



***PI Server 2010
Introduction to System Management***

OSIsoft, LLC

777 Davis St., Suite 250
San Leandro, CA 94577 USA

Tel: (01) 510-297-5800

Fax: (01) 510-357-8136

Web: <http://www.osisoft.com>

OSIsoft Australia • Perth, Australia

OSIsoft Europe GmbH • Frankfurt, Germany

OSIsoft Asia Pte Ltd. • Singapore

OSIsoft Canada ULC • Montreal & Calgary, Canada

OSIsoft, LLC Representative Office • Shanghai, People's Republic of China

OSIsoft Japan KK • Tokyo, Japan

OSIsoft Mexico S. De R.L. De C.V. • Mexico City, Mexico

OSIsoft do Brasil Sistemas Ltda. • Sao Paulo, Brazil

Copyright: © 1992-2010 OSIsoft, LLC. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of OSIsoft, LLC.

OSIsoft, the OSIsoft logo and logotype, PI Analytics, PI ProcessBook, PI DataLink, ProcessPoint, Analysis Framework, IT Monitor, MCN Health Monitor, PI System, PI ActiveView, PI ACE, PI AlarmView, PI BatchView, PI Data Services, PI Manual Logger, PI ProfileView, PI WebParts, ProTRAQ, RLINK, RtAnalytics, RtBaseline, RtPortal, RtPM, RtReports and RtWebParts are all trademarks of OSIsoft, LLC. All other trademarks or trade names used herein are the property of their respective owners.

U.S. GOVERNMENT RIGHTS

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the OSIsoft, LLC license agreement and as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12.212, FAR 52.227, as applicable. OSIsoft, LLC.

Version: 3.4.385

Published: 7/12/2010

Table of Contents

Chapter 1 Introduction	1
Chapter 2 Daily Health Checks.....	3
Chapter 3 Tools for System Management.....	5
PI System Management Tools (SMT)	5
PI Tag Configurator	7
PI Module Database Builder	7
PI Interface Configuration Utility (ICU).....	8
Collective Manager.....	8
Chapter 4 Introduction to the PI System	9
About the PI System.....	9
About the PI Server	10
PI System Data Flow.....	11
Chapter 5 Manage PI Points	17
About PI Points.....	17
Find Malfunctioning Points	22
Create New Points	22
Decommission Points.....	23
Delete Points	23
Chapter 6 Manage Archives	25
About Archives	25
Configure Archives	26
Create New Archives.....	30
Register an Archive	30
Unregister an Archive.....	31
Move an Archive.....	31
Chapter 7 Manage Backups.....	33
About PI Server Backups	33
How to Monitor and Maintain Your Scheduled Backups.....	34
How to Schedule Automatic Backups	37
How to Do a Manual Backup.....	40
How to Restore a Backup to an Existing PI Server.....	40

Chapter 8 Manage Interfaces	43
About PI Interfaces.....	43
Configure Interfaces.....	44
Monitor PI Interface Performance	47
Chapter 9 Manage Security	49
About Security.....	49
Manage Authentication	49
Manage Authorization	56
Tighten Security	65
Chapter 10 Monitor PI System Performance	67
How to Monitor Performance Counters.....	67
Which Performance Counters to Monitor	70
Chapter 11 Manage Data Source Equipment	73
Add New Equipment.....	73
Remove Obsolete Equipment	73
Replace Equipment.....	74
Appendix A PI Server Health Check	75
Appendix B Glossary	77
Appendix C Technical Support and Resources	99
Training	101
Index	103

Chapter 1

Introduction

This guide provides an introduction to PI Server system management. It explains the basics on how to configure and maintain the PI Server. These include backups, archive management, security options, and connections to client applications and PI interfaces.

- *Daily Health Checks* (page 3)
- *Introduction to the PI System* (page 9)
- *Tools for System Management* (page 5)
- *Manage PI Points* (page 17)
- *Manage Archives* (page 25)
- *Manage Backups* (page 33)
- *Manage Interfaces* (page 43)
- *Manage Security* (page 49)
- *Monitor PI System Performance* (page 67)
- *Manage Data Source Equipment* (page 73)

Chapter 2

Daily Health Checks

Each day, you should:

- **Check for Malfunctioning Tags:** Use the Stale and Bad Points tool in PI SMT to determine if there are any tags that have stopped working (stale) or are in an error condition.
- **Verify Backups:** Check either the log file in the target folder or the message logs on the server. See *Do a Daily Backup Check* (page 34).
- **Verify Archive Files:** Search the Archives tool in PI SMT for archive gaps. See *Manage Archives* (page 25).
- **Check Status on Collectives.** Verify the synchronization status of secondary members in a PI collective. Use *Collective Manager* (page 8).

Note: You can complete these health-check tasks with Windows scripting and PI Server command-line utilities such as **piconfig** or **piartool**. Refer to the *PI Server System Management Guide* for more information.

Tools for System Management

OSIsoft provides tools that make it easier to manage a PI System. These tools are included with every new PI Server installation:

- **PI System Management Tools (SMT)** for performing routine PI Server administration tasks
- **PI Tag Configurator** for creating and editing tags in an Excel spreadsheet
- **PI Interface Configuration Utility (ICU)** for configuring PI interfaces
- **Collective Manager** for creating and managing PI collectives for implementing high availability (HA) in your PI Server

With PI Server 2010 and later, PI AF is part of every PI System. PI AF replaces the PI Module Database (MDB). To manage PI AF, OSIsoft provides the **PI System Explorer**. For more on PI AF and the PI System Explorer, see the *PI AF Users Guide*.

Note: For backward compatibility, MDB is still included in the PI Server installation. MDB is automatically migrated to and synchronized with the AF Server. For more information on MDB to AF migration and synchronization, see the *PI MDB to AF Transition Guide*.

Check the *OSIsoft Technical Support Web Site* (<http://techsupport.osisoft.com/>) regularly for updates to these tools.

OSIsoft also provides some powerful command-line utilities. This book discusses command-line utilities only as they are needed for specific tasks. For more complete information on command-line utilities, refer to the *PI Server Reference Guide*.

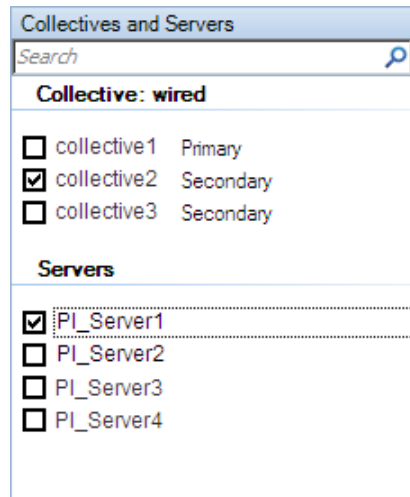
PI System Management Tools (SMT)

PI System Management Tools (SMT) is a set of easy-to-use tools that allow you to perform all the basic PI Server administration tasks. PI SMT is included in the PI Server installation, but you can get the latest version of PI SMT on the *OSIsoft Technical Support Web site* (<http://techsupport.osisoft.com>). To manage PI Server 3.4.380 and later you should use SMT version 3.3.0.4 or later. Older PI SMT versions will work but will not allow you to view or change any new features, such as security settings.

To start PI SMT from the Windows **Start** menu, choose **All Programs > PI System > PI System Management Tools**.

Select a Server in SMT

SMT includes a **Collectives and Servers** pane where you can select one or more PI Server to work with, including collective members. If you select more than one server, be careful not to perform tasks on the wrong PI Server.



Add a PI Server

If SMT does not list the server to which you want to connect, use Connection Manager to add the server to the list of available servers:

1. Choose **File > Connections**. PI Connection Manager opens. The dialog box lists the configured servers to which you can connect and the version of each server. It also shows the server you have chosen as the default server.
2. Choose **Server > Add Server**.
3. In **Network Node**, enter the network path (host name or IP address) of the PI Server. If the target server is a PI collective, enter the path to one of collective members. You can also choose a default user for the connection.
4. Click **OK**.
5. Close Connection Manager.

Managing PI Servers of Different Versions

PI Server version 3.4.380 represents a significant change in PI Server security configuration. You might need to manage PI Servers that use the old security model along with servers that use the new model. SMT 3.3.0.4 and later helps you do that seamlessly. Features that are not available on older versions of the PI Server do not appear in SMT if you are connected only to an older PI Server.

Note: If you are installing a new PI Server 3.4.380 or upgrading a PI Server to that version, refer to the manual *Configuring PI Server Security*. That guide explains the security model and your implementation options.

PI Tag Configurator

PI Tag Configurator is a PI SMT add-in for Microsoft Excel. You must add the PI Tag Configurator utility to the Excel **Add-Ins** menu before you can use it. Complete the following steps:

1. Open Excel.
2. Open the **Add-Ins** menu:
 - o In Excel 2003 and earlier: Select **Tools > Add-Ins**
 - o In Excel 2007: From the Office button click **Excel Options**, choose **Add-Ins**, then click the **Go** button
3. Click **Browse** to open the **Browse** explorer window.
4. Browse to the **\PIPC\SMT** folder.
5. Select **PITagCnf.xla**.
6. Click **OK** to return to the Add-Ins window.
7. Select **PI-TagConfigurator 32 bit**.
8. Click **OK** to exit the **Add-Ins** window.

This adds the **PI-SMT** menu to Excel.

Use the PI SMT Help file for instructions for using the **PI SMT** menu.

PI Module Database Builder

The PI Module Database Builder allows you to view and modify items from the Module Database in an Excel spreadsheet. You must add the Module Database add-in to the Excel menu before you can use it. Complete the following steps:

1. Open Excel.
2. Open the **Add-Ins** menu:
 - o In Excel 2003 and earlier: Select **Tools > Add-Ins**
 - o In Excel 2007: From the Office button select **Excel Options > Add-Ins**, then click the **Go** button
3. Click **Browse** to open the **Browse** explorer window.
4. Browse to the **\PIPC\MDBuilder** folder.
5. Select the **MDBuilder.xla** add-in.

6. Click **OK** to return to the Add-Ins window.
7. Select **Module Database Builder**.
8. Click **OK** to exit the **Add-Ins** window.

This adds the **Module Database Builder** to the **PI SMT** menu in Excel.

Use the PI SMT Help file for instructions on how the **PI SMT** menu.

PI Interface Configuration Utility (ICU)

PI Interface Configuration Utility (ICU) is a point-and-click tool for configuring interfaces. To configure a PI interface with the ICU, you must run it directly on that interface node. This means you need to install the ICU on each interface node. You can get the latest version of the ICU on the *OSIsoft Technical Support Web site* (<http://techsupport.osisoft.com>).

To run the ICU, click **Start > Programs > PI System > PI Interface Configuration Utility**. To learn how to configure a PI interface with the ICU, see *Configure Interfaces* (page 44).

Collective Manager

Use Collective Manager to create new PI collectives, configure existing collectives and their servers, and view the status of collectives. To run Collective Manager, click **Start > All Programs > PI System > Collective Manager**.

To view and edit a collective's properties, click the collective name under **Collectives**. The collective properties and a diagram of servers in the collective appear on the right side of Collective Manager.

An icon in the diagram represents each server in the collective. A green check mark on the icon indicates that the server is communicating properly. A red X indicates that the server is unavailable. A yellow warning icon indicates that the server is available but has errors.

Status and **Connection Status** show the associated errors. For further details about Collective Manager, see the Collective Manager Help files.

You can get the latest version of Collective Manager on the *OSIsoft Technical Support Web site* (<http://techsupport.osisoft.com>).

Chapter 4

Introduction to the PI System

This chapter provides some basic information about the PI System:

- *About the PI System* (page 9)
- *About the PI Server* (page 10)
- *PI System Data Flow* (page 11)

About the PI System

The PI System collects, stores, and manages data from your plant or process. You connect your data sources to one or more PI interface nodes. *Interface nodes* retrieve data from your data sources and send it to one or more PI Servers. Users on other computers can get data from the PI Server and display it with client tools (for example, PI ProcessBook, PI DataLink, and PI WebParts). The computers on which these tools run are sometimes called *client nodes*.

- **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 Performance Equations, PI ACE, and Totalizer are also considered data sources, even though they may be hosted on the PI Server computer. For more details about data sources, see *Manage Data Source Equipment* (page 73).
- **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 Interfaces* (page 43) and *Managing Interfaces* (page 43).
- **PI Server Nodes:** The PI Server stores the 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 stored in external systems, that is data that is not generated by the PI System. For more information, see *About the PI Server* (page 10).
- **PI Application Nodes:** The PI System comes with many middle-tier products that act as application servers. These include analytical products such as PI ACE, and PI Notifications, asset databases such as AF 2.0, and Web portals based on Microsoft SharePoint and SAP NetWeaver.
- **Clients Nodes:** Operators, engineers, managers and other plant personnel use a variety of client applications to connect to PI Servers and PI application servers to view plant data.

About the PI Server

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

What is in the PI Directory?

By default, the PI Server installs its files in a directory called `PI` on the disk with the most available space, but you can choose a different location during installation. Within the `PI` directory, the PI Server installs the following subdirectories:

Directory	Contains
<code>adm</code>	Administrative tools
<code>bin</code>	Subsystem or PI service executables
<code>dat</code>	Databases such as points and digital states. This is also the default directory for archives.
<code>log</code>	Log files
<code>setup</code>	Files for install and uninstall

File System Dos and Don'ts

- **Disable virus scanning on the `PI\dat` and archive folders.** Virus scanning may affect the integrity of archive or other database 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 data file.
- **Use the Windows File System Compression feature with caution.** Compressed file might slow down the PI Server's access to archive files. The compression might save disk space, but it requires more CPU resources.

Core Subsystems

The PI Server consists of several modules, including a set of core subsystems. To check whether the core subsystems are running, you can use the PI Services tool in PI System Management Tools (SMT).

Core Subsystem	What It Does
PI Archive Subsystem (piarchss.exe)	Stores and serves all time-series data after it comes out of PI Snapshot Subsystem.
PI Backup Subsystem (pibackup.exe)	Controls, performs, and maintains a history of PI Server backups.
PI Base Subsystem (pibasess.exe)	Performs user authentication and maintains most security settings. Maintains the PI Server configuration data, such as Point Database, Digital States, and Module Database.
PI License Manager (pilicmgr.exe)	Maintains and controls licensing information for the PI Server and all connected applications.
PI Message Subsystem (pimsgss.exe)	Logs and serves information and error messages for the PI Server and connected applications.
PI Network Manager (pinetmgr.exe)	Provides the connection between all subsystems in the PI Server. This subsystem also manages network connections between the server and client applications.
PI Shutdown Subsystem (pishutev.exe)	Determines when the PI Server was stopped and writes shutdown events to points configured to receive these events. This subsystem runs only at startup and then stops on non-clustered PI Servers.
PI Snapshot Subsystem (pisnapss.exe)	Stores the most recent event for each point. This subsystem applies compression, sends data to the event queue, and serves snapshot events and updates to client applications.
PI SQL Subsystem (pisqlss.exe)	Processes SQL query statements, including those submitted by the PI ODBC driver.
PI Update Manager (piupdmgr.exe)	Sends notifications of changes in time-series values or configuration data to any client application that is signed up for notification.

In addition to the core subsystems, the PI Services tool in SMT lists the status of optional subsystems, such as PI Batch Subsystem and PI Performance Equation Scheduler. These optional subsystems do not need to be running in order for the PI Server to be running.

PI System Data Flow

This section provides a broad overview of the data flow in the PI System:

1. **Interface sends data to the PI Server.** PI interfaces collect data from the data sources and pass them on to the server in the form of PI events. An event consists of a *timestamp*, a *value*, and a *status*. Before sending an event to the PI Server, the interface evaluates the event; it sends significant events on to the PI Server, and discards the others. This is called exception reporting (*What is Exception Reporting?* (page 13)). Interfaces complete the following steps:
 - a. Get data and create a PI event.

- a. Perform exception testing to determine if event is significant.
- b. Send significant events to the PI Server and discards the rest.
- c. If buffering is configured, save the PI events on disk in case the PI Server is unavailable.

Note: The buffering service may also perform compression testing and fans data to all members of a PI collective. This is called *n-way buffering*.

2. **PI Snapshot Subsystem receives all data.** On the PI Server, all new PI events are handled first by PI Snapshot Subsystem (*What is the Snapshot?* (page 14)). PI Snapshot Subsystem holds a single value for each PI point. If data comes in time order, incoming events become new *snapshot values*.

When a new value comes in, PI Snapshot Subsystem evaluates the old value by performing *compression testing* (*What is Compression Testing?* (page 14)). Based on the results of the compression testing, PI Snapshot Subsystem either discards the old value or sends it to the event queue. PI Snapshot Subsystem completes the following steps:

- a. Get a new event and make this event the new snapshot value.
- b. If the new event is out-of-order, send it directly to the even queue without performing any compression testing. Otherwise, perform compression testing between the old and new events.
- c. Send significant events to the event queue and discard the rest.

3. **PI Archive Subsystem reads data from the event queue.** PI Archive Subsystem continuously reads data in the event queue and stores it a memory cache, and eventually into archive files. However, when PI Archive Subsystem is shut down or unable to receive data, the event queue holds the data, serving as a high-speed buffer (*What is the Event Queue?* (page 15)). When available, PI Archive Subsystem resumes data processing. PI Archive Subsystem completes the following steps:

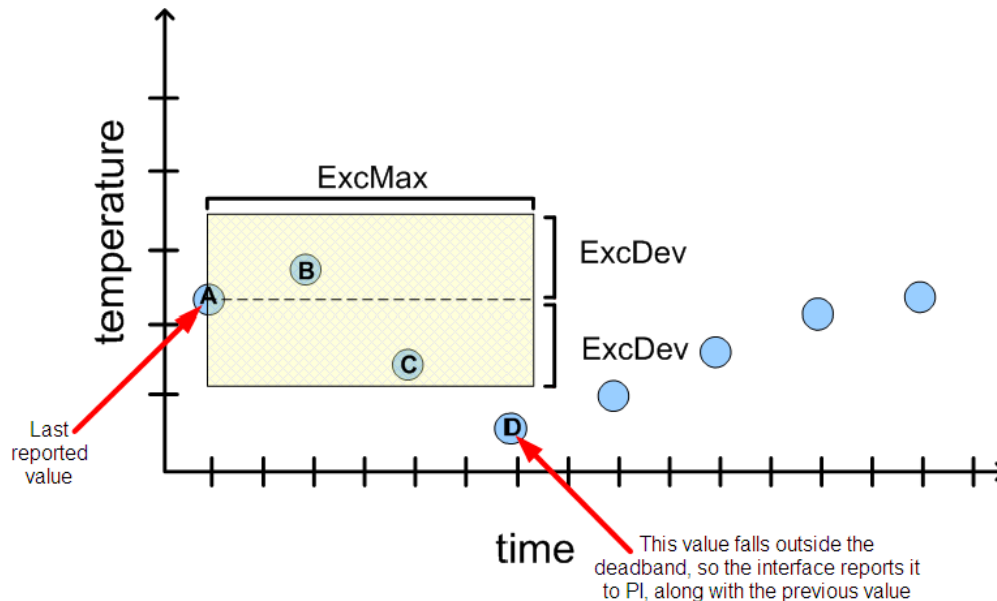
- a. Read events out of the event queue.
- b. Validate, sort, and store events in the write cache.
- c. Periodically move data from the write cache into archive files.

4. **Users access the data through client applications.** The ultimate goal of a PI System is to distribute information across the enterprise or organization to those who will use it to make key decisions. To achieve this goal, OSIsoft produces a number of different *client tools*, all of which display PI Server data in one form or another.

What is Exception Reporting?

Exception reporting ensures that a PI interface only sends meaningful data, rather than taxing the network connection by sending unnecessary data.

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, the interface reports values A, D, and C to 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 dead band, so the interface reports value D along with the previous value—in this case, value C. Compression testing will determine if C needs to be preserved, or if A and D are sufficient to recreate the original signal with the required fidelity (*What is Compression Testing?* (page 14)).

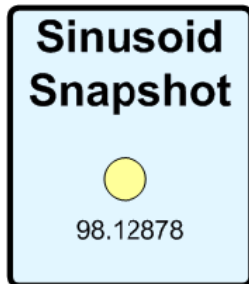
The interface uses the point's `ExcDev`, `ExcMin`, and `ExcMax` attributes to decide whether to report the new value to PI Server:

- `ExcDev` (or `ExcDevPercent`) determines 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.
- `ExcMax` sets a limit on how long the interface can go without reporting a value to PI Server. After the `ExcMax` time period, the interface sends the next new value to PI Server, regardless of whether the new value is different from the last reported value.
- `ExcMin` 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 PI Server, then you would set the `ExcMin` attribute to 600 seconds.

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

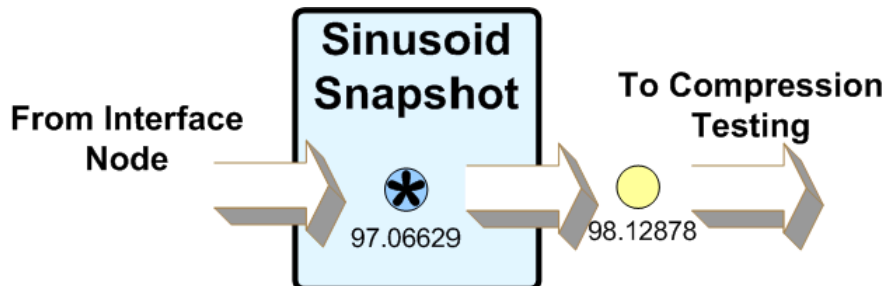
What is the Snapshot?

PI 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 old value to the event queue or discards it.

Note: Sometimes an event enters PI Snapshot Subsystem with a timestamp that is older than the current snapshot value. This is called an out-of-order event. PI Server sends out-of-order events directly to the event queue for archiving, *without* compression testing.



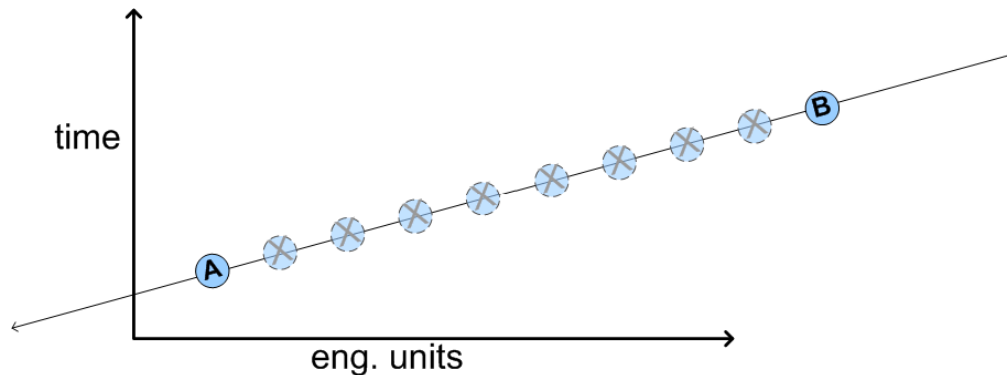
These values in PI Snapshot Subsystem are called *snapshot events* or sometimes, just *snapshots*. The collection of all the snapshot values for all the points is the Snapshot Table.

What is Compression Testing?

PI Snapshot Subsystem uses compression testing to determine what events need to be saved in the archive. Compression testing ensures that you store just enough data to accurately reproduce the original signal.

Note: PI Buffer Subsystem also uses compression testing.

For example, in the following illustration all the events fall on the same straight line. In a simple case like this, you do not 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.



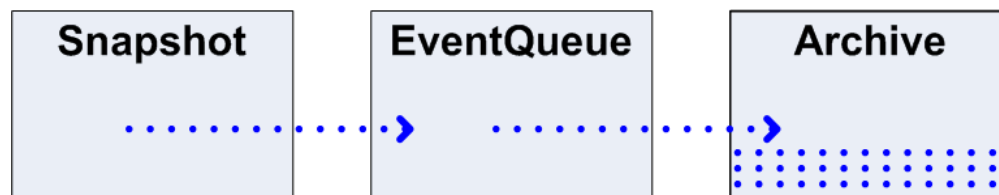
The same principle applies to compressing real-world data. PI Server uses a sophisticated compression algorithm to determine which events it needs to keep in order to provide an accurate data history. Use the **CompDev**, **CompMin**, and **CompMax** attributes to control the precision of the compression algorithm.

- **CompMin** and **CompMax** control how often PI Server saves a new value for a particular point. (This is similar to the **ExcMin** and **ExcMax** attributes in exception reporting.)
- **CompDev** or **CompDevPercent** determine how much a point's value needs to change for PI Server to save it.

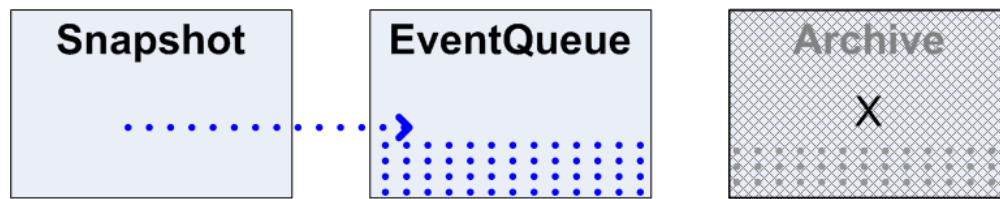
For details on setting compression testing attributes, see *Compression Specifications* (page 20).

What is the Event Queue?

The event queue serves as a memory and disk buffer between the snapshot and archive subsystems. PI Snapshot Subsystem adds data to the queue and PI Archive Subsystem removes data from the queue.



Normally the event queue passes events to the archive as quickly as they arrive, but in some circumstances PI Archive Subsystem might be busy or unavailable. When this happens, the event queue stores the data, filling until the archive is again available. This is called archive queuing.



Archive queuing commonly occurs when:

- Interface nodes or other data sources produce a burst of events. This might occur when an interface recovers data after a network outage.
- PI Archive Subsystem becomes busy storing out-of-order events, or processing very expensive user queries.
- The operating system or hardware supporting the PI Server has a reduced I/O or CPU capacity. Other applications that run on the PI Server machine or that share the same hardware host might cause this situation.

Chapter 5

Manage PI Points

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

- *About PI Points* (page 17)
- *Find Malfunctioning Points* (page 22)
- *Create New Points* (page 22)
- *Decommission Points* (page 23)
- *Delete Points* (page 23)

For complete information on PI points, see the *PI Server Reference Guide*.

About PI Points

Points, sometimes called *tags*, are the basic building blocks of a PI System because you use them to track the events that comprise your data history. When system managers or OSIsoft field services engineers install a PI Server, they create a PI point for every source of data that the PI System must track. PI Base Subsystem stores points and their attributes in the *point database*.

About Point Attributes

Use point attributes to configure how and when the PI System should collect data from a particular data source. Point attributes specify the data source location, how often the PI System should get new values from the data source, which values PI can ignore, which values represent valid data, and much more.

For complete details on point attributes, see the *PI Server System Management Guide*. This section gives you a brief overview of a few key attributes.

You can specify more than 50 different point attributes for each 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 can view and/or edit the point, and so on. The exact list of attributes that configures a point depends on the *class of the point* (page 18).

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

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:
`* ' ? ; { } [] | \ ` ' "`

Class of Point: PtClass Name Attribute

The attributes that you need to configure for a particular point depend on what the point is for. The PI System 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. For details, see the *PI Server Reference Guide*.

Points that represent data from a PI interface are always in the Classic point class. The default PI point classes are:

- **Alarm:** Used for alarms. See the *PI Server Applications Guide* for more information on Alarm points.
- **Base:** 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.
- **Classic:** Includes attributes used by interfaces.
- **SQC_Alarm:** Used for SQC alarm points. See the *PI Server Applications Guide* for more information on SQC alarm points.
- **Totalizer:** Used 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 PI SMT Help topic for the Totalizers tool.

Data Type of Point: PointType Attribute

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

Point Type	Use For
Digital	Points with values limited to one of several discrete states, such as ON/OFF or Red/Green/Yellow.
Int16	Points with integer values between 0 and 32767 (15-bit unsigned integers).
Int32	Points with integer values between -2147450880 and 2147483647 (32-bit signed integers).
Float16	Floating point values, scaled. The accuracy is one part in 32767.
Float32	Single-precision floating-point values (IEEE floating points).
Float64	Double-precision floating-point values (IEEE floating points).

Point Type	Use For
String	Strings of up to 976 characters.
Blob (Binary Large Object)	Any type of binary data up to 976 bytes.
Timestamp	Any time/date in the range 1-Jan-1970 to 1-Jan-2038 Universal Time (UTC).

Data Source: Point Source Attribute

Each interface has an associated point-source string, consisting of one or more characters (*Point Source* (page 45)). When you define a point, set the **Point Source** attribute to the match the point-source string for the interface. This identifies the interface as the data source for this point. 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.

Note: The SMT Point Source Table tool lists the currently used point sources.

Interface ID Number: Location1 Attribute

The **Location1** attribute applies only to interface points—that is, points that get their data from a PI interface rather than some other source. Most interfaces use the **Location1** attribute to specify the interface ID number.

Setting Scan Class: Location4 Attribute

The **Location4** attribute only applies to interface points—that is, points that get their data from a PI interface rather than some other source. Most PI interfaces have one or more *scan classes* (page 46) for scheduling data collection. You typically set the **Location4** attribute for a point to specify which of the interface scan classes you want to use.

Note: Most interfaces require you to use the **Location4** attribute to set the scan class. There are exceptions, however, particularly among older interfaces. Also, some interfaces get data on command, and do not require scan classes. Always check the interface documentation.

Exception Specifications

Exception reporting specifications determine which events the interface sends to PI Server and which it discards. For more details, see *What is Exception Reporting?* (page 13).

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

Specification	Attribute	Description
Exception Deviation	ExcDev	Specifies how much a point value must change before the interface reports the new value to PI Server. Use ExcDev to specify the exception deviation in the point's engineering units. Typically, you set ExcDev slightly smaller than the precision of the instrument system.
	ExcDevPercent	Use ExcDevPercent instead of ExcDev to specify the exception deviation as a percentage of the Span attribute. However, if you set the Span attribute incorrectly, your exception reporting will be wrong too. Typically, you set ExcDevPercent to about 1% of Span .
Exception Minimum	ExcMin	Minimum time (in seconds) between events that the interface reports to PI Server. For example, if you set ExcMin to five, then the interface discards any values collected within five seconds of the last reported value. Typically, you set ExcMin to zero.
Exception Maximum	ExcMax	Maximum time (in seconds) between events that the interface reports to PI Server. After this time elapses, the interface reports the next event to PI Server without applying the exception deviation test.

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

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

Compression Specifications

PI Server uses the compression specifications to filter the data passed from the snapshot to the archive. Compression ensures that you store just enough data to accurately reproduce the original signal. By filtering out unneeded data, you get more efficient archive storage and the archive can serve the data to the clients more efficiently.

While exception reporting filters data with a simple dead-band method, compression testing filters data with a more complex method that follows the slope of the data (the swinging-door compression algorithm). For details on the compression testing algorithm, see the *PI Server System Management 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	Description
Compression Flag	Compressing	Turns compression on (when set to 1) or off (when set to 0).
Compression deviation	CompDev	Specifies the compression deviation in the point's engineering units. Typically, you 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.
	CompDevPercent	Specifies the compression deviation as a percent of the point's Span attribute.
Compression minimum time	CompMin	Sets minimum time between events in the archive. Set CompMin to zero for any point coming from an interface that does exception reporting. Typically, you use CompMin to prevent an extremely noisy point from using a large amount of archive space.
Compression maximum	CompMax	Sets maximum time between events in the archive. If the time since the last recorded event is greater than or equal to CompMax , then PI Server automatically stores the next value in the archive, regardless of the CompDev setting. Typically, you set CompMax to the same value for all points in the system. You might set CompMax to equal one work shift (such as 8 hours).

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

Point Value Range: Zero, Span and Typical Value

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

Attribute	Description
Zero	Point's lowest possible value. Logically, you might set to the same value as the instrument's zero. Required for all numeric data type points; critical for float16 points.
Span	Difference between the top of the range and the bottom of the range. Required for all numeric data type points.
Typical Value	Documents an example of a reasonable value for this point. For a numeric tag, value must be greater than or equal to the Zero value, and less than or equal to the Zero value plus the Span value.

Configure Shutdown Events: Shutdown

The **Shutdown** attribute has two possible values: 1 (On) and 0 (Off). If the PI Server shuts down, it writes a shutdown event to all points that have **Shutdown** set to 1 (On). Set

Shutdown to `OFF` for points on buffered interfaces. The buffering service restores the data for these points once it connects to the PI Server again.

Point Security

How to set point security depends on what version of the PI Server you are using. See *Understanding Access Permissions: Old and New* (page 57) for a description of the old and new security models. See *PI Point Access Permissions* (page 61) for instructions on setting point security.

Find Malfunctioning Points

Use the Stale and Bad Points tool in PI SMT to search for stale and/or bad points. This tool identifies points that have not received data for a long time (stale) or that have current values representing error or bad conditions, such as I/O timeout, PtCreated, bad input or, in many cases, Shutdown.

When you find bad or stale points, research the reason for the condition. Some possible scenarios are:

- No network connection between the PI Server and the interface.
- Someone has shut down the interface computer, or the interface computer has lost connection with the device.
- Someone has changed the point attributes.

If point values are stale or bad for no known reason, you should immediately determine the cause.

When you find points that are no longer useful, such as points that represent data from obsolete equipment, decommission them.

Create New Points

As a PI system manager, you might need to create a new PI point. The easiest way to create a new point is to copy an existing point that is very similar to the point you want to create. Then edit the **Tag** attribute and any other attributes that you want to change.

Point Builder in PI SMT provides an easy way to edit and create a small number of PI points. If you have Microsoft Excel installed, you might find the Excel add-in, PI Tag Configurator, a better tool to use. OSIsoft recommends PI Tag Configurator when creating a large number of points.

Note: To create new points, you need read/write access to the PIPOINT table. See *Where to Set Access Permissions* (page 59)..

Decommission Points

To decommission a point:

1. Start PI SMT and select the PI Server for that point.
2. Under **System Management Tools**, choose **Points > Point Builder**.
3. Search for the point.
4. Click the **Archive** tab.
5. Under **Scan**, choose **Off**.

To decommission many points at once, use PI Tag Configurator:

1. Start PI SMT and select the PI Server for that point.
2. Click **Tools > Tag Configurator**. A Microsoft Excel spreadsheet opens.
3. In Microsoft Excel, click **PI-SMT > Tag Attributes** to open the **Tag Attributes** dialog box. (In Microsoft Excel 2007, you can find the PI-SMT menu on the **Add-Ins** tab.) In **Point Class**, select **classic**. Under **Attributes**, select the **Scan** check box. Click **OK**.
4. Click **PI-SMT > Tag Search**, search for and select the points, and then click **OK** to close the **Tag Search** dialog box. You see the tag names on the spreadsheet.
5. Click **PI-SMT > Import Tags**. On the **Import PI Tags** dialog box, select **Import tags and attributes on spreadsheet**, click **OK**. The **Tag Configurator** dialog box display the number of points. Click **OK**. The spreadsheet shows the scan attributes for the tags.
6. In the **scan** column on the spreadsheet, change **1** to **0** for all the points.
7. Click **PI-SMT > Export Tags**. In **Mode**, select **Edit** and click **OK**. The **Tag Configurator** dialog box displays the number of points changed. Click **OK**.

Note: Some interfaces do not use the scan bit to turn off points. If you want to decommission a point for such an interface, change the point's **Point Source** attribute to a value that you use only for decommissioned points.

Delete Points

When you delete a point, you lose all data for that point and will break any client displays that use the point. Furthermore, once you delete a point, you cannot get it back. If you are unsure about the purpose of a point or about the need for any historical data associated with the point, consider *decommissioning the point* (page 23) rather than deleting the point.

Note: To delete points, you need read/write access to the PIPOINT table. See *Where to Set Access Permissions* (page 59).

Chapter 6

Manage Archives

PI Server stores data in archive files. See *About Archives* (page 25) to learn more about how this works. You can do all or most of your archive management with the PI SMT Archives tool. The basic tasks for managing archives are:

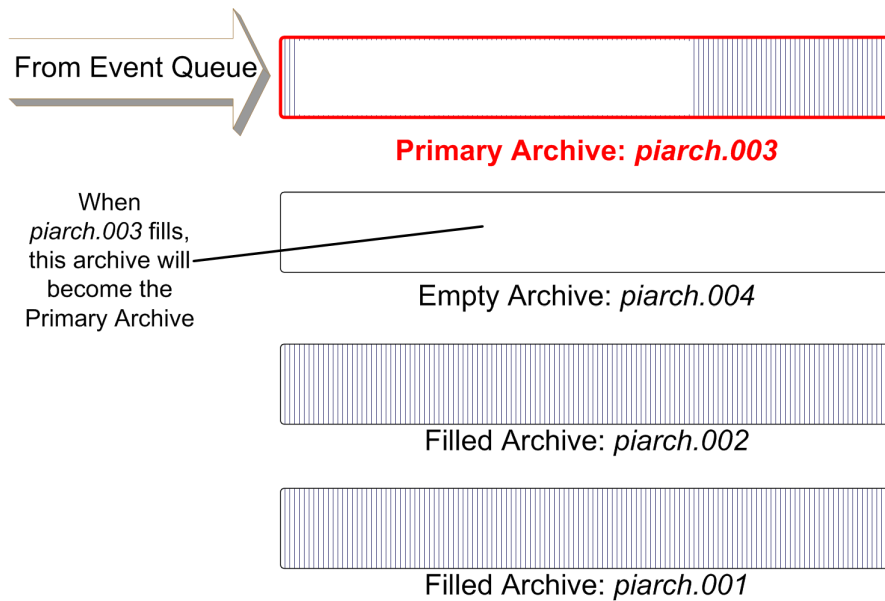
- *Configure Archives* (page 26)
- *Create New Archives* (page 30)
- *Registering an Archive* (page 30)
- *Unregistering an Archive* (page 31)
- *Move an Archive* (page 31)

About Archives

Typically, archives are files of a fixed size that can hold PI data. Fixed archives allocate the full amount of space in advance, meaning that an empty archive and a full archive require 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 Server performs the archive shift before the primary archive is completely full so that you can add older data later, if necessary.



Archive files eligible to be the new primary archive must be *registered* (page 30), writable, shiftable, and large enough to handle the current size of the PI point database.

If no eligible empty archive files are available for an archive shift, PI Server uses the oldest available filled archive file as the new primary archive, and overwrites the data in the old archive file. For example, in the preceding illustration, after the shift from `piarch.003` to `piarch.004`, no empty registered archive files remain. If you do not create a new archive file, then `piarch.001` becomes the next primary archive.

It takes PI Server a few minutes to complete an archive shift. During this time, you cannot add, edit, or delete points. PI Server stores incoming data in the event queue until the shift completes, and then writes the queued events into the new primary archive.

Note: If PI Archive Subsystem loses its connection to the primary archive, the subsystem does not try to reconnect to the archive. Data will flow into the event queue, but will not be stored in the archive file. You must either stop and restart the subsystem or unregister and re-register the archive file in order to reconnect and store data in the archive.

Configure Archives

When you configure archives, pay special attention to two goals:

- **Prevent data loss.** Data moves from PI Snapshot Subsystem to the event queue to the archive file. If no archive file is available, the event queue holds new data until an archive file becomes available. To prevent data loss, locate the event queue on a different drive than the archive files. With this configuration, if the archives become unavailable due to a disk failure, the event queue safeguards new data.
- **Prevent archive overwrites.** PI Server overwrites the oldest available full archive file when:

- o There is not enough space on the drive where the archives are located. To prevent, locate the archive files on a drive with sufficient space (*Locate the Archive Files* (page 27)) and monitor that drive.
- o There is no empty, writable, shifttable, archive available when the archive shift occurs. To prevent, set up PI Server to *create archive files automatically* (page 28). If you do not want PI Server to create new archives automatically, determine how often your archives fill and create new archives as needed so that PI Server does not run out of space and start overwriting data (*Create New Archives* (page 30)).

Prevent Archive Data Loss

After receiving data from an interface, PI Snapshot Subsystem places the data into the event queue. PI Archive Subsystem retrieves this data almost instantly. With this waypoint for data, PI Server continues data collection if the archives or disk should become unavailable.

To take advantage of this data waypoint, *do not place the event queue on the same physical disk* as the archives. Check the location of the event queue on your server and make sure it is on a different disk from the primary archive.

To move the event queue:

1. Open **PI SMT**.
2. Under **System Management Tools**, select **Operation > Tuning Parameters**, and select the **Snapshot** tab.
3. Double-click the **Snapshot_EventQueuePath** parameter. The parameter dialog box opens.
4. Set **Value** to the desired path for the event queue.

Note: To edit tuning parameters, you need read/write access to the PITUNING table. See *Where to Set Access Permissions* (page 59).

5. Restart PI Snapshot Subsystem.

This moves the event queue file location. You can delete the old event queue file.

Locate the Archive Files

By default, the installation places the archives in the `PI\data` directory, but you can put them anywhere you like. The SMT Archives tool lists the location of each registered archive file. To open the Archives tool:

1. Start **PI SMT** and select the PI Server on which you want to view the archives.
2. Under **System Management Tools**, click **Operation > Archives**.
3. The Archives tool opens.

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.

Warning: On systems collecting production data, do not use anti-virus software to scan the directories containing PI Server database and archive files. For details, see *File System Dos and Don'ts* (page 10).

Set Up Automatic Archive Creation

You can configure the server to automatically create new archive files if you run PI Server 3.4 or later. If you do this, then you must monitor available disk space for the archives. If there is not enough disk space to create the new archives, the PI Server overwrites data in the old archives.

To automate archive file creation:

1. If you are running a PI Server of version earlier than 3.4.380, you must perform some preliminary steps (*Preliminary Steps* (page 28)). If you are running 3.4.380 or later, skip this step.
2. Start **PI SMT**.
3. Select the server on which you want to automatically create archives.
4. Under **System Management Tools**, select **Operation > Tuning Parameters**.
5. Click the **Archive** tab and double-click the **Archive_AutoArchiveFileRoot** parameter.
6. In **Value**, enter the path to the directory where you want to create automatic archives, along with the archive file prefix.

By specifying a value for this parameter, you turn on auto-archiving. If necessary, PI Archive Subsystem creates the directory.

By default, PI Archive Subsystem names the automatically-generated files with the specified prefix and the date the archive shifts. For example, if you specify `D:\PI\arc\auto`, then a file name might be: `auto_2008-10-30_16-28-33.arc`. You can change the file name format (*Change File Name Format* (page 29)).

7. Click **OK**.

Preliminary Steps

On PI Server version 3.4.380 and later, these steps are not required. For earlier versions of PI Server, perform these steps before you set up automatic archive creation:

1. Verify that you have a fixed-size primary archive.

Note: Automatic archive generation will not work if you have a dynamic primary archive. The automatically-generated archives will be the same size as the primary archive.

2. Verify that you have a valid, shiftable target archive available. This gives you a backup in case the file creation fails. This target archive can be dynamic or fixed.

Change File Name Format

By default, automatically-generated files are named with the specified prefix and the date the archive shifts. By default, the format is:

```
root_YYYY-MM-DD_HH-MM-SS.ext
```

where *root* is the prefix defined in the **Archive_AutoArchiveFileRoot** parameter and *.ext* is the file extension defined in the **Archive_AutoArchiveFileExt** parameter. For example, if the root is *auto_* and the extension is *.arc* then a file name might be: *auto_2008-10-30_16-28-33.arc*.

To change the *root* prefix:

1. Start PI SMT and select the server.
2. Select **Operation > Tuning Parameters**.
3. Click the **Archive** tab and select the **Archive_AutoArchiveFileFormat** parameter.
4. In **Value**, enter the value corresponding to the format you want.

Value	Format	Example (assume <i>root</i> prefix is <i>auto</i> and ext is <i>arc</i>)
1	root_YYYY-MM-DD_HH-MM-SS.ext	auto_2008-10-30_16-28-33.arc
0	root_D_Mon_YYYY_H_M_S.ext	auto_30_10_2008_16_28_33.arc
2	root_UTCSECONDS.ext	auto_1221064113.arc

5. Click **OK**.

To change the *.ext* file name extension:

1. Start **PI SMT**.
2. Select the server on which you want to automatically create archives.
3. Under **System Management Tools**, select **Operation > Tuning Parameters**.
4. Click the **Archive** tab and select the **Archive_AutoArchiveFileExt** parameter.
5. In **Value**, enter the extension you want.

Change File Name Extension


By default, automatically-generated archives have the file extension *.arc*, but this is configurable.

To change the archive file-name extension:

1. Start **PI SMT**.
2. Select the server on which you want to automatically create archives.
3. Under **System Management Tools**, select **Operation > Tuning Parameters**.
4. Click the **Archive** tab and double-click the **Archive_AutoArchiveFileExt** parameter.
5. In **Value**, enter the file extension you want to use.
6. Click **OK**.

Create New Archives

The SMT Archives tool provides an easy way to create, edit, and monitor your PI archives. To create a new PI archive:

1. Run PI SMT and select the server on which you want to view the archives.
2. Under **System Management Tools**, select **Operation > Archives**.
The Archives tool lists all the archives registered on the selected server. The primary archive is first on the list.
3. To create a new archive, click the **Create a New Archive** button .
4. In the **Create New Archive** dialog box that opens, enter the new archive file name and select **Clone the primary archive fixed size**.


Note: Archives that can grow in size to accept variable amounts of data are called dynamic archives. For details, see the *PI Server System Management Guide*.

5. If you want to choose a different size for the archive, be sure you understand the issues in archive sizing. For details on managing archives, see the *PI Server System Management Guide*.
6. Click **OK**. The Archives tool creates and registers the archive.

Register an Archive

If you want PI Server to recognize a file as an archive file, you must register the file. By registering an archive file, you tell PI Server that the file exists and is available for PI Archive Subsystem to use for storing data. PI Server cannot access data in unregistered archives, nor can the PI client applications.

To register an archive:

1. Run PI SMT and select the server on which you want to view the archives.
2. Under **System Management Tools**, select **Operation > Archives**.
The Archives tool lists all the archives registered on the selected server. Unregistered archive files do not appear in the list.
3. To register an archive, then click the **Register an Archive** button .
4. In the dialog box that opens, double-click the archive file you want to register.
The archive file appears in the list of registered archive files.


Unregister an Archive

To move or reprocess an archive file, you must unregister the archive, make your changes, and then re-register the file. You cannot unregister a primary archive.

To unregister an archive:

1. Run PI SMT and select the server on which you want to view the archives.
2. Under **System Management Tools**, select **Operation > Archives**.

The Archives tool lists all the archives registered on the selected server. Unregistered archive files do not appear in the list.

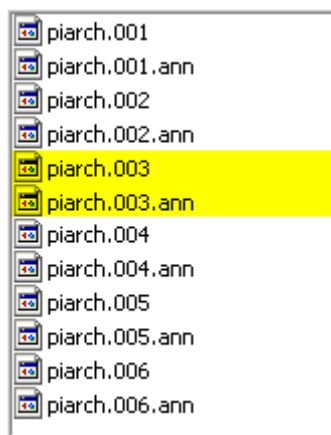
3. Select the archive file you want to unregister, and click the **Unregister Selected Archive** button .

The Archives tool removes the archive from the list.

Move an Archive

To move an archive file, *unregister it* (page 31), move it to the new location, and then *register it again* (page 30).

When you move an archive file, be sure to move the associated annotation file as well. The annotation file has the same name as the archive file, appended by `.ann`. For example, if the archive is named `piarch.003`, then the associated annotation file is `piarch.003.ann`.



Manage Backups

To avoid losing data and configuration information, you must back up your PI Server and its archives daily. The basic tasks for managing backups are:

- *Monitor and Maintain Your Scheduled Backups* (page 34)
- *Set Up Automatic Backups* (page 37)
- *Do a Manual Backup* (page 40)
- *Restore a Server from a Backup* (page 40)

See *About PI Server Backups* (page 33) for information on the types of backups and the accessibility of the PI Server during backups.

About PI Server Backups

PI Server does not run regular backups unless you specifically schedule a backup task. If you do not have an automatic backup task scheduled on the PI Server machine, your first priority should be to set one up (see *How to Schedule Automatic Backups* (page 37) for instructions). Note the following points concerning the automatic PI Server backup:

- By default, the archives are backed up incrementally—that is, only files that have been modified since the last backup date are backed up.
- If supported, the PI Backup scheduled task performs a VSS (Volume Shadow Copy Service) backup. VSS provides a backup infrastructure that allows the PI Server to perform seamless online backups.
- You can continue to use PI Server while you are backing it up. However, there are some limitations for non-VSS backups (see *VSS and Non-VSS Backups* (page 33)).

VSS and Non-VSS Backups

Depending on your operating system, PI Server uses one of two different backup methods: VSS or non-VSS. If VSS is supported on the PI Server computer, then your PI Server backup task is a VSS backup. Otherwise, it is a non-VSS backup. Non-VSS backups have some restrictions on writing data to PI Server while the backup is taking place.

- *VSS Backups*. PI Server's preferred online backup mechanism uses Volume Shadow Copy Services (VSS), which is included in Microsoft XP and Microsoft Server 2003 and later. With VSS, PI Server works as usual during backups. You can create points, push data to the archives, and so on.

- *Non-VSS Backups.* If your operating system does not support VSS, then PI Server still provides online backup functionality by doing non-VSS backups. Non-VSS backups are online backups—you do not need to take the server or archives offline. However, non-VSS backups have the following limitations:
 - Can only read files while they are backed up
 - Cannot create new points during the backup
 - Cannot push new data into the archives during the backup

You can check to see which type of backups are scheduled.

If possible, run PI Server on an operating system that has VSS support. This is especially important for batch data because batch data does not pass through the event queue. In this case, you are relying entirely on buffering at your interfaces.

How to Monitor and Maintain Your Scheduled Backups

Do a Daily Backup Check

Every day, check the backup history in the PI SMT Backups tool (select **Operation > Backups**) to ensure that the last backup completed successfully.

Periodically, you should:

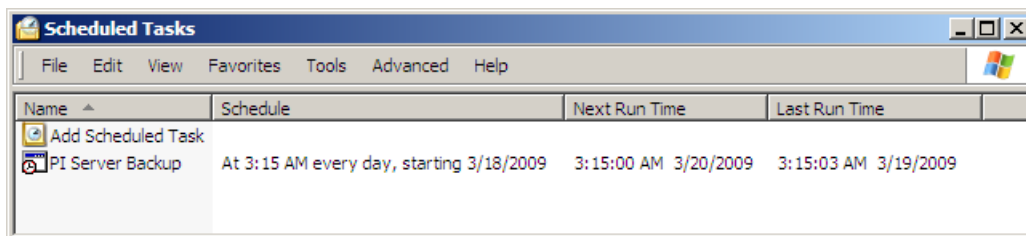
- Check that the last backup was itself successfully backed up to a storage medium separate from PI Server.
- Check the backup logs in the PI Server backup directory. These logs have a name of the form `pibackup_DD-MMM-YY_hh:mm:ss`. Check the backup log to make sure the site-specific backup completed successfully.
- Check that you are not running out of space on the disk where PI Server creates the backups.
- Check that the backup files are in the proper location, and that the file size looks right. These files should be about the same size each time (substantially smaller backup files indicate that the backup might not have successfully completed).
- Restore your backups to a test machine to make sure that the backups are completing successfully (see *How to Restore the Server from a Backup* (page 40)).

Check Whether Backups are Scheduled

In the Windows Control Panel, open **Scheduled Tasks**. If your automatic PI Server backups are scheduled as a Windows scheduled task, then the **PI Server Backup** task appears in the list of tasks, along with information about the backup schedule:

- Time the backups occur each day
- Time the last backup was run

- Time next backup is scheduled to run



Note: If you are using third-party software to automate PI Server backups, then the PI Server Backup task might not appear in the Scheduled Tasks window.

Monitor Backup History in the SMT Backups Tool

Use the SMT Backups tool to monitor your PI Server's backup history. You can also create on-demand backups in SMT by clicking the **Backup Now** button. However, only use on-demand backups with SMT for troubleshooting purposes. They are not a substitute for regularly scheduled backups.

To check the PI Server backup history, use the SMT Backups tool:

1. Open PI SMT.
2. Under **Collectives and Servers**, select the server you want to check.
3. Under **System Management Tools**, select **Operation > Backups**.
4. In the **PI Server** drop-down list, select the server you want to examine. The list includes all of the servers selected under **Collectives and Servers**.

The backup history for that server appears.

5. Right-click a column heading to see a complete list of columns you can display.

Index	Start Time	Status
1	3/19/2009 3:15:09 AM	[0] Success
2	3/20/2009 3:15:05 AM	[0] Success
3	3/21/2009 3:15:05 AM	[0] Success
4	3/22/2009 3:15:05 AM	[0] Success
5	3/23/2009 3:15:05 AM	[0] Success
6	3/23/2009 5:23:15 PM	[0] Success
7	3/24/2009 3:15:05 AM	[0] Success
8	3/24/2009 11:16:17 AM	[0] Success
9	3/24/2009 4:23:15 PM	[0] Success
10	3/25/2009 3:15:06 AM	[0] Success
11	3/25/2009 9:52:40 AM	[0] Success
12	3/26/2009 3:15:05 AM	[0] Success
13	3/26/2009 4:25:29 PM	[0] Success
14	3/26/2009 5:00:32 PM	[0] Success
15	3/26/2009 5:01:00 PM	[-16915] Back
16	3/27/2009 3:15:05 AM	[0] Success
17	3/27/2009 11:55:53 AM	[0] Success
18	3/27/2009 3:53:49 PM	[0] Success
19	3/27/2009 3:54:26 PM	[0] Success
20	3/27/2009 3:54:55 PM	[0] Success
21	3/28/2009 3:15:05 AM	[0] Success
22	3/29/2009 3:15:05 AM	[0] Success
23	3/30/2009 3:15:05 AM	[0] Success

Index	Start Time	Status	Files Copied	Size (MB)	Duration	File Copy Failures	Total Files	Type	VSS	Component Mode	Third Party	Initialization Duration (s)
52					52		257					
53					53		257					
54					54		257					

6. Double-click a backup entry to see details about that particular backup. You can view a backup summary or the entire list of backed up files.

By default, you can view reports for the last 100 PI Server backups. These reports only tell you whether or not the backup of PI Server itself was successful. The reports do not tell you whether or not your `pisitebackup.bat` script ran successfully or whether or not a third-party backup of the backup directory was successful.

The history tells you the type of backups completed. If the **Type** column does not appear, right-click the column header and select **Type**. The following backup types are possible.

Backup Type	Description
INCREMENTAL	Any new backup task that is installed will perform incremental backups.
NUMARCH/CUTOFF	If you have upgraded your PI Server from 3.4.375 and you have not re-installed your backup task, you will see back types of NUMARCH/CUTOFF. Backups of this type use as selection criteria either a particular hard-coded number of archives or an archive cutoff date. All modified archives are not guaranteed to be included in the backup.
COPY	Collective Manager and the PI SMT Backups tool do COPY backups. A COPY backup does not update the last backup time for archive files. Backups with the SMT Backups tool are not a substitute for regularly scheduled backups.
FULL	If you have upgraded from PI Server 3.4.370 and you are still using NtBackup.exe to backup your PI Server, you will see FULL backups reported. However, you should consider backups with NtBackup.exe to be of type NUMARCH/CUTOFF.
DIFFERENTIAL	This backup type will typically not appear in the list.

Check the Message Logs

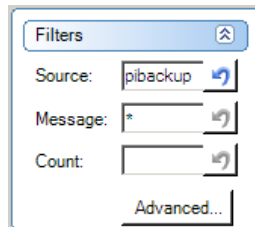
Use PI SMT to search the message logs for messages from PI Backup Subsystem. If you think there might be a problem with your automated backups, or with PI Backup Subsystem, this is a good place to look. Follow these steps:

1. Open PI SMT.
2. Under **Collectives and Servers**, select your PI Server.
3. Under **System Management Tools**, select **Operation > Message Logs**.

You can use the Message Logs tool to examine all the messages that PI Server produces.

4. Under **Time**, select the time period that concerns you.
5. Under **Filters**, type the following in **Source**:

pibackup



6. Click the **Retrieve Messages** button  on the toolbar.

The log messages appear. Check the log messages for errors. You can select a message to see more details.

How to Schedule Automatic Backups

PI Server installation does not create a scheduled backup task. You need to set up a PI Server backup to run as an automatic task in Windows. This section explains how to set up automatic PI Server backups:

1. *Schedule the automatic backup* (page 38).
2. *Schedule backups of the files in the pipc directory* (page 39).
3. Use third-party software to automatically copy the contents of your backup files to a safe location.

Once you have successfully created a scheduled backup task, you can *change the time of the backup* (page 38).

Schedule the Automatic Backup

To set up an automatic PI Server backup, follow these steps:

1. On the PI Server computer, log into a Windows account that has administrator privileges.
2. Open a Windows command window.
3. Change to the `PI\adm` directory. For example, if PI Server is installed on the D drive, type:

```
cd /d %piserver%adm
```

4. Use the **pibackup** command to install a backup task to the desired target directory.

For example, suppose you want to back up to the `e:\pibackup` directory. (Ideally, the `e:` drive does not correspond to the system drive or the drive that stores your archives.)

Type:

```
pibackup e:\pibackup -install
```

This sets up the PI Server backup as a Windows scheduled task. A task called *PI Server Backup* now appears in Scheduled Tasks (you can open Scheduled Tasks from Control Panel).

Note: On Windows 2000 Server, the task name will be of the form **Atn**, where *n* is the next available task number when the task was created. If you have installed the scheduled task on Windows 2000, rename the scheduled task to **PI Server Backup** by right-clicking the task name and choosing **Rename**.

5. Next, devise a method to backup the `e:\pibackup` directory (see *Copy the Backup Files to a Safe Location* (page 38)). This is a crucial step because it ensures that you have a PI Server backup that is stored somewhere separate from the PI Server itself.
6. If desired, change the default backup time (3:15 AM daily), by editing the PI Server Backup scheduled task (see *Change Scheduled Time of Automated Backup* (page 38)).

Copy the Backup Files to a Safe Location

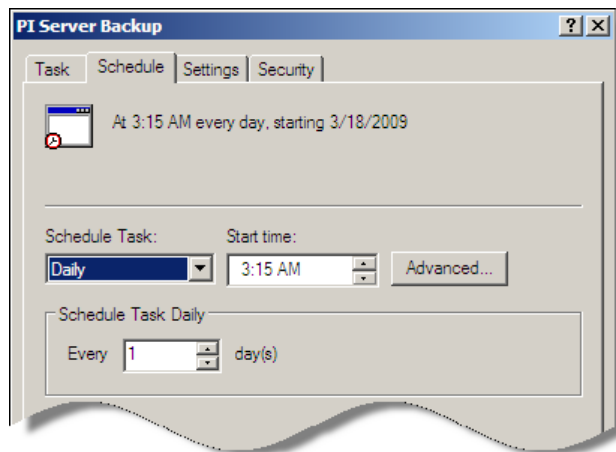
The PI Server Backup task creates a backup directory on PI Server. Your backup task is not truly complete until you copy these files to a medium that is not local to PI Server. OSIsoft recommends that you use a 3rd-party backup solution for this purpose.

If 3rd-party backup software is not available, you can use the `pisitebackup.bat` file to automatically copy the backup directory to a remote computer. The `pisitebackup.bat.example` file contains instructions for setting this up.

Change Scheduled Time of Automated Backup

To change the scheduled time of the automatic backup:

1. On the PI Server computer, log on to a Windows account that has administrator privileges.
2. In Control Panel, open **Scheduled Tasks**.
3. Double-click the **PI Server Backup** task.
A configuration dialog box opens.
4. Click the **Schedule** tab.
5. Set the scheduled time for the backup.



6. Click **OK**.

Schedule PIPC Backups

The scheduled backup task backs up the archive files and the files in the `PI` directory. It does not back up the files in the `pipc` directory. The `pipc` directory contains site-specific configuration files, PI interface startup command files, log files, and so on.

To automate the backup of these files:

1. On the PI Server computer, log on to a Windows account that has administrator privileges.
2. Open a Windows command window.
3. Change to the `PI\adm` directory:

```
cd %piserver%adm
```
4. Find the file called `pisitebackup.bat.example`.
5. Rename this file to `pisitebackup.bat` (remove the `.example` extension).

This sets up a backup of a default set of files in the `pipc` directory. This backup runs immediately after the automated PI Server Backup task executes. The backup files are placed in a subdirectory called `sitebackup` within the PI Server backup directory.


6. By default, `pisitebackup.bat` backs up the `.bat`, `.log`, `.ini`, `.txt`, and `.sql` files from the `pic` directory. To change which files are backed up, edit the `pisitebackup.bat` file.

Note: You can edit `pisitebackup.bat` to include any scripts you want to run immediately after the scheduled PI Server Backup task executes. For more information, read the instructions in the `pisitebackup.bat.example` file.

How to Do a Manual Backup

Use the PI SMT Backups tool for manual backups.

To do a manual backup:

1. Open PI SMT.
2. Under **Collectives and Servers**, select the server you want to back up.
3. Under **System Management Tools**, select **Operation > Backups**.
4. In the **PI Server** drop-down list, select the server you want to back up. The list includes all of the servers selected under **Collectives and Servers**.
5. Click the **Backup Now** button  to open the server's **Backup** dialog box.
6. In **Backup Location**, type the path for the backup location or browse to the appropriate folder.
7. To restrict the backup to specific components, select the components under **PI Server Components**.
8. Click **Backup**.

The backup runs. Backups executed with these steps do not change the last backup time. Your regularly scheduled incremental backup will not be affected.

How to Restore a Backup to an Existing PI Server

This section explains how to restore your PI Server from a backup. Follow these instructions to restore the PI Server to the same computer that it was running on:

1. Isolate your PI Server from the network.
2. Stop PI Server.
`\pi\adm\pisrvrstop.bat`
3. Delete the following file:
`\pi\dat\PIModuleUnitDb.dat`

This file is automatically regenerated when you restore from backup.

4. Restore the backup to a temporary directory, such as `C:\TempRestoreDir`.

For example, if you back up your backup directory with a third-party backup application, restore the desired backup to `C:\TempRestoreDir`. Alternatively, if you are restoring the latest backup, you can restore PI Server directly from the latest backup directory. This procedure assumes that you have restored the desired backup to a folder of the name `C:\TempRestoreDir`.

5. Copy the files from `C:\TempRestoreDir\dat` to `PI\dat`.
6. Copy the files from `C:\TempRestoreDir\adm` to `PI\adm`.
7. Copy the files from `C:\TempRestoreDir\bin` to `PI\bin`.
8. Copy the files from `C:\TempRestoreDir\log` to `PI\log`.
9. Copy the archive files from the `C:\TempRestoreDir\arc` directory in your backup folder to their original location on PI Server.

If you are not sure where your archive files were located on PI Server, look in the backup log file in `C:\TempRestoreDir\`. The log contains the archive list at the time of the backup.

Since you are restoring to an existing server, you do not have to restore all archives. At a minimum you must restore the primary archive. Restore other archives as needed.

10. If a site backup was performed (if, for example, `C:\TempRestoreDir\sitebackup` exists), then copy the files from the site backup directories to the corresponding 32-bit and 64-bit `pipc` directories.
11. Restart PI Server.
12. Restore PI Server's connection to the network.
13. Use the MDB to AF Synchronization tool in PI SMT to check the status of the synchronization between MDB and AF.
14. If MDB and AF are out of sync, then use the MDB to AF Synchronization tool to reset MDB.

Manage Interfaces

After you install and configure a PI interface, you can typically leave it running indefinitely without any intervention. If you perform software upgrades or security patches, or if the network infrastructure changes, you might need to perform a few basic tasks.

- *Configure Trusts* (page 52): Interfaces connect to PI Server through trusts, typically based in part on network information for the interface node. When you install an interface, you need to configure a trust for it. If your network infrastructure changes (change in host name or IP address for example) then you need to reconfigure the trust.
- *Configure Interfaces* (page 44): When you install an interface, you need to register it with the ICU. If the interface node does not have buffering configured, you should set that up as well. See the *PI Buffering User Guide* for instructions.
- *Monitor Interface Performance* (page 47): If you suspect a problem with an interface, you might want to monitor its performance.

About PI Interfaces

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

PI interfaces typically run on a dedicated system, called an interface node, which connects both to the data sources and to PI Server. 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 (PI to PI). Interface nodes can run multiple interfaces and connect to multiple PI Servers.

Note: Interface nodes are sometimes referred to as API nodes or data source nodes.

Since data sources run without human intervention, user logins are not applicable. To allow PI interfaces to connect to PI Server, you typically configure a *PI trust*. This is sometimes called a *trust login*. For more about PI trusts, see *Manage Interface Authentication with PI Trusts* (page 52).

Start and Stop Interfaces

The first time you start a PI interface, start it from the Windows Services control panel. After that, you can start and stop the interface from *PI ICU* (page 44).

To start and stop the interface directly on the interface node:

1. Click **Start > Administrative Tools > Services**.
2. In the Services window, find the interface that you want to start or stop.

In the Services window, PI Interface services are listed with the **PI-** prefix. For example, the buffering service is listed as **PI-Buffer Server**.

3. Right-click the interface service and choose **Start** or **Stop**.

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/>):

- Interface documentation — Each PI interface has its own document that describes how to configure that interface.
- *PI Buffering User Guide* — Contains comprehensive information about using and configuring buffering.
- *UniInt Interface User Manual* — Found in the `PIPC\Interfaces\UniInt` directory after interface installation, this manual describes Universal Interface (UniInt), on which many interfaces are based.
- *PI Server System Management Guide* — Includes information on managing interfaces.


The *Technical Support Web Site* (<http://techsupport.osisoft.com/>) also provides a product page dedicated to PI interfaces and a PI System Manager Resources page. In addition to information on new releases, you can get information on PI ICU and other tools.

Configure Interfaces

Use PI Interface Configuration Utility (ICU) to manage your PI interfaces. You must run PI ICU directly on the computer where the PI interface runs.

Configure the interface initially in PI ICU to register the interface with PI ICU.

To register an interface with PI ICU:

1. On the PI ICU toolbar, click the **Create a New Interface Instance from EXE** button  to open the **Configure a New Interface** dialog box.
2. Specify the interface executable.
 - a. Click **Browse**.

- b. Navigate to the directory containing the interface executable. By default, PI installs your interface executables in an interface subdirectory under the `Program Files\PIPC\Interfaces` directory.
 - c. Select the executable file.
 - d. Click **Open**.
3. In **Host PI Server/Collective**, select the host PI Server or collective that contains the points for this interface.
If the desired server or collective does not appear, use PI Connection Manager to connect to that server or PI collective.
 4. Enter a descriptive name for the interface (optional).
 5. In **Point Source**, enter the unique string that identifies the interface in configured points (see *Point Source* (page 45)).
 6. In **Interface ID**, enter the number that associates points with a copy of the interface (see *Interface ID* (page 46)).
 7. In **Service ID**, enter a number that identifies this instance of the interface.
Multiple instances of an interface can use the same executable (.exe) without copying and renaming the interface executable file.

Note: Service ID is not related to Interface ID. However, you can use the same value for Service ID that you use for Interface ID.

8. Click **Add**.

Point Source

When you configure an interface, you specify a point source, a unique string that identifies the interface. All the points that belong to that interface must use that point-source string as the value for the **Point Source** attribute. Some interfaces can use multi-character values for the point source and some require a single character. Check the interface documentation if you are not sure.

PI Server comes preconfigured with applications that use reserved point-source characters. When you choosing a point-source string for your interfaces, do not use any of the following reserved strings:

String	Reserved For
9	RampSoak Simulator
@	Alarm
G	
C	Performance Equations scheduler
Lab	Default point source string
R	Random Interface Simulator
T	Totalizer program

The PI SMT Point Source Table tool lists all the currently used **Point Source** values for the selected PI Server. In PI SMT, select **Points > Point Source Table** to see this tool.

Interface ID

Interface ID is a number that associates a point with a particular copy of an interface. Set Interface ID 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 before setting Interface ID.

Scan Class

A *scan class* is a code that PI interfaces use to schedule data collection. Scan classes consist of the following components:

Component	Description	Optional	Example
Period (Scan Frequency)	Specifies how often the interface collects data.	No	01:00:00 Get data every hour
Offset	Specifies a start time for the calculation. PI interprets the value starting from midnight of the current day.	Yes	01:00:00,13:00:00 Get data every hour, starting at 1:00PM
UTC Time	Requires that the scheduling is synchronized with UTC. To use it, add ", U" to the end of the scan class. UTC scan classes are not affected by daylight saving time because they force the scan class scheduling to synchronize with UTC, rather than local time. Using UTC time has no effect when the period of the scan class is less than or equal to 1 hour.	Yes, but recommended	01:00:00,13:00:00,U Get data every hour, starting at 1:00PM UTC time
Local Time	Means that the period of the scan class will be 24 hours during daylight saving time to standard time transitions and 22 hour during standard time to daylight saving time transitions. To use it, add ", L" to the end of the scan class. Using Local time has no effect when the period of the scan class is less than or equal to 1 hour.	Yes, using will force Wall Clock Scheduling	23:00:00,08:00:00,L Get data every 24 hours during daylight saving time to standard time transitions and 22 hours during standard time to daylight saving time transitions.

Here is an example scan class:

hh:mm:ss, hh:mm:ss, U

period offset UTC

The scan class in the example above uses the hh:mm:ss format and specifies a period, an offset, and UTC time.

When you specify a scan class in the ICU, you can use any of the following formats:

```
ss
ss,ss
hh:mm:ss
hh:mm:ss,hh:mm:ss
hh:mm:ss,hh:mm:ss,t
```

where `hh` is hours, `ss` is seconds, `mm` is minutes, and `t` can be either a `U` (for UTC time) or an `L` (for Local Time). Offset and UTC/Local Time are optional.

Note: In PI ICU, scan class specifications cannot contain any spaces.

Monitor PI Interface Performance

To monitor the performance of each interface, you can:

- *Check IO-Rate and Performance Points* (page 47)
- *Check Log Files* (page 47)

Check IO-Rate and Performance Points

Create a PI ProcessBook display that shows the IO-rate and performance points for each interface. Some interfaces do not have performance points, so for these interfaces, you rely on IO-rate points alone.

Point Type	What it Does
IORates	Monitors the flow of data from an interface. Every 10 minutes each IO-rate point registers the 10-minute average data transfer rate to PI Server in events/second.
Performance	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.

Note: You can create both IO-rate and performance points for an interface using *PI ICU* (page 44).

Check Log Files

The PI System logs interface and buffering errors in the `pipc.log` and the `pigetmsg.log` on the interface node. Use PI SMT's Message Logs tool to see these messages. Some interfaces also produce an interface output file that might contain information about interface performance.

Most interfaces also write a performance summary every 8 hours to `pipc.log`. For each scan class, the summary lists:

- Duration of the most recent scan
- Percent of scans missed
- Percent of scans skipped

Interfaces count a scan as missed if it was started after its scheduled start time because a previous scan took too long. Interfaces count 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.

Use performance points to tune scan classes. If a scan takes too long, the next scan might 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 your interface user manual.

Manage Security

This chapter explains the basic management tasks for a PI Server security configuration. It does not explain how to plan and implement a new security configuration. If you want to configure security on a new PI Server, or if you are upgrading an older PI Server to version 3.4.380 or later, then refer to *Configuring PI Server Security* guide.

- *About Security* (page 49)
- *Manage Authentication* (page 49)
- *Manage Authorization* (page 56)
- *Tighten Security* (page 65)

About Security

PI Server security configuration has two main components:

- *Authentication* (page 49): Which users and applications are allowed to connect to PI Server?
- *Authorization* (page 56): What access permissions do authenticated users (or applications) have on PI Server?

As system manager, you should consider taking additional steps to improve security on PI Server. The most basic of these steps are explained in *Tighten Security* (page 65).

Manage Authentication

There are three methods of authentication available on PI Server. They are listed here in order from most secure to least secure (not recommended). The recommended configuration is to use PI trusts for interface connections and Windows authentication for users.

1. **Windows User Accounts and Passwords.** With this method of authentication, users log onto their Windows users accounts and are automatically authenticated on PI Server. They do not need to enter a separate PI Server account name and password. This is the most secure authentication method available on PI Server. Use Windows authentication for users connecting to PI Server.

Note: The Windows authentication method is available only on PI Server 3.4.380 or later. It also requires PI SDK version 1.3.6 or later.

2. **PI Trusts.** PI trusts allow applications to access PI Server without typing in a user name and password of any kind. Use trusts to authenticate PI interfaces, which run unattended. Do not use PI trusts for client applications that support Windows authentication (almost all client applications).
3. **PI Server User Accounts and Passwords.** With this method of authentication, users can log onto PI Server by typing in a PI user account name and password. This is called an *explicit login*. Explicit logins are the least secure way to authenticate on PI Server. Configure your PI Server for Windows authentication and then disable explicit logins.

Manage User Authentication through Windows

To configure PI Server for Windows authentication you use *PI identities* and *PI mappings*. You create a PI mapping between the PI identity and a Windows group (this might be an Active Directory or local Windows security group). Authenticated Windows users are automatically authenticated on PI Server through the mapping.

Note: PI identities are ideal for mappings because they do not imply management of users directly on PI Server. However, you can also use PI groups and users to create mappings.

Basic management tasks for Windows authentication are:

- *Maintain Mappings between Windows and the PI Server* (page 50)
- *Manage PI Identities* (page 52)
- On upgraded systems, you might have legacy users and groups that you need to use in mappings (*Manage PI Server User Accounts and Groups* (page 56))

If you need fundamental instructions on planning and implementing this type of configuration, refer to *Configuring PI Server Security* guide.

Manage Mappings

A mapping creates an association between an entity on Windows (such as an AD group) with an entity on PI Server (for example, a PI identity). You can create, delete and edit mappings using the PI SMT Mappings & Trusts tool. This section explains:

- *The Windows Side of the Mapping* (page 50)
- *The PI Server Side of the Mapping* (page 51)
- *About the Mappings & Trusts Tool* (page 51)
- *How to Create a Mapping* (page 51)

The Windows Side of the Mapping

Your PI mappings can use any of the following things on Windows:

- Any Active Directory (AD) Principal (user or group)
- Any local Windows user or group

If you use local Windows security, then the Windows user accounts on PI Server must exactly match the Windows user accounts on *each* client workstation. The account names, and also the passwords must be identical on the server and all client machines. When a password changes for a Windows user account, you must make that change on PI Server and all participating client machines.

Local Windows authentication is not as secure as AD authentication. If at all possible, use Active Directory (AD) rather than local Windows security. (Local Windows authentication is still far more secure than individual PI user accounts.)

The PI Server Side of the Mapping

On the PI Server side of the mapping, you can use a PI identity, a PI user, or a PI group. It is best to use a PI identity, for consistency. PI users and PI groups imply direct management of users on PI Server. If you are no longer managing users on PI Server, this could cause confusion for other PI Server administrators. If you use PI identities only for mappings or trusts and PI users/groups only for PI account management, then your security configuration is very easy for other administrators to interpret.

Note: If you need to use a PI user account in a mapping, then disable explicit logins for that account. PI user passwords are not especially secure.




About the Mappings & Trusts Tool

The Mappings & Trusts tool in PI SMT allows you to configure mappings on your PI Servers. To use it, choose **Security > Mappings & Trusts**. The tool has two tabs: one for trusts and one for mappings. The **Mappings** tab lists all the PI mappings for all selected PI Servers.

The **Mappings** tab does not appear unless you are running PI SMT version 3.3.0.4 or later *and* you are connected to at least one PI Server version 3.4.380 or later. Earlier versions of PI Server do not support mappings.

How to Create a Mapping

To create a mapping in PI SMT:

1. Under **Collectives and Servers**, select the server.
2. Under **System Management Tools**, select **Security > Mappings & Trusts**.
3. Select the **Mappings** tab.
4. In the toolbar, click the **New** button  to open the **Add New Mapping** dialog box.
5. In **Windows Account**, enter an AD principal or a local Windows group or user. To select the account either:
 - o Click the browse button  to browse for the account.
 - o Type in the account name. If you choose to type in the account name, click the resolve SID button  to verify that this is a valid account. If the account is valid, an SID appears in the field. Otherwise, a dialog box with an error message opens.

6. In **Description**, enter a description of the mapping (optional). There are no restrictions on the contents of this field.
7. In **PI Server**, choose the PI Server where you want to create the mapping. This drop-down list contains all the PI Servers that are selected under **Collectives and Servers** and that are version 3.4.380. Earlier versions of PI Server do not support mappings.
8. In **PI Identity**, enter a PI identity, group, or user.

To choose, click the browse button and open the **Select PI Identity, PI Group, or PI User** dialog box. Make a choice in **Type** to filter the choices. In the list, choose either a PI identity, a PI group, or a PI user, and then click **OK**.

OSIsoft recommends choosing a PI identity. PI users and PI groups represent an older security model that involves managing individual user accounts on PI Server. PI user accounts and passwords are not as secure as Windows accounts.

9. Click **OK** to create the mapping.

Manage PI Identities

PI identities are the link between Windows authentication and PI Server authorization (access permissions). Each PI identity represents a level of access to resources on PI Server. For example, one PI identity might be allowed to create points, while another PI identity could be allowed to read point data but not create new points.

When a mapping exists between a PI identity and a Windows group, all the users in the Windows group are automatically authenticated on PI Server and granted the access permissions defined for that PI identity.

You manage PI identities through the PI SMT **Identities, Users, and Groups** tool. This tool is located under **Security** in PI SMT. By default, the identities, users, and groups for all selected PI Servers appear in separate tabs. See the PI SMT online Help for instructions on using the tool.

Note: If you are not connected to one or more PI Servers of version 3.4.380 or later, then the **Identities** tab does not appear.

Manage Interface Authentication with PI Trusts

Most interfaces do not support Windows authentication. To authenticate interfaces, use PI trusts. Trust authentication works by comparing the connection credentials of a connecting application to records in the trust database. The connection credentials might include:

- Name of the connecting application
- IP address and netmask of the interface node
- Fully qualified host name of the interface node (such as apollo.osisoft.com)
- Short host name of the interface node (such as apollo)

As the system manager, you must update PI trusts when the interface node changes host name or IP address. Use the PI SMT Mappings & Trusts tool to view and manage your PI trusts.


Each PI trust is defined against a single PI identity, PI group, or PI user. When an interface successfully authenticates through a trust, it gets the access permissions defined for the associated identity, group, or user.

Note: You can define a PI trust against an identity or group only in PI Server version 3.4.380 or later. In earlier versions of the server, you can use only a PI user to define a trust.

For details on creating tighter security, see the section on PI security in the *PI Server System Management Guide*.

How to Create a Trust

To create a new PI trust in PI SMT:

1. Under **Collectives and Servers**, select the server.
2. Under **System Management Tools**, select **Security > Mappings & Trusts**. The Mappings & Trusts tool appears.
3. Select the **Trusts** tab.
4. Click the **New** button  to open the Add Trust Wizard.
5. Select the PI Server name and type in a name for the trust (and, optionally, a description). Click **Next**.
6. Select the type of trust to add:
 - o **PI-API application** (this is the right choice for most PI interfaces)
 - o **PI-SDK application on a Windows NT based OS**
7. Click **Next**. The next screens allow you to define optional information for the PI trust. If you leave a field blank, then that information is not checked for the trust authentication. When you fill in fields, then only applications with matching information can authenticate against this PI trust.
 - o **Application Name:** This is slightly different for API and SDK connections.
 - API: Connecting PI API applications send an identifier called an application process name, or procname. This is a four-character string with an E appended (for example, the procname for the Perfmon interface is: **PIPeE**).
 - SDK: This is the full name of the connecting application, including the extension, but not the path (for example: **PI-ICU.exe**).
 - o **Network Path:** Fully-qualified domain name of the interface node (for example **my_laptop.my_company.com**).
 - o **IP Address:** The IP address of the interface node.
 - o **Net Mask:** The net mask for the interface node (for example, **255.255.255.255**).
 - o For SDK connections only, you also have the following optional fields:
 - **Windows Domain:** the Windows domain of the user who runs the application (for example: **osi**).

- **Windows Account:** the Windows user name of the user who runs the application (for example: **my_account**).

Select the PI identity that you want to use for this trust. Applications authenticated through this trust will have all the access permissions granted to this PI identity. Alternatively, you can select a PI group or a PI user for this step.

Connection Types

When you configure a PI trust, you need to know the type of connection the trust will be used for. There are two different PI Server connection types. Each PI interface is configured to use one of these mechanisms (there are also a few interfaces that use both.) The two mechanisms are:

- **PI API Connection:** Most PI interfaces use the PI API to connect to PI Server. PI API does not support Windows authentication. PI trusts are the standard way to authenticate PI API connections.
- **PI SDK Connection:** Most client applications use PI SDK to connect to PI Server. PI SDK versions 1.3.6 and later support Windows authentication, so use Windows authentication for these connections if possible.

If you are not sure which connection type an interface uses, then consult the documentation for that interface.

The Application Name

A PI trust can require a specific application name. When you specify an application name in a trust, you have to use the appropriate format for the connection type:

- Applications that connect through the API send an identifier called an application process name, or procname. This is a four-character string with an E appended. For example, the procname for the Perfmon interface is: **PIPeE**

Note: PI API versions before 1.6.0 always send a five-character string: 4 characters plus a capital **E**. For PI API versions 1.6.0 and later, the name can be up to 8 characters, without a trailing capital **E**.

- For applications that connect through the SDK, use the full name of the connecting application, including the extension, but not the path. For example, the application name for PI ICU is: **PI-ICU.exe**

If you are running the same PI interface on another PI Server, then you can use PI SMT to determine the correct application name. Select **Operation > Network Manager Statistics**. Find the interface in the list. The application name is listed in the **Name** field.

IP Information

A PI trust can specify IP information about the computer running the PI interface or client application for which you are defining the trust. To collect this information, you can run `pidiag -host` on the computer where the interface or client application runs. This returns the connection credentials as retrieved from the local operating system.

Note: Using `pidiag -host` is helpful, but it is not a *guarantee* of getting the right information, depending on many factors, including the type of interface, the version of the SDK (if SDK-based), and whether there are firewalls / NAT devices in between the interface computer and the PI Server computer. If you have difficulty configuring the trust, contact OS/soft Technical Support.

- **Network Path.** The fully-qualified domain name. For PI API, this should match what the PI Server thinks based on a reverse-name lookup using the interface's IP address. For PI SDK (1.3.6.x and later), this should match what the client thinks, based on the Windows configuration (you can use `pidiag -host` on the client to see this information). For example, `my_laptop.my_company.com`
 - **IP Address.**
 - **Netmask.** If you specify an IP address, you must also explicitly provide a netmask value. Failure to do so will generate an error. If you require an exact match on an IP address, specify the netmask as **255.255.255.255**. If you specify a class C subnet, specify the netmask as **255.255.255.0** and the fourth field of the IP address as **0**.
-

Note: When applications run on machines with multiple network cards, you cannot predict which credentials the application will send to the PI Server for the trust authorization. OS/soft thus recommends that you either avoid such configurations, or create a PI trust for every IP address on the machine where the application runs.

Windows Account Information

For SDK connections only, you can specify Windows account information as part of the PI trust. This type of trust is not needed in the new security model because a PI mapping serves the same purpose as a trust based on OS user name and Windows domain membership.

- **Windows Domain:** the Windows domain of the user who runs the application.
- **Windows Account:** the Windows user name of the user who runs the application.

Default Trusts

PI Server includes default trusts that guarantee access to all applications running on the local machine. These default trusts are automatically recreated every time the system starts, to ensure that they are always configured in their default state. Different versions of PI Server have different default trusts. The following list includes default trusts for PI Server 3.4.375 and later.

Trust Name	Description	PI Server Version
!Proxy_127!	Allows access for local PI applications.	All versions
!PIServerName_FQDN!	Used for communication between primary and secondary servers in a PI collective.	Not needed for PI Server versions 3.4.380 and later.

PI Server versions 3.4.380 and later do not need the FQDN trust and it is not included in new installations. However, that trust is not removed if PI Server is upgraded from an earlier version.

Manage PI Server User Accounts and Groups

PI Server has its own user identification and password security, which you manage through PI user accounts. PI user accounts are not as secure as PI trusts or Windows authentication. Use the more secure methods of authentication where possible.

You can assign users to one or more PI groups; each PI user has the access permissions defined for that user, as well as the access permissions defined for all groups to which the user belongs.

PI users and PI groups are essentially legacy components. You can use PI users and PI groups in trusts and in mappings. However, when you do that, you might create some confusion about the role of the PI user or the PI group on the server. Do you use these components to manage actual PI user accounts? Do you use them only for mappings and trusts? Or do you use them for both? If you are creating a new component to use in a mapping or a trust, then use a PI identity to avoid confusion.

You manage users, and groups through the PI SMT **Identities, Users, & Groups** tool. This tool is located under **Security** in PI SMT. By default, the identities, users, and groups for all selected PI Servers appear in separate tabs. See the PI SMT online Help for instructions on using the tool.

Note: If you are not connected to one or more PI Servers of version 3.4.380 or later, then the **Identities** tab does not appear.

Manage Authorization

The authorization part of the security configuration consists of the access permissions defined for resources on PI Server. You can control access to a wide variety of PI Server resources, including points, modules, archive configuration, backups, batches, audit trails, and so on. The following sections explain what you need to know about access permissions:

- *Understanding Access Permissions: Old and New* (page 57)
- *Where to Set Access Permissions* (page 59)
- *How to Set Access Permissions* (page 63)

Understanding Access Permissions: Old and New

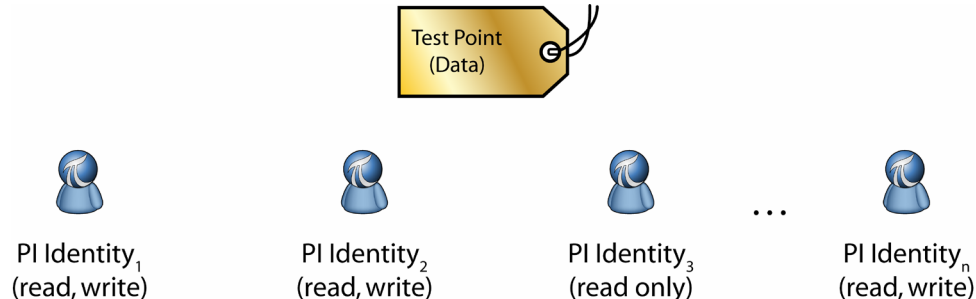
PI Server version 3.4.380 represents a fundamental change in the access permissions model. This means that how you can set access permissions is dependent on what version of PI Server you are running:

- *For Servers Earlier than 3.4.380* (page 58)
- *For Servers 3.4.380 and Later* (page 57)

The possible levels of access are the same in both models (*PI Access Levels* (page 59)).

For Servers 3.4.380 and Later

On PI Server 3.4.380 and later, resources can define access permissions for an unlimited number of PI identities, as the following illustration shows.



Although the illustration shows PI identities, you can also define access permissions for PI users and PI groups. There is no limitation on the number of users or groups. There is no longer any concept of a resource *owner* or *group* as there was in previous versions of PI Server.

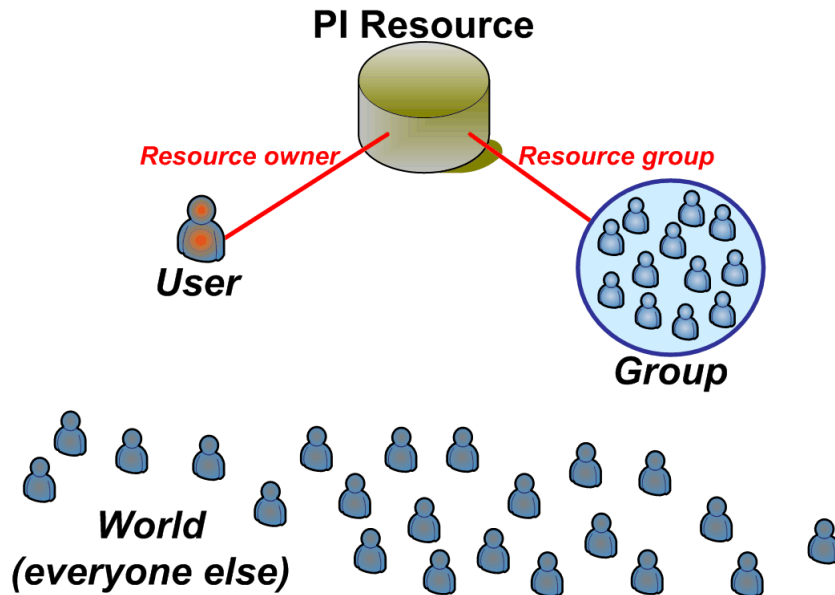
PI Server stores the settings for each object in an access control list (ACL). Each secure object on PI Server has an ACL that defines access permissions for that object. The ACL lists each identity for which access permissions are set on that object. The ACL for the TEST_POINT data in the illustration above would look like this:

```
Identity1:A(r,w) | Identity2:A(r,w) | Identity3:A(r) |
IdentityN:A(r,w)
```

Access permissions for each PI identity are separated by a pipe (|) symbol. Each entry consists of the PI identity name, then a colon (:), followed by the *access specifier*. The access specifier is defined in the format: A(r,w). The A in this notation stands for *Allow* and "r,w" indicates the allowed access rights – read and write, in this example.

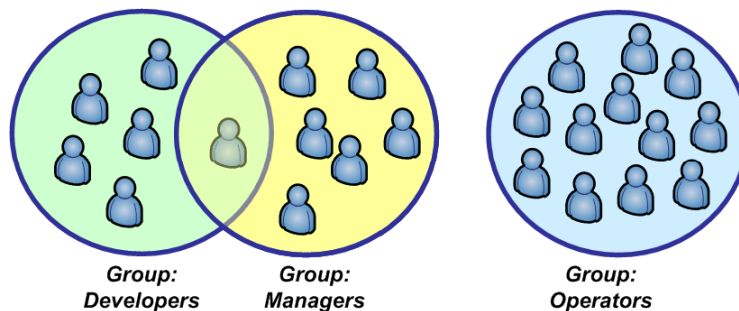
For Servers Earlier than 3.4.380

Older PI Servers use the owner/group/world model for access permissions. Each object can have one *owner*, which must be a PI user and one *group*, which must be a PI group. You can set the access for the owner, the access for the group, and the access for everyone else (called *world* access).

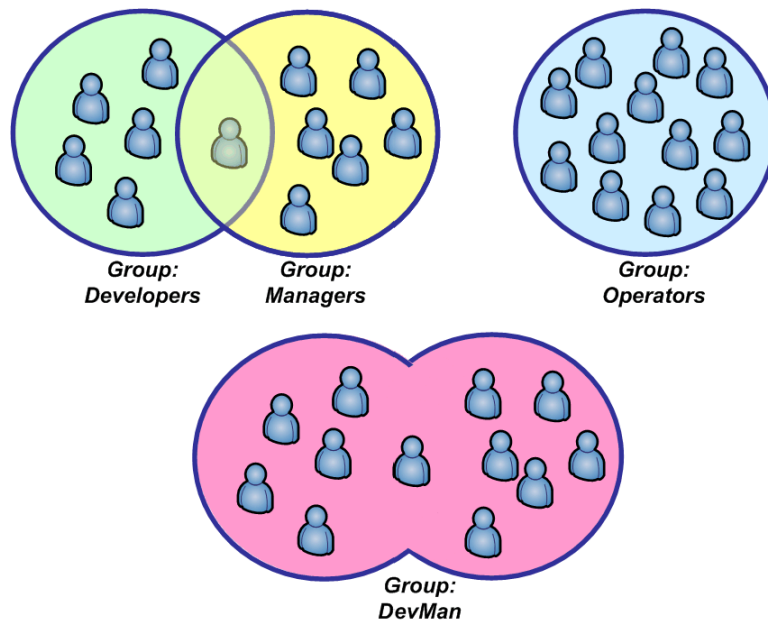


Each resource has one (and only one) associated group. When a user is not the owner of a particular PI resource (such as a point or database), PI Server checks to see if the user is a member of the group that is associated with that resource. If so, then the user gets whatever access level the group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

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. 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.



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 could create a group called **DevMan** 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.

PI Access Levels

PI Server provides these standard levels of access permissions:

- **Read-only access.** Users can view the item, but they cannot edit it
- **Read-write access.** Users can view and edit the item
- **No access.** Users cannot view or edit the item

Note: *No access* is not the same as *deny*. There is no level for *deny*, as there is in Windows.

Where to Set Access Permissions

The entries in the PI SMT Database Security tool control access to PI Server resources at the highest level. You can further configure access permissions for each point and module.

- *Database Security Access Permissions* (page 60)
- *PI Point Access Permissions* (page 61)
- *PI Module Access Permissions* (page 62)

Database Security Access Permissions

The entries in the PI SMT Database Security tool control access to PI Server resources at the highest level. The Database Security tool lists the resources for which you can configure access and shows the current access permissions for each entry. You can change these access permissions if you have read/write access to the PIDBSEC entry. PIDBSEC stands for PI database security, and this entry controls access to all the entries in the table.

The table also includes an entry called PIPOINT and an entry called PIModules. These control access to PI points and PI modules respectively. You can further restrict access to points and modules by editing the security directly on the point or module itself. However, you cannot grant access beyond what is granted in the Database Security tool.

For example, suppose an identity called **PIEngineers** has read/write access for a point called TEST_POINT but has read-only access to PIPOINT in the Database Security tool. The **PIEngineers** Identity can view but cannot make changes to TEST_POINT. **PIEngineers** must also have read/write access to PIPOINT in order to edit the point.

Access Permissions Required for Common Tasks

The following table lists some basic PI Server administration tasks and tells you which tables in the Database Security tool control access to that task.

Administration Task	Which Entries Control the Task
Managing archives	PIARCAADMIN (basic archive administration tasks: archive creation, registration and shifts") and PIARCDATA (archive data that is not tag-specific, such as listing the archives; archive trouble-shooting tasks)
Managing backups	PIBACKUP
Managing identities, users, and groups	PIUSER
Manage mappings	PIMAPPING
Managing trusts	PITRUST
Managing auditing	PIAUDIT
Creating/deleting points	PIPOINT
Creating/deleting modules	PIModules
Editing the database security table	PIDBSEC
Managing firewall table, tuning parameters	PITUNING
Managing message logs	PIMSGSS
Managing PI collectives	PIReplication, PIBACKUP

PI Point Access Permissions

At the top level, access to points is controlled by the PIPOINT entry in the Database Security tool in PI SMT. You can further *restrict* access permissions for individual points, but you cannot grant *more* access than is granted for PIPOINT.

PI point security is divided into two separate pieces, *data security* and *point security*.

Data Security	Specifies who has access to a point's data values (snapshot and archive data).
Point Security	Specifies who has access to the point's configuration (Zero , Span , Descriptor , and other point attributes).

You can have different access permissions for a point's attributes than for the point's data. 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.

- **Data Security:** For viewing and editing point data, you also need read access to point security. If users do not have permission to view a point's attributes, they cannot see that point's data, in most cases. (This is because client applications need access to the point attributes in order to get the data.)
- **Point Security:** In order to view point attributes, you need read access to PIPOINT, as well as read access to the point security for the point itself. Similarly, in order to edit a point's attributes, you need read/write access to PIPOINT, and also read/write access to the configuration for the point itself.

The following table lists required access permissions for basic tasks.

Task	Required Permissions
View point data	Read-only access to <i>all</i> of the following: PIPOINT, data security for that point, point security for that point
Edit point data	Write access to PIPOINT and to data security for that point; read access to point security for that point
View point attributes	Read-only access to PIPOINT and to point security for that point
Edit point attributes	Read-write access to PIPOINT and to point security for that point
Create a point	Read-write access to PIPOINT
Delete a point	Read-write access to PIPOINT and to point security for that point

Set Default Access for New Points

You can set default access permissions for points and modules. When you create a new point or module without explicitly setting access permissions, the point or module gets the default access permissions.

- Default access permissions for all new points (both point data and point configuration) match the access permissions for the point database (PIPOINT). You can set permissions for PIPOINT in the Database Security tool.

Similarly, default access permissions for root modules match the access permissions for the module database (PIModules). You can set permissions for PIModules in the Database Security tool. New modules below the root level inherit from their parent.

Note: This feature was added in version 3.4.380 of PI Server. Earlier versions of PI Server do not allow you to specify default security for new points.

PI Module Access Permissions

At the top level, access to modules is controlled by the PIModules entry in the Database Security tool in PI SMT. You can further restrict access permissions for individual modules, but you cannot grant more access than the PIModules table grants.

Furthermore, modules have a hierarchical structure and each module inherits permissions from parent modules. You cannot grant more access to a module than you have granted to the module's parent. The following table explains the necessary permissions for basic tasks.

Task	Required Access
Create a module	Write access to PIModules, write access to module's parent
Delete a module	Write access to PIModules, write access to module's parent, write access to module
Rename a module	Read access to PIModules, write access to module
Edit a module (except link operation)	Read access to PIModules, write access to module
Edit a module (link operation)	Read access to PIModules, write access to module's parent, write access to module

How to Set Access Permissions

A variety of client tools allow you to set access permissions. For ease of reference, we loosely refer to these as *administrative applications*. The most commonly used administrative applications are:

- For top-level access, use the PI SMT Database Security tool. See *How to Set Permissions In SMT* (page 63).
- For points, you typically use either Point Builder in PI SMT or PI Tag Configurator (to edit multiple tags at once). See *How to Set Permissions In SMT* (page 63).
- For modules, you typically use Module Database Builder.

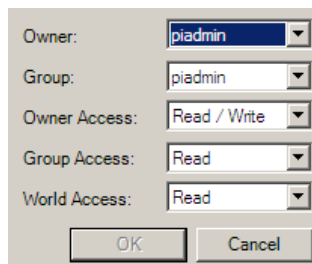
The change in the access permissions model for PI Server 3.4.380 means that you set permissions differently for newer and older versions of the PI Server. To set permissions for the 3.4.380 Server, you should use only tools that are compatible with the new access permissions model (*Which Tools are Compatible with PI Server 3.4.380?* (page 64)).

If you are trying to use an old version of SMT (or any other client) to set permissions on a new version of PI Server, you will probably run into compatibility issues (*Client Tool Incompatibilities* (page 64)).

How to Set Permissions In SMT

To edit the access permissions for a PI point, select the point in PI SMT Point Builder and click the **Security** tab. To edit the access permissions for an item in the Database Security tool, double-click the item to open the Properties dialog box. In both cases, you see a simple interface that allows you to configure security. The interface depends on what version of PI Server you are using.

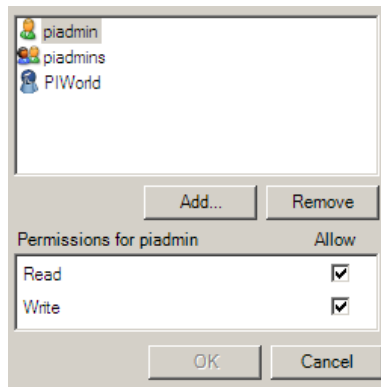
- For PI Servers 3.4.375.99 and earlier, the interface shows access permissions like this:



Owner:	piadmin
Group:	piadmin
Owner Access:	Read / Write
Group Access:	Read
World Access:	Read

OK Cancel

- For PI Servers 3.4.380 and later, the interface shows access permissions like this:



Client Tool Incompatibilities

Administrative applications are applications that allow you to configure access permissions. Examples are PI SMT, Point Builder, Module Database Builder, and so on. When you upgrade to PI Server 3.4.380, your existing access permissions are converted to the new model. New versions of most administrative tools can display access permissions for either the old or the new model, depending on the version of the connected PI Server.

Older versions of administrative applications cannot interpret new-model access permissions unless the access permissions are compatible with the old model. The display of incompatible access permissions depends on the specific client tool. Typically the tool will show:

```
owner: PIUserIncompatible
group: PIGroupIncompatible
```

PIUserIncompatible and **PIGroupIncompatible** are built-in PI identities included in the PI Server 3.4.380 installation.

Which Tools are Compatible with PI Server 3.4.380?

Older versions of administrative tools cannot properly display access permissions unless they follow the owner/group/world model. To work with new-model permissions, you need to run SDK 1.3.6 or later and you need a tool version that supports the new model. Here are the required versions for common administrative tools:

- PI SMT version 3.3.1.3 or later (includes Point Builder, Module Database tool, and Database Security tool)
- PI Tag Configurator version 2.1.3 or later
- Module Database Builder version 1.2.1.3 or later
- PI ICU 1.4.7 or later
- PI APS 1.2.5.0 or later

Tighten Security

This section explains a few steps that you can take to improve security on your PI Server. This is in no way a comprehensive list. To learn more about PI Server security, see *Configuring Security on the PI Server*.

- *Secure the piadmin Account* (page 65): The piadmin account is the PI Server's built-in super-user account. You should protect it as much as possible.
- *Require Passwords on All User Accounts* (page 65): If you are still using explicit logins on PI Server, make sure that you at least require a password for all user accounts. You can create temporary passwords for your users and instruct them to change the passwords within a specific period of time. Once all accounts have passwords, disable access for accounts with blank passwords.
- *Disable All Explicit PI Server Logins* (page 66): When a user logs onto PI Server by typing in a PI user name and password, this is called an *explicit login*. Explicit logins on the PI Server are the least secure authentication method available to you. As you migrate to the new security model, you will no longer need them. At this point, you should disable explicit logins.
- *Disable Explicit Logins for a User* (page 66): If you cannot disable all explicit logins, consider disabling explicit logins for PI user accounts that are accessed through a trust or a mapping.

Secure the piadmin Account

Each PI Server has an administrator account, called **piadmin**. This super user account has permission to do anything on the PI System, regardless of security settings. Thus, you should restrict access to the piadmin account to a small group of trusted PI system managers. Ideally, you should disable explicit logins for this account (*Disable Explicit Logins for a User* (page 66)). If you are not ready to disable explicit logins for **piadmin**, at least put a password on the account. New PI Server installations require a password for **piadmin**, but upgrades from older versions do not.

Require Passwords on All User Accounts

PI Server user passwords provide a minimal level of security for PI Server. If you are going to use individual PI user accounts to manage PI Server, at least require that the user accounts be protected by passwords. When you do this, PI user accounts that do not have passwords cannot authenticate on PI Server.

Follow these steps:

1. In PI SMT, select a server under **Collectives and Servers**. You can change settings for only one PI Server at a time.
2. Open the **Security Settings** tool (**Security > Security Settings**.) The Security Settings tool appears (only for PI Servers running version 3.4.380 or later).
3. Choose the **Blank password not allowed** option.
4. Stop and then restart PI Base Subsystem to apply the changes.

Disable All Explicit PI Server Logins

A good security practice is to disable explicit PI Server logins. On new PI Server installations, explicit logins are disabled by default. When you upgrade PI Server, you are prompted to disable explicit logins, but you are not required to do so. After you configure mappings to replace PI user logins, you should plan to disable explicit logins altogether. Before doing this, make sure that the explicit logins are no longer needed.

Follow these steps:

1. In PI SMT, select a server under **Collectives and Servers**. You can change settings for only one PI Server at a time.
2. Open the **Security Settings** tool (**Security > Security Settings**.) The Security Settings tool appears (only for PI Servers running version 3.4.380 or later).
3. Choose the **Explicit login not allowed** option.

Stop and then restart PI Base Subsystem to apply the changes.

Disable Explicit Logins for a User

When you disable explicit logins for a PI user, then users cannot access PI Server by typing in that user name and password. You can still use the user in mappings and trusts. Because these passwords are not as secure as trusts or Windows authentication (mappings), it is a good idea to disable explicit logins where possible.

Note: You can disable explicit logins only on PI Servers version 3.4.380 or later. Earlier versions of PI Server do not have this feature.

To disable explicit logins for a PI user account:

1. Open the Identities, Users, & Groups tool (select **Security > Identities, Users, & Groups**).
2. Click the **PI Users** tab.
3. Double-click the username to open that PI user's **Properties** dialog box.
4. On the **General** tab, select the **User cannot be used for an explicit login** check box.
5. Click **OK**.

To re-enable explicit logins for a PI user account, clear the same check box.

Monitor PI System Performance

One important way to monitor your PI 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.

- *How to Monitor Performance Counters* (page 67)
- *Which Performance Counters to Monitor* (page 70)


How to Monitor Performance Counters


PI gets performance counter data through the PI Performance Monitor Interface, PIPerfmon. To monitor performance counters:

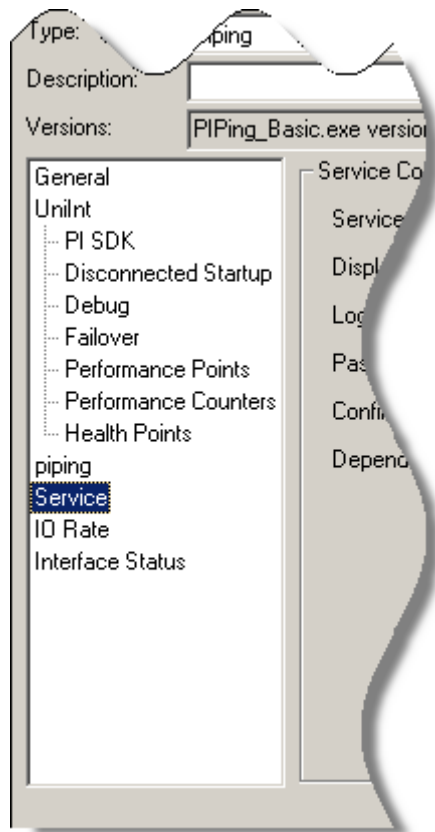
1. Configure the Performance Counter interface (*Configure the PIPerfmon Interface* (page 67)).
2. Create PI points for performance counters (*Build Performance Points* (page 69)).
3. Create a PI ProcessBook display that contains those points (*Trend Performance Points* (page 70)).

Configure the PIPerfmon Interface

To configure PIPerfmon on your PI Server:

1. Start PI ICU.
2. Click the button for creating a new interface . The Configure a New Interface dialog box opens.
3. Click **Browse...** and browse to the location of the PIPerfmon executable. By default, the installation program locates the PIPerfmon executable in the PIPC directory under Interfaces\PIPerfmon_basic. Select the executable and click **Open**.
4. Enter the interface properties:
 - o In **Host PI Server/Collective**, select the host PI System. If you are running the basic version of PIPerfmon then this must be the PI Server itself (*Restrictions on the PIPerfmon Interface* (page 69))
 - o In **Interface name as displayed in the ICU**, enter a descriptive name for the interface, such as Performance Monitor Interface or PIPerfmon.

- o In **Point Source**, enter a **point source** (page 45). The default point source for PIPerfmon is the number sign (#).
 - o In **Interface ID**, enter an **Interface ID number** (page 46). If you are not sure what number to enter, click the **Suggest** button and PI ICU will automatically enter one for you.
5. Click **Add**. A dialog box informs you that the interface is ready to be configured.
 6. Click **OK**.
 7. Now you can configure the interface in PI ICU. You need to add a scan class frequency. Click the new scan class button . Enter the scan class frequency for the interface. For example, to scan once a minute:
00:01:00
 8. Click **Apply**.
 9. Click the **Service** heading on the left side of PI ICU.



10. Enter a `user` name and password for a Windows account with administrative privileges on PI Server. This account is necessary because the interface runs as a Windows service.
11. Click **Create**. This installs the interface. To start the interface, click the **Start interface service** button at the top of PI ICU.

Restrictions on the PIPerfmon Interface

There are two versions of the PIPerfmon interface, *full* and *basic*. PI Server comes with the basic version, which is similar to the full version, with these exceptions:

- Must run on the machine with PI Server
- Limited to 512 points
- Allows one instance of the interface
- Collects data for local performance counters only

Build Performance Points

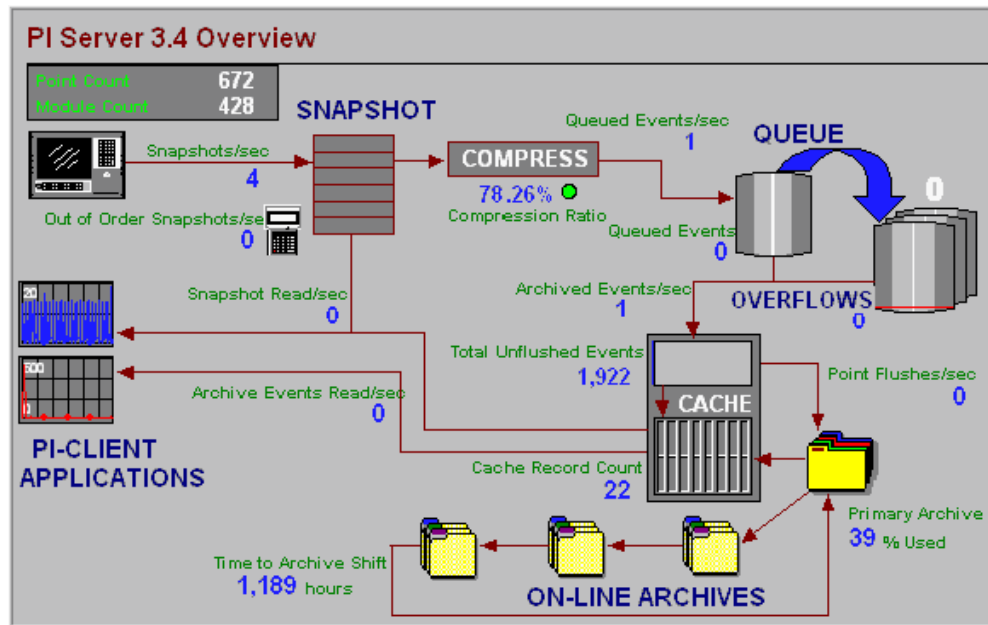
PIPerfmon points are different from interface performance points. You do not use Point Builder to build them. To build a performance monitor point, follow these steps:

1. Start PI SMT.
2. *Select the PI Server* (page 6) where you want to create the performance point.
3. Under **System Management Tools**, select **IT Points > Performance Counters**.
4. On the **Tag Settings** tab, choose an existing **PI Performance Monitor Interface** from the drop-down list. If no interfaces appears in the list, make sure PIPerfmon is installed and running on PI Server (*Configuring the PIPerfmon Interface* (page 67)).
5. Click the **Build Tag** tab and select the performance monitor points you want to create from the list of available counters. See *Which Performance Counters to Monitor* (page 70).

Click the **Create Tags** button. The Performance Counters tool creates the performance monitor tags for you.

Trend Performance Points

If you put your PI Performance Monitor points on a PI ProcessBook display, you can check your PI System performance at a glance, as shown in the following illustration.



You might also include *interface performance points* (page 47) and representative points from each of the batch, alarm, performance equation, and ACE components.

Which Performance Counters to Monitor

The following list contains the basic set of performance counters that we recommend that you monitor. This list includes some counters that are available only on PI Server version 3.4 and higher.

PI Performance Counter Tag	Under Heading	Description
Archived Events/sec	PI Archive Subsystem	Rate of successful event addition to the archive.
Cache Flush Operations/sec	PI Archive Subsystem	Rate at which points are flushed from the archive cache to disk.
Cache Record Count	PI Archive Subsystem	Archive cache records in memory.
Events Read/sec	PI Archive Subsystem	Rate of archive events read.
Primary Archive % Used	PI Archive Subsystem	Percent of used records in primary archive file. Available in PI Server version 3.4 and higher.

PI Performance Counter Tag	Under Heading	Description
Time to Archive Shift	PI Archive Subsystem	Number of seconds until the archive is projected to shift. This time is not calculated if the archive is less than 20% full.
Total Unflushed Events	PI Archive Subsystem	Total number of unflushed events. Available in PI Server version 3.4 and higher.
Module Count	PI Base Subsystem	Total number of modules in the PI module database.
Point Count	PI Base Subsystem	Total number of defined points. This number includes the connector point count.
Events in Overflow Queues	PI Snapshot Subsystem	Total of events in the overflow queue files. Available in PI Server version 3.4 and higher.
Events in Primary Queue	PI Snapshot Subsystem	Number of events in the primary queue file. Available in PI Server version 3.4 and higher.
GetSnapshots/sec	PI Snapshot Subsystem	Rate of events read from the snapshot.
Number of Overflow Queues	PI Snapshot Subsystem	Number of overflow queue files (0 if only the primary queue is active). Available in PI Server version 3.4 and higher.
OutOfOrderSnapshots/sec	PI Snapshot Subsystem	Out of order events sent to the snapshot.
Queued Events/sec	PI Snapshot Subsystem	Events sent to event queue.
Snapshots/sec	PI Snapshot Subsystem	Events sent to the snapshot.

This table lists only a small subset of the PI performance counters. See the *PI Server Reference Guide* for a comprehensive list of all available PI performance counters.

Manage Data Source Equipment

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. The basic system management tasks for data sources are:

- *Add New Equipment* (page 73)
- *Remove Obsolete Equipment* (page 73)
- *Replace Equipment* (page 74)

Add New Equipment

When new equipment comes online, you must configure your PI System 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 must install PI ICU and *set up the trusts* (page 55).
- Install the interface.
- Register and *configure the interface with PI ICU* (page 44).
- Configure buffering, if it is not already running on this interface node. See the *PI Buffering User Guide* for instructions.
- *Build new points* (page 22) to get the data from the equipment into PI Server.

Remove Obsolete Equipment

When equipment goes offline, *decommission any points associated with that equipment* (page 23).

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

Replace 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.

Appendix A

PI Server Health Check

The checklist is organized into functional areas, with a list of items to check in each area. This table does not provide detailed instructions for checking each item, but provides references to sections that do. You perform most of the checks using PI System Management Tools (SMT).

Area	What to Check	How to Check It
PI Server Subsystems	Core PI subsystems and interfaces are running	<i>SMT Server Process Manager</i> (page 11)
Archives	<ul style="list-style-type: none">▪ 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	<i>SMT Archive Manager</i> (page 25) Update Manager
Backups	<ul style="list-style-type: none">▪ PI System backups have been run▪ There is enough disk space for future backups▪ The backup files are copied to the backup media or device	<i>SMT Backups</i> (page 35)
Event Queue	<ul style="list-style-type: none">▪ 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	<i>PI Performance Monitor points</i> (page 67)
Message Log	There are no errors or unusual events in the message logs	SMT Message Log viewer
Connections	<ul style="list-style-type: none">▪ No unusual connection losses/reconnections▪ No stale connections are accumulating▪ There are no network errors	SMT Network Manager Statistics
Data Sources	<ul style="list-style-type: none">▪ The I/O rate tag trends look good▪ There are no error messages in the pipc.log file	<i>Check I/O rate tags</i> (page 47)
Points (Tags)	There are no stale or bad tags	<i>SMT Stale and Bad Tags</i> (page 22)

Area	What to Check	How to Check It
Technical Support Site	<ul style="list-style-type: none">▪ There are no new bulletins▪ There are no software upgrades or patch releases you should install	<i>OSIsoft Technical Support Web site</i> (http://techsupport.osisoft.com)

Glossary

Annotation

Arbitrary information (such as text comments and other binary data) that can be associated with any archive value. Annotations are accessed exclusively with 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.

API Node

A synonym for interface node.

Archive

The historical record of time-series data maintained by PI Server. This term may be used to refer to the entire logical data record itself, a specific archive file, or 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, create archive files 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 does not participate in archive shifts. Unlike a fixed archive, once a dynamic archive receives any data, the shift flag is automatically and permanently disabled.

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 PI Server like **piartool** and **pidiag** require the use of one or more command-line arguments.

Attribute, Point

A characteristic or parameter of a point that directs an interface and 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 be 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.

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 PI Archive Subsystem. All access to the batch database is provided exclusively by PI SDK. The batch database is independent of the older batch information store maintained by PI Batch Subsystem.

Batch Generator (PIBaGen)

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

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 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

A process that typically runs on an interface node for the purpose of storing Interface data during periods when network communication to PI Server is unavailable. When network communication is restored, the buffering service relays the stored data to 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 PI API executable or process name that performs the buffering service. Called API Buffer Server.

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 PI ACE calculations. Another method is event scheduling.

COM Connector

A COM object designed to allow 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 Microsoft 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 least, should typically be set to the error of the underlying instrument.

CompDevPercent

The base attribute that specifies compression deviation as a percentage of the Span attribute, another base attribute. The relationship is defined by the following equation: $\text{CompDev} = (\text{CompDevPercent} / 100) * \text{Span}$. 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 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 PI Server itself never generates events, the archiving rate for the associated point will be at most one event every CompMin seconds.

Component

A component is a logical grouping of files. The purpose of a component is to identify a group of files for backup. For example, all of the files for PI Base Subsystem are grouped under the `pibasess` component.

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 are sent to the archive for storage. Applying compression is one of the main responsibilities of PI Snapshot Subsystem and PI Buffer Subsystem. The specific algorithm used is known as swinging door compression.

Compression Specification

The 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 PI Server. This information can include the client computer's IP address or host name, the client application's name, or the Windows Domain name and Windows user name under which the client application is running. PI API applications are restricted in the credentials that they can specify. 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 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 is 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 PI Base Subsystem.

Distributed Data Collection

Gathering data from multiple sources and from more than one computer on a network. The PI 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 PI Server. Each event consists of two main components: a value and a timestamp. The value can be one of several different data types (such as 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. PI Snapshot Subsystem writes events into the event queue, and PI 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 PI 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 PI 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: $\text{ExcDev} = (\text{ExcDevPercent} / 100) * \text{Span}$. 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 PI Snapshot Subsystem on 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 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 is 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.

Failover

The act of switching a connection from one member server of a collective to another. This can occur manually at the request of a user or automatically if one of the collective members becomes unavailable.

Firewall

A table hosted by Network Manager that provides the first level of security access to 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 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.

High Availability

An implementation that ensures a certain degree of operational continuity during a given measurement period. You can enable HA features in PI components to enhance the data-protection features of a basic configuration and provide near-continuous access to your PI data.

Home Node

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

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 PI Server. Some interfaces also have the ability to read data from 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 PI Server.

Ipsql Utility

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

Mapped Point

A synonym for COM connector point.

Module Database (MDB)

A hierarchical information store hosted by PI Base Subsystem that consists of one or more modules. Each module contains collections of properties, aliases, and other modules.

Module Database Builder

A PI SDK-based Microsoft Excel add-In that allows the creation, modification, or viewing of elements of the module database in a spreadsheet. Module Database Builder is the ideal user interface for 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 PI Server.

Node

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

N-Way Buffering

The capability of storing and relaying data to the multiple member servers in a collective. Both Bufserv (beginning with PI API 1.6.x) and PI Buffer Subsystem provide this functionality.

ODBC

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

Offline Archive Utility

The same program that runs PI Archive Subsystem, but 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 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 PI 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 runs on VAX and Alpha computers with the OpenVMS operating system.

PI3

Nickname for the current generation of the PI Server software.

PI API

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

PI Archive Subsystem

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

PI Base Subsystem

The core PI Server component that is responsible for hosting several configuration stores such as the point table, the user and group table, and the trust table. PI Base Subsystem also hosts the hierarchical module database.

PI 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 PI Batch Subsystem is provided by **piconfig**. Read-only access to the batches recorded by PI Batch Subsystem is typically provided by the PI API, but it is possible to map PI Batch Subsystem units to the batch database so that their batches can also be accessed by PI SDK. The batch information store maintained by PI Batch Subsystem is independent of the newer batch database.

PI BatchView

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

PI Buffer Subsystem

A collective-aware PI3 process that performs the buffering service. This was first introduced in PR1 (PI Server version 3.4.375) and has some notable advantages and disadvantages compared to API Buffer Server (Bufserv).

PI Collective

A set of PI Servers that acts as the logical PI Server in your PI System to provide high availability, disaster recovery, load distribution, and increased scalability. Each server in a collective is called a member of the collective.

PI Message Subsystem

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

PI SDK

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

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.

PI 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 receives a shutdown event. After PI Shutdown Subsystem writes all the shutdown events for the appropriate points, it stops running.

PI 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, PI Snapshot Subsystem responds to client requests for snapshot events and forwards snapshot events for requested points to Update Manager.

PI SQC

The PI SDK-based add-in to PI ProcessBook that enables users to create and view a variety of statistical quality control (SQC) charts on their PI ProcessBook displays. PI SQC chart limits can be PI points, manually entered constants, or values from ODBC datasets defined within ProcessBook.

PI SQC Alarm Manager

A standalone client application used for managing real-time PI SQC Alarms on PI Server. This application has been replaced with the PI SMT **SQC Alarms** tool that provides equivalent functionality.

PI SQL Subsystem

The component of the PI Server that prepares and executes Structured Query Language statements directed against it from mainly ODBC and PI SDK applications. The existence of PI 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.

PI 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, PI Server, and client applications.

PI Tag Configurator

A PI SDK-based add-in to Microsoft Excel that facilitates creating, editing, and viewing points from a spreadsheet. This is the ideal application for bulk point operations.

PI 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.

Piarchss

The executable or process name that implements both PI 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 PI 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. PI Server must be started for nearly all of the commands to function properly.

Pibasess

The executable or process name that implements PI Base Subsystem.

Pibatch

The executable or process name that implements PI Batch Subsystem.

Pibufss

The executable or process name that implements PI Buffer Subsystem.

Piconfig

An interactive command-line utility program that provides access to nearly all the configuration and data stores maintained by PI Server. Several client applications now 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 PI 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 information 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 PI Server for data archiving.

Ping

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

PlonPINet/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 the point database. In piconfig, the table that provides access to the point database.

Pishutev

The executable or process name that implements PI Shutdown Subsystem.

Pisnapss

The executable or process name that implements PI Snapshot Subsystem.

Pisqlss

The executable or process name that implements PI 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, and control limits. Each point must be assigned a unique tag on 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 PI 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 PI Base Subsystem.

Point Identifier (PointID)

The unique number used to identify a point. This is primarily for internal use within PI Server, but is often needed in PI API and PI 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 and 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 int16, int32, float16, float32, float64, digital, string, Blob, and timestamp. You can edit the **PointType** attribute 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 PI Snapshot Subsystem to PI Archive Subsystem. With the introduction of the memory-mapped event queue in PI Server, posting is now rarely performed.

Postprocessing

Processing by 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.

Primary PI Server

The server in a collective that stores configuration and initiates configuration changes to one or more secondary servers. A collective includes exactly one primary server.

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, Float32 point type, or Float64 point type.

Ramp Soak

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

Random Simulator

A standard interface program included with 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 PI Server without affecting actual process data.

Real-time SQC

The component of PI Server that provides continual evaluation of statistical quality control (SQC) pattern tests and the management of alarms generated from these tests. Use of this component helps in monitoring how well a process stays within its control limits.

Recalculator

The component of 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 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 Microsoft Windows platforms.

Registering an Archive

Informing PI 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 PI SMT.

Replication

The ongoing synchronization of configuration changes between the primary PI Server and a secondary PI Server in a collective.

Satellite Node

Any remote computer on a network running PI software other than the PI Server software (such as interfaces, PINet, and 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 is not 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.

Secondary PI Server

A type of collective member server that retrieves all configuration changes from the primary PI Server. Direct configuration changes are disallowed on a secondary PI Server. A collective includes one or more secondary PI Servers.

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 PI Server or an interface or some other application or device is not available.

Snapshot Event

An overloaded term that can refer to either any event sent to PI 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.

SNMP Interface

An interface included that collects performance data from any device that supports the simple network management protocol (SNMP) 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 PI Server for Microsoft Windows.

SNMP Point Builder

A PI SMT tool 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.

State Set

A synonym for digital state set or DigitalSet.

Statement Handle

An object allocated by PI 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. For points with non-numeric type (digital, string, and timestamp) the Step attribute is always non-zero.

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. PI Server has several core subsystems such as Network Manager, Update Manager, PI Base Subsystem, PI Snapshot Subsystem, and PI Archive Subsystem.

Swinging Door Compression

A data compression algorithm used by PI Snapshot Subsystem and PI Buffer 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 PI 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 your PI System. PI SMT 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, including *, ?, \, and ;. The term "tag" and "point" are often used interchangeably.

Timeout Table

The information store that contains all the configuration parameters for PI Server. When tuning the performance of PI Server, you typically need to adjust several of these timeout table parameters.

Timestamp

A date and time, almost always associated with a data value through an event. PI Server stores timestamps internally in UTC.

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.

Trust

A record stored on PI Server that automatically grants access for a program connecting to PI Server without requiring an explicit PI user login. A trust consists of one or more connection credentials 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 PI Base Subsystem. Trust lookup is always performed when an application first connects to 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 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, PI ProcessBook requests updates of snapshot events for a point on a trend so that the trace will remain current; all such events 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.

Appendix C

Technical Support and Resources

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>

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 log files at that time

Help Desk and Telephone Support

You can contact OSISOFT Technical Support 24 hours a day. Use the numbers in the table below to find the most appropriate number for your area. Dialing any of these numbers will route your call into our global support queue to be answered by engineers stationed around the world.

Office Location	Access Number	Local Language Options
San Leandro, CA, USA	1 510 297 5828	English
Philadelphia, PA, USA	1 215 606 0705	English
Johnson City, TN, USA	1 423 610 3800	English
Montreal, QC, Canada	1 514 493 0663	English, French
Sao Paulo, Brazil	55 11 3053 5040	English, Portuguese
Frankfurt, Germany	49 6047 989 333	English, German
Manama, Bahrain	973 1758 4429	English, Arabic
Singapore	65 6391 1811 86 021 2327 8686	English, Mandarin Mandarin
Perth, WA, Australia	61 8 9282 9220	English

Support may be provided in languages other than English in certain centers (listed above) based on availability of attendants. If you select a local language option, we will make best efforts to connect you with an available Technical Support Engineer (TSE) with that language skill. If no local language TSE is available to assist you, you will be routed to the first available attendant.

If all available TSEs are busy assisting other customers when you call, you will be prompted to remain on the line to wait for the next available TSE or else leave a voicemail message. If you choose to leave a message, you will not lose your place in the queue. Your voicemail will be treated as a regular phone call and will be directed to the first TSE who becomes available.

If you are calling about an ongoing case, be sure to reference your case number when you call so we can connect you to the engineer currently assigned to your case. If that engineer is not available, another engineer will attempt to assist you.

Search Support

From the OSIsoft Technical Support Web site, click Search Support.

Quickly and easily search the OSIsoft Technical Support Web site's support solutions, documentation, and support bulletins using the advanced MS SharePoint search engine.

E-Mail–Based Technical Support

techsupport@osisoft.com

When contacting OSIsoft Technical Support by e-mail, it is helpful to send the following information:

- Description of issue: Short description of issue, symptoms, informational or error messages, history of issue.
- Log files: See the product documentation for information on obtaining logs pertinent to the situation.

Online Technical Support

From the OSIsoft Technical Support Web site, click My Support > **My Calls**.

Using OSIsoft's Online Technical Support, you can:

- Enter a new call directly into OSIsoft's database (monitored 24 hours a day)
- View or edit existing OSIsoft calls that you entered
- View any of the calls entered by your organization or site, if enabled
- See your licensed software and dates of your Service Reliance Program agreements

Remote Access

From the OSIsoft Technical Support Web site, click Contact Us > **Remote Support Options**.

OSIsoft Support Engineers may remotely access your server in order to provide hands-on troubleshooting and assistance. See the Remote Support Options page for details on the various methods you can use.

On-Site Service

From the OSIsoft Technical Support Web site, click Contact Us > **On-site Field Service Visit**.

OSIsoft provides on-site service for a fee. Visit our On-site Field Service Visit page for more information.

Knowledge Center

From the OSIsoft Technical Support Web site, click 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** on the Technical Support Web site.

- The Search Support 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 checks, daylight saving time configuration, PI Server security, PI System sizing and configuration, PI trusts for interface nodes, and more.

Upgrades

From the OSIsoft Technical Support Web site, click Contact Us > **Obtaining Upgrades**.

You are eligible to download or order any available version of a product for which you have an active Service Reliance Program (SRP), formerly known as Tech Support Agreement (TSA). To verify or change your SRP status, contact your Sales Representative or *Technical Support* (<http://techsupport.osisoft.com/>) for assistance.

Training

OSIsoft offers training classes and computer-based training (CBT) CDs on managing PI Servers. For more information, see the *OSIsoft training website* (<http://training.osisoft.com>).

Index

A

- Access permissions • 59
- Access privileges
 - Points • 22
- Adding new data sources • 73
- adm directory • 10
- Administrator account • 64
- Alarm point class • 18
- Archive Manager • 25
- archive queuing • 15
- Archives
 - Creating • 30
 - Managing • 25
- Attributes
 - Compression specification • 20
 - Exception reporting • 19
 - Location1 • 46
 - Location4 • 19
 - Points • 17
 - PointSource • 19, 45
 - PointType • 17, 18
 - PtClass • 18
 - Shutdown • 21
 - Span • 21
 - Tag • 18
 - Typical value • 21
 - Zero • 21

B

- Backups
 - Monitoring • 34
- Bad points
 - Finding • 22
- Base point class • 18
- bin directory • 10
- Blob point type • 18

C

- Checking Backups • 34
- Checking interface log files • 47
- Checklist for System Managers • 3
- Class attribute • 18
- Classic point class • 18
- CompDev • 14
- CompDev attribute • 20

- CompDevPercent • 14
- CompDevPercent attribute • 20
- CompMax • 14
- CompMax attribute • 20
- CompMin • 14
- CompMin attribute • 20
- Compressing attribute • 20
- Compression deviation • 20
- Compression Flag • 20
- Compression specifications
 - Attributes • 20
- Compression testing • 14
- Configuring interfaces
 - In the ICU • 44
- Creating performance points • 69
- Creating points • 22

D

- Daily Health Check • 3
- dat directory • 10
- Data
 - How often to get new values • 19
- Data Access privileges • 22
- Data flow
 - Overview • 9
 - Server • 11
- Data sources
 - Adding new • 73
 - Removing equipment • 73
- Decommissioning points • 22
- Deleting points • 22, 23
- Demonstration account • 64
- Digital point type • 18
- Directories
 - Server • 10
- Documentation
 - On interfaces • 44

E

- Equipment
 - Adding new • 73
 - Removing • 73
- Event Queue • 15
- ExcDev • 12
- ExcDev attribute • 19
- ExcDevPercent attribute • 19

Exception reporting • 12
 Attributes • 19
ExcMax • 12
ExcMin • 12

F

File system
 Server • 10
Filtering data
 Exception reporting • 12
Float16 point type • 18
Float32 point type • 18
Float64 point type • 18
Frequency
 Data collection • 19

G

Groups
 Managing • 64

H

Health Check • 3

I

ICU • 8
 Configuring interfaces with • 44
ID number, Interface • 46
Information
 On interfaces • 44
Int16 point type • 18
Int32 point type • 18
Interface ID number • 46
Interfaces
 Configuring • 44
 Getting more information • 44
 Log files • 47
 Monitoring performance • 47
 PIPerfmon • 67
 Point source • 45, 93
 Registering • 44
 Starting and stopping • 43
IORates points • 47

L

Location1 attribute • 46
Location4 Attribute • 19
log directory • 10
Log files

Interfaces • 47
 pigetmsg • 47
 pipc.log • 47

M

Monitoring Backups • 34

N

Naming points
 Tag attribute • 18
No access • 59

O

Obsolete points • 22, 23
Offset • 46, 95

P

Performance
 Interfaces • 47
Performance counters
 Which to use • 70
Performance points
 Creating • 69
 Interfaces • 47
 PIPerfmon interface • 67
Period • 46, 95
Permissions
 Access • 59
PI
 Overview • 9
piadmin account • 64
piarchss_Archived Events/sec • 70
piarchss_Cache Record Count • 70
piarchss_Events Read/sec • 70
piarchss_Primary Archive % Used • 70
piarchss_Time to Archive Shift • 70
piarchss_Total Unflushed Events • 70
pibasess_Module Count • 70
pibasess_Point Count • 70
pidemo account • 64
pigetmsg • 47
pipc.log • 47
PIPerfmon interface • 67
pisanpss_Events in Overflow Queues • 70
pisanpss_Events in Primary Queue • 70
pisanpss_GetSnapshots/sec • 70
pisanpss_Number of Overflow Queues • 70
pisanpss_OutOfOrderSnapshots/sec • 70
pisanpss_Queued Events/sec • 70

pisnapss_Snapshots/sec • 70

Point Access

Privileges • 22

Point classes • 18

Point Security

Configuration • 22

Point Source • 45, 93

PIPerfmon • 67

Points

Attributes • 17

Creating • 22

Decommissioning • 22

Deleting • 22, 23

Finding Malfunctioning • 22

points vs. tags • 18

PointSource attribute • 19

PointType attribute • 17, 18

Privileges

Points • 22

PtClass attribute • 18

Q

Queue, Event • 15

R

Read only access • 59

Read/Write access • 59

Registering interfaces • 44

Removing equipment • 73

Retiring points • 22

S

Scan Class • 46, 95

Location4 attribute • 19

PIPerfmon interface • 67

Security

Point configuration • 22

Server

About • 10

Compression • 14

Compression testing • 14

Data flow • 11

File system • 10

Snapshot • 14

Shutdown attribute • 21

SMT

About • 5

Archive Manager • 25

Snapshot • 14

Span attribute • 21

SQC_Alarm point class • 18

Stale points

Finding • 22

Starting interfaces • 43

Stopping interfaces • 43

String point type • 18

T

Tag attribute • 18

tags vs. points • 18

Timestamp point type • 18

Totalizer point class • 18

Training

Where to find • 101

Type attribute • 17, 18

Typical Value attribute • 21

U

UniInt

Documentation • 44

Universal Interface

Documentation • 44

Users

Managing • 64

UTC Time • 46, 95

Z

Zero attribute • 21