the *pibackup.bat* file is as follows.

```
PIbackup.bat <path> [number of archives]
 [archive cutoff date] [-install]
```

where <> indicates a required parameter and [] indicates an optional parameter. The command line parameters must be specified in the above order. If the -install flag is not specified, a backup will be performed immediately. The more restrictive of [number of

archives] and [archive cutoff date] takes precedence. Regardless of [number of archives] and [archive cutoff date] archives that do not contain data will not be backed up.

Parameter	Example	Description
<path>	E:\PI\backup	Path must be the complete drive letter and path to a directory with sufficient space for the entire backup.
[number of archives]	2	The number of archives to backup. For example, "2" will backup the primary archive and archive 1.
[archive cutoff date]	*-10d	The cutoff date is specified in PI time format. For example, "*-10d" restricts the backup to archives archives that contain data between 10 days prior to current time and current time. The more restrictive of [number of archives] and [archive cutoff date] takes precedence.
[–install]		Installs a scheduled task to run *pibackup.bat* daily at 2:00 am. If the –install flag is not specified, then a non-VSS backup is performed immediately.

7.7 Site-Specific Backup Tasks

If the *pisitebackup.bat* file exists, then the *pibackup.bat* backup script calls it before exiting. If you have any tasks you want *pibackup.bat* to execute each day after the backup, then add these tasks to a file called *pisitebackup.bat* in the *PI\adm* directory.

Typically, FactoryTalk Historian System Managers use the *pisitebackup.bat* file to move the backup directory to tape. FactoryTalk Historian System Managers may also use the script to back up specific files that are not included in the PI Server backup.

Chapter 8.   MANAGING INTERFACES

Once you install and configure an interface on a PI Interface Node, you can typically then leave it to run indefinitely without any intervention. If you perform software upgrades or security patches or if the network infrastructure changes, you might need to know how to perform a few basic tasks.

- ❑ *About PI Interfaces*
- ❑ *Configuring Interfaces*
- ❑ *Starting and Stopping Interfaces*
- ❑ *Monitoring Interface Performance*
- ❑ *Configuring Interfaces for Buffering*
- ❑ *Where to Go for More Information on Interfaces*

8.1   About PI Interfaces

PI interfaces are the software applications that take the data from your data source and send them to the PI Server. There are hundreds of PI interfaces and they each have their own specific documentation. However, because most interfaces are based on the OSIsoft Universal Interface (UniInt), they share a common set of features.

The three basic variables that define an interface are interface ID, scan class, and point source.

8.1.1   What's a Point Source?

The **PointSource** attribute is a single character that identifies a PI point as belonging to a particular interface. When you configure an interface, you specify a point source for that interface. All the points that belong to that interface must use that point source code as the value for the **PointSource** attribute.

The PI Server comes pre-configured with applications that use some of the otherwise available point source characters. When you're choosing a character to use as the point source for an interface, avoid using the following characters:

Character	Reserved For
9	RampSoak Simulator

Character	Reserved For
C	Performance Equations scheduler
G	Alarm
L	default point source character
R	Random Interface Simulator
T	Totalizer program
@	Alarm

8.1.2 What's an Interface ID Number?

The **Interface ID** number is a number that associates a point with a particular copy of an interface. Set the Interface ID number to any positive integer. Points that use the interface then typically use the ID number as the value of the **Location1** attribute. Refer to the interface documentation though, to be sure.

8.1.3 What's a Scan Class?

A **scan class** is a code that PI interfaces use to schedule data collection. Scan classes consist of a period, which tells PI how often to collect the data and, optionally, an offset, which tells PI when to start collecting data. A scan class can also optionally contain a code that requires that the interface use UTC time:
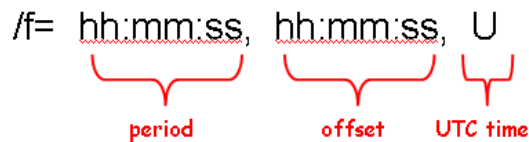
Scan Class Component	Description	Example
Period	The period specifies the interval between calculations. The first two digits are the hours, the second two the minutes, and the third two the seconds.	01:00:00 Get data every hour.
Offset	The offset specifies a start time for the calculation. The offset is optional. PI counts the offset from midnight of the current day. As with the period, the first two digits are the hours, the second two the minutes, and the third two the seconds.	01:00:00, 13:00:00 Get data every hour, starting at 1PM

Scan Class Component	Description	Example
UTC Time	The UTC time specifies that the scheduling sync with Universal Coordinate Time (UTC). The UTC time is optional. To use it, add a comma followed by a capital U to the end of the scan class.	01:00:00, 13:00:00, U Get data every hour, starting at 1PM, UTC time.

Note: If a scan class has a frequency of more than an hour, make it a UTC scan class. Otherwise, your scheduling might be thrown off the next time there's a switch change to or from daylight savings time. UTC scan classes don't have this problem because they force the scan class scheduling to sync with UTC, rather than local time.

8.2   Configuring Interfaces

If you have access to the Interface Nodes for your FactoryTalk Historian System, then the Interface Configuration Utility (ICU) is the best way to manage your PI interfaces. You need to do an initial configuration in the ICU for each interface. This is called registering the interface with the ICU:

1. On the ICU menu, select **New**. The **Configure a New Interface** dialog appears.
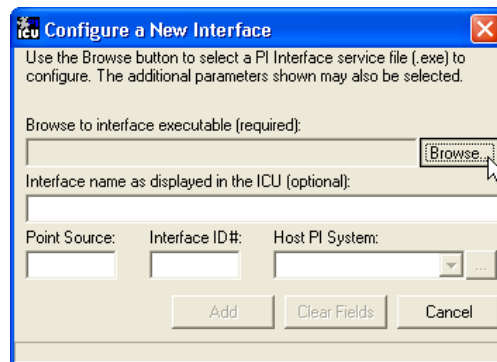
2. Click the **Browse…** button and browse to the location of the interface executable. By default, PI installs your interface executables in the *Program Files\PIPC\Interfaces* directory.

3. Type in a descriptive name for the interface. The ICU uses this name in its displays, so choose a name that you can remember.

4. Type in a **Point Source**.

5. Type in an **Interface ID** number.

6. Select the **Host FactoryTalk Historian System**.

7. Click **OK**.

8.3 Starting and Stopping Interfaces

The first time you start a PI Interface, you need to start it from the Windows Services control panel. After that, you can start and stop the Interface from the ICU (*Configuring Interfaces* on page 43). Start and stop the interface directly on the Interface Node:

1. From the Windows **Start** menu on the Interface Node, open the **Control Panel** and then open **Administrative Tools**.

2. In the **Administrative Tools** folder, double-click on **Services**.

3. In the **Services** window, find the interface that you want to start or stop. In the Windows Services panel, PI interface services are listed with the PI- prefix. For example, the buffering service is listed as **PI-Buffer Server**.

4. Right-click on the interface service select either **Start** or **Stop** from the resulting pop-up menu.

8.4 Monitoring Interface Performance

There are two main ways to check how your interfaces are working:

❑ *Checking IORates and Performance Points*

❑ *Checking Log Files*

8.4.1 Checking IORates and Performance Points

Create a FactoryTalk Historian ProcessBook display that shows the IOrates and performance points for each interface. Some interfaces do not have performance points, so for these interfaces, you rely on IORates points alone.

Type of Point	What it Does
IORates point	Monitors the flow of data from an interface. Every 10 minutes each IOrates point registers the 10-minute average data transfer rate to PI in events/second.
Performance Point	Reads the value in seconds that it takes the interface to complete one round of data collection for a set of points. You can create one performance point for each scan class of each interface.

You can easily create both IORates points and performance points for an interface using the ICU (*Configuring Interfaces* on page 43).

8.4.2 Checking Log Files

The FactoryTalk Historian System logs interface and buffering errors in the *pipc.log* and/or the *pigetmsg log* on the interface machine. You can use SMT's **Message Log Viewer** tool to see these messages. Some interfaces also produce an interface output file that might contain information about how the interface is performing.

Most interfaces also write a performance summary every 8 hours to *pipc.log*. For each scan class, the summary shows the duration of the most recent scan, the percent of scans missed, and the percent of scans skipped. PI counts a scan as missed if it was started after its scheduled start time (due to a previous scan taking too long). PI counts a scan as skipped if it did not have an opportunity to run at all. Note that a previous scan can be from any of the defined scan classes.

Performance points are an important tool for tuning scan classes, because if a scan takes too long, it can cause the next scan to be skipped, resulting in data loss. You can tune scan classes by changing the scan frequency, the scan offset, and the number of tags in the scan list. For more information on configuring scan classes and scan lists, see the user manual for your interface.

8.5   Configuring Interfaces for Buffering

If you are starting the buffering service on an Interface Node for the first time, you also need to re-configure the interfaces so that they work correctly with buffering:

1. Configure the interfaces on this node so that they're dependent on the buffering service.

2. Set the **shutdown** attribute to **off** for points on these interfaces.

3. Restart the interfaces, so that they can send their data through the buffering service.

Note: When installing an interface for the first time, it is a good idea to disable buffering until you are sure the interface is correctly collecting data and storing it in the archive. Bufserv stands in for the archive when it is not available and may mask errors your interface would see when running unbuffered.

8.6   Where to Go for More Information on Interfaces

There are many different sources of information on PI interfaces. You can download all of the following documents from the Technical Support Web Site (http://techsupport.osisoft.com).

❑   Each interface has its own documentation. If you're not sure how to configure a particular interface, check the documentation for that specific interface.

❑   Many interfaces are based on the Universal Interface (UniInt). You can learn a lot by reading the *UniInt End User manual*.

❑   The *PI Server System Management Guide* contains a chapter on managing interfaces.

❑   The Technical Support Web Site also provides a product page dedicated to PI interfaces and a FactoryTalk Historian System Manger Resources page. You can get information on new releases, as well as information on the Interface Configuration Utility and other tools.

Chapter 9. MANAGING PI SECURITY

PI has two main mechanisms for security: log-in accounts (users and groups) and trusts.

❑ *About PI Security*

❑ *Managing Users and Group*

❑ *Managing PI Trusts*

❑ *Managing PI Database Security*

9.1 About PI Security

The two most commonly used methods of PI Security are user identification security (login accounts with access permissions) and trusts (trusted computers that are granted access without having to log in).

❑ **User Identification Security:** The PI Server has its own user identification and password security. Users typically access client applications, server connections and other PI resources by typing in a valid user name and password. For some resources, such as point data and attributes, you can set access permissions based on user name and group association. User identification security is explained in *Managing Users and Group* on page 47. PI databases rely on user identification security (Managing PI Database Security).

❑ **Trusts:** In some situations, you don't want to require a username and password to establish a PI connection. A typical example of this is a PI interface application requesting a connection to a PI Server. PI provides trust security for this type of situation. Trusts are explained in *Managing PI Trusts*.

Note: PI also provides a firewall mechanism that is not covered in this guide. The PI firewall is explained in the *FactoryTalk Historian System Management Guide.*

9.2 Managing Users and Groups

This section explains how to manage user accounts and how to set up groups to manage access permissions to PI resources. It contains the following sections:

❑ *About Users and Groups*

❑ *Setting up Groups to Manage Resource Access*

❑ *Adding, Editing and Deleting Users and Groups*

9.2.1 About Users and Groups

The PI Server has its own user account security. Each user account is protected by a password. Users can also belong to one or more groups. PI groups are collections of users who need the same level of access to a particular PI resource, such as point data or attributes.

As System Manager, you can configure which users and groups have access to which PI resources (point data and attributes, for example) and how much access they have. To configure this access, you set permissions for a particular resource's owner and group.

9.2.2 What are Access Permissions?

PI Permissions, like Windows permissions, determine whether a particular user is allowed to view and/or edit an item in PI. PI provides the three standard levels of access permissions:

Access	Description
Read-only access	Users can view the item, but they cannot change it.
Read-write access	Users can view the item and they are also allowed to edit it.
No access	Users cannot view or edit the item.

9.2.3 What are PI Access Categories?

PI grants access to certain resources (databases, point data and point attributes) based on user access category. There are three categories of user access in PI: owner, group, and world. For each PI resource, you can configure the permissions for each user category:

PI grants resource access based on whether a user is the Owner of the item being accessed, a member of its Group, or neither. Each resource has one Owner and one associated Group.
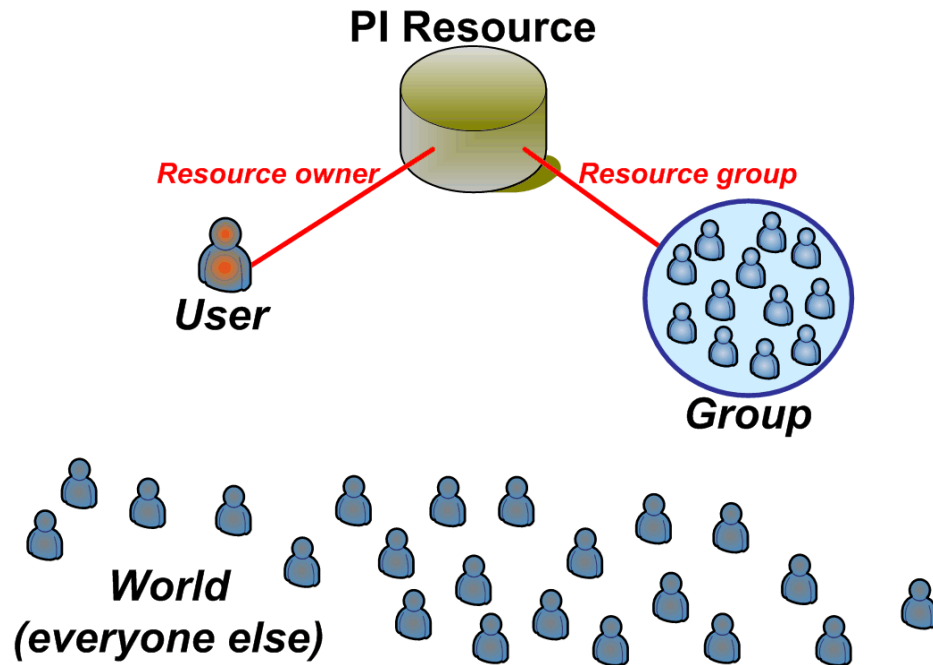
Category	Description
Owner	Each resource has one (and only one) owner. The owner is always a single user, not a group. Owner access is usually **read-write**, but it doesn't have to be.
Group	Each resource has one (and only one) associated group. Group access is often **read-only**, but it doesn't have to be. Since each resource has only one associated group, you sometimes need to create additional groups to give access to all the users who need it (*Setting up Groups to Manage Resource Access* on page 50).
World	The World category is where you set the permissions for users that are not the owner and not in the associated group. By default, world access is **none**, but it doesn't have to be.

The *piadmin* user always has read-write access to all resources regardless of access settings. By default, the *pidemo* user has read-only access to all objects. For more information about these two accounts, see *What are the piadmin and pidemo User Accounts* on page 49.

9.2.4 What are the piadmin and pidemo User Accounts?

By default, each PI Server has two accounts:

❑ **The Administrator account**, called *piadmin*, is the super user account. It has permission to do anything on the FactoryTalk Historian System, regardless of security settings. This means that you need to restrict access to the *piadmin* account to a very small group of trusted managers. You cannot delete the *piadmin* account.

❑ **Demonstration account**: the *pidemo* user is the demonstration account. The *pidemo* user has permission to look at all the data, but is not allowed to make any changes. For example, the *pidemo* user cannot edit a point. You can delete the *pidemo* account, if you wish.

The default password is blank for both the *piadmin* and the *pidemo* accounts. Since the *piadmin* account can do just about anything on the PI Server, you should choose a good password for the *piadmin* account and change it periodically. You cannot delete the *piadmin* account.

9.2.5 Setting up Groups to Manage Resource Access

When a user is not the owner of a particular PI resource (such as a point or database), PI checks to see if the user is a member of the group that is associated with that resource. If so, then the user gets whatever access level the group has.

You can associate only one group with a particular PI resource. This means that, if you want to configure access for more than one group, you typically need to create a new group that includes all the users for whom you want to configure access.

For example, the following figure illustrates an organization with three groups: Developers, Managers and Operators. One user is a member of both the Developers and the Managers group.

Group:
Developers

Group:
Managers
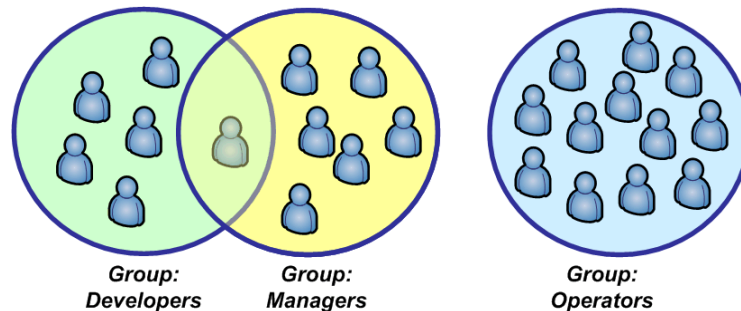
Group:
Operators

Suppose that all the users in the Developers and Managers groups need read-write access to a particular resource, such as the attributes for the Sinusoid point. Because a resource can have only one associated group, you need to create a group that contains all the developers and all the managers and then associate that resource with the new group.
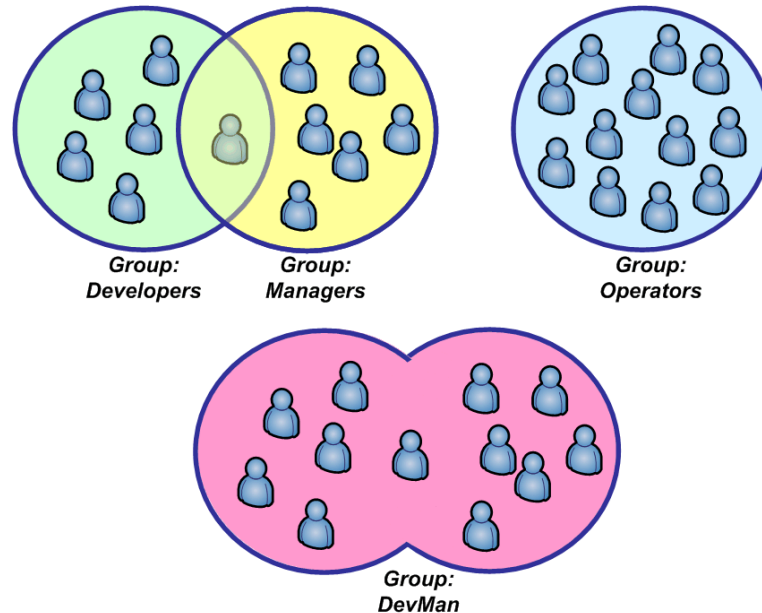
Typically, you create different PI groups for groups in your organization that need different point access.

9.2.6 Simple Case Example for Managing Groups

Consider the simple example in which you have a single group in your company, developers, that needs the ability to see and edit point attributes and data. Everyone else in your company needs to see all the points—but they are not allowed to edit them.

In this case, you create a PI group, called developers and you make all the developers members of this group. You configure every point's data and attributes to belong to the developers group (this does not effect the point owner) and you set the group permissions for all the points to read-write. You set the World permissions for each group to read-only. Now all the members of the developers group have read-write access to the Data and Attributes for all the points, and everyone else has read-only access to all the points.

9.2.7 Example for Managing Multiple Groups

Now consider the example in which you have three groups in your company, developers, managers and operators. Operators need read-only access to the data for the Sinusoid point. Developers and managers need read-write access to the data for the Sinusoid point. You set World permissions to read-only, so that the operators can see the point. But, since you can associate only one group with the Sinusoid point data, you need to create a combination group, which you might call devMan. All the users in the developers group and all the users in the managers group belong to the devMan group. You associate the Sinusoid data with the devMan group, giving all these users read-write access to that data.

9.2.8 Adding, Editing and Deleting Users and Groups

Use the SMT User and Group Editor to add and delete users and groups, to change user passwords, and to add and remove users from groups:

1. Run the System Management Tools (*How to Run SMT* on page 5).

2. Select the Server on which you want to manage the accounts (*How to Select a Server in SMT* on page 6).

3. Click to expand the **Security** item in the System Management Plug-ins pane.

4. From the list of Security plug-ins, double-click on **User and Group Editor**. The User and Group Editor plug-in appears in the Active Plug-in pane.

   The Users tab lists all the users who have accounts on the selected server. Similarly, the Groups tab lists all the groups registered on the selected server.

5. To work with users (to add or delete users or to change a password) select the **Users** tab. To work with groups (to add or delete a group, or to add users to or delete users from a group) select the **Groups** tab.

6. Different icons appear in the toolbar, depending on whether you select the Users or Groups tab. Some icons are enabled only when you click on an item in the list.
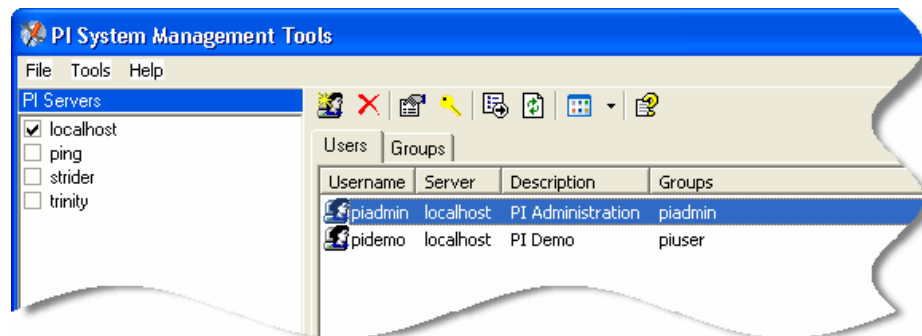
9.3 Managing PI Trusts

As System Manager, you need to update the PI trusts anytime any trusted machine changes host name or IP address. Use SMT's PI Trust Editor to view and manage your PI trusts.

❑ *About Trusts*

❑ *Managing Trusts with SMT*

9.3.1 About Trusts

PI uses trusts to allow access to resources without requiring that someone enter a user name and password. PI typically uses trusts to grant access to interfaces.

At a bare minimum, every PI Server has the following four trusts:

❑ The PI Server itself, by *host name*

❑ The PI Server itself, by *IP address*

❑ The PI Server itself by the special name *localhost*

❑ The special *IP address*: *127.0.0.1*

In addition, you set up a trust for each Interface Node that connects to the PI Server. For each Interface Node, configure the following three trusts:

- ❑ The Interface Node by IP address and netmask
- ❑ The Interface Node by fully-qualified host name (for example, apollo.osisoft.com)
- ❑ The Interface Node by the short host name (for example, apollo)

The preceding interface trust configuration is the most general configuration. If you want tighter security, read the section on PI security in the *FactoryTalk Historian System Management Guide*.

9.3.2 Managing Trusts with SMT

Use the SMT Trust Editor to view existing trusts and to add, edit or delete trusts:

1. Run the **System Management Tools** (*How to Run SMT* on page 5).

2. Select the **Server** on which you want to manage the trusts (*How to Select a Server in SMT* on page 6).

3. Click to expand the **Security** item in the System Management Plug-ins pane.

4. From the list of Security plug-ins, double-click on **Trust Editor**. The Trust Editor plug-in appears in the Active Plug-in pane.

5. To view the settings for an existing trust, right-click on that trust in the list and select **Properties** from the resulting pop-up menu. The Trust Properties dialog box appears.

Each trust has a name and three basic areas of information: *IP information, Windows account information,* and *application information*. In addition, each trust has an associated PI user.

You do not need to fill in all the areas in the Properties form. The more details you fill in, the more restrictive the trust becomes. For example, if you create a trust for a particular hostname, then any request from that hostname gets access, based on that trust. However, if in addition to the hostname, you add an application name, then that trust lets through only the specified application from that hostname.

9.4   Managing PI Database Security

PI stores its configuration information and process data in databases. You can configure access to the databases with SMT's Database Security Editor:

❑   *About Databases*

❑   *Managing Databases with SMT*

9.4.1   About Databases

The PI Server includes several databases that store PI configuration information and process data. The two main databases are the Point Database (also called the Point Configuration) and the Data Archive (usually called simply, the Archive).

Database security is similar to point security, in that you can set Owner, Group and World permissions for most PI databases. Users need read access to be able to read information within a database. Users need to have write access in the database to be able to create an item. For example, write access to the point database allows you to create a point.

You can configure security for the following PI Databases by setting Owner, Group and World permissions (see *What are PI Access Categories* on page 48):

Database	What it Stores
PI Batch Database (PIBatch)	Database for PI Batch objects
PI Digital State Table (PIDS)	Contains the digital state sets
PI Heading Sets Database (PIHeadingSets).	Stores the heading sets for the Module Database
PI Module Database (PIModules)	Stores and maintains all the data objects associated with the Module Database, including PIModules, PIHeadingSets, and PIHeadings
PI Point Database (PIPOINT)	Contains the list of points that are either recorded in the Data Archive or mapped to points in foreign data systems
PI Transfer Record Database (PITransferRecords)	Stores the transfer records for plant materials
PI User Database (PIUSER)	Database where PI users are defined

For comprehensive information on databases, refer to the *PI Server Reference Guide.*

9.4.2 Managing Databases with SMT

The SMT Database Security Editor plug-in provides a simple interface for managing user and group access to PI databases.To use the Database Security Editor:

1. Run the **System Management Tools** (*How to Run SMT* on page 5).

2. Select the Server on which you want to manage the database security (*How to Select a Server in SMT* on page 6).

3. Click to expand the **Security** item in the System Management Plug-ins pane.

4. From the list of Security plug-ins, double-click on **Database Security Editor**. The Database Security Editor plug-in appears in the Active Plug-in pane, listing all the PI databases for which you can configure the access permissions (*What are Access Permissions* on page 48).

5. **Select the database** for which you want to configure access. The Database Properties dialog box appears.
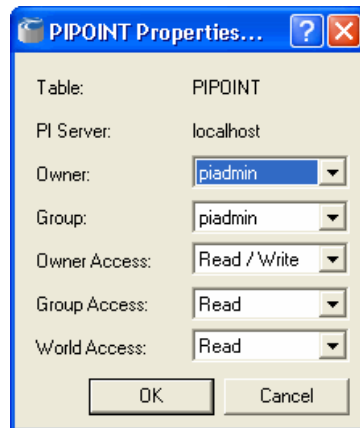
6. Select your access configuration from the pull-down menus and click **OK**.

Chapter 10. MONITORING FACTORYTALK HISTORIAN SYSTEM PERFORMANCE

As System Manager, you should check regularly on how PI is performing. By creating PI performance points (called PIPerfmon points) and trending them in a ProcessBook display, you can see how a system is doing at a glance.

❑ *About FactoryTalk Historian System Performance Monitoring*

❑ *Which Performance Counters to Monitor*

❑ *Building Performance Monitor Points*

❑ *Trending Performance Points*

❑ *Performance Monitoring Kit (PI 3.4 and Later)*

❑ *Configuring the PIPerfmon Interface*

Note: PI performance points (PIPerfmon points) are different from interface performance points, which are discussed in *Monitoring Interface Performance* on page 44.

10.1   About FactoryTalk Historian System Performance Monitoring

One important way to monitor your FactoryTalk Historian System's performance is to record key performance counters. Performance counters can provide important insights into a number of performance management problems, including memory, disk, and process management problems.

PI gets performance counter data through the PI Performance Monitor interface, PIPerfmon. PIPerfmon reads the PI point database to determine which performance counters to monitor. It then scans for the performance counter and sends exception reports to the PI system.

There are two versions of the PIPerfmon interface, *full* and *basic*. The PI Server comes with the basic version, which is just like the full version, except that it:

❑ Must run on the machine with the PI Server.

❑ Is limited to 512 points.

❑ Allows one instance of the interface.

❑ Will collect data for local performance counters only.

To use PIPerfmon to monitor performance, you build a point for each performance counter you're interested in (*Which Performance Counters to Monitor* on page 58) and then create a ProcessBook display that contains those points (*Trending Performance Points* on page 59). You can then open the ProcessBook display and check system performance at a glance.

Note: You can also view the statistics for the PI performance counters in Microsoft's Performance Monitor utility (System Monitor in Windows 2000).

10.2   Which Performance Counters to Monitor

The most confusing thing about performance counters for new System Managers is figuring out which ones to monitor. The following list contains the basic set of performance counters that OSIsoft recommends you monitor. This list includes some counters that are available only on PI 3.4 and later. If you are running an earlier version of PI, and don't want to upgrade, just use the other counters in the list.

PI Performance Counter Tag Name	Description
PI Archive Subsystem\Archived Events/sec	Rate of successful event addition to the archive.
PI Archive Subsystem\Cache Flush Operations/sec	Rate at which points are flushed from the archive cache to disk.
PI Archive Subsystem\Cache Record Count	Archive cache records in memory.
PI Archive Subsystem\Events Read/sec	Rate of archive events read.
PI Archive Subsystem\Time to Archive Shift	Number of seconds until the archive is projected to shift. This time is not calculated if the archive is less than 20% full.
PI Base Subsystem\Module Count	Total number of modules in the PI Module Database.
PI Base Subsystem\Point Count	Total number of defined points. This number includes the Connector Point Count.
PI Snapshot Subsystem\GetSnapshots/sec	Rate of events read from the Snapshot.
PI Snapshot Subsystem\OutOfOrderSnapshots/sec	Out of order events sent to the Snapshot.
PI Snapshot Subsystem\Queued Events/sec	Events sent to Event Queue.
PI Snapshot Subsystem\Snapshots/sec	Events sent to the Snapshot.
PI Archive Subsystem\Primary Archive % Used	Percent of used records in Primary Archive file. Available in PI Server v. 3.4 and later.
PI Archive Subsystem\Total Unflushed Events	Total number of unflushed events. Available in PI Server v. 3.4 and later.

PI Performance Counter Tag Name	Description
PI Snapshot Subsystem\Events in Overflow Queues	Total of events in the overflow queue files. Available in PI Server v. 3.4 and later.
PI Snapshot Subsystem\Events in Primary Queue	Number of events in the primary queue file. Available in PI Server v. 3.4 and later.
PI Snapshot Subsystem\Number of Overflow Queues	Number of overflow queue files (0 if only the primary queue is active). Available in PI Server v. 3.4 and later.

Note: This table lists only a very small subset of the PI performance counters. The *PI Server System Management Guide* provides a comprehensive list of all available PI performance counters.

10.3   Building Performance Monitor Points

The FactoryTalk Historian System Management Tools (*The FactoryTalk Historian System Management Tools (SMT) on page 5*) includes a plug-in for building PIPerfmon tags, called the Performance Monitor Tag Builder. To build a performance monitor point, follow these steps:

1.   Run the **System Management Tools** (*How to Run SMT* on page 5).

2.   **Select the Server** on which you want to create the performance point (*How to Select a Server in SMT* on page 6).

3.   In the list of Points plug-ins, double-click on **Performance Monitor Tag Builder**. The Performance Monitor Tag Builder appears in the Active Plug-in pane.

4.   Under the **Tag Settings tab**, choose an existing **PIPerfmon interface** from the pull-down menu. If no interfaces appears in the list, make sure PIPerfmon is installed and running on the PI Server (*Configuring the PIPerfmon Interface* on page 60).

5.   Click the **Build Tag** tab and **select the performance monitor points** you want to create from the list of available counters.

6.   Click the **Create Tags** button. The plug-in creates the performance monitor tags for you.

10.4   Trending Performance Points

If you put your PI-Performance Monitor points on a ProcessBook display, you can check your FactoryTalk Historian System performance at a glance. You might also include interface performance points (*Monitoring Interface Performance* on page 44) and representative points from each of the batch, alarm, performance equation, and ACE component.

The following illustration shows the ProcessBook display for the Performance Monitoring Kit from the OSI Developer's Network (*Performance Monitoring Kit (PI 3.4 and Later* on page 60). This is a good example of a ProcessBook display for performance monitoring.
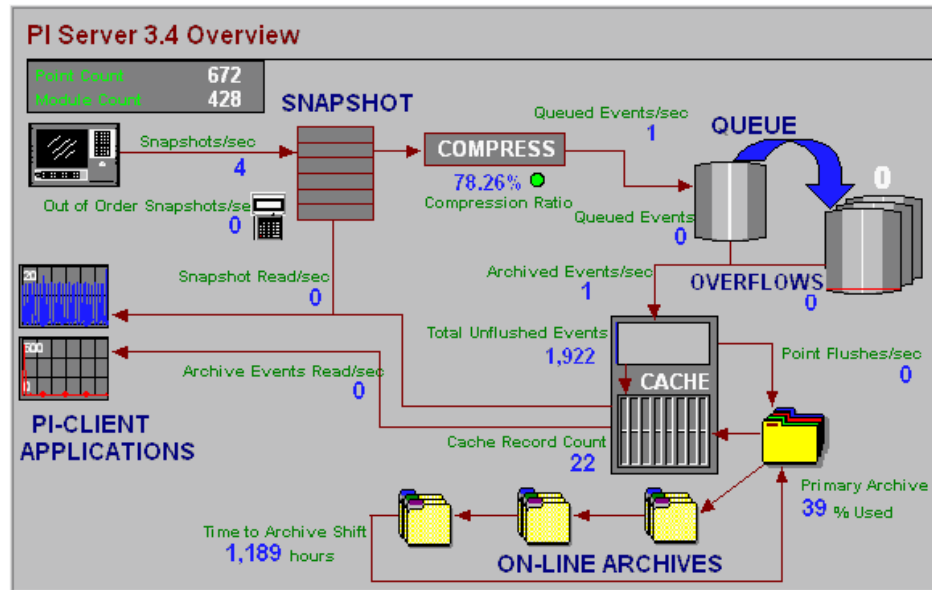
## 10.5   Performance Monitoring Kit (PI 3.4 and Later)

If you're running PI 3.4 or later, then you can download and use the PI Server Performance Monitor Kit. This kit gives you a point configuration spreadsheet that allows you to create your performance points in a matter of seconds, and it also gives you a great Performance Monitoring display for ProcessBook. You can get this kit from OSIsoft's Developers Network (http://devnet.osisoft.com).

> **Note:** The Developer's Network is a useful source of tools and information for FactoryTalk Historian System Managers, but the downloads are not actively supported by OSIsoft. If you have trouble getting, installing or running the Performance Monitor Kit, you cannot get help from Technical Support. Also, this kit might not work with future releases of PI.

10.6   Configuring the PIPerfmon Interface

The executable for the basic version of the PIPerfmon interface is included in your PI Server installation. To configure PIPerfmon on your PI Server, follow these steps:

1.   Run the **ICU** (*How to Run the ICU* on page 6).

2.   On the ICU menu, select **New**. The Configure a New Interface dialog appears.
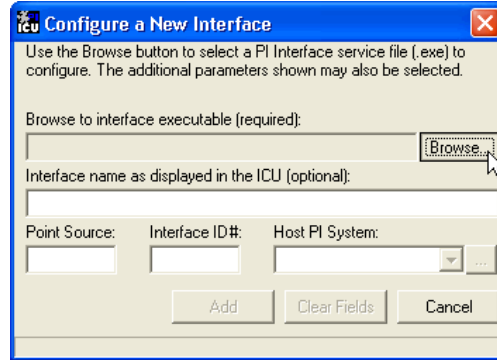
3. Click the **Browse…** button and browse to the location of the **PIPerfmon** executable. By default, PI installs the PIPerfmon executable in the PIPC directory under:

   `Interfaces\PIPerfmon_basic`

4. Type in a **descriptive name** for the interface, such as Performance Monitor Interface or PIPerfmon.

5. **Type in a Point Source** (*What's a Point Source?* on page 41). The default point source for PIPerfmon is the pound sign (#).

6. **Type in an Interface ID number** (*What's an Interface ID Number?* on page 42).

7. Select the Host FactoryTalk Historian System. This can be the PI Server itself.

8. Click **OK.** A dialog appears saying that the interface is ready to be configured.

9. Click **OK**.

10. If you're using the Performance Monitor Kit from DevNet, **add the following scan classes for the interface** (*What's a Scan Class?* on page 42):

    `00:00:05`
    `00:01:00`

11. Click **Apply**.

12. Under the **Service Tab**, type in a **user name** and **password** for a Windows account with Administrative privileges on the PI Server, then click the **Create** button. This installs the interface.
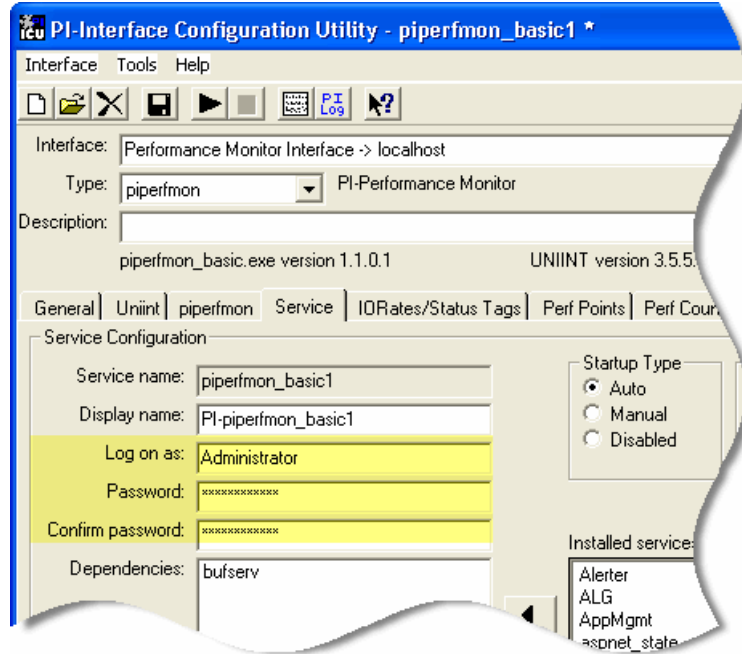
13. Click the **Start the Interface** arrow button at the top of the ICU. This starts the interface.

Chapter 11. MANAGING BUFFERING

If you are not currently using the buffering service on your Interface Nodes, refer to *Configuring Interfaces for Buffering* on page 45. Once it's running, buffering typically takes care of itself.

❑ *About Buffering*

❑ *Checking whether the Buffering Service Is Running*

❑ *Testing the Buffering Service*

❑ *If the Buffering Service Isn't Working*

❑ *Starting the Buffering Service*

11.1 About Buffering

OSIsoft provides a buffering service that can save your data if the Interface Node loses its connection to the PI Server. When an Interface Node is running the buffering service (**bufserv**), data flows from the data source, through the interface to the buffering service and from there to the Snapshot Subsystem on the PI Server.
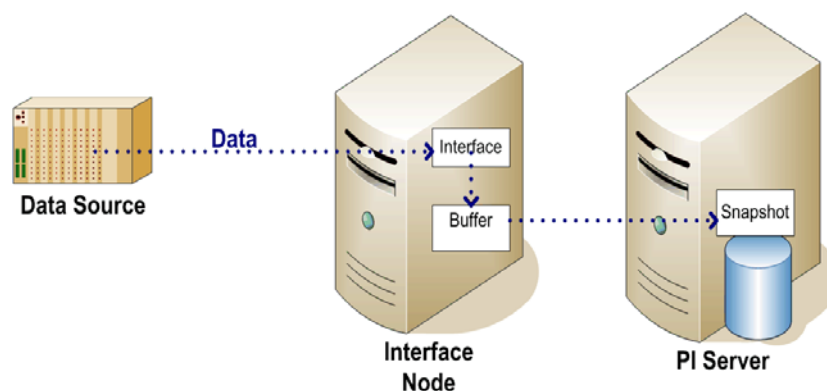
Data Flow with Buffering

If the PI Server is not available (e.g., for an upgrade on the Server) then **bufserv** stores the data in a file buffer on the Interface Node. When the PI Server becomes available again, **bufserv** sends all the stored data from the buffer, in chronological order, back to the PI

Server. At this point, if you look at the data in ProcessBook, you see a continuous flow of data, with no gaps.

As System Manager, you should make sure that the buffering service is running on each Interface Node. The main exception to this rule is for PINet Nodes, which perform their own buffering. Also, a few interfaces, such as batch file and event file, work better without the buffering service. If you're not sure whether a particular interface is compatible with the buffering service, check the documentation for that interface.

You need only one buffering service running on each Interface Node to buffer all the interfaces for a particular Server, however you can only buffer data to one Server at a time from each Interface Node. You can set the maximum size of the file buffer (up to 2GB) in the ICU (*Configuring Interfaces* on page 43).

11.2   Checking whether the Buffering Service Is Running

To check whether buffering is running on an Interface Node look for the APIBE process for that particular Interface Node in the Network Manager Statistics SMT plug-in:

1. Run the **System Management Tools** (*How to Run SMT* on page 5).

2. **Select the Server** that is connected to the Interface Nodes you want to check (*How to Select a Server in SMT* on page 6).

3. It's easier to check one PI Server at a time, so while you're in the PI Servers pane, click to deselect all other PI Servers. Now there should be only one checkmark in the PI Servers pane.

4. Click to expand the **Operation** item in the System Management Plug-ins pane.
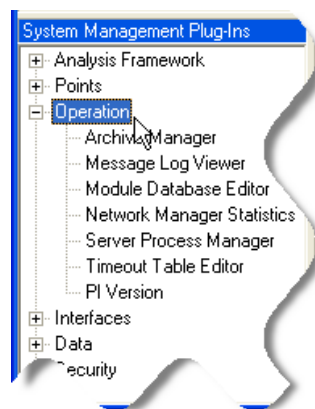
5. From the list of Operation plug-ins, double-click on **Network Manager Statistics**. The Network Manager Statistics plug-in appears in the Active Plug-in pane.

6. Click on the **Name** column header to sort the list alphabetically by service (or application) name.

7. Now look in the Name column for the buffer service, which in this display is called **APIBE**. You should see one APIBE connection for every Interface Node that is running the buffering service.

8. For each APIBE row in the table, look under the **PeerName** column to find the hostname for the Interface Node associated with that particular buffering service. (The IP address is also listed, under the PeerAddress column.)

9. Interface Nodes that do not have an APIBE application associated with them are not running the buffering service.

10. **Start the buffering service** on all Interface Nodes where it is not running (*Starting the Buffering Service* on page 68).

Note: If you are looking for a particular Interface Node among many, click the PeerName heading to sort the table by host name.

11.3   Testing the Buffering Service

The only way to actually test whether the buffering service is working is to interrupt the connection between the PI Server and the Interface Node for a period of time and then, after restoring the connection, check to see whether you have data:

11.3.1 When should I test the buffering service?

The best time to check the buffering service is when you're shutting down the PI Server anyway. For example, if you stop your PI Server for software upgrades or other maintenance, be sure to check for data loss after you restart.

Also, if your network connection goes down so that the PI Server loses its connection to the Interface Node anyway, that's a great time to check that your data was saved by the buffering service.

If you decide to interrupt the connection between an Interface Node and the PI Server for a buffering test, choose a time when your PI users will be least inconvenienced by missing data and access to the PI Server. Do this test once or twice a year.

11.3.2 How do I test the buffering service?

To test the buffering service, unplug the PI Server from the network:

1. Choose a time when your PI users will be least inconvenienced by missing data and access to the PI Server.

2. Send a memo or email warning users in advance of the test and telling them how long the Server will be off the network.

3. Unplug the network connections from the back of your PI Server for a few minutes.

4. Plug the connections back into the PI Server.

5. Notify your users that the PI Server is back on the network.

6. Open ProcessBook and look to see if you have continuous data (it might take several minutes for PI to fully restore the data). If you do have continuous data, then your buffering is working correctly. If you have gaps in your data, then the buffering isn't working properly (*If the Buffering Service Isn't Working* on page 66).

Note: Why unplug the PI Server and not the Interface Node? Because many Interface Nodes get their data from the network. If you unplug such an Interface Node from the data source, then the interface gets no data and you can't test data buffering.

11.4 If the Buffering Service Isn't Working

If the buffering service is not working correctly, first check the login information for the buffering service, then check the *pipc.log* file.

11.4.1 How to Check the Login Information for Buffering

Be sure that bufserv service is set to run under a local Windows Administrator account on the Interface Node, not under System Account, or any other unprivileged account. To check this, follow these steps:

1. From the Windows **Start** menu, open the **Control Panel** and then open **Administrative Tools**.

2. In the **Administrative Tools** folder, double-click on **Services**.

3. In the **Services** window, right-click on **PI-Buffer Server** and select **Properties** from the resulting pop-up menu.

4. Click the **Log On** tab and enter the account information for a Windows account that has administrative privileges on this Interface Node. Do not use the System account.
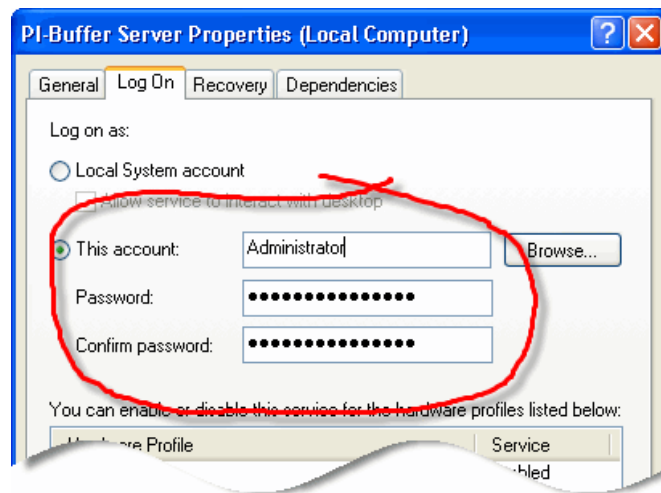
5. Click on the **General** tab and click the **Start** button under Service status.
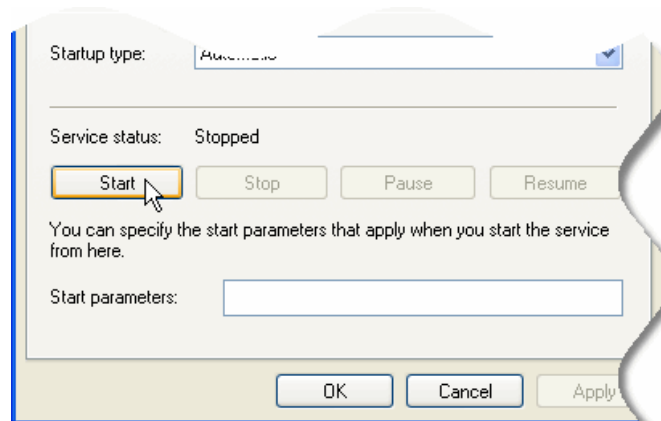
6. Click **OK**.

Note: Make sure the interface that you are buffering is running under this same Windows account. The buffering program shares memory between bufserv and the interface. If they are not running under the same account then they can not share memory and buffering will not work.

11.4.2 How to Check the pipc.log File

PI writes messages about the interfaces in the *pipc.log* file. If you're having trouble with the buffering service, it's a good idea to check the log file:

1. From the Windows **Start** menu on the Interface Node, point to **Programs**, point to **FactoryTalk Historian System**, and then click **PI-Interface Configuration Utility**. The ICU appears.

2. On the **Tools** pull-down menu, click **Log Files…** The Log File dialog box appears.
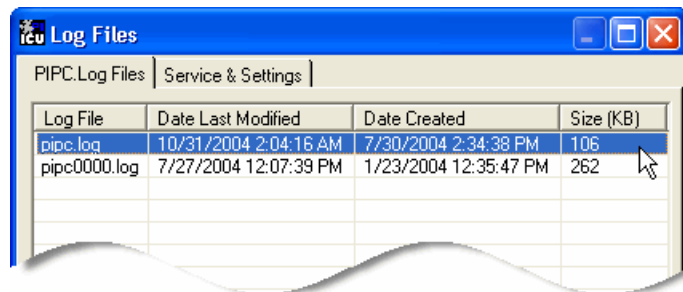
3. In the Log Files dialog box, double-click on the **pipc.log** entry. The *pipc.log* file appears.

4. If you see the error message, "Unable to create shared memory buffers" you're probably trying to run the buffering service under an account that does not have administrative privileges. Follow the instructions in *Starting the Buffering Service* on page 68.
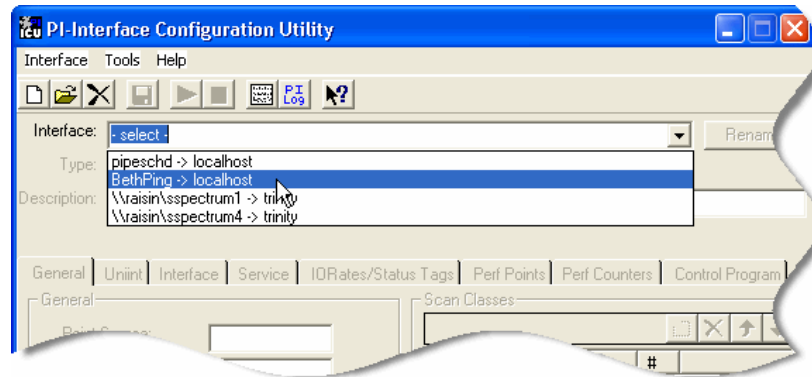
11.5 Starting the Buffering Service

These instructions assume that you are starting the buffering service on an Interface Node that is configured for buffering (*Managing Buffering* on page 63) and that the following items are installed on the Interface Node:

❑ The interface software for the interface(s) you want to buffer

❑ PI Interface Configuration Utility (ICU)

❑ PI API (this is installed with the interface software)

❑ PI-SDK (this is installed with the ICU)

To start the buffering service on an Interface Node, follow these steps:

1. From the Windows **Start** menu on the Interface Node, point to **Programs**, point to **FactoryTalk Historian System**, and then click **PI-Interface Configuration Utility**. The ICU appears.

2. **Select an interface** from the Interface pull-down menu. If the interface you want does not appear in the Interface pull-down menu, then you need to register that interface with the ICU *(Configuring Interfaces* on page 43).

3. On the **Tools** menu, point to **API Buffering**. The Buffering dialog box appears.

4. Under the **Settings** tab, make sure the **Enable buffering** checkbox is checked.
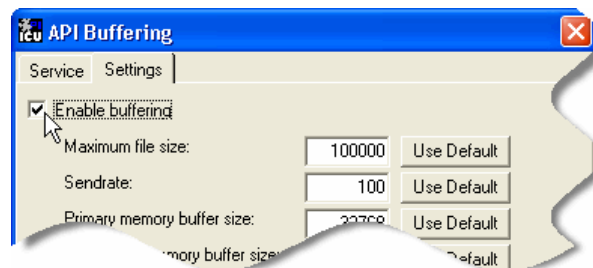

5. Change the **maximum file size**, if necessary. The maximum is 2GB.

6. Under the **Service** tab, type in the **name** and **password** for a Windows account that has administrative privileges on this computer.
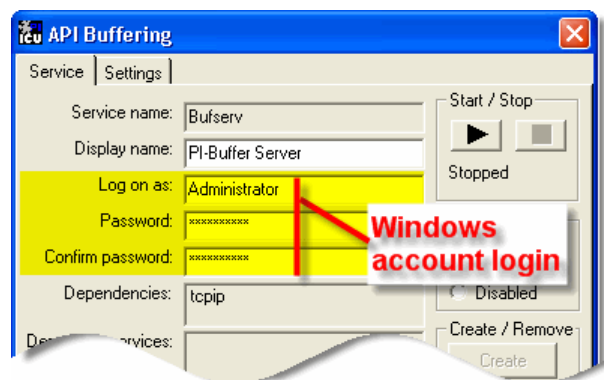
7. Click **OK.**

8. From the Windows **Start** menu, open the **Control Panel** and then open **Administrative Tools**.

9. In the Administrative Tools folder, double-click on **Services**.

10. In the Services window, right-click on **PI-Buffer Server** and select **Start** from the resulting pop-up menu. The buffering service should start now. If it doesn't, see *If the Buffering Service Isn't Working* on page 66.
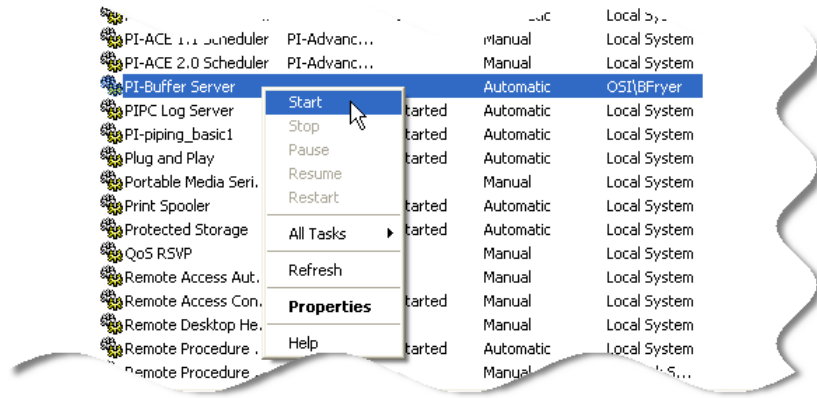
Chapter 12.   MANAGING DATA SOURCE EQUIPMENT

Changes to your plant or process equipment sometimes mean you need to make changes to PI.

- ❑   About Data Sources
- ❑   Adding New Equipment
- ❑   Replacing Equipment
- ❑   Removing Obsolete Equipment

12.1   About Data Sources

Your data sources can be almost anything, including Distributed Control Systems (DCSs), Programmable Logic Controllers (PLCs), lab systems, Supervisory Control and Data Acquisition systems (SCADA), process models, and other business information systems.

12.2   Adding New Equipment

When new equipment comes online, you need to configure PI to recognize it and set up points to collect the data:

- ❑   Connect the equipment to an Interface Node and install the appropriate interface software. You can download interface documentation from the Technical Support Web Site (http://techsupport.osisoft.com). If this is a new Interface Node, you also need to install the ICU (*How to Install the ICU* on page 6) and set up the trusts (*Managing PI Trusts* on page 52).

- ❑   Install the interface according to the instructions in the interface documentation.

- ❑   Register and configure the interface with the ICU (*Configuring Interfaces* on page 43).

- ❑   Start the buffering service, if it isn't already running on this Interface Node (*Starting the Buffering Service* on page 68).

- ❑   Build a new point(s) to get the data from the equipment into PI (*Creating New Points* on page 23).

12.3   Removing Obsolete Equipment

When equipment goes offline, you need to *decommission* any points associated with that equipment (*Decommissioning Points* on page 24).

If you do not decommission obsolete points, the FactoryTalk Historian System continues to try to get values for them, which is bad for system performance and can sometimes even lead to data loss for other points.

12.4   Replacing Equipment

If you replace an instrument with a different one that measures the same process value, it's usually best to continue using the same PI point. Edit the point as required so that it will collect the new data. If the instrument is significantly different, you might need to adjust the compression and exception attributes, among others. Don't change the **Tag** attribute.

When you change the point, insert a digital event into the data to indicate when the transition from the old to the new instrument took place.

Chapter 13. WHERE TO GO TO GET MORE HELP

13.1 Training

OSIsoft offers training classes and computer-based training (CBT) CDs on managing PI Servers. For more information, look at the OSIsoft training website:

http://training.osisoft.com

13.2 Technical Support Web Site

The OSIsoft Technical Support Web site (http://techsupport.osisoft.com) has great information for FactoryTalk Historian System Managers. Check the site daily for support bulletins and for information about software updates.

You can also download software and documentation from the Download Center on the Technical Support Web site. To do this, you must first sign in (or register and then sign in). This Web site also has a System Manager Resources area that provides how-to information, as well as tips and tricks for FactoryTalk Historian System Managers.

13.3 Developers Network

The OSIsoft Developers Network (DevNet) is a Web site dedicated to helping PI users and developers create productive PI applications. PI users post a lot of these homegrown applications on the DevNet site so, even if you're not interested in developing your own PI applications, you're still likely to find some useful tools and information on DevNet. The DevNet Web site is here:

http://devnet.osisoft.com

Chapter 14. GLOSSARY

Annotation

Arbitrary information (e.g. text comment, other binary data) that can be associated with any archive value. Annotations are accessed exclusively with the PI-SDK and are stored in an archive file's corresponding annotation file (<archive_filename>.ANN). The maximum size of an annotation is equal to the page size in the Event Queue, which is 1 MB by default with a maximum of 8 MB.

API

A C-based Application Programming Interface library of functions that enable programs to access PI Servers locally or remotely across a network. FactoryTalk Historian ProcessBook 2.x (and earlier), FactoryTalk Historian DataLink 2.x (and earlier), a majority of interfaces, and many custom programs depend on this library.

API Node

A synonym for Interface Node.

Archive

The historical record of time-series data maintained by the PI Server. The term is often overloaded: sometimes it's used to refer to the entire logical data record itself; sometimes it's used to refer to a specific archive file; and sometimes it's used to refer to the subsystem responsible for actively hosting the historical data record.

Archive Event

Any Event that is stored in the Archive.

Archive File

A binary file that contains a section of the data archive covering some finite time range. These files, defined by start and end times, should be contiguous and non-overlapping. Two types of archive files may be created: Dynamic and Fixed.

Archive Gap

A non-zero period of time between the end time of one archive file and the start time of the chronologically next archive file. Archive gaps are not desirable because archive events with a timestamp during the gap cannot be stored on disk in an archive file and will be discarded. To avoid archive gaps, archive files should always be created such that the end time of one archive equals the start time of the chronologically next archive.

Archive Queue

A less commonly used synonym for Event Queue.

Archive Shift

The process of clearing the oldest writeable and shiftable archive file and making it the new primary archive. An archive shift typically happens automatically when the previous primary archive becomes full, but it sometimes must be performed manually for maintenance and troubleshooting purposes.

Archive Shift Flag

A flag that controls whether or not a particular archive file can participate in archive shifts. If the flag is disabled (set to 0), then that specific archive file will not participate in archive shifts. Unlike a fixed archive, once a dynamic archive receives any data, the shift flag is automatically and permanently disabled.

Archive Subsystem

The core PI Server component that is responsible for writing data to, reading data from, and otherwise managing the complete data archive. The Archive Subsystem is very tightly coupled with the Snapshot Subsystem, which is actually responsible for performing compression on the incoming data.

Argument, Command-Line

User input specified after the name of a program to control or modify the behavior of that program in some fashion. A command line argument must typically be separated from the program name and other command line arguments by at least one space. Depending on the program, command line arguments must typically be prefixed by a hyphen ('-') or a slash ('/'). Several of the diagnostic utility programs that are distributed with the PI Server like piartool and pidiag will require the use of one or more command line arguments.

Attribute, Point

A characteristic or parameter of a point that directs an interface and the PI Server in the collection and processing of data values for that point.

Attribute Set

A named collection of attributes. One or more attribute sets are used to define point classes to establish the complete list of attributes that can specified when creating or modifying a point of that class.

Base Point Class

The common set of point attributes that all other point classes include. The Base class includes both system- and user-assigned attributes.

Base Subsystem

The core PI Server component that is responsible for hosting several configuration stores such as the Point database, the User and Group database, and the Trust table. The Base Subsystem also hosts the hierarchical Module Database.

Batch

A batch represents a span of time on a unit.

Batch Alias

An additional name, usually the common name, for a unit attribute. A batch alias allows batch users and applications to reference the more natural common name of a unit attribute instead of its more obscure instrument name, which may only be readily understood by the instrument or process engineer.

Batch Database (BDB)

A logical collection of batch objects hosted by the Module Database and the Archive Subsystem. All access to the Batch Database is provided exclusively by the SDK. The Batch Database is independent of the older batch information store maintained by the Batch Subsystem.

Batch Generator (PIBaGen)

An SDK-based interface that writes batch information into the Batch Database. The interface detects batch activity by monitoring specific points on the PI Server for events that trigger the beginning and ending of a batch.

Batch Subsystem (BSS)

A PI Server process that is responsible for configuring, monitoring, and recording batch activity in the data archive. The main interface to the Batch Subsystem is provided by piconfig. Read-only access to the batches recorded by the Batch Subsystem is typically provided by the API, but it is possible to map Batch Subsystem units to the Batch Database so that their batches can also be accessed by

the SDK. The batch information store maintained by the Batch Subsystem is independent of the newer Batch Database.

BatchView

A Windows client application that allows the viewing of batch data from the Batch Database and the Batch Subsystem. BatchView consists of three different components: an SDK-based ProcessBook Add-In, an API-based Excel Add-in, and a stand-alone SDK-based application for quick batch searching.

Blob Point Type

The acronym for Binary large object and the point type typically chosen for an arbitrary unstructured array of bytes. Interpretation of the bytes can only be performed by the retrieving application. Because the record size in archive files is currently fixed in size at 1024 bytes, the value of a single event for a Blob point can contain binary data of up to only 976 bytes in length. If larger binary objects must be written, then either the binary data must be split into multiple events for the Blob point or the data must be stored as an annotation.

Buffering Service

An API process that typically runs on an Interface Node for the purpose of storing interface data during periods when network communication to the PI Server is unavailable. When network communication is restored, the buffering service relays the stored data to the PI Server. Because the buffering service works to help prevent data loss but is disabled by default, enabling and configuring buffering is a critical task for an administrator.

Bufserv

The executable or process name that performs the Buffering Service.

Calculated Point

A synonym for PE Point.

Calculated Tag

A synonym for PE Point.

Classic Point Class

The common set of attributes required by most standard OSIsoft interfaces, such as Location1 and Location4.

Clock Scheduling

A method of triggering program execution to occur based on a fixed time or clock schedule such as that defined by a scan class. Clock scheduling is one method available for triggering PE or ACE calculations. Another method is Event Scheduling.

COM Connector

A COM object designed to allow the PI Server to access data from foreign data sources and make it available to any PI client application in a seamless fashion. Some currently available COM Connectors include those for data historians from AspenTech and Honeywell as well as one for any data source with an OLEDB provider. In order to function, all COM Connectors require the services provided by the Redirector. COM Connectors are only available on Windows platforms.

COM Connector Point

A point belonging to a special point class that has at least three attributes defining how to reference a particular value on a foreign data source. One of the attributes is the name of the COM Connector needed to communicate with the foreign data source. These points are only available on Windows platforms.

CompDev

The base attribute that specifies the compression deviation in engineering units. This represents the maximum error when historical data values need to be interpolated and, at the very least, should typically be set to the error of the underlying instrument.

CompDevPercent

The base attribute that specifies compression deviation as a percentage of Span, another base attribute. The relationship is defined by the following equation: CompDev = (CompDevPercent / 100) * Span. If both CompDev and CompDevPercent are specified when creating or editing a point, CompDevPercent takes precedence.

CompMax

The base attribute that specifies the compression maximum time, in seconds. CompMax is the maximum time difference from the previous archive event before the next event will be sent to the archive. Because the PI Server itself never generates events, a lower bound on the archiving rate for the associated point cannot be determined from CompMax alone.

CompMin

The base attribute that specifies the compression minimum time, in seconds. CompMin is the minimum time difference from the previous archive event before the

next event is eligible to be archived. Because the PI Server itself never generates events, the archiving rate for the associated point will be at most one event every CompMin seconds.

Compressing

The base point attribute that controls whether or not compression is performed for a particular point. If Compressing is disabled (set to 0), then all events will bypass compression.

Compression

The process of selecting which Snapshot events will be sent to the Archive for storage. Applying compression is one of the main responsibilities of the Snapshot Subsystem, and the specific algorithm used is known as Swinging Door Compression.

Compression Specification

The three base attributes that control the compression process for a particular point: CompDev, CompMax, CompMin. Although they are technically not included in the specification, CompDevPercent and Span affect CompDev, and Compressing determines whether the specification is needed at all.

Connection Credentials

The set of identifying information about a client application seeking connection to the PI Server. This information can include the client computer's IP address or hostname, the client application's name, or the Windows Domain name and Windows user name under which the client application is running. API applications are restricted in the credentials that they can specify. The PI Server uses connection credentials to determine if there is a matching Trust.

D Point Type

Interface manuals sometimes refer to the D point type. This is synonymous with Digital Point Type.

Data Archive

The fundamental and most important information store of the PI Server that contains the historical data record of all events for all points. The Data Archive is commonly referred to as simply the Archive.

Data Source Node

A synonym for Interface Node.

Data Type

The kind of value that will be used. Both points and point attributes have a data type. Some of the possible types include several kinds of numbers, digital, string, and Blob.

Descriptor

The base point attribute that can be used to provide a textual description of a point. The Descriptor is a common attribute to display in various client applications and user reports.

Deviation Blanket

In compression, the conceptual parallelogram with a width that extends from the previous archive event to the current event and a height equal to twice the compression deviation.

DigitalSet

The base attribute required for all digital points to indicate the appropriate digital set containing the list of possible digital state values for the point.

Digital Point Type

A point type typically used when values can only be one of several discrete states, such as ON/OFF or Red/Green/Yellow. This point type is the nearest equivalent to the PI 2.x Digital type.

Digital State Set

A named collection of digital states. For example, a digital state set called ValveStates may contain the two possible discrete states of a valve: OPEN and CLOSED.

Digital State Table

A table that contains the complete definition of all defined digital state sets. This table is hosted by the Base Subsystem.

Distributed Data Collection

Gathering data from multiple sources and from more than one computer on a network. The FactoryTalk Historian System is designed to work extremely well in an environment with distributed data collection.

Dynamic Archive

A type of archive file that does not pre-allocate its disk space at creation time but instead grows as needed. A dynamic archive can be configured to grow up to a maximum size and support a maximum number of points, but a non-empty dynamic archive cannot participate in archive shifts. Another archive type is a Fixed Archive.

Event

The fundamental unit of information used in the PI Server. Each event consists of two main components: a value and a timestamp. The value can be one of several different data types (e.g. string, digital, int32, float64). The timestamp is always represented as UTC seconds and can contain a sub-second component.

Event Queue

A buffer consisting of one or more memory-mapped files that stores events that have passed or bypassed compression and are destined for archive storage. The Snapshot Subsystem writes events into the Event Queue, and the Archive Subsystem reads events out of the Event Queue. While events are still in the Event Queue, they are not visible by any client applications. Under normal operating conditions, the Event Queue should typically be empty.

Event Scheduling

A method of triggering program execution when some specific condition occurs such as the arrival of a new Snapshot event for a particular point. Event scheduling is one method available for triggering PE or ACE calculations. Another method is Clock Scheduling.

ExcDev

The base attribute that specifies exception deviation in engineering units. ExcDev specifies the deadband or how much a new value must differ from the previous value sent to the Snapshot Subsystem on the PI Server in order to determine whether the new value is significant and should also be sent.

ExcDevPercent

The base attribute that specifies exception deviation as a percentage of Span, another base attribute. The relationship is defined by the following equation: ExcDev = (ExcDevPercent / 100) * Span. If both ExcDev and ExcDevPercent are specified when creating or editing a point, ExcDevPercent takes precedence.

Exception Reporting

The process, normally executed by an interface program or external system, of sending events to the Snapshot Subsystem on the PI Server only when there has been

a significant change in the monitored value. Significance is determined with a simple deadband algorithm.

Exception Specification

The three base attributes that control the exception reporting process for a particular point: ExcDev, ExcMax, and ExcMin. Although they are technically not included in the specification, ExcDevPercent and Span affect ExcDev.

ExcMax

The base attribute that specifies exception maximum time, in seconds. ExcMax is the maximum time difference from the last sent event before the next event will be sent. ExcMax thus effectively limits the length of time that events can be discarded because their values did not exceed exception deviation.

ExcMin

The base attribute that specifies exception minimum time, in seconds. ExcMin is the minimum time difference from the last sent event before the next event is eligible to be sent. Thus, the send rate of events for the associated point can be at most one event every ExcMin seconds.

Firewall

A table hosted by Network Manager that provides the first level of security access to a PI Server. Access can either be allowed or disallowed based on the IP address or hostname of a client computer.

Fixed Archive

A type of archive file that allocates all of its disk space at creation time. Thus, both an empty and full archive occupy the same amount of disk space. Unless shifting has explicitly been disabled, non-empty fixed archives will participate in archive shifts. Another archive type is a Dynamic Archive.

Float16 Point Type

The only floating-point type that is scaled. The accuracy is one part in 32767, and the range is defined by the Zero and Span base attributes. This type is the nearest equivalent to the PI 2.x Real type.

Float32 Point Type

The floating-point type typically chosen for single-precision floating-point values. This type is not scaled.

Float64 Point Type

The only floating-point type capable of storing double-precision floating-point values. This type is not scaled.

Home Node

A computer running the PI Server software or the network location (IP address or hostname) of such a computer.

I Point Type

Interface manuals sometimes refer to the I point type. This is synonymous with either Int16 Point Type or Int32 Point Type.

Initializing Archive

The process of writing all the primary records, one for each existing point, to an archive file and cleaning and preparing overflow records in order to receive data.

Int16 Point Type

The integer point type typically chosen for values that are 15-bit unsigned integers (0 to 32767). This type is the nearest equivalent to the PI 2.x Integer type.

Int32 Point Type

The only integer type capable of representing 32-bit signed integers. Because the lowest 32K values of the 32-bit range are reserved for digital states, the effective useful range of possible integer values is -2147450880 to 2147483647.

Interface Configuration Utility (ICU)

A Windows GUI application installed on an Interface Node for the purpose of easing the burden of managing interfaces and their configuration files.

Interface Node

A computer running one or more PI interfaces or the network location (IP address or hostname) of such a computer.

Interface

A software program that collects data from some type of data source and sends the data to a PI Server. Some interfaces also have the ability to read data from a PI Server and write back to the data source.

Interface Status Utility (ISU)

A standalone program for determining whether or not an interface is sending fresh data to the PI Server.

Ipisql Utility

An interactive command line program that executes SQL statements directed at the PI Server. The utility depends on the API to communicate to the PI Server.

Mapped Point

A synonym for COM Connector Point.

Message Subsystem

The PI Server component that records informational and error messages from various PI Server subsystems in a series of log files. The Message Subsystem can also serve these messages to various client applications.

Module Database (MDB)

A hierarchical information store hosted by the Base Subsystem that consists of one or more Modules. Each Module contains collections of Properties, Aliases, and other Modules.

Module Database Builder

An SDK-based Excel Add-In that allows the creation, modification, or viewing of elements of the Module Database in a spreadsheet. The Module Database Builder is the ideal user interface when performing bulk operations on the Module Database.

Network Manager

The core PI Server component that handles all communication between the PI Server subsystems. Network Manager also manages all connections from client applications and their communication with the PI Server.

Node

A computer on a network or the network location (IP address or hostname) of such a computer.

ODBC

The driver software that exposes a PI Server as an ODBC-compliant data source and thus provides the PI Server with the ability to communicate with any ODBC-compliant client application that needs to access the process data stored on the PI Server.

Offline Archive Utility

The same program that runs the Archive Subsystem, but just in a different mode. The offline archive utility is used for a variety of archive maintenance tasks such as merging multiple archives files into a single archive file or reprocessing an archive file to recover it from a corrupt or failed state.

Offset

An optional field used when defining a scan class that specifies the first time at which a scan should occur. If no offset is specified, the first scan occurs immediately after the specified interval. After the initial scan, subsequent scans continue to occur after every specified interval.

Out of Order Event

An incoming event whose timestamp is prior to the timestamp of the event currently residing in the Snapshot table for a particular point. All such events bypass compression and are written directly to the Event Queue.

Owner

The individual user that has permission to view and edit a resource on the PI Server. Each resource can have only one owner. Two examples of resources include a point's data (attribute name: DataOwner) and a point's configuration (attribute name: PtOwner).

PE Point

A PI point whose value is calculated by the Performance Equation (PE) Scheduler based on the point's configured performance equation specified in the ExDesc base attribute.

Perfmon (PIPerfmon)

The Windows-only Performance Monitor interface which reads Windows Performance Counters and stores the values in PI points. The basic version of the interface can only monitor a limited number of Windows Performance Counters from the local computer.

Performance Equation (PE)

An expression that allows a user to implement an arbitrary and potentially sophisticated calculation without formal programming. A performance equation has an intuitive syntax and may consist of standard mathematical and logical operators as well as a wide variety of built-in functions. The result of a performance equation can be archived for a PE point just like data for any other point. Performance equations are also available programmatically via the SDK for archive calculations and other data filtering operations.

Performance Equation Scheduler (PIPESCHD)

The hybrid interface and subsystem on the PI Server responsible for evaluating performance equations for all points that specify its point source (default: 'C').

Performance Point

An overloaded term that can mean either a point associated with the Perfmon interface or a special point used to monitor interface performance on a per-scan class basis. In the case of monitoring interface performance, the point tracks how long (in seconds) the interface took to collect data for all tags in that scan class for each scan.

PI2

Nickname for the original generation of the PI Server software that (still) runs on VAX and Alpha computers with the OpenVMS operating system.

PI3

Nickname for the current generation of the PI Server software that runs on computers with the Windows or UNIX operating systems.

PI Server

The set of several software subsystems packaged together that constitute a single logical server application capable of storing time-series data from distributed data sources and serving this same data to client applications in real-time.

FactoryTalk Historian System

The complete collection of OSIsoft software applications running on one or more computers that function to collect, store, retrieve, analyze, view, or manage process data. Examples of these software applications include interfaces, the PI Server, and client applications.

Piarchss

The executable or process name that implements both the Archive Subsystem and the Offline Archive Utility.

Piarcreate

A command line utility program for creating both fixed and dynamic archive files. After creation, the archive files must be registered with the Archive Subsystem in order to be made available for use.

Piartool

A command line utility program that provides a number of diagnostic and management functions. The PI Server must be started for nearly all of the commands to function properly.

Pibasess

The executable or process name that implements the Base Subsystem.

Pibatch

The executable or process name that implements the Batch Subsystem.

Piconfig

An interactive command line utility program that provides access to nearly all the configuration and data stores maintained by the PI Server. Several client applications have come along to provide a graphical interface to these various tables and databases, but certain tasks can still only be performed with this command line utility. To achieve some degree of automation, a series of piconfig commands can be saved to a text file which can then be passed as input to piconfig.

Pigetmsg

A command line utility program that allows the viewing of PI Server messages stored by the Message Subsystem. Messages can be retrieved based on characteristics like timestamp, a search string, or the program name that generated the message.

Pilistupd

A command line utility program that displays information about the registered consumers and producers maintained by Update Manager. Consumer info includes the number of outstanding events in its buffer.

Pinetmgr

The executable or process name that implements Network Manager.

PINet on OpenVMS

Software running on a remote VMS computer that collects data from interfaces and sends it to a PI Server for data archiving.

Ping

An interface that monitors the network availability of computers by directing an ICMP ping request agai them and then storing the response times in PI points. A basic version of the interface is included with the PI Server.

PIonPINet/VMS

Software running on a remote VMS computer that includes all the PINet on OpenVMS functionality as well as additional utility programs for analysis, reporting, and graphical displays of process data.

Pipeschd.bat

The script file containing the startup configuration for the Performance Equation Scheduler.

PIPOINT Table

A synonym for Point Database. In piconfig, the table that provides access to the Point Database.

Pishutev

The executable or process name that implements the Shutdown Subsystem.

Pisnapss

The executable or process name that implements the Snapshot Subsystem.

Pisqlss

The executable or process name that implements the SQL Subsystem.

Piupdmgr

The executable or process name that implements Update Manager.

Point

A variable whose value is measurable and typically dynamic. Examples include transmitter readings, status indicators, manual inputs, control limits, etc. Each point must be assigned a unique tag on the PI Server, and measurements of the point captured over time are effectively stored as an array of timestamped values in the data archive.

Point Class

A collection of one or more attribute sets. Examples of point classes include Base, Classic, Alarm, and Totalizer. All point classes include all the attributes from the Base class, which has a core set of attributes needed by various processes in the FactoryTalk Historian System. Other point classes add attributes needed to provide functionality for certain processes. The base attribute PtClassName specifies the point class for every point.

Point Configuration

The complete list of attributes characterizing a point.

Point Database

The information store that contains the list of all points and their complete point configuration. The list includes both typical points that have their data stored in the archive and COM connector points that have their data stored on foreign data sources. The point database is hosted by the Base Subsystem.

Point Identifier (PointID)

The unique number used to identify a point. This is primarily meant for internal use within the PI Server, but it is often needed in API / SDK programming and troubleshooting scenarios. Point identifiers are assigned sequentially as points are created, and they are not reused if points are deleted. The base attribute PointID stores a point's assigned number after creation.

Point Security

Access control for a point which consists of specifying an owner and a group and the respective read / write permissions. Each point has one security specification for controlling access to its attribute configuration and a second separate security specification for controlling access to its archive data. The base attribute PtAccess holds the security specification for configuration, and the base attribute DataAccess holds the security specification for archive data.

PointType

The base attribute that specifies the data type for the values that a point stores. The possible point types include the following: int16, int32, float16, float32, float64, digital, string, Blob, and timestamp. PointType can be edited after point creation, but not all type transitions are allowed.

PointSource

The base attribute that identifies the interface or other scanning software responsible for providing data for the associated point.

Posting

Sending events packaged into messages that contain either 128 or 256 events (depending on the server platform), from the Snapshot Subsystem to the Archive Subsystem. Posting is typically no longer performed with the introduction of the memory-mapped Event Queue in later versions of the PI Server.

Postprocessing

Processing by the Totalizer on the values stored in the Snapshot table that enables accurate counting and summary calculations. The results of these operations are then stored in other points.

Primary Archive

The archive file with an end time of current time. All events recorded with a timestamp after the start time of the primary archive are stored in this archive file. Thus, the primary archive typically contains the most recent data for all points. At most one primary archive may be registered at any given time.

Product

In batch processing, the description of a specific material or class of materials. This term is used in batch applications that use equipment to produce a variety of different materials.

R Point Type

Interface manuals sometimes refer to the R point type. This is synonymous with the Float16 Point Type or Float32 Point Type or Float64 Point Type.

Ramp Soak

A standard interface program included with the PI Server that generates signals that might have come from a batch process. This interface is useful for testing and validating a PI Server without affecting actual process data.

Random Simulator

A standard interface program included with the PI Server that is capable of generating a sinusoidal wave and several kinds of pseudo-random data. This interface is useful for testing and validating a PI Server without affecting actual process data.

Real-time SQC

The component of the PI Server that provides continual evaluation of Statistical Quality Control pattern tests and the management of alarms generated from these tests. Use of this component will assist in monitoring how well a process stays within its control limits.

Recalculator

The component of the PI Server that adjusts the values of PE points automatically whenever the values of points used in their expressions are added, edited, or deleted by any application.

Redirector

The component of the PI Server that functions as the intermediary between server subsystems and all COM connectors. The redirector is an out-of-process COM server that is only available on Windows platforms.

Registering an archive

Informing the Archive Subsystem of the name and location of an archive file that is available for use. Registering an archive can be performed with several different programs including piartool and SMT.

Satellite node

Any remote computer on a network running PI software other than the PI Server software. Examples of the software the computer might be running include interfaces, PINet, or PIonPINet.

Scan

The base attribute that specifies whether or not the interface or scanning program should collect new data for the associated point. If Scan is disabled (set to 0), then new data will not be collected.

Scan Class

A specification that provides an interface with the schedule for performing data collection for its associated points. The scan class specification consists of a period and an optional offset. The period determines the recurring interval when data collection should occur, and the offset determines when data collection should first start. A scan class can also optionally contain a code to force the interface to use UTC time for scheduling. A point can only be in one scan class, and assignment to a scan class is typically configured through the classic attribute Location4.

SDK

A COM-based Software Development Kit that provides rich access to objects and data stored on the PI Server. The SDK is used for other PI applications like ProcessBook 3.x and DataLink 3.x and also for custom user applications. The SDK is only available for Windows platforms, but it can access a PI Server running on any supporting operating system. The distribution kit for the SDK also includes the API.

Shutdown Event

An Event whose value consists of the Shutdown digital state from the SYSTEM digital set and whose timestamp is intended to mark when the PI Server or an interface or some other application or device is not available.

Shutdown Subsystem

The component of the PI Server that writes a shutdown event for all points that match a particular tag mask and attribute selection. By default, any tag with its base attribute Shutdown set to 1 will receive a shutdown event. After the Shutdown Subsystem writes all the shutdown events for the appropriate points, it will stop running.

Snapshot Event

An overloaded term that can refer to either any Event sent to the Snapshot Subsystem or the event currently residing in the Snapshot table for a particular point. The event stored in the Snapshot table for each point has the most recent timestamp of all events received so far for that point; when a new event arrives with a more recent timestamp, the previous event is passed through the compression filter.

Snapshot Subsystem

The core component of the PI Server that receives all the new data events for all points regardless of the sending application. The most recent of these events for each point is maintained in the Snapshot table along with additional information necessary to perform compression. Besides performing compression and writing events to the Event Queue, the Snapshot Subsystem responds to client requests for Snapshot events and forwards Snapshot events for requested points to Update Manager.

SNMP Interface

An interface included that collects performance data from any device that supports the Simple Network Management Protocol and then stores the values in PI points. Examples of devices include computers, printers, and routers. A basic version of the interface is included with the PI Server for Windows.

SNMP Point Builder

An SMT plug-in useful for creating and editing points for the SNMP Interface.

Span

The base point attribute that specifies the range or the difference between the maximum and minimum values for a point. Span is required for all numeric points and is linked to compression and exception specifications through CompDevPercent and ExcDevPercent, respectively. Span is only enforced for values for float16 points.

SQC

The SDK-based Add-In to ProcessBook that enables users to create and view a variety of Statistical Quality Control charts on their ProcessBook displays. SQC chart limits can be PI points, manually entered constants, or values from ODBC datasets defined within ProcessBook.

SQC Alarm Manager

A stand-alone client application used for managing Real-Time SQC Alarms on a PI Server. This application has been replaced with the SMT SQC Alarms plug-in that provides equivalent functionality.

SQL Subsystem

The component of the PI Server that prepares and executes Structured Query Language statements directed against it from mainly ODBC and SDK applications. The existence of the SQL Subsystem allows clients to access PI Server information stores like the Data Archive and Point Database using the same SQL syntax used to interact with relational databases.

State Set

A synonym for Digital State Set or DigitalSet.

Statement Handle

An object allocated by the SQL Subsystem to enable servicing of a SQL request or statement.

Status

The classification of an Event depending on the nature of its value. If the event has a valid value considering the type of point, then the event is considered to have a Good status. If the event has a SYSTEM digital state as a value, then the event is considered to have a Bad status.

Steam Functions

A set of built-in functions available within a Performance Equation that calculate the thermodynamic properties of steam.

Step

The base attribute that specifies how to interpolate between successive archive events. If Step is non-zero, the value is assumed to change in a stepwise or staircase fashion.

String Point Type

The point type used for storing strings, sequences of alphanumeric characters, up to 976 characters in length.

Subnet

A networking term that refers to a range of numerical IP addresses.

Subsystem

A functionally distinct software component or module that executes in its own process space. The PI Server has several core subsystems such as Network Manager, Update Manager, Base, Snapshot, and Archive.

Swinging Door Compression

A data compression algorithm used by the Snapshot Subsystem that guarantees all of the original samples were within a specified value, the compression deviation, of a straight line drawn between any two events selected for archiving. In other words, this compression algorithm allows for the reconstruction of the original signal as a series of straight lines, and the maximum error between the reconstructed and original signals is guaranteed to be no more than the compression deviation.

System Digital State Set

The default digital state set that contains a few hundred digital states that may apply to any tag. States may be added to this set, but states in the offset range 193-320 are reserved for use by the FactoryTalk Historian System and should not be modified.

System Management Tools (SMT)

A set of easy-to-use programs for performing a wide variety of common administrative tasks in a FactoryTalk Historian System. The tools can be downloaded from the Technical Support Web Site (http://techsupport.osisoft.com).

Tag

The base attribute that is the unique alphanumeric name for a point. Certain characters are not allowed like '*', '?', '\', and ';'. The term Tag and Point are often used interchangeably.

Tag Configurator

An SDK-based Add-In to Excel that facilitates creating, editing, and viewing points from a spreadsheet. This is the ideal application for bulk point operations.

Timeout Table

The information store that contains all the configuration parameters for the PI Server. When tuning the performance of the PI Server, several of these Timeout Table parameters will typically need to be adjusted.

Timestamp

A date and time, almost always associated with a data value through an Event. The PI Server stores timestamps internally in UTC (Universal Coordinated Time).

Timestamp Point Type

The point type used to store values that are timestamps. The possible range of timestamps that can be stored is 1-Jan-1970 through 1-Jan-2038.

Totalizer Subsystem

The component of the PI Server that can be used to continuously calculate a variety of quantities like totals, averages, minimum and maximum values, and standard deviations.

Trust

A record stored on the PI Server that automatically grants access for a program connecting to the PI Server without requiring an explicit PI user login. A trust consists of one or more Connection Credentials criteria and the name of an existing PI user to be used for access. All trusts are stored in the Trust Table which is hosted by the Base Subsystem. Trust lookup is always performed when an application first connects to the PI Server.

Unit

In batch processing, the name of the equipment set on which batch activity takes place. The definition of a unit is not limited to a single piece of equipment. For example, a unit could be a single reactor or a group of reactors and related equipment.

Universal Data Server (UDS)

An obsolete name for the PI Server.

Update Manager

The core component of the PI Server that buffers data events and notifications of configuration changes for programs that have requested this service. For example, ProcessBook will request updates of Snapshot events for a point on a trend so that the trace will remain current; all such events will pass through Update Manager.

Zero Attribute

The base point attribute that indicates the lowest possible value for a point. Zero is only enforced for values for float16 points.

TECHNICAL SUPPORT AND RESOURCES

Technical Support Options

OSIsoft provides dedicated technical support internationally, 24 hours a day, 7 days a week. You can read complete information about technical support options, and access all of the following resources at the OSIsoft Technical Support Web site:

http://techsupport.osisoft.com

OSIsoft provides the following support options and resources.

Help Desk and Telephone Support

Telephone support is available 24 hours a day, 7 days a week. Please note that in some cases you may need to leave a message, and you will receive a call back within one hour.

- Call the Technical Support Center at (01) 510-297-5828
- FAX Technical Support at (01) 510-352-2349

Email Support

Email support is available 24 hours a day, 7 days a week. Send technical support inquiries, including the problem description and message logs, to: techsupport@osisoft.com. Technical Support will respond to your inquiry within 24 hours.

Personalized Online Technical Support

The Online Call Center allows you to create a support call, which will be responded to in 24 hours. It also allows you to review information from your previous support calls. Click **My Support > My Calls**. *My Support* also allows you to review *My Products, My Download History*, and *SRP Terms*, which covers Service Reliance Program (SRP) service agreements.

Knowledge Center

The Knowledge Center provides a searchable library of documentation and technical data, as well as a special collection of resources for system managers. For these options, click **Knowledge Center** in the Technical Support Web site.

- The **Search** feature allows you to search Support Solutions, Bulletins, Support Pages, Known Issues, Enhancements, and Documentation (including User Manuals, Release Notes, and White Papers).
- **System Manager Resources** include tools and instructions that help you manage: Archive sizing, Backup scripts, Daily Health Check, Daylight Saving Time configuration, PI Server security, FactoryTalk Historian System sizing and configuration, PI Trusts for Interface Nodes, and more.

Remote Server Access

Technical support engineers can remotely access your PI Server to provide diagnostics, hands-on troubleshooting, and assistance. For remote assistance, go to **Contact Us > Remote Access Options** in the Technical Support Web site.

On-site Technical Support

OSIsoft provides on-site service according to SRP service level agreements. To see current SRP status, go to **My Support > SRP Terms** in the Technical Support Web site.

Before You Call or Write for Help

When you contact OSIsoft Technical Support, please provide:

- Product name, version, and/or build numbers
- Computer platform (CPU type, operating system, and version number)
- The time that the difficulty started
- The message log(s) at that time

Find Version and Build Numbers

To find version and build numbers for each FactoryTalk Historian System subsystem (which vary depending on installed upgrades, updates or patches) use either of the following methods:

- If you have FactoryTalk Historian System Management Tools (PI SMT) installed, select **Start > Programs > FactoryTalk Historian System > FactoryTalk Historian System Management Tools**. In PI SMT, select the server name, then under *System Management Plug-Ins*, open **Operation > PI Version**. The PI Version tree lists all versions.
- If you do not have PI SMT installed, open a command prompt, change to the *pi\adm* directory, and enter `piversion -v`. To see individual version numbers for each subsystem, change to the *pi\bin* directory and type the subsystem name followed by the option `-v` (for example, `piarchss.exe -v`).

View Computer Platform Information

To view platform specifications:

- In Windows, right-click on **My Computer** and choose **Properties**. For more detailed information, select **Start > Run**, and enter `msinfo32.exe`

- In UNIX, enter `uname -a`

INDEX OF TOPICS