



PI DataLink User Guide

2010

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.

Published: 6/14/2010

Table of Contents

Chapter 1 Introduction	1
PI DataLink Editions	1
System Requirements	3
Installation and Upgrade	3
Configuration	4
About this Document	6
Chapter 2 Basics	7
User Interface	7
PI Server Connections	11
Graphic Function Components	13
Preference Settings	21
Spreadsheet Construction	23
Chapter 3 PI Functions	25
Current Value	25
Archive Value	26
Compressed Data	27
Sampled Data	30
Timed Data	31
Calculated Data	33
Time Filtered	37
Chapter 4 PI Tags and Attributes	39
Tag Search	39
Tag Functions	41
Chapter 5 Module Database Objects	45
Module Database Browse	45
Module Database Functions	47
Chapter 6 Spreadsheets	49
Calculation Frequency	49
Functions and Array Values	51
Array Management	51
Share Spreadsheets	53
Chapter 7 Trend Displays	55
Create a Trend	55

Specify the Time Range	58
Title and Placement.....	59
Context Menu	60
Chapter 8 Notifications	63
Search and Display Notifications	63
View Notifications	65
Insert Notifications.....	66
Appendix A Supplementary Information	69
Setup	69
PI Time	71
PI Data Type Support.....	73
PI Expressions	74
Manual Function Entry	80
Write Data to PI.....	83
Troubleshooting.....	84
Appendix B PI DataLink Function Reference	87
Single Value Functions.....	87
Multiple Value Functions	89
Calculation Functions.....	96
Tag Functions.....	109
Module Database Functions	111
Input Functions.....	112
Appendix C Technical Support and Resources	115
Index	119

Chapter 1

Introduction

PI DataLink is a Microsoft Excel add-in that enables you to retrieve information from your PI Server directly into a spreadsheet. With PI DataLink, you can:

- retrieve point values from a PI server
- retrieve system metadata to create a structured view of PI data:
 - PI tag names and attributes
 - PI Module Database paths, aliases and properties
- reference these items using PI DataLink functions to calculate and filter data
- keep values updated when the spreadsheet recalculates
- retrieve PI Notifications you are subscribed to from a PI System
- build a trend from the spreadsheet data or selected PI points

PI DataLink provides a graphical interface to retrieve data and build functions and calculations. DataLink functions are embedded in spreadsheet cells and can provide active updates of real-time data from the PI server.

You can also use the rich calculation and formatting capabilities of Excel to organize and present PI system data to fit your purpose or audience.

Combined with the computational, graphic and formatting capabilities of Microsoft Excel, PI DataLink offers powerful tools for gathering, monitoring, analyzing, and reporting PI data.

PI DataLink Editions

Note: Throughout this book, references to **Microsoft Excel 2007** apply to **Microsoft Excel 2007 and later**. References to **Microsoft Office 2007** apply to **Microsoft Office 2007 and later**.

Two editions of PI DataLink are installed on your system:

- An add-in is installed for Microsoft Excel 2007. This add-in cannot run in earlier versions of Excel. This edition is configured automatically if you have Excel 2007 on your system.
- An add-in is installed for Microsoft Excel XP or 2003. This add-in resembles earlier versions of PI DataLink and must be *configured manually* (page 4).

The differences between the add-ins reside primarily in *design and user interface* (page 7) changes for Office 2007 and later.

PI SDK

PI SDK is installed with PI DataLink, and connects PI DataLink to your PI server to retrieve PI point data.

PI DataLink uses both PI SDK and the Microsoft Excel SDK. When an error occurs at the PI SDK level, PI DataLink displays the error in the cell. For example, if a function attempts to retrieve the current value of a tag that does not exist, the output cell displays a "Tag not found" message.

PI SDK also allows users with Microsoft Visual Basic for Applications (VBA) experience to develop Excel routines that call PI SDK functions. Users who plan to develop programs outside of Excel must purchase a separate PI SDK or PI API license. Contact your OSIsoft sales representative for more information.

PI DataLink Server (DLS)

PI DataLink Server (PI DLS) is a server version of PI DataLink for use with Microsoft SharePoint Server 2007 and later. PI DLS supports all DataLink and PI BatchView functions provided by Excel Add-ins in a typical PI client configuration.

PI DLS enables you to retrieve and view data from PI Servers in Microsoft Excel spreadsheet format using only a web browser. Access to real-time PI data can be expanded to many different consumers, regardless of location, or familiarity with either Microsoft Excel or PI DataLink functions.

This server version of PI DataLink uses Microsoft SharePoint technologies including Excel Services and the Excel Web Access web part. Together, these Microsoft components support a majority of standard spreadsheet features. PI DataLink Server augments these services, adding connectivity and support for PI functions.

Excel workbooks published to SharePoint document libraries can be accessed independently, or added to web part pages. The Excel Web Access web part supports connections that allow parameters to be exchanged between a workbook and other web parts, integrating spreadsheets with PI DataLink functions into the PI WebParts environment.

Contact your OSIsoft sales representative for more information about PI DLS.

PI DataLink Language Support

This release of PI DataLink provides multi-language support. The user interface of the add-in appears in the same language as the Microsoft Excel that it runs in, if the language is supported by PI DataLink. Otherwise the add-in appears in English.

To view the add-in in a different language, use the Microsoft Office language setting tool to change the language of Microsoft Excel.

To get the multi-language support, you must install the separate language pack.

System Requirements

For up to date system requirements, see the OSIsoft Tech Support site:

<http://techsupport.osisoft.com/Products.htm> (<http://techsupport.osisoft.com/Products.htm>)

Installation and Upgrade

Installation

To install, download the installation kit from <http://techsupport.osisoft.com>, and run the PI DataLink installation kit.

The setup program installs PI DataLink under the root directory PIPC, typically in:

C:\Program Files\PIPC\Excel

The installation also includes online help files and release notes, both of which are also available for download at <http://techsupport.osisoft.com>.

Upgrade

If you have a previous version of PI DataLink installed on your computer, the installer automatically upgrades your installation. Preference settings from previous versions are retained.

The following points may be helpful when upgrading from previous versions of PI DataLink:

- PI DataLink is backwards compatible. Workbooks created in older versions can be read by the current version without any conversion. However, the reverse is not necessarily true. Once you modify a spreadsheet in the current version, it may no longer work in older versions of PI DataLink.
- Current versions of PI DataLink use PI SDK, while 2.x and previous versions used PI API.
 - Error messages returned may be different, so if the spreadsheets look for certain strings (for example, in Excel VBA code or macros) the text may not be the same.
 - 3.x and 4.x versions do not create an API connection to the PI Server, so VBA code in workbooks that depends on an existing API connection must be edited to create a new, explicit connection to PI API.

Configuration

After installation, you may need to configure the PI DataLink add-in in Excel.

Microsoft Excel 2010

The PI DataLink setup program automatically installs the DataLink add-in for Microsoft Excel 2010 if this version is found on your system. No configuration is necessary.

Installation makes PI DataLink available to all users of the client machine. If an individual user deactivates PI DataLink or a related add-in, updates do not reactivate the add-in; it must be restored manually. To restore add-ins:

1. Click the **File** tab and select **Excel Options**.
2. Click **Add-ins > Manage Disabled Items > Go** and enable *PIDataLink.UI.dll.manifest*.
3. Go to **Add-ins > Manage Com Add-Ins > Go** and enable *PI DataLink* or *PI Notifications*.
4. Click **OK**.

Microsoft Excel 2007

The PI DataLink setup program automatically installs the DataLink add-in for Microsoft Excel 2007 if this version is found on your system. No configuration is necessary.

Installation makes PI DataLink available to all users of the client machine. If an individual user deactivates PI DataLink or a related add-in, updates do not reactivate the add-in; it must be restored manually. To restore add-ins:

1. Click the **Microsoft Office Button** and select **Excel Options**.
2. Click **Add-ins > Manage Disabled Items > Go** and enable *PIDataLink.UI.dll.manifest*.
3. Go to **Add-ins > Manage Com Add-Ins > Go** and enable *PI DataLink* or *PI Notifications*.
4. Click **OK**.

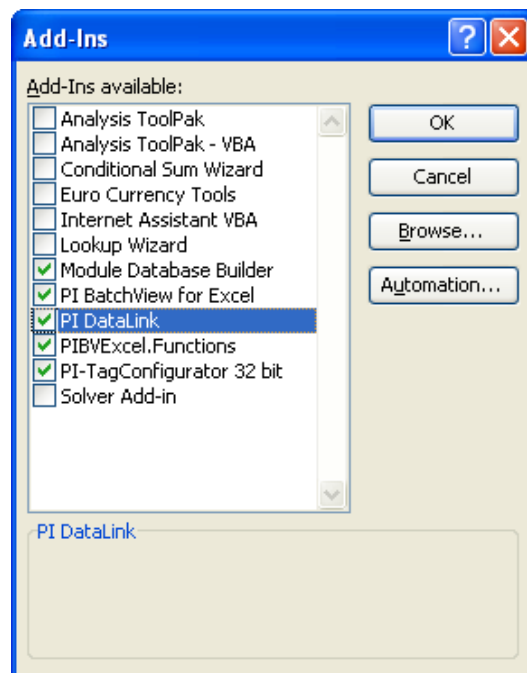
Microsoft Excel XP-2003

If you install PI DataLink for Excel XP-2003, you will need to configure the add-in for Excel following installation. You can also use this procedure to restore the add-in if the PI menu does not appear in the Excel menu bar.

To add PI DataLink to Excel XP-2003 versions:

1. In Excel, choose **Tools > Add-ins**.

The **Add-Ins** dialog appears.



2. If the PI DataLink add-in is not listed, click **Browse** to locate the *PIPC32.XLL* file on your local hard drive.

The default path for PI DataLink is *PIPC\EXCEL*.

3. Click the check box next to *PI DataLink*, and click **OK**.

About this Document

The PI DataLink User Guide provides a detailed description of product features and a complete reference to PI DataLink functions. The content of print and online versions of the guide is identical. Trend control objects also launch a separate help file which includes information on programmatic controls for trends.

Printed versions of the PI DataLink User Guide include tutorial exercises used by OSIsoft for PI DataLink training courses. These exercises are also available in PDF format (along with accompanying training exercise files) from the OSIsoft support web site at techsupport.osisoft.com.

This guide does not duplicate or distinguish between PI DataLink add-ins for Excel 2007 and Excel XP-2003 when discussing common functionality unless there are significant functional differences.

Specific references to PI DataLink for Excel XP-2003 are labeled *(2003)*, and those specific to PI DataLink for Excel 2007 are labeled *(2007)*, when necessary.

Chapter 2

Basics

The following sections introduce basic PI DataLink concepts and features. You should familiarize yourself with these sections before using PI DataLink to build a spreadsheet and retrieve PI data, particularly if you are new to PI DataLink or Office 2007.

These sections describe:

- Common interface components, and how they differ between PI DataLink add-ins for Excel XP-2003 and Excel 2007
- Core features required to begin working with PI DataLink
- Different approaches to building a PI DataLink spreadsheet, depending on your goals, needs, and resources

User Interface

PI DataLink is an add-in application to Microsoft Excel, and appears within the Microsoft Excel user interface in the form of menus, dialog boxes and task panes specific to PI DataLink.

While the underlying functionality of PI DataLink is the same for Excel XP-2003, and 2007, the user interface and tools used to build and manage PI DataLink spreadsheets differ slightly to reflect the design and objects available from the corresponding version of Microsoft Excel.

Depending on your version of Excel, you may work with either of the following groups of PI DataLink components:

PI DataLink for Excel XP-2003	PI DataLink for Excel 2007
PI menu	PI ribbon
function dialog boxes	function task panes
right-click context menu	right-click context menu
Trend Control Wizard	Trend Control Wizard
PI Notifications dialog box	PI Notifications dialog box

The following section describes the visual and operational differences between PI DataLink add-ins.

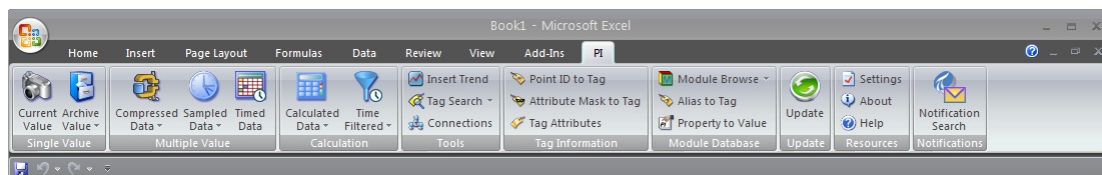
PI DataLink for Excel 2007

PI DataLink for Excel 2007 includes design elements introduced in Microsoft Office 2007.

If you are upgrading from a previous version of PI DataLink, familiarize yourself with the new design elements before you begin working with PI DataLink.

PI Ribbon

The PI DataLink add-in for Microsoft Excel 2007 uses a ribbon menu.

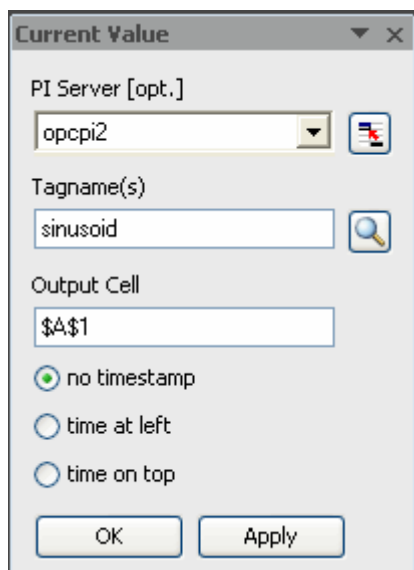


The ribbon menu is a graphical menu that combines features of both a standard menu and a toolbar.

- Click **PI** in the Excel menu bar to select the tab for the PI DataLink ribbon menu.
- Click an item in the PI ribbon to open a corresponding task pane or dialog box.
- Place the cursor over an item to display a descriptive tooltip.

Function Task Panes

PI DataLink for Microsoft Excel 2007 provides custom task panes to define PI DataLink functions.



A task pane is a moveable and dockable panel of controls that functions much like a dialog box, except that you can continue to work on a spreadsheet while a task pane is open.

PI DataLink opens a task pane when you add a function or select a cell in a function array for editing. To open a function task pane:

- Click in the desired output cell, and then click a function on the **PI** ribbon to add a function.
- Click a cell in an existing function array to display a corresponding task pane and edit function arguments.
- Right-click a function array cell and choose the function name to manually display the task pane.

Note: You can *turn off automatic display* (page 21) of task panes if you prefer.

Once a function task pane is open:

- Enter or change values, and then click **OK** to save your changes and close the task pane. Click **Apply** to save changes without closing the task pane.
- Click the **X** button in the task pane title bar to close the task pane without saving changes.
- Click the arrow button in the task pane title bar to detach and **Move**, or **Resize** the task pane.

You can also click the title bar to drag and dock a task pane, or place your cursor over the edge and drag to resize the pane.

Note: Dialog boxes are used for other tasks including **Tag Search**, **Preferences** and **Connections** in the add-in for Excel 2007.

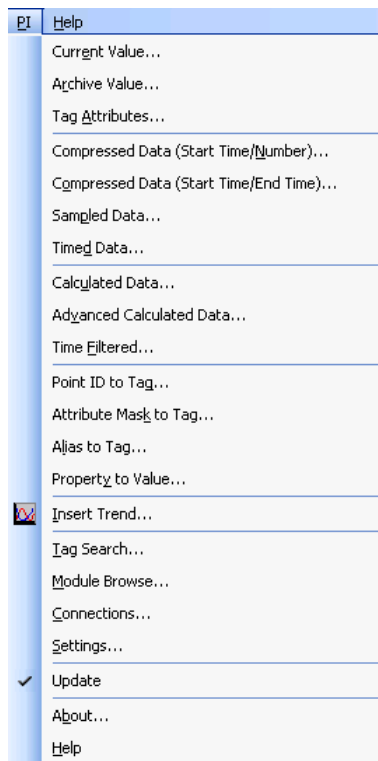
PI DataLink for Excel XP-2003

PI DataLink for Excel XP-2003 appears in Microsoft Excel as a standard menu and related dialog box controls.

If you have used previous releases of PI DataLink, the add-in for Excel XP-2003 maintains the same look and feel.

PI Menu

PI DataLink for Excel XP-2003 adds a **PI** menu to the Excel menu bar once the PI DataLink add-in is loaded. If you do not see the **PI** menu, you may need to *configure* (page 4) the PI DataLink add-in for Excel.

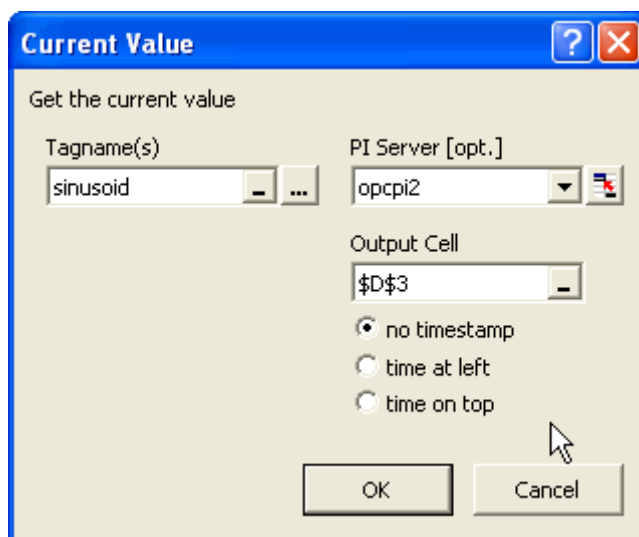


PI DataLink for Excel XP-2003 uses a standard menu configuration:

- Choose an item from the **PI** menu to open a corresponding dialog box.

Function Dialog Boxes

PI DataLink for Microsoft Excel XP-2003 provides dialog boxes to define PI DataLink functions.



PI DataLink opens a corresponding function dialog box when you add or edit a function. To open a function dialog box:

- Click in a cell and then choose a function from the **PI** menu to add a function.
- Right-click a cell in an existing function array to display the context menu, and then choose the function name from the menu to edit function arguments.

Once a function dialog box is open:

- Enter or change values, and then click **OK** to save your changes and close the dialog box.

PI DataLink automatically retrieves values from PI and populates the function array in your spreadsheet when you click **OK**.

- Click **Cancel** or **X** to close the dialog box without saving changes.

Context Menu

PI DataLink adds several commands to the standard Microsoft Excel context menu. You can use these commands to manage function arrays that are already inserted into a spreadsheet.

Note: Context menus are the same for both add-in versions of PI DataLink.

To display the context menu:

- Right-click anywhere in a cell or group of cells that contains a PI DataLink function array.

A function array is the cluster of cells that contain the output from a single PI DataLink function. You can also right-click on an embedded trend control to see a *related context menu* (page 60).

The following items appear in the context menu:

- Choose **Select DataLink Function** to select the entire function array.

You should select an array before attempting to copy, cut or drag it to a new spreadsheet location.

- Choose **Recalculate (Resize) Function** to *recalculate* (page 50) the array, retrieving new values from PI.

For some functions, a different number of values may be available with each recalculation, causing the array to resize.

- Choose **<function name>** to open a corresponding *function dialog box* (page 10) and update arguments to collect different data.

PI Server Connections

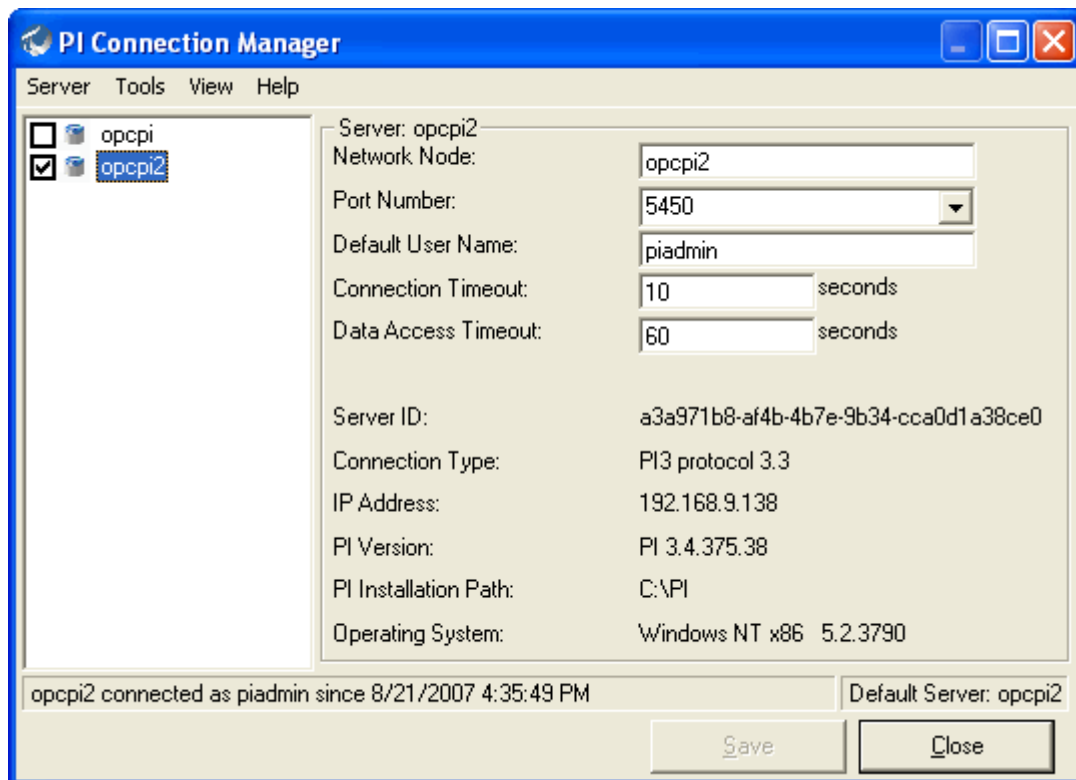
Use the **Connections** dialog box to manage connections to PI servers in your computing environment. To connect to a PI server:

- Click the **Connections** icon on the ribbon bar (2007), or choose **PI > Connections** (2003) to display the **Connections** dialog box.

Each configured PI server available to PI DataLink appears in the **Server** pane to the left.

If you do not see the desired server, you can add a connection. To add additional PI server connections:

Choose **Server > Add Server** or right-click in the server pane and choose **Add Server** from the context menu to display the **Add Server** dialog box and configure a new PI server connection. A selected checkbox next to a server name in the **Connections** dialog box indicates an open connection to the PI Server.



To manage connection settings:

- Click to select a checkbox and open a connection to a PI Server in the list. Clear a checkbox to close a connection.
- Select a PI Server in the server pane to access connection settings.

Note: An open connection to a PI server is required to add DataLink functions to a spreadsheet and retrieve values from PI. The name of the default PI server appears at the bottom right of the dialog box. You can maintain connections to more than one PI server at a time, although you may need to specify the target server when you define a PI DataLink function.

Graphic Function Components

PI DataLink provides a graphical user interface to build functions in Excel. Common dialog box features make it easy to supply arguments to define the function.

The tool used to build a function depends on the version of PI DataLink installed:

- The add-in for Excel XP-2003 provides *dialog boxes* (page 10) to define functions.
- The add-in for Excel 2007 provides *task panes* (page 8) to define functions.
- Advanced users familiar with PI DataLink and the PI Server can *enter function syntax* (page 80) directly into the Excel formula bar.

The following sections describe common features in PI DataLink function dialog boxes and task panes, how real-time data is acquired and presented, and how PI DataLink functions can be used within a spreadsheet.

Entry Fields

Labeled entry fields accept specific arguments used to define a function. Most arguments may be entered directly, or referenced in spreadsheet cells. While the options differ by argument, all entry fields allow one or more of the following choices:

- Enter text directly in an edit field , such as a tag name or the address of an output cell.
- Select an item from choices in a list , such as a calculation or a sampling method.
- Reference spreadsheet cells   that contain arguments, such as a tag name, output cell location, or timestamp.

Note: Be sure to enter strings in cells that you may reference with a leading apostrophe ('). This forces Excel to interpret the contents as a string.

- Select values from a PI server  or other sources  through a tag or module database search.

For example, you can enter a tag name string into the **Tagname** field, or click the button next to the field to display the **Tag Search** dialog box, and search the PI server for tags.

You can also manually supply a reference to a spreadsheet cell that contains a tag name:

- Click first in the edit field, and then click the cell (or click the cell and drag to an adjacent cell to select a range) on the spreadsheet.

DataLink automatically enters the cell reference into the edit field.

Note: Some entry fields display a default entry when the dialog box or pane opens. Fields marked **Optional** are optional, and a value is not required.

Standard Arguments

Some arguments are common to most PI DataLink functions in dialog boxes or task panes, and are usually required to define the function:

Tagname(s) or Expression

The **Tagname(s)** or **Expression** field is required by most functions to evaluate PI point (tag) data or the results of a *PI expression* (page 74). One or more tag names, or the syntax of a PI expression may be:

- Entered directly in a field.

For example, `sinusoid`.

- Entered in one or more spreadsheet cells, which are then referenced in the field.

For example, a reference to the array `Sheet1!$B$3:Sheet1!$B$4`, which in turn contains the tag name strings `'sinusoid'` and `'cdt158'`.

Note that cell references allow multiple tag names to be specified for a function, whereas you cannot specify multiple tag names directly unless the field is labeled **Tagname(s)**.

Note: Note that single quotes are added to each tag name to denote string values in Excel. Both strings and time expressions referenced in cells should be enclosed in single quotes. For example:

```
TimeEq('CD:M158','y','t','Manual')
```

- Selected in the *Tag Search dialog* (page 39), based on a search for tags by name, alias or attribute.

Start Time

The beginning of a *time range* (page 15) over which the function is evaluated, from Start Time to End Time. All event values corresponding to the time range are retrieved from the PI archive or calculated to create the resulting function array.

End Time

The end of the time range, frequently expressed as the current time, where the Start Time is the current time minus a specified time interval.

Start and end times may be expressed in absolute or relative terms to establish a time range in the past, or up to and including the current time.

PI Server

An instance of a PI DataLink function runs against only one PI server. The **PI Server** specifies a target PI server from a known servers list. If you do not select a server or leave the field blank, the *default PI server* (page 11) is used.

Servers that appear above the dotted line in the servers list are currently connected; those below are disconnected.

Output Cell

Specify where you want to place the resulting function array using the **Output Cell** field. If you select a cell before the function dialog or task pane is displayed, the currently-selected cell is used as the default **Output Cell** value.

The output cell value always specifies the top left corner of the function array. If you *append* (page 16) timestamps and other data, keep in mind that the data column may be shifted downward or to the right of the output cell location, overwriting data in those adjacent cells.

If you click in the **Output Cell** field and then select a multi-cell array in the spreadsheet, you can override the default display of all function results. Instead, only the results that fit within the specified array dimensions are displayed.

Time Arguments

Many PI DataLink functions require **Start Time** and **End Time** arguments to retrieve an array of event values over a specific time range. Follow these guidelines when specifying time arguments:

- Time strings may be entered directly in edit fields in either absolute or relative *PI time* (page 71) formats (for example, 10-Dec-99 19:12 or -3h).
- If the start time is more recent than the end time, results are displayed in reverse chronological order.
- Timestamps referenced in spreadsheet cells may also use fixed or relative PI time formats, and should be preceded by an apostrophe to indicate a string (for example, '10-Dec-99 19:12 or '-3h).
- Cell references may also use the absolute Excel time format (such as 39299.6146, equivalent to 8/5/2007 2:45:00 PM). Excel stores timestamps in this format, which represents the cumulative number of days since 1900. Excel can display the same timestamps using any date-time format assigned to the cell.
- Some arguments call for an interval or duration of time, represented by a single value rather than start and end times. Whether referenced or entered directly, interval values should use relative PI time strings such as a number followed by a time variable (for example, 1d or 30m), and may not specify or refer to a fixed PI or Excel timestamp. For example, to specify an interval of 32 minutes, enter or reference a cell containing the string 32m.
- To enter intervals in terms of frequency, convert the frequency to equivalent seconds. For example, a frequency of 25 Hz should be entered as a 0.04s interval (=1/25 of a second).
- When using relative formats (for example, -2h), the reference time used to translate the time format is different for function start and end times. The current wall clock time is the reference for a relative start time, but the start time is then used as the reference for a relative end time.

Note: PI DataLink supports only the default 1900 date system supported by Excel. PI DataLink does not support Excel's 1904 date system, and returns incorrect time stamps if this system is used.

Appended Data

Timestamps are particularly relevant to PI point events. Each event value is accompanied by a timestamp that gives it context.

Most PI DataLink functions can return corresponding timestamp values and other point data with PI event or calculated expression values. These data can be appended to values returned in a PI function array, and include:

- event timestamps
- timestamps indicating start and end times of an interval
- timestamps indicating the occurrence of minimum and maximum values
- the percentage of good values over a sampling interval
- value attributes
- manually entered event annotations
- source PI server names

When specified, appended data appear in columns (or rows) adjacent to the primary values returned by a function according to the following rules:

- Time data are added in columns to the left (or rows above) primary values.
- Other related data are added in columns to the right (or rows below) primary values.

Appended Data Array

The following table illustrates how data are appended per function. For functions that can display output in *rows or columns* (page 19), substitute Columns 1-5 with Rows 1-5.

	Function Array Columns				
Function/Mode	Output Cell	Column 2	Column 3	Column 4	Column 5
Any function without appended data	value				
Current and Archive Value, Sampled Data	timestamp	value			
Compressed Data	timestamp	value	annotations		
Time Filtered	start time	end time	value	pct good	
Calculated (minimum)	start time	end time	min time	min value	
Calculated (maximum)	start time	end time	max time	max value	
Calculated (range)	start time	end time	min time	max time	value
Calculated (other)	start time	end time	value	pct good	
Alias	server	value			

Note: The table assumes all appended data are specified for display, and that some function arrays can expand to five columns if all related data are appended.

Show Timestamps

A **show timestamps** check box appears when a function returns an array of PI values over time. If selected, this option returns a corresponding timestamp with each PI event value.

For functions that return multiple event values (for example, *Sampled Data*), the *orientation controls* (page 19) determine how timestamps are displayed in relation to each value.

For some functions that return a single value (for example, *Current Value*), radio buttons are provided to provide equivalent options:

- Select **no timestamp** to suppress the timestamp display.
- Select **time at left** to return a timestamp into a cell to the left of the event value.
- Select **time on top** to display the timestamp in a cell above the retrieved event value.

When a reference to an array of tag names in cells is specified in the **Tagname** field, these preferences are ignored. Timestamp and event values are instead positioned to match the orientation of the input tag array. For tag names in a single column, timestamps appear to the **left** of values. For tag names in a single row, timestamps appear **on top** of values.

Interval Timestamps

Functions that return values at specific time intervals (for example, *Calculated Data*) provide options to display different timestamps:

- Choose **show start time** to display the start time of each interval.
- Choose **show end time** to display the end time of each interval.

For calculated minimum, maximum, and range values:

- Choose **show min/max time** to display timestamps for minimum and maximum values over each interval.

Timestamps are displayed to match the specified row or column orientation. If all timestamps are shown, the order from left column to right or top row to bottom is start time, end time, minimum value time, maximum value time, event value.

Show Percent Good

Select **show percent good** to display the percentage of time for which good values are returned over the total time range of the array. Percent good values are typically added in a column to the right or a row below retrieved values.

Good values are event values determined to be valid by the PI server, and not in an error state. The percentage of good values helps in assessing the reliability of calculations built on PI point values, particularly if calculated values are to be used in further calculations.

For example, totals may still be calculated over a time range even if some of the data are bad. The resulting time-weighted total value equals the total divided by the fraction of the interval where good data is available.

However, this normalization assumes that the average over the time range with bad data is equivalent to the average value over the entire time range, which becomes less reliable if a large fraction of the time range contains bad data.

Show Value Attributes

Select **show value attributes** to display extended status bits associated with returned event values. The returned bits correspond to value attributes stored in PI, and may be one of the following:

- A = annotated, indicating a comment has been added to the event
- S = substituted, indicating the event value has been changed from its original value
- Q = questionable, indicating that there is some reason to doubt the accuracy of the value

Value attributes are displayed in a column to the right of event values.

Show Annotations

Select **show annotations** to display any annotations associated with returned event values in a column to the right.

Annotations are notes or comments appended to a data value in PI for descriptive purposes.

Orientation Controls

For functions that return an array of PI values, use the **column** and **row** buttons to determine the orientation of returned data.

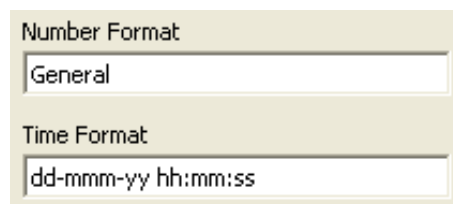
- Choose **column** to display data in columns.
- Choose **row** to display the data in rows.

Additional columns (page 16) or rows are added for timestamps and related data if they are selected for display. The function's **Output** field always indicates the upper left-hand corner of the entire range used for the array.

Note: For functions that return a range of values, the orientation of the input tag array overrides row and column selection to determine the orientation of output data.

Display Formats

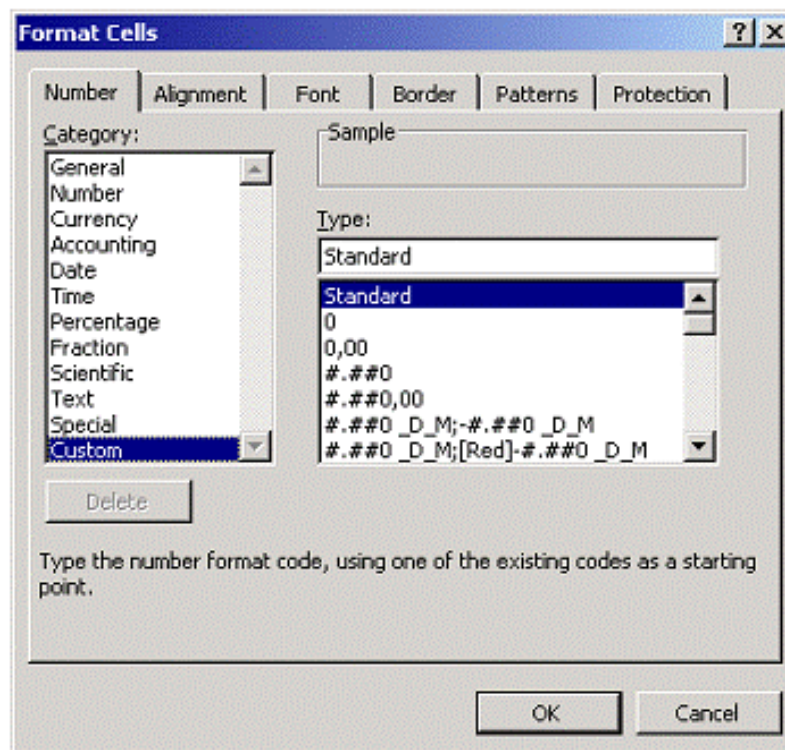
You can change the default time and number formats used by PI DataLink to format data in function arrays. Default formats are specified in the *Settings* (page 21) dialog box, where two settings are available for formats:



The screenshot shows a portion of a settings dialog box with a light beige background. It contains two sections. The first section is labeled 'Number Format' and has a text input field containing the word 'General'. The second section is labeled 'Time Format' and has a text input field containing the string 'dd-mmm-yy hh:mm:ss'.

- The default number format *General* formats numbers (and all non-timestamp data) to match the formatting **Category General** in the Excel **Format Cells** dialog box.
- The default time format *dd-mmm-yy hh:mm:ss.000* matches the standard PI timestamp format. You can add *.000* to the end of the string (*dd-mmm-yy hh:mm:000*) to display sub-second timestamps. Note that Excel does not support microsecond precision formats.

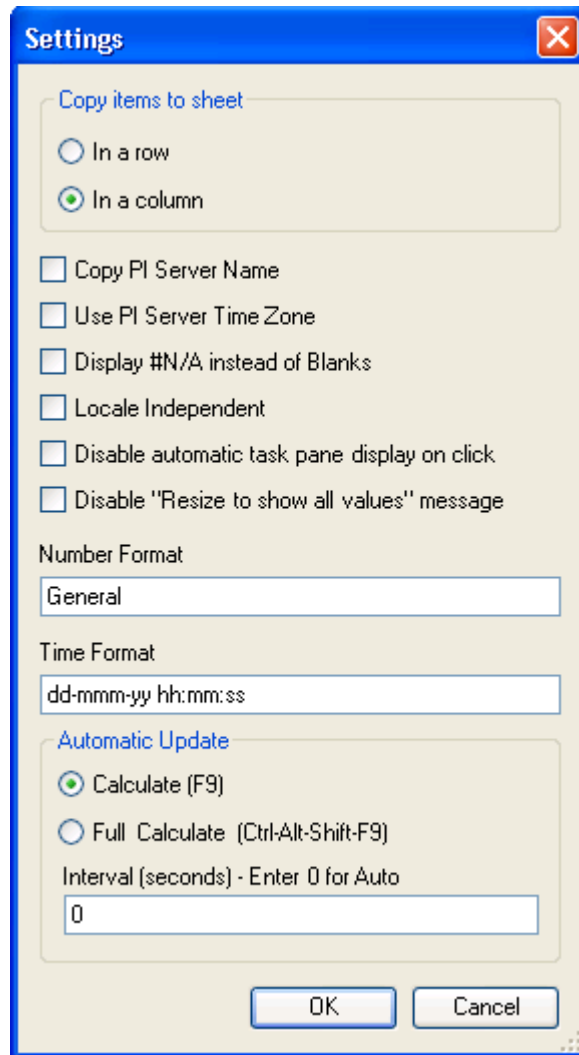
You can customize default format strings using any valid Excel format codes from the **Format Cells** dialog box in Excel. For example, German equivalents of *General* and *dd-mmm-yyyy hh:mm:ss* are *Standard* and *TT-MM-JJJJ hh:mm:ss*, respectively.



You can also apply individual time and date formatting to any spreadsheet cell, including those containing PI DataLink functions, by choosing **Format > Cells > Number**. See your Excel documentation for more information on formatting dates and times.

Preference Settings

Use the **Settings** dialog box to specify global preferences and default formatting for the output of PI DataLink functions.



To access settings:

1. Click **Settings** on the **PI** ribbon (2007), or choose **PI > Settings** (2003).
2. Set the desired preferences and click **OK**.

Copy Items to Sheet

Choose **In a row** or **In a column** to copy the names of multiple tags selected in a **Tag Search** (or items in a **Module Browse** search) to the spreadsheet in a row or column of values, respectively. The default setting is **In a column**.

This command is also available as a shortcut to the **Tag Search** ribbon item (2007).

Copy PI Server name

Select the checkbox to copy the name of the PI server data source into an adjacent spreadsheet cell when tag names are copied to a spreadsheet.

Use PI Server Time Zone

Select the checkbox to force PI DataLink to use the PI Server time zone as a reference value to interpret functions. If cleared, PI DataLink uses the time zone setting of the client machine.

This option is relevant only if a PI server has a time zone setting different from the client machine.

Display #N/A instead of Blanks

Select the checkbox to substitute #N/A (Not Applicable) for blank cells when there are fewer data points than cells in a function array. Functions added to a spreadsheet maintain their original array size, even if variance over a particular time period produces fewer data values, resulting in one or more empty cells.

This feature is particularly useful when function results are plotted through the Excel charting package.

Locale Independent

Select the checkbox to force PI DataLink to interpret input time strings according to PI time format rules, regardless of the locale settings of the client machine. PI time assumes all strings are English and use the date-time order *dd-MMM-yyyy hh:mm:ss*.

Clear the checkbox to parse date-time formats according to regional settings on the client machine, falling back to PI time format rules only if necessary.

Disable automatic task pane display on click

For Excel 2007 only, select this checkbox to prevent the automatic display of the function task pane when clicking in a function cell. Right-click to choose the function name and display the pane manually.

Disable "Resize to show all values" message

Select this checkbox to display no warning message when the number of values retrieved by a recalculated array exceed the number that can be displayed.

Number format

Enter a number format to indicate how numeric function output should be displayed by default. The format string may be any valid number format code from the **Excel Format dialog box** (page 19).

On install, PI DataLink assigns the default number format from Excel to this field, unless you have set your own format preference previously in DataLink. If your version of Excel runs a different locale, this format includes correct syntax to reflect the locale. For example:

35.03
would appear as
35,03
in French Excel.

Time format

Enter a timestamp format for function output. The time format string may be any valid *date-time format* (page 15) code from the **Excel Format** dialog box.

On install, DataLink assigns the default PI timestamp format to this field, unless you have set your own format preference previously in DataLink. If your version of Excel runs a different locale, this format includes correct syntax to reflect the locale. For example:

dd-mm-yy (03-07-09)

would appear as

jj-mm-aa (03-07-09)

in French Excel.

Automatic Update

Use this section to set preferences for **Automatic Update** (page 49) when this feature is enabled:

Choose a calculation method:

- **Calculate (F9)** forces all volatile functions (and any functions that reference them) to recalculate.
- **Full Calculate (Ctrl+Alt+Shift+F9)** forces all embedded functions, regardless of volatility, to recalculate.

Then enter a calculation interval in seconds. The minimum value is 5 seconds. Enter 0 to accept the automatic interval selected by PI DataLink based on duration of previous calculation times.

Spreadsheet Construction

Once you are familiar with the basic tools and concepts of PI DataLink, and are ready to build a spreadsheet and use PI DataLink functions, where do you begin?

Your goals

First consider your goals.

- What data do you want to display in a spreadsheet in order to monitor performance or answer a business question?
- In which PI server does the data reside?
- How can you most effectively display the information, both to communicate essential points and provide needed background context?

The answers will help you to determine which PI DataLink functions can return the data most relevant to your task.

Functional

One way to build a spreadsheet is simply to add *functions* (page 25), building the display around them as needed. Later on you can add tag information to help clarify what the data represents. This may be the best approach if you are still exploring your requirements or learning how PI DataLink works.

Structured

Another way to build a spreadsheet is to add structure from the beginning. Use *tag search and tag functions* (page 39) to add tags and other attributes as metadata to frame your subject, then add functions to retrieve the corresponding PI point data. This requires a bit more planning and familiarity with PI DataLink, but once tag information is present in the spreadsheet, you can use it to build functions more easily through cell references. Spreadsheets built in this manner are also more flexible for re-use.

Module-Driven

If you have a configured PI Module Database for your PI server, you can build maximum flexibility into your spreadsheet by incorporating objects from your *module database* (page 45). Like tag data, module database objects can be referenced in functions. You can also add module context features to maximize the potential re-use of the spreadsheet across parallel or redundant subjects.

Targeted Reports

If you are a system administrator, or a builder of spreadsheets for other users, you may want to place PI DataLink functions on a second spreadsheet in a workbook, and copy/paste the results to the first spreadsheet for display. Function syntax and business logic can be hidden and protected by using Excel to secure the second sheet. This strategy also works well for documents distributed through *PI Data Link Server* (page 3).

Chapter 3

PI Functions

PI DataLink functions enable you to query, calculate and return PI point values and attributes to spreadsheet cells. Like other Excel functions, PI DataLink function results are displayed in arrays which can be recalculated to display updated event values as needed.

To access PI functions:

- Click in the desired output cell, and then choose **PI > (function name)** to open a function dialog (*XP-2003*), or click a function on the **PI** ribbon to open a new function task pane (*2007*).

The function descriptions in this section describe arguments specific to each function in terms of these common features.

Current Value

Retrieves the current or most recent (snapshot) value of a PI point, based on tag name.

REQUIRED ARGUMENTS

- Tagname

SPECIAL NOTES

- Reference a range of cells containing tag names to display current snapshot values for each.
- Updates whenever Excel calculates or recalculates any cell in the spreadsheet. To force an immediate recalculation, press **F9**.

EXAMPLE

To see the current value of the `sinusoid` tag, the following arguments are set for the Current Value function:

Tagname = `sinusoid`

time at left

and the resulting function array appears as:

03-Oct-07 11:41:46	42.07747

RELATED TOPICS

- `PICurrVal()` (page 87)

Archive Value

Retrieves a PI point value, or evaluates an expression corresponding to a specified timestamp.

REQUIRED ARGUMENTS

- Tagname or Expression
- Timestamp

TIMESTAMP

The time reference used to retrieve the archive value from PI, which can be a fixed *time value* (page 15) or expressed relative to the current time.

RETRIEVAL MODE

One of five modes can be selected to produce an archive value:

- *Previous* - retrieves the value exactly matching or preceding the specified timestamp.
- *Previous only* - retrieves the value preceding the specified timestamp.
- *Interpolated* - interpolates the value at the specified time, or follows *Previous only* behavior.
- *Auto* - interpolates the value at the specified time, or follows *Previous* behavior.
- *Next* - retrieves the value exactly matching or following the specified timestamp.
- *Next Only* - retrieves the value following the specified timestamp.
- *Exact Time* - retrieves only a value exactly matching the timestamp, or returns *No events found* if no value exists.

SPECIAL NOTES

- Reference a range of cells containing tag names to display archived values for each.
- For tags with step attributes, *Previous* and *Previous only* modes are substituted for *Auto* and *Interpolated* calculation modes, respectively.
- `PIExpVal()` is used if an expression is substituted for a tag; otherwise `PIArcVal()` is used.

EXAMPLE

To see the value of the `sinusoid` tag corresponding to a specific time, the following arguments are set for the Archive Value function:

Tagname = `sinusoid`

Timestamp = `30-Sep-07 15:13`

Retrieval Mode = *previous*

time at left

and the resulting function array appears as:

30-Sep-07 14:39:16	99.18375

RELATED TOPICS

- `PIArcVal()` (page 88)
- `PIExpVal()` (page 89)

Compressed Data

Returns either all values of a PI point occurring within a specified time range, or a number of point values beginning at a certain time.

Compressed values are data recorded by the PI server after a compression algorithm has removed all values that represent the same slope.

Note: Separate dialog boxes are provided for Compressed Data (Start Time/End Time) and Compressed Data (Start Time/Number) in PI DataLink for Excel XP-2003. For 2007, you can select a Time Range or Number of Values calculation in the task pane.

REQUIRED ARGUMENTS

- Tagname
- Start Time
- End Time or Number of Values, depending on the function type

FILTER EXPRESSION

Add a filter expression to filter event values using a mathematical expression, eliminating data for which the expression evaluates as *false*. The filter expression is applied to the raw data from PI (and not values that result from the calculation itself).

For example, the simple filter expression:

```
sinusoid < 70
```

would remove all values over 70 from the calculation. You can also use any valid PI performance equation in the filter expression to build more complex expressions to remove atypical peaks in data values, for example.

Select **mark as filtered** to substitute *Filtered* as a placeholder for each value or block of values filtered from the array based on the filter *expression* (page 74).

BOUNDARY TYPE

Specify a boundary type to determine how the **Archive Editor** searches for data values are handled near the start and end times of the value range:

- **Inside (default):** Returns values at start and end times, if they exist, or the nearest values occurring within the range.
- **Outside:** Returns the closest values occurring immediately outside the range.
- **Interpolated:** Returns interpolated values at start and end times.
- **Auto:** Interpolated, but using **Inside** behavior for tags with step attributes set on Windows-based PI Servers.

COUNT OF VALUES

Normally the first row or column of the array includes the count of values retrieved for the time range. Select **hide count** to hide the count of values and show only retrieved values in the array.

SHOW VALUE ATTRIBUTES

Select **show value attributes** to display extended status bits associated with returned event values. The returned bits correspond to value attributes stored in PI, and may be one of the following:

- A = annotated, indicating a comment has been added to the event
- S = substituted, indicating the event value has been changed from its original value
- Q = questionable, indicating that there is some reason to doubt the accuracy of the value

Value attributes are displayed in a column to the right of event values.

SHOW ANNOTATIONS

Select **show annotations** to display any annotations associated with returned event values in a column to the right.

Annotations are notes or comments appended to a data value in PI for descriptive purposes.

SPECIAL NOTES

- Reference a range of tag names to display compressed values for each. Note that by default, values are displayed in rows for a column of referenced tags, and in columns for a row of referenced tags.
- If the Start or End Time is the current time (*), then the first (or last) value is actually the current snapshot value, and not an archive value. Once the snapshot value passes through the PI Server's compression algorithm, it may not actually be recorded in PI.

- The `PICompFilDat()` function is used for End Time functions if a filter expression are specified; otherwise, the `PICompDat()` function is used.
- The `PINCompFilDat()` function is used for Number of Values functions if a filter expression is specified; otherwise, the `PINCompDat()` function is used.

EXAMPLE

To see the most recent 10 values of the `sinusoid` tag, the following arguments are set for the Compressed Data function:

Number of Values

Tagname = `sinusoid`

Start Time = <cell reference to current time>

Number of Values = 10

backwards in time

Boundary Type = *inside*

show timestamps

column

and the resulting function array appears as:

03-Oct-07 12:39:16	66.79971
03-Oct-07 10:47:16	20.35228
03-Oct-07 09:35:46	2.415744
03-Oct-07 08:36:46	1.024204
03-Oct-07 07:33:16	13.6514
03-Oct-07 04:40:46	81.88248
03-Oct-07 03:31:16	98.15032
03-Oct-07 02:31:16	98.43637
03-Oct-07 01:31:16	85.74396
02-Oct-07 22:57:46	24.16043

Note that all events between two specified times could also be retrieved by the same function using the **Time Range** argument.

RELATED TOPICS

- *Filter Expressions* (page 74)
- `PICompDat()` (page 91)
- `PICompFilDat()` (page 92)
- `PINCompDat()` (page 89)
- `PINCompFilDat()` (page 90)

Sampled Data

Returns evenly-spaced, interpolated sample values for a PI point or expression over a regular interval.

REQUIRED ARGUMENTS

- Tagname or Expression
- Start Time
- End Time
- Time Interval

TIME INTERVAL

Enter a time interval as a *PI time* (page 71) expression. The interval is the sampling frequency used to collect or calculate values over the course of the time range.

For example, an interval of 15m (15 minutes) returns a value for every 15 minute interval.

FILTER EXPRESSION

Add a filter expression to filter event values using a mathematical expression, eliminating data for which the expression evaluates as *false*. The filter expression is applied to the raw data from PI (and not values that result from the calculation itself).

For example, the simple filter expression:

```
sinusoid < 70
```

would remove all values over 70 from the calculation. You can also use any valid PI performance equation in the filter expression to build more complex expressions to remove atypical peaks in data values, for example.

Select **mark as filtered** to substitute *Filtered* as a placeholder for each value or block of values filtered from the array based on the filter *expression* (page 74).

SPECIAL NOTES

- You can use a *PI Expression* (page 74) instead of a tag name for this function.
- Reference a range of tag names to display sampled values for each. Note that by default, values are displayed in rows for a column of referenced tags, and in columns for a row of referenced tags.
- The `PI SampFilDat()` function is used if a filter expression is specified; otherwise, the `PI SampDat()` function is used.
- `PI ExpDat()` is used if a PI Expression is substituted for a tag.

EXAMPLE

To see periodic, interpolated values over the previous 24 hours for the `sinusoid` tag, the following arguments are set for the Sampled Data function:

Tagname = `sinusoid`
Start Time = `-1d`

End Time = <cell reference to current time>

Time Interval = 3h

show timestamps

column

and the resulting function array appears as:

02-Oct-07 13:57:08	91.09476
02-Oct-07 16:57:08	75.72464
02-Oct-07 19:57:08	9.009873
02-Oct-07 22:57:08	23.98436
03-Oct-07 01:57:08	91.21608
03-Oct-07 04:57:08	75.40824
03-Oct-07 07:57:08	8.905168
03-Oct-07 10:57:08	24.27547

where a value is displayed for each 3-hour interval.

RELATED TOPICS

- `PI SampDat()` (page 92)
- `PI SampFilDat()` (page 93)
- `PI ExpDat()` (page 94)

Timed Data

Returns actual or interpolated sample values for a PI point to match an array of specified timestamps.

REQUIRED ARGUMENTS

- Tagname
- Timestamp

TIMESTAMP

A reference to one or more spreadsheet cells containing timestamp values.

RETRIEVAL MODE

There are two modes to select from:

- *Interpolated* - interpolates values corresponding to specified timestamps.
- *Exact Time* - retrieves only values exactly matching specified timestamps, or returns *No events found* if no value exists.

SPECIAL NOTES

- You can use a *PI Expression* (page 74) instead of a tag name for this function.
- Reference a range of tag names to display timed values for each. Note that by default, values are displayed in rows for a column of referenced tags, and in columns for a row of referenced tags.
- For tags with step attributes, the value preceding a specified timestamp is substituted for an interpolated value when the calculation mode is *Interpolated*.

EXAMPLE

To see values corresponding to an array of timestamps for the `sinusoidu` tag, the following arguments are set for the Timed Data function:

Tagname = `sinusoidu`

Timestamps = <cell reference to existing array of compressed data for `sinusoid` tag>

Retrieval Mode = *interpolated*

and the resulting function array appears as:

03-Oct-07 13:01:46	75.66432		7.194396
03-Oct-07 10:47:16	20.35228		55.57911
03-Oct-07 09:35:46	2.415744		83.05303
03-Oct-07 08:36:46	1.024204		96.86302
03-Oct-07 07:33:16	13.6514		98.42677
03-Oct-07 04:40:46	81.88248		41.41196
03-Oct-07 03:31:16	98.15032		16.77808
03-Oct-07 02:31:16	98.43637		3.110793
03-Oct-07 01:31:16	85.74396		2.372
02-Oct-07 22:57:46	24.16043		51.6191

where the column of timed data appears in the column to the right, based on timestamps retrieved for a different tag in the first two columns at left.

RELATED TOPICS

- `PITimeDat()` (page 95)
- `PITimeExpDat()` (page 96)

Calculated Data

Returns one or more evenly-spaced, calculated values based on PI point values or evaluated expressions. A range of different calculation and sampling preferences are provided.

Note: Separate dialog boxes are provided for *Calculated Data* and *Advanced Calculated Data* in PI DataLink for Excel XP-2003. For the Excel 2007 version, click the + sign next to **Advanced** to access all function options in a single *Calculated Data* task pane.

REQUIRED ARGUMENTS

- Tagname or Expression
- Start Time
- End Time
- Conversion Factor

FILTER EXPRESSION

Add a filter expression to filter event values using a mathematical expression, eliminating data for which the expression evaluates as *false*. The filter expression is applied to the raw data from PI (and not values that result from the calculation itself).

For example, the simple filter expression:

```
sinusoid < 70
```

would remove all values over 70 from the calculation. You can also use any valid PI performance equation in the filter expression to build more complex expressions to remove atypical peaks in data values, for example.

Select **mark as filtered** to substitute *Filtered* as a placeholder for each value or block of values filtered from the array based on the filter *expression* (page 74).

TIME INTERVAL

Enter a time interval as a *PI time* (page 71) expression. The interval is the sampling frequency used to collect or calculate values over the course of the time range.

For example, an interval of 15m (15 minutes) returns a value for every 15 minute interval.

CALCULATION MODE

The following calculations are provided:

- *Total*
- *Minimum*
- *Maximum*
- *Standard deviation*
- *Range*
- *Average*

- *Count*
- *Mean*

All calculation modes are time-weighted except for the *Mean* function, which is an event-weighted version of the *Average* function.

CALCULATION BASIS

Select either a time- or event-weighted basis for the calculation:

- Calculations are time-weighted by default. Each event value is weighted for purposes of the calculation by the duration of time over which the event applies.
- Event-weighted calculations weigh each event value equally. At least one event (two for standard deviation calculations) must exist within a time range for a successful calculation.

CONVERSION FACTOR

The conversion factor is used for Total calculations where the time unit of the point calculated is not the same as the server default time unit. A conversion factor may be necessary to convert to the units desired for the calculation from the server default of units per day.

The following table illustrates the use of conversion factors for points using different units of flow. Note that a conversion factor of one means that the conversion factor is not needed, and that units cannot be specified.

Flow unit	Conversion Factor needed to match server default (units/day)
Units/day	1.0
Units/hour	24
Units/minute	1440
Units/second	86400

EXPRESSION SAMPLING MODE

Different sampling modes are available for functions that include expressions:

- For PI tag calculations, select one of three sampling modes.
 - *Point Compressed* mode is the default used for PI tag calculations. This mode considers the timestamps for the PI tag for which the calculation is being performed. It does not consider timestamps for other PI tags used in the filter expression. This mode performs the calculation using the time intervals between timestamps that the filter expression evaluates to true. The *Point Compressed* mode is not as accurate as the *Expression Compressed* mode, but it remains the default to ensure compatibility with earlier versions of the PI Server.
 - *Expression Compressed* mode considers the timestamps for PI tags used in the filter expression. It does not consider the timestamps for the PI tag for which the calculation is being performed unless that PI Tag happens to be used in the filter expression. This mode determines the intervals for which the filter expression is true and then performs the calculation using the interpolated value of interval endpoints.

The *Expression Compressed* mode produces more accurate results than the *Point Compressed* mode.

- *Interpolated* mode evaluates the expression at evenly-spaced sampling intervals determined by the Expression Sampling Frequency.
- For PI expression calculations, select one of two sampling modes:
 - *Compressed* mode evaluates the expression for each event value at corresponding timestamps.
 - *Interpolated* mode evaluates the expression at evenly-spaced sampling intervals determined by the Expression Sampling Frequency.

EXPRESSION SAMPLING FREQUENCY

Specify the frequency at which a calculation or filter expression is evaluated when the **Expression Sampling Mode** is set to *Interpolated*.

For example, a frequency of *10m* (10 minutes) returns an interpolated value for every 10 minute interval while the filter expression evaluates as *true*.

MINIMUM PERCENT GOOD

Specify the minimum *percentage of good data* (page 18) required in each time range to calculate and return a value. *Insufficient good data* is substituted as a placeholder when a value is not returned.

TIMESTAMPS

- Select **show start time** to display the Start time of each interval used to calculate a value. Start times are added in a column to the left or a row above calculated values.
- Select **show end time** to display the End time of each interval used to calculate a value. End times are added in a column to the left or a row above calculated values.
- For Maximum, Minimum and Range calculations, select **show min/max time** to display timestamps corresponding to minima and/or maxima over each interval used to calculate a value:
 - For Minimum and Maximum calculations, the timestamp of the corresponding value is displayed.
 - For Range functions, both timestamps are displayed.

PERCENT GOOD

Select **show percent good** to display the percentage of time for which good values are returned over the total time range of the array. Percent good values are typically added in a column to the right or a row below retrieved values.

Good values are event values determined to be valid by the PI server, and not in an error state. The percentage of good values helps in assessing the reliability of calculations built on PI point values, particularly if calculated values are to be used in further calculations.

SPECIAL NOTES

- You can use a *PI Expression* (page 74) instead of a tag name for this function.
- Reference a range of tag names to display calculated values for each. Note that by default, values are displayed in rows for a column of referenced tags, and in columns for a row of referenced tags.
- For Maximum or Minimum calculations with specified intervals, timestamps indicate the start time of each calculation interval.
- Timestamps cannot be displayed for Total, Standard Deviation, Average, Count or Mean calculations when the interval is unspecified.
- The `PIAdvCalcVal()` function is used to retrieve a single value if you do not specify an interval; otherwise the `PIAdvCalcDat()` function is used to retrieve multiple interval values. Function names including 'Exp' and 'Fil' are used for calculations that include PI Expressions and Filter Expressions, respectively.

EXAMPLE

To see the hourly range of values beginning the same day at midnight until the present time for the `sinusoid` tag, the following arguments are set for the Calculated Data function:

PI Tag

Tagname = `sinusoid`

Start Time = `t`

End Time = `*`

Time Interval = `1h`

Calculation Mode = `range`

Calculation Basis = `time-weighted`

show start time

show end time

column

and the resulting function array appears as:

03-Oct-07 00:00:00	03-Oct-07 01:00:00	24.07173679
03-Oct-07 01:00:00	03-Oct-07 02:00:00	18.62230522
03-Oct-07 02:00:00	03-Oct-07 03:00:00	6.614157613
03-Oct-07 03:00:00	03-Oct-07 04:00:00	6.874663229
03-Oct-07 04:00:00	03-Oct-07 05:00:00	17.14983817
03-Oct-07 05:00:00	03-Oct-07 06:00:00	23.73254958
03-Oct-07 06:00:00	03-Oct-07 07:00:00	23.73254958
03-Oct-07 07:00:00	03-Oct-07 08:00:00	18.47439853
03-Oct-07 08:00:00	03-Oct-07 09:00:00	7.311182223
03-Oct-07 09:00:00	03-Oct-07 10:00:00	6.922760253
03-Oct-07 10:00:00	03-Oct-07 11:00:00	17.07054751
03-Oct-07 11:00:00	03-Oct-07 12:00:00	24.56483508
03-Oct-07 12:00:00	03-Oct-07 13:00:00	24.56483508

where the range is calculated for each one hour interval. Note that any available calculation could be specified over any interval.

RELATED TOPICS

- *Calculation Expressions* (page 74)
- `PIAdvCalcVal()` (page 98)
- `PIAdvCalcFilVal()` (page 99)
- `PIAdvCalcExpVal()` (page 100)
- `PIAdvCalcExpFilVal()` (page 101)
- `PIAdvCalcDat()` (page 103)
- `PIAdvCalcFilDat()` (page 104)
- `PIAdvCalcExpDat()` (page 105)
- `PIAdvCalcExpFilDat()` (page 106)

Time Filtered

Returns the amount of time over which a PI expression evaluates as *true* for a specified time range.

Note: Time Filtered results vary slightly depending on your PI Server version.

REQUIRED ARGUMENTS

- Expression
- Start Time
- End Time
- Time Units

TIME INTERVAL

Enter a time interval as a *PI time* (page 71) expression. The interval is the sampling frequency used to collect or calculate values over the course of the time range.

For example, an interval of 15m (15 minutes) returns a value for every 15 minute interval.

TIME UNITS

Specify the units of time used to display the result of the calculated expression.

SPECIAL NOTES

- You can enter the expression in a spreadsheet cell and reference the cell name in the function. Begin expressions with a single quote if you want Excel to interpret the expression as a string.

EXAMPLE

To see the amount of time for which the value of the `sinusoid` tag was over 75 over a 7-day period, the following arguments are set for the Time Filtered function:

Expression = `sgn('sinusoid'-75)=1`

Start Time = -7d

End Time = *

Time Units = h

show percent good

column

and the resulting function array appears as:

7.853746	100	

where the results show the tag value was over 75 for 7.85 hours over the previous 7 days.

RELATED TOPICS

- *Calculation Expressions* (page 74)
- `PITimeFilterVal()` (page 108)
- `PITimeFilter()` (page 107)

Chapter 4

PI Tags and Attributes

A PI point is a stream of real-time data from a defined source, and is described by a corresponding tag name and other attributes.

In PI DataLink and this guide, PI points are frequently referred to as PI tags, and the terms are used somewhat interchangeably. However, a tag is simply a name for a PI point. When you retrieve PI data into an Excel spreadsheet, the tag name is the most commonly-used PI attribute to refer to data from a PI point.

PI tag names can help identify point data, describe the source of function data values and arrays, and be referenced as attributes to retrieve data. This section describes the use of tags and other point attributes in building a PI DataLink spreadsheet, including both:

- direct retrieval of tags into spreadsheet cells, and
- functions used to resolve tag names from other point attributes, or retrieve attributes based on a tag name.

Tag Search

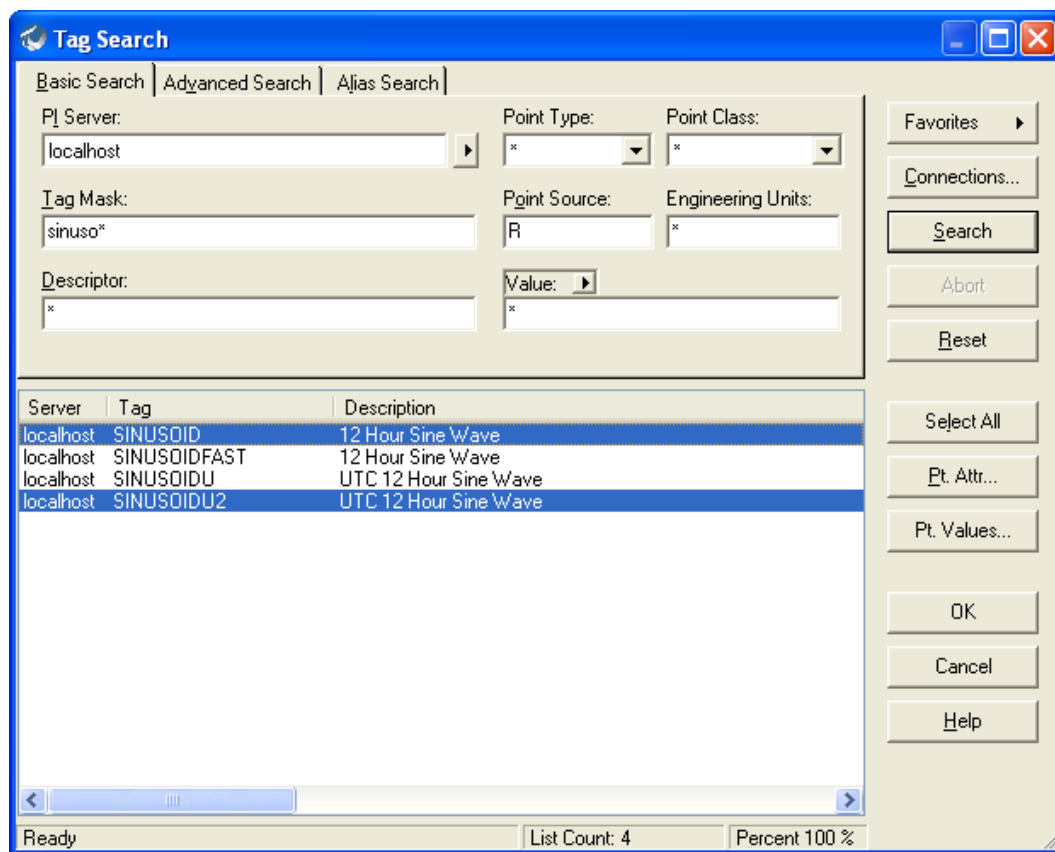
Use the **Tag Search** dialog box to query the PI server directly for PI tag names to copy into functions and spreadsheets. Selected tags are copied to the spreadsheet, or added to a **Tagname** edit field as a function argument.

To access tag search:

- Click **Tag Search** on the PI ribbon (2007), or choose **PI > Tag Search** (2003), or click **Tag Search**  next to a **Tagname** field.

The **Tag Search** dialog box provides three types of searches:

- **Basic Search** allows you to create a tag mask by specifying PI point attributes. The mask is used to find a list of tags on the server with matching attributes.
- **Advanced Search** provides a query-building interface with access to more point attributes for complex searches.
- **Alias Search** provides a logical tree view of a PI Server through the PI Module Database, which you can use to select tags by their descriptive aliases.



To search for tags:

1. Click a tab to choose a **Basic**, **Advanced** or **Alias** search.
2. Enter the required search criteria and click **Search**.

Use '*' or '?' as wildcard characters to search for tag names and attributes. For example, the tag mask *Tem** returns all point names that start with *Tem* while *Tem?* returns only points that start with *Tem* and end with another single character. All point mask fields are case insensitive.

You can also click **Favorites** to access previous searches.

3. Tags returned from a search appear listed in a search results panel. Select the desired tags in the results panel, and click **OK**.

Click column headers in the search results panel to sort the results. Ctrl-click or Shift-click to select multiple tags.

Tag Search Tips

- If multiple tags are selected, the first selected tag name in the list is copied to the **Tag name** field for a function reference.
- The current cell is taken as the starting point for a list of tags copied into a spreadsheet along a single column or row, depending on the **Copy items to sheet** setting.

- Choose **In a row** or **In a column** from the Tag Search shortcuts on the PI ribbon menu (2007) to copy the names of multiple tags to the spreadsheet in a row or column of values, respectively.
- The PI Server name corresponding to a tag is appended to the tag name if the **Copy PIServer Name** setting is enabled.

Tag Functions

The following section describes PI DataLink tag functions. These functions can be used to resolve PI point IDs or other point attributes into a matching tag name. You can also retrieve other point attribute values based on a tag name.

To access tag functions:

- Click in the desired output cell, and then choose **PI > (function name)** to open a function dialog (XP-2003), or click a function on the **PI** ribbon to open a new function task pane (2007).

Tag functions use common *function components* (page 13).

Point ID to Tag

Returns the tag name that corresponds to a specific PI point ID.

REQUIRED FIELDS

- Point ID

EXAMPLE

To retrieve the name of a tag corresponding to a PI point ID, the following arguments are set for the Point ID to Tag function:

Point ID = (reference to a cell containing a point ID value)

and the resulting function array appears as:

1	SINUSOID

where the tag name is displayed in the column to the right, based on a reference to the cell containing the point ID on the left.

RELATED TOPICS

- PIPointIDToTag () (page 109)

Attribute Mask to Tag

Returns a list of tag names in PI that match one or more specified tag attribute values.

REQUIRED FIELDS

One or more of the following fields is required to retrieve matching values:

- Tagname
- Descriptor
- Point Source
- Point Type
- Point Class
- Engineering Units

EXAMPLE

To retrieve the names of tags matching one or more specified point attributes, the following arguments are set for the Attribute Mask to Tag function:

Tagname = BA*

Point Type = Float32

and the resulting function array appears as:

	BA:CONC.1
	BA:LEVEL.1
	BA:TEMP.1

Note that attribute strings must match attribute values in PI, and wildcards can be used. The function operates the same as a *tag search* (page 39).

RELATED TOPICS

- `PIAttributeMaskToTag ()` (page 109)

Tag Attributes

Retrieves an attribute associated with a specified PI tag name.

REQUIRED FIELDS

- Tagname
- Attribute

ATTRIBUTE

Specify the tag attribute to be retrieved from PI.

SPECIAL NOTES

- Reference a range of tag names to display attributes for each.
- The list of tag attributes available depends on the point class associated with the tag.

EXAMPLE

To retrieve attributes of a PI point based on the tag name, the following arguments are set for the Tag Attributes function:

Tagname = (reference to a cell array containing tag names)

Attributes = *pointid* (column 2), *pointsource* (column 3), *pointtype* (column 4), *descriptor* (column 5)

and the resulting function arrays appear as:

BA:CONC.1	8 9	Float32	Concentration Reactor 1	
BA:LEVEL.1	7 9	Float32	Level Reactor 1	
BA:TEMP.1	6 9	Float32	Temperature Reactor 1	

where the tag names are referenced in the column to the left, and a separate Tag Attributes function is used for each attribute column.

RELATED TOPICS

- `PITagAtt()` (page 110)

Chapter 5

Module Database Objects

The PI Module Database provides an entirely different way to build the infrastructure of a spreadsheet. If your PI server's module database is configured and includes objects that represent the tags you want to use, your spreadsheets can be easier to build and provide more flexibility and potential for re-use.

The PI Module Database is typically configured by your system or PI administrator, and stores aliases and properties that refer to PI points and their attributes, respectively. The hierarchical structure of these objects provides a relational context between points.

For example, you may have five PI points that represent the levels of five individual, identical tanks. Each point has a different tag name and represents a different physical object, but their usage and monitoring requirements are the same.

By retrieving objects from the module database, you can create a single spreadsheet built on the aliases and properties that represent all five tanks. Module database functions resolve these objects into tag names and attribute values. Parallel module paths for each tank provide a different context for the alias references.

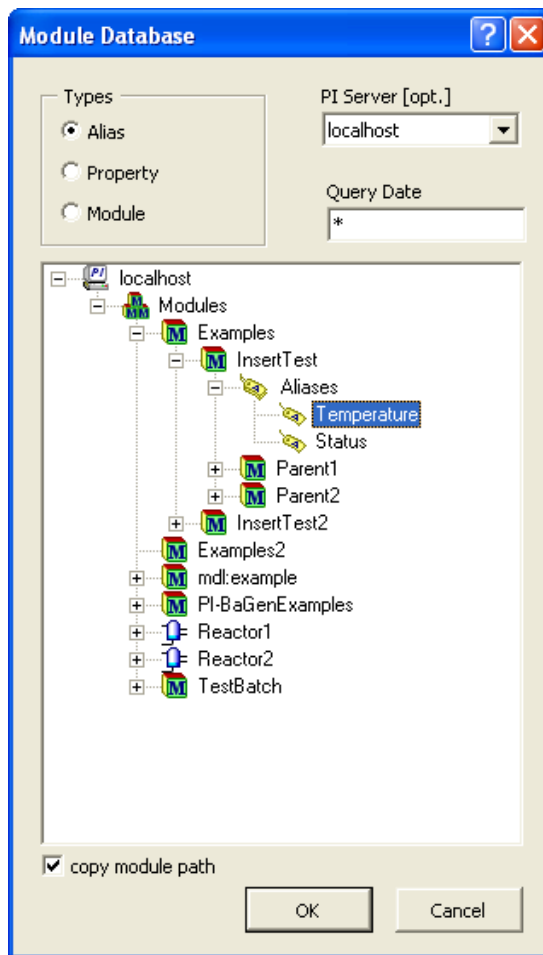
With this type of structure, you can use the same spreadsheet to alternate between views of multiple assets. For instance, when you select the name of any of the five tanks in a spreadsheet cell, the alias mapping to the matching PI point allows DataLink functions to retrieve values for the appropriate tank.

The following topics describe the tools provided by PI DataLink to access PI Module Database objects. See your PI Server documentation for more information about module databases and configuration.

Module Database Browse

The Module Database Browse dialog box allows you to browse for PI aliases, properties and module paths in the PI Module Database, a hierarchical tree structure based on the contents of a PI Server. You can select these items and copy them directly into a spreadsheet.

Note: Aliases and properties corresponding to PI points must be configured in your PI server Module Database to use the browse and Module Database functions.



To browse and select from the PI Module Database:

1. Select a cell and click **Module Browse** on the **PI** ribbon, or choose **PI > Module Browse**. The Module Database Browse dialog box appears.
2. Under **Types**, select the type of object you are looking for:
 - o Alias
 - o Property
 - o Module

Use the **PI Server** drop-down list to display module databases for different connected PI servers.

3. Specify a **Query Date** and time to supply a temporal context for objects displayed in the module database, if necessary, to select the correct object version.
4. Browse the tree control to find the desired item, and add it or a parent collection object to the spreadsheet:
 - o Select an item to copy it to the spreadsheet.
 - o Select a module's *Aliases* or *Properties* parent to copy all associated aliases or properties to the spreadsheet.

5. Select **Copy module path** to copy the full module path of the alias or property to an adjacent spreadsheet cell, along with the selected object.

The module path provides the context for the selected object

If the **Copy PIServer Name** setting is activated, the corresponding PI server name is also copied.

6. Click **OK** to close the dialog box and copy the selected items.

Module Database Functions

The following section describes PI DataLink module database functions. Module Database functions resolve aliases and properties from a module database tree into tag names through a module path. You can enter a module path as a function argument or reference it from a spreadsheet cell.

To access module database functions:

- Click in the desired output cell, and then choose **PI > (function name)** to open a function dialog (*XP-2003*), or click a function on the **PI** ribbon to open a new function task pane (*2007*).

Note: You cannot resolve an alias or property object into a tag name without a module path. The path provides a context for the alias that maps it to a specific PI point.

Module Database functions use common *function components* (page 13).

Alias to Tag

Returns a PI tag name corresponding to a specified PI alias.

REQUIRED ARGUMENTS

- Alias
- Module Path
- Query Date

ALIAS

Specify a PI alias name. An alias can be entered directly or referenced in a cell. You can also browse to an alias in the PI Module Database.

MODULE PATH

Enter the module path from the PI Module Database where the specified alias can be found. The module path is entered automatically if the alias argument is selected by browsing the Module Database.

QUERY DATE

Specify a date to retrieve a specific version of the alias.

RELATED TOPICS

- `PIAliasToTag()` (page 111)

Property to Value

Returns the value or values corresponding to a specified PI property.

DataLink can return values for Boolean, Date, Double, Integer, Long, Null, Single and String property types, or Double, Integer, Long and Single array property types.

REQUIRED ARGUMENTS

- Property
- Module Path
- Query Date

PROPERTY

Specify a PI property name. A property can be entered directly or referenced in a cell. You can also browse to a property in the PI Module Database.

MODULE PATH

Enter the module path from the PI Module Database where the specified property can be found. The module path is entered automatically if the property argument is selected by browsing the Module Database.

QUERY DATE

Specify a date to retrieve a specific version of the property.

RELATED TOPICS

- `PIPropertyToValue()` (page 111)

Spreadsheets

When you add a function to a spreadsheet, PI DataLink retrieves the requested data from the PI Server into an array. The size of the array can vary depending on the function, how many matching archive values are available in PI, the number of values requested, or the number of input references.

The following topics provide an overview of how PI DataLink functions work within a spreadsheet, and how to manage and maintain your spreadsheet over time.

Calculation Frequency

You can choose an automatic update interval to keep function data current in your spreadsheets, or use function-triggered and manual recalculation methods used in previous versions of PI DataLink.

For detailed information on how Excel manages calculation, see the *Excel Recalculation* (<http://msdn.microsoft.com/en-us/library/bb687891.aspx>) topic in the MSDN library.

Automatic Update

Automatic Update forces Excel to recalculate functions and trends on open worksheets at a specified interval. To activate Automatic Update:

- Click the **Update** toggle button on the PI Ribbon (2007), or choose **Update** from the PI menu (2003).

All workbooks open in the current Excel session are calculated immediately, and then at specified intervals based on your Automatic Update preference settings.

Automatic Update remains activated (with an indicator message in the Excel status bar) until it is toggled off. During a recalculation event the status bar also displays the calculation interval.

Note: Automatic Update cannot update protected workbooks, which are read-only.

Calculation Interval

Calculation interval is a *preference setting* (page 21) that is measured in seconds, and must be greater than 5 seconds.

A value of 0 represents an automatic interval in which DataLink adjusts the time between calculation cycles to twice the calculation duration time, with a minimum interval of 5

seconds. The automatic interval is recommended to insure that Excel remains responsive 50% of the time.

You can adjust the Automatic Update interval manually to allow more time between updates. If recalculation takes longer than your specified interval, a message prompts to switch to the automatic interval setting, or to turn off Automatic Update entirely.

Note that updates are paused when a menu, task pane or dialog box is open or the current spreadsheet is in edit mode.

Cancellation

Press **Esc** to cancel a recalculation in process. DataLink finishes calculation for the currently-processing function, and *Calculation aborted* appears in unprocessed function array cells. If you cancel during an update while Automatic Update is on, DataLink turns Automatic Update off.

Triggered Recalculation

In Excel, functions recalculate based on a triggering event. If you do not use the **Automatic Update** functionality, you can maximize the refresh frequency and keep function arrays current by referencing volatile time functions within non-volatile PI DataLink functions.

Volatile Excel time functions such as `now()` and `today()` recalculate with the most frequency. Whenever a user edits a spreadsheet cell, or presses **F9**, Excel updates all volatile time functions in the spreadsheet. The update in turn triggers any functions that reference a timestamp based on a volatile time function.

To base a PI DataLink function on a volatile Excel function:

1. Use the Excel function bar to enter a volatile function in a spreadsheet cell.

For example, you can use `(today()+1/3)` to represent 8 am the same day, or `now()` as a cell reference to replace the current PI time *.

To create an absolute timestamp that updates at the same time, use `2-feb-97 00:30:30 + now() - now()`.

2. Reference the cell when defining the **Start Time** or **End Time** arguments of a DataLink function.

Note: To maximize the frequency of updates when Automatic Update is not in use, check your Excel calculation preferences to ensure that both the spreadsheet and application are set to *Automatic* rather than *Manual* calculation. Keep in mind that this may impact performance and make large spreadsheets difficult to use.

Manual Recalculation

Most PI DataLink functions are non-volatile, but the **Current Value** (page 25) function is an exception. **Current Value** is a volatile function, and updates whenever a spreadsheet recalculates. The values of non-volatile functions do not change unless an argument changes,

and must be updated manually, through automatic update using **Full Calculate** mode, or through a reference to a volatile function value.

Manual recalculation is the best method for *variable-size arrays* (page 53) where the number of values returned may differ. If more values are available than the current size of the array, an array does not expand to display the additional values on recalculation unless it is manually recalculated.

To manually recalculate a single function:

- Right-click in any part of an array and choose **Recalculate (Resize) Function** from the context menu.
- Update function arguments in either version of PI DataLink. The array values refresh when you click **OK** or **Apply** to close the dialog box or task pane, even if no changes are made to function arguments.

Excel key combinations can also be used to recalculate all spreadsheets in a workbook, essentially a manual, on-demand version of Automatic Update. To recalculate all spreadsheet functions:

- Press **F9** to force all volatile functions (and any functions that reference them) to recalculate.
- Press **Ctrl+Alt+Shift+F9** to force all functions to recalculate.

Functions and Array Values

When you add a function to a spreadsheet, PI DataLink enters the active function syntax into the target cell. Once defined, a function queries the PI server and returns values, which appear in the same cell used to enter the function.

The difference between the actual content of cells and the surface display of data values retrieved by a function should be familiar to users of Microsoft Excel. For new users of Excel, it helps to keep this distinction in mind, and remember that the actual content of a spreadsheet cell always appears in the Excel formula bar, and not necessarily in individual spreadsheet cells.

For example, when you select a cell and add a **Current Value** function, you specify the name of the PI tag whose current value you want to see in the **Tagname** field. If you enter the *sinusoid* tag, PI DataLink pastes the following function into the target cell:

```
=PICurrVal("sinusoid",0,)
```

Excel evaluates this function, retrieves the latest value of *sinusoid* from PI, and displays it in the cell. If you select the cell, the function syntax above appears in the Excel formula bar.

Array Management

Function arrays can be moved within, copied, or removed from a spreadsheet. To select an array, do one of the following:

- Click and drag to select every cell in the array. Keep in mind that the array may have empty values, which must also be selected.
- Right-click on any part of the array and choose **Select DataLink Function** from the context menu to select all cells in the array.

Once selected, you can easily manage the array:

- Right-click any cell in the array and choose the desired **Cut**, **Copy**, **Delete**, **Clear** or **Format** operation from the context menu. Manually remove initial \$ characters in the formula bar to make cell references relative.
- Right-click in a new cell and choose **Paste** to reinsert a cut or copied array.
- Place the cursor over the edge of the selected array until a crosshair cursor appears, then click and drag the array to a new cell to move it.
- Open the function dialog box or task pane and *update the Output cell* (page 52) to place the array in a new location.

Note: You must move or clear an entire Excel array including all cells, or DataLink displays an error message: *You cannot change part of an array.*

You can also copy the values from a function array if you want to use them elsewhere in a spreadsheet. Once you copy and paste the values, they are no longer part of a function array; they are simply cell values and are not updated. To copy array values:

- Select the function values you want, copy them, and use **Paste Special** to paste them in the desired location.

You do not need to select the entire array column or row to copy values.

Update Arguments

You can update a function array with new argument values:

1. Right-click and select the function name from the context menu (*2003* or *2007*), or display the function dialog box or task pane.

The corresponding function dialog box appears and displays the current arguments in the appropriate edit fields.

2. Change any parameters as desired to update the array with new arguments, and click **OK** or **Apply**.

If the **Output Cell** argument is changed, the results depend on the location of the new cell reference:

- If the new cell reference is part of the original array, then the entire array is moved such that the upper-leftmost cell of the array is placed in the newly-referenced cell.
- If the new cell reference is not part of the original array, then the array is copied and pasted with the upper-leftmost cell located in the newly-referenced cell. The original array remains in its original location.

Note that you can also edit function syntax directly by pressing **F2** to edit in the Excel formula bar. Press Ctrl+Shift+Enter to save the edits when complete.

Resize Arrays

When a spreadsheet is called up at different times, a different number of values may be available for a defined function. If the number of values exceeds the current size of the array, the array does not expand to display the additional values unless you *manually recalculate* (page 50) the function.

If fewer than the original number of retrieved values exist, empty values are returned to fill the recalculated array, and are marked #N/A if specified, or left blank. If additional values exist that cannot be displayed, the text *Resize to show all values* appears at the bottom of the function array, unless you set a preference to *disable* (page 21) this message.

To resize the array to fit available data:

- Right-click in any part of an array and choose **Recalculate (Resize) Function** from the context menu.

Empty values are typically added when a function references a time range that extends to the current time, using relative start and end times. When the function is recalculated, the time range covers a new interval which may include more data variance, and thereby a different number of recorded events in the archive.

For example, consider a compressed data function created to request 40 values beginning at midnight the previous day. When the function is recalculated a week later, and only 35 event values (including filtered events) are recorded, the final five cells of the original array are filled with blank strings to preserve the array size.

Empty values may change the appearance of a spreadsheet, but help to ensure that variances in recorded data are incorporated without updating the function. To maintain a constant size for *Compressed Data* (page 27) functions, specify the number of values to retrieve rather than a time range. Or to display only a predetermined portion of results, specify a limited *output cell range* (page 14).

Share Spreadsheets

If you want to share PI data and make spreadsheets accessible to other users, keep the following considerations in mind:

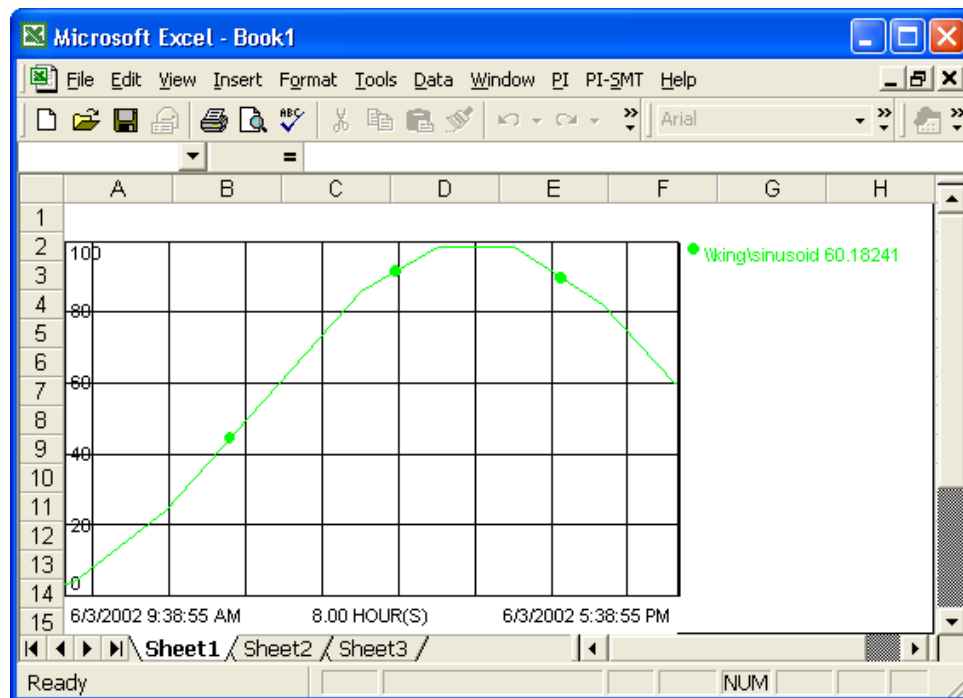
- You can send a PI DataLink spreadsheet to another Microsoft Excel user. The user must have PI DataLink installed and an active connection to the same PI server in order to recalculate functions and see dynamically updated PI point values.
- If users do not have PI DataLink installed, they can still see the last data saved in the spreadsheet provided the Excel **Calculation options** are set to *Manual* prior to the spreadsheet being opened in Excel. In Excel 2007, click the Office 2007 button, then click **Excel Options > Formulas** to access these options. In Excel XP-2003, the manual setting is available under **Tools > Options** on the Calculation tab.
- You can also save in HTML or PDF format in Excel 2007 to share a static spreadsheet.

- You can *copy and paste* (page 51) values from function arrays to new locations on a spreadsheet using **Paste Special**. Although the spreadsheet data can no longer be recalculated once copied, this method allows you to distribute a customized snapshot of data to anyone who has a copy of Excel.
- If you have Microsoft SharePoint Server in your computing environment, you can publish a DataLink spreadsheet to a SharePoint site and use *PI DataLink Server* (page 3) to display and recalculate PI function data.

Chapter 7

Trend Displays

PI DataLink includes an ActiveX trend control object that can be inserted into any Excel spreadsheet to display the trend of event values over time. Embedded trend objects can display data from both the spreadsheet and selected PI points.



Create a Trend

A wizard makes it easy to insert and configure trends. To insert a trend:

1. Click in the cell where you want to place the top left corner of the trend object, click **PI > Insert Trend (XP-2003)**, or click **Insert Trend** on the **PI** ribbon (2007) to display the **Trend Add-in Wizard**.

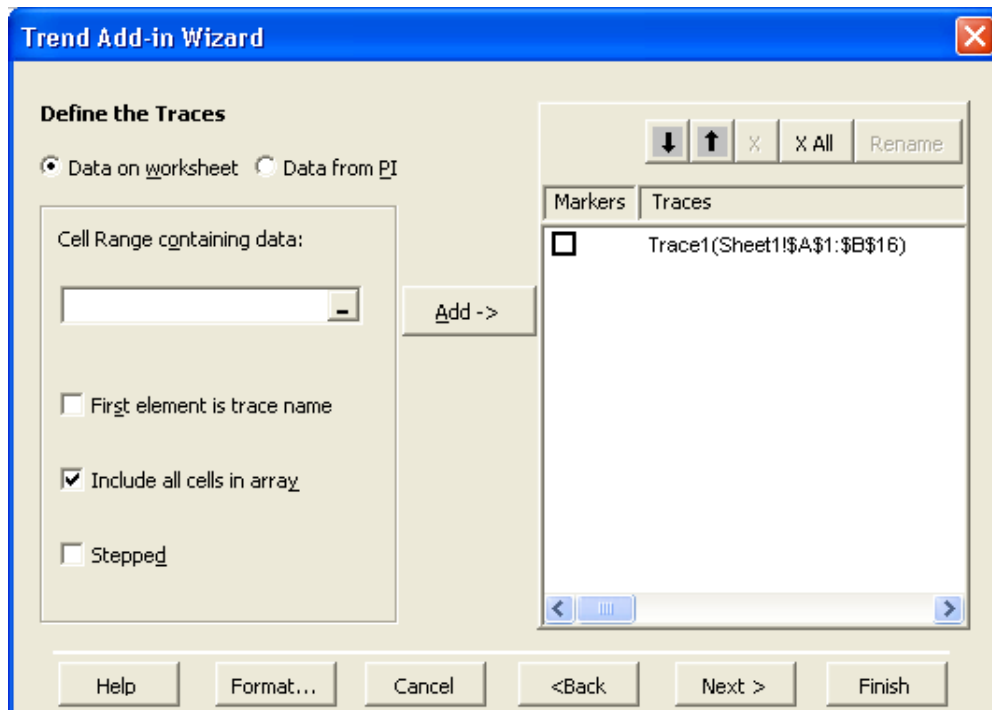
When selecting the input cell, consider that the default size of the trend object requires approximately 14 x 8 standard cells.

2. Choose the source of tag names:
 - o Select **Data on worksheet** to base the trend on *array values* (page 56) that already appear in spreadsheet cells.

- o Select **Data from PI** to base the trend on specific point data retrieved *directly from PI* (page 57).
3. Complete the remaining steps of the trend wizard.

Worksheet Data

If you choose **Data on worksheet**, the trend wizard prompts you to select the cell range containing the data you want to trend.



To specify worksheet data for the trend:

1. Click in the **Cell Range containing data** field, then click and drag to select a cell range in the spreadsheet.

Unless you deselect **Include all cells in array**, you only need to select one cell of a function array. The rest of the array is added automatically to the cell range reference when you click **Add**.

For discrete data points that are not part of an array you still need to select every desired cell.

2. Select **First element is trace name** to use the value of an adjacent cell to name the trace on the trend.

The first element is the cell above the first value of the selected range if the data is arranged in column format, and the first cell to the left of the first value of the range for data in row format.

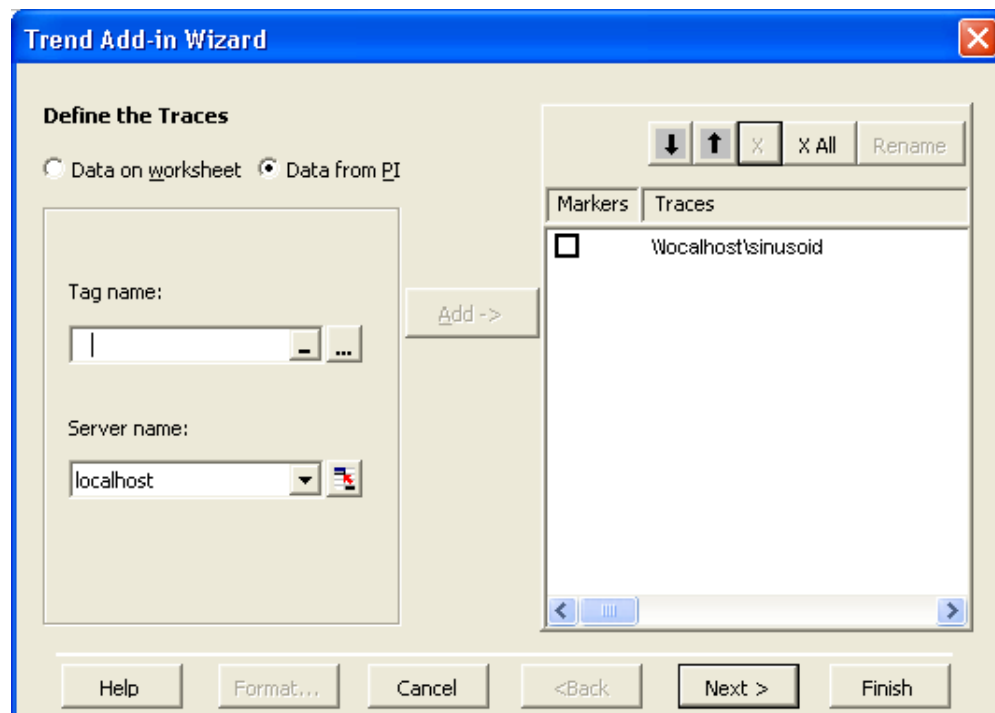
3. Select **Stepped** to plot a stepped trend instead of a continuous trace.

This option is usually specified for discrete tags.

4. Click **Add** to add the selected cell range to the trend.
5. Add cell references as necessary, or click **Next** or **Finish** to continue.

PI Data

If you choose **Data from PI**, the trend wizard prompts you to reference or search for PI tag names.



To specify PI data for the trend:

1. Click in the **Tag name** field, then click and drag to select a cell range in the spreadsheet.

You can also enter a cell reference directly, or use the reference button   to display a reference dialog used to browse and add cell references.

You can also click the selection button  to *search for tags* (page 39) in PI. Tags selected through a tag search are added automatically added to the list of traces.

2. Change the default server name (if necessary) in the **Server name** field, and click **Add** to add the selected tag name references to the trend.

You can use the reference button   to display a reference dialog used to browse and add cell references.

3. Add tag names as necessary, or click **Next** or **Finish** to continue.

Manage Traces

When building a trend from worksheet or PI data, you accumulate traces in the trace panel to the right of the wizard screen. Several controls are provided to manage the traces that appear in the trend:

- Select the **Markers** checkbox next to a trace name to display markers on the trace to indicate the location of archived data points.

Note: Traces are drawn in different colors. By default a trend also includes color-blind markers to differentiate traces on monochrome screens. Color-blind markers are evenly spaced along a trace and do not indicate archived data points.

- Click the arrow buttons above the trace panel to move traces up and down in the trace legend display.
- Click the **X** and **X All** buttons to remove selected and all traces, respectively.
- Click the **Rename** button to rename a selected trace in the trace legend.

Specify the Time Range

When the trend traces are determined, the wizard prompts you to specify a time range for the trend if any of the traces are drawn directly from PI.

To specify a time range:

1. Click in the **Start Time** and **End Time** fields, then click to select a timestamp cell in the spreadsheet.

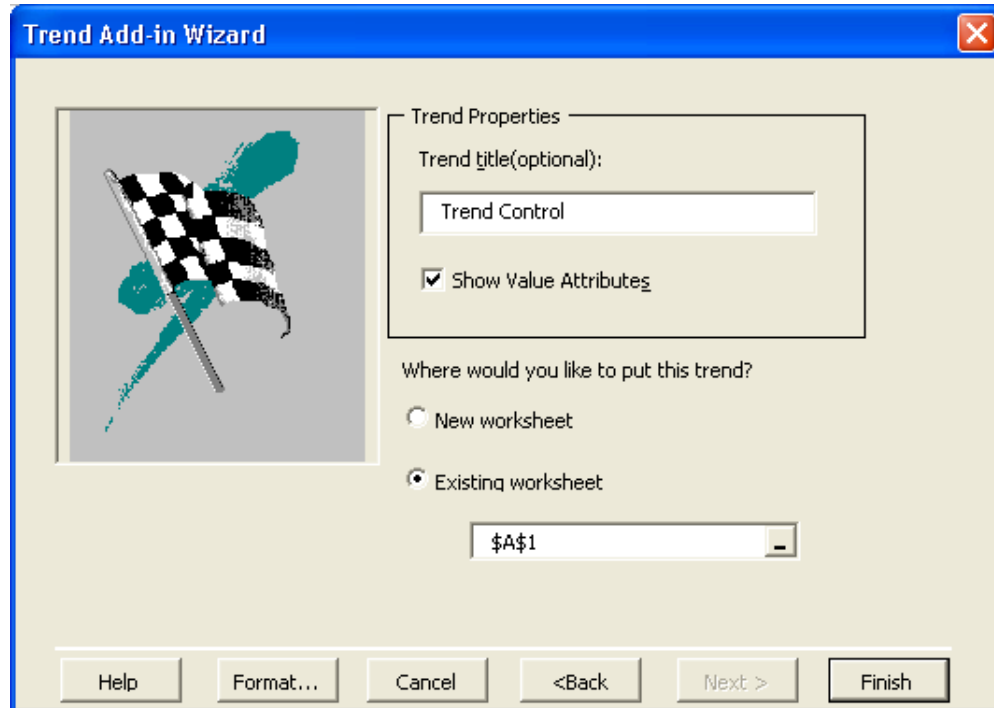
You can also enter a time string directly or use the reference button   to display a reference dialog used to browse and add cell references.

Note: If you enter a *combined or absolute time string* (page 71), be sure to use the correct timestamp format.

2. Select **Enable Updates** to refresh the trend with new PI data if the time range changes or extends to the present.
3. Choose **Next** or **Finish** to continue.

Title and Placement

Complete the trend and determine its location in the final screen.



To complete the trend:

1. Enter a **Trend title**.
2. Deselect **Show Value Attributes** if you do not want value attributes to appear along trend traces.

The icons appear when the range of data contains events that have the Questionable, Substituted, or Annotated bits set.

3. Select a different location for the trend, if desired, by placing it on a separate worksheet or changing the default placement cell.

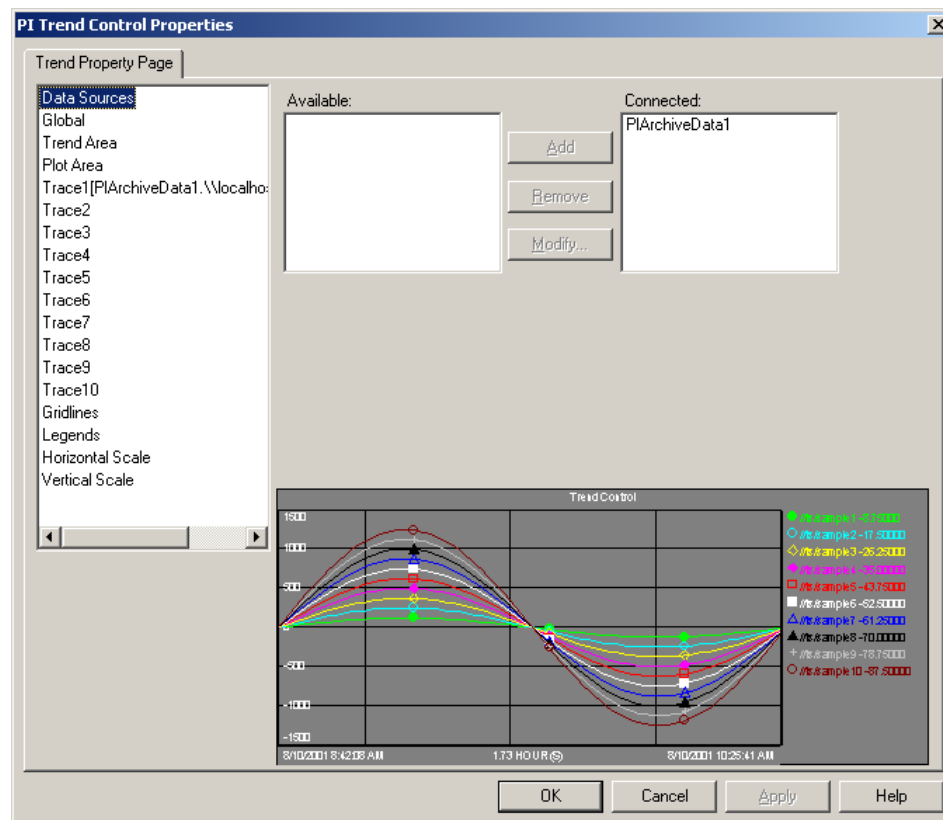
If **New worksheet** is selected, the trend is placed on a blank worksheet at the top left corner.

If **Existing worksheet** is selected, then the cell (or range used to size the trend) where the trend is to be placed must be specified or referenced in the accompanying field. If you change your default placement cell, make sure the new cell does not contain data (which will be covered by the trend display).

Context Menu

Once a trend is inserted into a spreadsheet, you can update, edit or manage it using the context menu. To access context menu options:

- Right-click on the trend display and choose an option from the menu:
 - Choose **Change Scale** to change the y-axis scaling and time range.
 - Choose **Revert** to undo any changes and revert to the original trend settings defined in the Trend Add-in Wizard.
 - Choose **Scroll Bar** to enable a scroll bar at the bottom of the trend to scroll the display through adjacent time ranges.
 - Choose **Format** to edit the display of individual traces or the trend as a whole using the **Trend Control Properties** dialog. You can change the color or line thickness of a trace, or add legend items to the trend.



- Choose **Define Trend** to display the trace definition screen of the trend wizard to update or change the trace sources.
- Choose **Define Trend Time Range** to display the time range definition screen of the trend wizard to update the time range.
- Choose **Delete Trend** to remove the trend from the spreadsheet.

Note: A deleted trend cannot be restored and must be recreated.

- Choose **Export Data** to paste the values used to plot the trend into spreadsheet cells. You can specify row or columnar format, and also export trace names as headings.
- Choose **Move/Resize** and enter a cell location to move the upper left hand corner of the trend to the specified cell. Enter a cell range to resize the trend.
- Choose **Copy/Paste** to copy the entire trend display to the clipboard and prompts for a destination cell to paste the control.

Chapter 8

Notifications

PI DataLink includes a PI Notifications Search add-in you can use to retrieve notifications subscribed to from your PI System, and insert them directly into spreadsheets.

The PI Notifications Excel add-in is installed with PI DataLink 4.0, runs on Windows XP and later operating systems, and requires Excel XP or later versions.

Note: You must have an account on at least one PI System running a PI AF 2.0 server with PI Notifications 1.0, and be subscribed to at least one notification rule to receive notifications.
For further information on PI Notifications, see the *PI Notifications User Guide*.

To launch the PI Notifications viewer:

- Click the **Notifications** icon on the **PI** ribbon (2007), or choose **PI > Notification Search** (2003).

Search and Display Notifications

You can use the PI Notifications viewer to search for and retrieve notifications you have subscribed to on an accessible PI System. A PI System is a database of structural elements and notification events that help you work with data in your PI server.

	Start Time	End Time	Duration	State	Priority	Acknowledged	Action
[-]	7.11.2008 6:00:00	7.11.2008 12:00:00	06:00:34	OutsideContr...	Normal	No	
	7.11.2008 6:00:00						Sent
	7.11.2008 6:00:00						Sent
	7.11.2008 6:00:00						Sent
	7.11.2008 6:00:00						Sent
	7.11.2008 6:00:00						Sent
	7.11.2008 6:00:00						Sent
[-]	7.10.2008 18:00:00	7.11.2008 0:00:00	06:01:01	OutsideContr...	Normal	No	
	7.11.2008 0:00:00						Sent

To search for notifications:

1. On the **Search** tab, select a **PI System** to search for notifications:
 - o Choose a system from the **PI System** menu and click **Connect** to connect to the selected PI System.
Click the ellipsis button  to access the **Systems List** and *add PI Systems* (page 64) to the menu.
 - o Click **Reference**  to reference the name of a PI System in a spreadsheet cell. Click the button again to display the **PI System** menu.
2. Choose an option from the **View Notifications** menu to filter the list of returned notifications.
 - o Choose *Active* to display all currently-active notifications.
 - o Choose *Timerange* and enter **Start Time** and **End Time** values to return notifications occurring over a specific time range.
 - o Choose *Recent* and enter a number and a time interval to return the most recent notifications over a specific time period.
3. Change the **Output Cell** to place the list of notifications into a different spreadsheet cell, if desired.
4. Click **Search**.

Notifications returned by the search appear in the *Notification History viewer* (page 65).

5. Click **OK** to insert the contents of the Notification History viewer *into the spreadsheet* (page 66) and close the dialog box.

The notification query is inserted into the spreadsheet at the designated location as a function. Like other PI functions, notification data is updated when the spreadsheet *recalculates* (page 50).

Click **Cancel** to close the dialog box, discarding the search results.

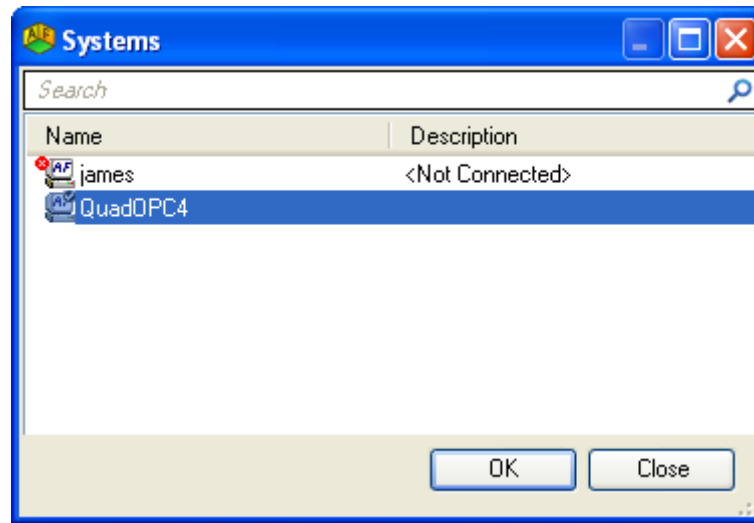
Add PI Systems

There may be multiple PI Systems in your environment, each of which may have PI Notifications installed, and be a source of notifications relevant to you. For example, you may subscribe to notifications from PI systems running in different facilities, or representing different business units.

To receive these notifications you must subscribe yourself, or have your administrator subscribe you to notifications from each PI system using the **PI System Explorer** application, and add each PI System to the PI Notifications viewer.

To add systems to the **PI System** menu:

1. Click **Systems List** .
The **Systems** dialog box appears.



2. Right-click in the systems panel and choose **New PI System**.
Right-click an existing PI system entry to change properties or specify a default system.
3. Enter the relevant network information to locate and authenticate your machine to the PI system.

The **Name** field contains the name used to describe the PI system in the PI Notifications viewer.

4. Click **Configure Active Directory** to pull PI System information from an Active Directory location, if applicable.
5. When configuration is complete, click **OK** to save the new PI System entry.

The PI system appears in the PI System menu, or at the top of the menu if specified as the default system.

For more information about PI Systems, see the *AF 2.0 User Guide*.

View Notifications

The Notifications History viewer appears in the lower half of the Notifications **Search** tab, and displays notifications that match the search criteria.

Use the Notifications History viewer to assess, manage and respond to notifications.

- Select a notification and right-click to display the context menu, then choose an option:
 - Choose **Acknowledge Instance** to acknowledge that action has been taken regarding the notification, and that no further escalation is needed. Enter a comment in the provided dialog box, if desired, and click **OK**.
 - Choose **Acknowledge Subscription** to acknowledge receipt of a notification you are subscribed to. Enter a comment in the provided dialog box, if desired, and click **OK**.
 - Choose **Add Comment** to comment on a notification.
 - Choose **View Notification Rule Summary** to display the properties of the notification rule that triggered a selected notification.

You can also manage the display of notifications in the viewer:

- Click column headings to sort notifications by a particular attribute, or right-click to show and hide columns.
- Right-click in the notifications window and choose an option:
 - Choose **Expand All** or **Collapse All** to expand or collapse a notification grouping.
 - Choose **Show Contact Events** to toggle the display of contact events for each notification.

Insert Notifications

Notifications data in spreadsheet cells may be referenced in PI DataLink functions. For example, you may want to use the *Start Time* value from a retrieved notification to retrieve sampled data indicating the progress of a batch process.

Notification Count:	22							
Start Time	Contact Event Time	End Time	Duration	State	Priority	Action	Acknowledge	
10-Dec-07 01:02:42		10-Dec-07 05:04:42	04:02:00	high	Normal			
	10-Dec-07 01:02:42					Instance acknowledged automatically		
	10-Dec-07 01:02:42					Sent		
09-Dec-07 23:50:11		10-Dec-07 01:02:42	01:12:31	low	Normal			
	09-Dec-07 23:50:11					Instance acknowledged automatically		
	09-Dec-07 23:50:11					Sent		
09-Dec-07 13:02:42		09-Dec-07 17:04:41	04:01:59	high	Normal			
	09-Dec-07 13:02:42					Instance acknowledged automatically		

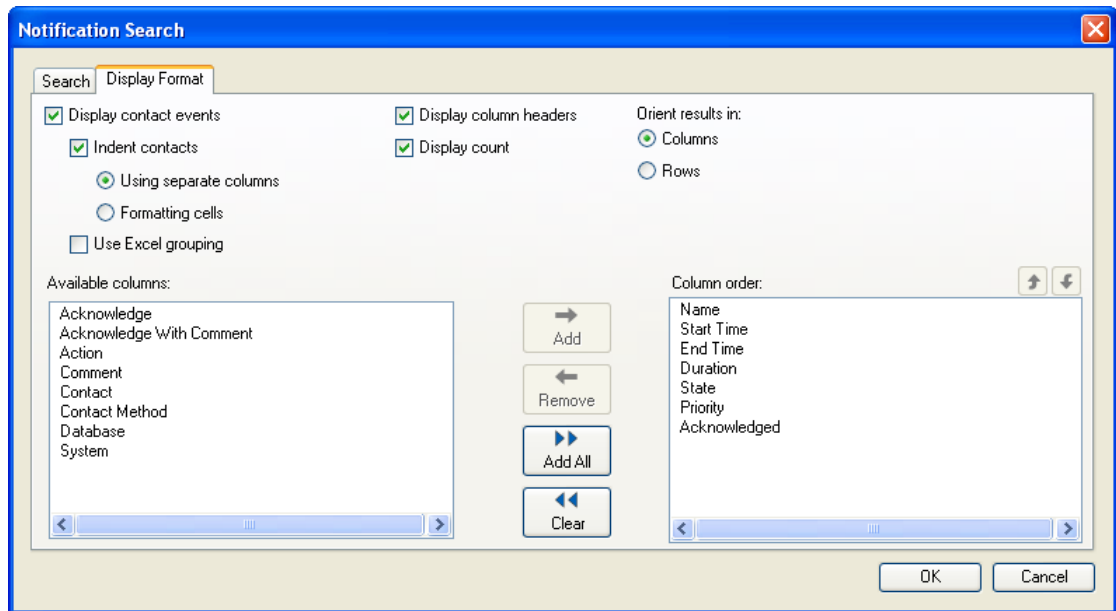
If your notifications are set to display acknowledgements, these columns include links to PI Notifications web pages where you can acknowledge or comment on a notification.

Note: Warning messages appear when you acknowledge a notification in Excel. You can disable these messages for *Office 2003* (<http://support.microsoft.com/kb/829072>) and *Office 2007* (<http://support.microsoft.com/kb/925757>), but not for *other versions* (<http://support.microsoft.com/kb/291912>) of Microsoft Office.

For more information on notifications, subscription and acknowledgement, see the *PI Notifications User Guide*.

Notification Display Settings

Use the **Display Format** tab to specify how notifications appear when inserted into spreadsheet cells. You can select the columns used to display standard notification data and change their organization and formatting.



Columns

- Select columns from the **Available Columns** list and click **Add** to move them to the **Column Order** list to determine which data is inserted into the spreadsheet.
- Use the arrow buttons above the **Column Order** list to set column (or row) order.
- Click **Remove** to move selected columns from the **Column Order** list or **Clear** to remove all columns.
- Add *Acknowledge* or *Acknowledge With Comment* columns to add links to acknowledge receipt or comment on a notification.

Contact Events

- Select **Display contact events** to add contact events related to each notification to the display, such as sending to each recipient, comments and acknowledgements.
- Select **Indent contacts** to distinguish contact events in the display by indenting them. Choose **Formatting cells** to display them indented in the same column as notification events, or **Using separate columns** to add a second column for contact events.
- Select **Use Excel Grouping** to use Excel's grouping functionality to nest contact events beneath individual parent notifications that spawned them.

Note: Excel Grouping is not supported for spreadsheets published to *PI DataLink Server* (page 3).

Display and Orientation

- Select **Display column headers** to add row or column headers above notifications data.
- Select **Display count** to add a count of notifications above notifications data.
- Under **Orient results in**, choose to display notification data in **Columns** or **Rows**.

Supplementary Information

Setup

There are two ways for users to work with PI DataLink functions in Excel spreadsheets:

- Users can install a stand-alone version of PI DataLink on their local PC.
- Web users may view spreadsheets with PI DataLink functions using *PI DataLink Server* (page 3).

Contact your OSIsoft sales representative for more information.

Note: Previous versions of PI DataLink allowed a shared version to be installed and run from a central file server. This feature is no longer supported.

You can install PI DataLink from a distribution CD or an installation kit downloaded from OSIsoft. The installation program automatically installs both Excel XP-2003 and 2007 versions of the PI DataLink add-in to your system, regardless of which version of Excel you have installed.

- If you have Excel XP-2003 on your PC, you must *configure the add-in manually* (page 4) after installation by adding it to Excel.
- If you have Excel 2007 on your PC, the install program automatically configures the add-in for Excel 2007.

Silent Installs

For silent installs, use the `silent.ini` file provided in the PI DataLink distribution kit by typing:

```
Setup.exe -f silent.ini
```

You can make site-specific alterations to the file as needed. See the PI SDK `setup.ini` for further information and descriptions of available arguments.

High Availability Connection Preferences

The connection preference is set at the time of the first connection.

For DataLink 3.1.6 and above, the connection preference for DataLink functions is set to **Any**. However, since the connection preference is set at the time of the connection, the order in which the connection is made matters. If the first connection is made with DataLink

functions, then it makes a connection with connection preference of **Any**. If the first connection is made with the trend control, tag search, or the connection manager, then the connection preference is set to **Prefer Primary**.

See *Connection Preferences* in the *High Availability Advanced User Guide* for an explanation of connectivity preferences.

PI Server Connectivity

Client computers running Microsoft Excel and PI DataLink can retrieve data from PI 3 servers.

For more information on PI Server connectivity and security, consult your PI Server documentation, or download documentation from the *OSIsoft technical support site* (<http://techsupport.osisoft.com/downloadcenter.aspx>).

Firewall Database

The firewall database on a PI server machine must be configured to allow access from client computers running PI DataLink. For example,

```
PIConfig> @table pi_gen,pifirewall PIConfig> @ostr hostmask,value  
PIConfig> @select hostmask=* PIConfig> @ends *.*.*.*, ALLOW
```

blocks no IP ranges, indicating that all client machines may connect to PI 3 Server. In contrast,

```
*.*.*.*, DISALLOW 192.168.100.*, ALLOW
```

allows connections only from TCP/IP addresses starting with 192.168.100.

PI Trusts

A PI trust allows clients to connect to a PI server machine and retrieve data without explicit user login. A trust may instead allow a user to log in based on an IP address, Windows domain, user name, application or other criteria.

PI trusts are established on the PI server and stored in a trust database.

Point Access Permissions

PI DataLink prompts the user for a username and password upon connection to a PI server. The username is used in conjunction with the PtOwner, PtAccess, DataOwner, and DataAccess fields.

For example, if the `sinusoid` tag is configured with PtAccess O:RW, G:NONE, W:NONE, and PtOwner is `user1`, a user must log in as `user1` in order to access the tag. Otherwise, the tag can not be found.

If `sinusoid` DataAccess is O:R, G:NONE, W:NONE, and DataOwner is `user1`, a user must log in as `user1` in order to see `sinusoid` tag data. Otherwise, a read access error message is returned.

However, the same user may not write data to `sinusoid` because the `DataAccess` attribute is `O:R` (read only). The PI System Manager would need to change `DataAccess` to `O:RW` (read and write) to enable a user logging in as `user1` to read and write data to `sinusoid`.

Note: The `PIAdmin` account is a super-privileged user and has read and write access to all tags. For this reason, System Administrators should not normally allow `PI DataLink` users to log in as `PIAdmin`.

PI Time

PI Time abbreviations and PI Time expressions allow you to specify times and time ranges for data using constants, variables, and short expressions.

PI Time Abbreviations

An interval is a unit of time that can be used in time entries. Intervals that support fractional values are listed below. For intervals where the Fractions column indicates No, fractional amounts cannot be used in time strings.

Name	Short name	Plural name	Member names	Fractions
second	s	seconds	no	yes
minute	m	minutes	no	yes
hour	h	hours	no	yes
day	d	days	no	no
month	mo	months	yes (for example, December)	no
year	y	years	no	no
week	w	weeks	no	no
weekday	wd	weekdays	yes (for example, Tuesday)	no
yearday	yd	yeardays	no	no

You can spell out month and weekday names, or enter the first three letters (for example, Dec, Tue).

PI Times can also be expressed using certain constants:

Constant	Result
*	The current time.
Today or t	12:00 am of the current day.
Yesterday or y	12:00 am of the previous day.
Sunday or sun	00:00:00 (midnight) on the most recent past Sunday (in reference to the PI Server).

PI Time Expressions

PI allows three types of time expressions: relative time, combined time, and absolute time. These time expression types are defined in the following table.

Expression	Description	Examples
Relative Time	Relative time expressions specify a number of days, hours, minutes, or seconds with either a leading plus sign or a leading minus sign. In the absence of a reference time (such as * or an absolute time) in either the start or end time strings, the end time is calculated relative to current clock time and the start time is calculated relative to the end time.	+1d -24h -3m +24s
Combined Time	A combined time expression is a specific reference time followed by a relative time expression.	*+8h 18-dec-02 - 3m t+32s
Absolute Time	An absolute time expression is any time expression that is neither a relative nor a combined time expression.	* 14-Dec-97 11-Nov-96 2:00:00.0001 t y

When using PI times, follow these guidelines:

- Use absolute or combined time expressions. Avoid using relative time expressions. Multiple relative time expressions in a time range may cause an incorrect start time or an error message, depending on the context of the expression.
- Relative and combined time expressions contain only a single operator: either a single plus sign (+) or a single minus sign (-). Additional operators can lead to unpredictable results. For example, the following are not valid time expressions:
 *+1d+4h
 T-1d+12h
- The name or short name for an interval used to denote PI time is not case-sensitive.

PI Time String Examples

Time Syntax Examples

PI Time String	Meaning
06-dec-91 15:00:00	3:00:00 pm on December 6, 1991
*	Current time (now)
25	00:00:00 (midnight) on the 25th of the current month
25-aug-92	00:00:00 (midnight) on August 25th, 1992
8:	08:00:00 on the current date
25 8:	08:00:00 on the 25th of the current month
t	00:00:00 on the current date (today)

PI Time String	Meaning
y	00:00:00 on the previous date (yesterday)
sun, mon, tue, wed, thu, fri, sat	00:00:00 on the most recent Sunday, Monday, ... Saturday
*-1h	One hour ago
t+8h	8:00:00 am today
y-8h	4:00:00 pm on the day before yesterday
mon+14.5h	2:30:00 pm last Monday
sat-1m	11:59:00 pm last Friday

Time Interval Examples

In interval expressions, a positive or unmarked interval is based on the starttime, and a negative interval is based on the endtime of a time expression.

For example, if starttime is y , endtime is t , and interval is $+5h$ for a Sampled Data function, then interpolated values are generated at y , $y+5h$, $y+10h$, $y+15h$, and $y+20h$. If the interval is $-5h$, the interpolated values are generated at $y+4h$, $y+9h$, $y+14h$, $y+19h$ and t .

PI Time String	Meaning
1.5h	One and one-half hours
32m	Thirty-two minutes
49s	Forty-nine seconds
+5h	Five hours added to the time beginning with the starttime
-5h	Five hours subtracted from the time beginning with the endtime

PI Data Type Support

PI DataLink supports the following PI point data types:

Data Type	Support
Digital (defined states)	supported
Int (16 and 32)	supported
Float (16, 32 and 64)	supported
String (text)	supported
Timestamp	supported
Blob	not supported

PI Expressions

PI expressions (also known as PI Performance Equations) are used for functions that incorporate mathematical operations and calculations based on PI points.

For some functions you can substitute a **PI Expression** for a tag name to calculate values as they are retrieved from the PI Server. Functions that allow an expression argument include both **PI Tag** and **PI Expression** buttons in the function dialog box or task pane.

To base a function on a PI calculation expression:

1. Click the **PI Expression** button at the top of the function dialog box or task pane.
The **Tagname(s)** field becomes an **Expression(s)** field. Other fields may activate, or become inactive.
2. Enter the expression directly in the **Expression(s)** field, or use the field to reference a cell containing an expression.

Using field references allow you to specify multiple expressions for a function.

Remember that an expression in a referenced cell is a string, and should be preceded by an apostrophe.

Some functions also provide a field for a **Filter Expression** argument. You can supply a filter expression to limit the values returned by a function, and reference a filter expression in a worksheet cell.

Functions that may use these expressions include:

- *Archive Value* (page 26)
- *Compressed Data* (page 27)
- *Sampled Data* (page 30)
- *Timed Data* (page 31)
- *Calculated Data* (page 33)
- *Time Filtered* (page 37)

The following sections briefly outline PI expression basics. PI DataLink supports all functional syntax, operators and functions supported by the PI Server.

For a full description of PI Performance Equations and examples, see the *PI Server Applications Guide* or the *PI 3 Performance Equations* online help file.

Syntax

Follow these guidelines when writing PI expressions:

- Begin the expression with a single quote to force Excel to interpret the expression as a string.
- Enclose tag names in the expression in single quotes.
- Enter two single quotes for an expression that begins with a tag name.

Operators

The following operators are supported in PI Expressions:

Type	Operator	Syntax Example	Meaning
Arithmetic	+	A + B	Addition: A + B
	-	A - B	Subtraction: A minus B
	*	A * B	Multiplication: A times B
	/	A / B	Division: A divided by B
	^	A ^ B	Raising to a power: A to the power of B (AB)
	Mod	A mod B	Modulus: the remainder of A divided by B
Relational	<	A < B	Less than: returns true if A is less than B
	=	A = B	Equal to: returns true if A equal to B
	>	A > B	Greater than: returns true if A is greater than B
	<=	A <= B	Less than or equal to: returns true if A is less than or equal to B
	<>	A <> B	Not equal to: returns true if A is not equal to B
	>=	A >= B	Greater than or equal to: returns true if A is greater than or equal to B
Prefix	Not	NOT A	Complementation: returns true if A is 0 and False otherwise
	-	- A	Negation (as prefix operator): returns the negative of A
Conjunction, Disjunction and Inclusion	And	A and B	Conjunction: returns true if operands A & B both evaluate to true. If both A and B are integers, returns the result of a bitwise AND operation.
	Or	A or B	Inclusive disjunction: returns true if either operand A or operand B evaluates to true. If both A and B are integers, returns the result of a bitwise OR operation.
	in ..	A in B..D	Membership in a range: returns true if the value of A is between B and D
	in ()	A in (B1, B2, ...BN)	Membership in a discrete set: returns true if the value of A matches any of the values enclosed in the parentheses.
If-Then-Else Expressions	if then else	if A then B else D	If-then-else expression: returns B if A is true—otherwise it returns D

Performance Equation Functions

You can also evaluate expressions that contain PI Performance Equation functions:

Math Functions

Name	Description
Abs	Absolute value
Asin	Arc sine
Acos	Arc cosine
Atn	Arc tangent
Atn2	Arc tangent (two arguments)
Cos	Cosine
Cosh	Hyperbolic cosine
Exp	Exponential
Float	Conversion of string to number
Frac	Fractional part of number
Int	Integer part of number
Log	Natural logarithm
Log10	Common logarithm
Poly	Evaluate polynomial
Round	Round to nearest unit
Sgn	Numerical sign
Sin	Sine
Sinh	Hyperbolic sine
Sqr	Square root
Tanh	Hyperbolic tangent
Tan	Tangent
Trunc	Truncate to next smaller unit

Aggregate Functions

Name	Description
Avg	Average
Max	Maximum
Median	Median selector
Min	Minimum
PStDev	Population standard deviation
SStDev	Sample standard deviation
Total	Sum

Miscellaneous Functions

Name	Description
BadVal	See if a value is bad (not a number or time)
Curve	Get value of a curve
DigState	Get digital state from a string
IsDST	Test whether a time is in local daylight savings time period
IsSet	Test if a PI value is annotated, substituted, or questionable
StateNo	The code number of a digital state
TagBad	See if a point has an abnormal state

PI Archive Retrieval

Name	Description
NextEvent	Time of a point's next Archive event
NextVal	Point's next value after a time
PrevEvent	Time of a point's previous Archive event
PrevVal	Point's previous value before a time
TagVal	Point's value at a time

PI Archive Search

Name	Description
FindEq	Timestamp when point = value
FindGE	Timestamp when point >= value
FindGT	Timestamp when point > value
FindLE	Timestamp when point <= value
FindLT	Timestamp when point < value
FindNE	Timestamp when point != value
TimeEq	Total period when point = value
TimeGE	Total period when point >= value
TimeGT	Total period when point > value
TimeLE	Total period when point <= value
TimeLT	Total period when point < value
TimeNE	Total period when point != value

PI Archive Statistics

Name	Description
EventCount	Number of Archive events
PctGood	Percent of good time in a period
Range	Range of minimum to maximum value
StDev	Time-weighted standard deviation

Name	Description
TagAvg	Time-weighted average
TagMean	Event-weighted average
TagMax	Maximum value in a period
TagMin	Minimum value in a period
TagTot	Time integral over a period

Point Attributes

Name	Description
TagDesc	Get a point's descriptor
TagEU	Get a point's engineering unit string
TagExDesc	Get a point's extended descriptor
TagName	Get a point's name
TagNum	Get a point's ID
TagSource	Get a point's point source string
TagSpan	Get a point's span
TagType	Get a point's type character
TagTypVal	Get a point's typical value
TagZero	Get a point's zero value

Time Functions

Name	Description
Bod	Timestamp for beginning of the day for given time
Bom	Timestamp for beginning of the month for given time
Bonm	Timestamp for first of the next month for given time
Day	Day of the month from a time
DaySec	Seconds since midnight from a time
Hour	Hour from a time
Minute	Minute from a time
Month	Month from a time
Noon	Timestamp for local noon of day of a time
ParseTime	Convert character string to time
Second	Second from a time
Weekday	Day of the week from a time
Year	Year from a time
Yearday	Day of the year from a time

Dynamic Response

Name	Description
Arma	Dynamic response from Auto Regressive Moving Average model
Delay	Introduce time delay
MedianFilt	Select the median value of time series
Impulse	Dynamic response characterized by impulse response shape

Alarm Status Functions

Name	Description
AlmAckStat	Alarm acknowledgment status code
AlmCondition	Condition code number for Alarm State
AlmCondText	Alarm condition as text
AlmPriority	Alarm priority number

String Functions

Name	Description
Ascii	ASCII character code for a character
Char	String for ASCII character code(s)
Compare	Wild comparison of two strings
DigText	Text for a digital state
Format	Formatting of a numerical number
InStr	Instance of a sub-string
LCase	Conversion of all characters to lower case
Len	Length of a string
Left	First characters in a string
LTrim	Removal of blanks on the left side of a string
Mid	Extraction of a sub-string from a string
Right	Last characters in a string
RTrim	Removal of blanks on the right side of a string
Trim	Removal of blanks on both sides of a string
UCase	Conversion of all characters to upper case

String Conversion

Name	Description
Concat	Concatenate two or more strings
String	String representing any PI value
Text	Concatenation of strings for a series of PI value arguments

PI Expression Examples

Note: These examples use the tag delimiters : (colon) and . (period). Your PI system may use different delimiters.

The following expression determines the sum of the value of `sinusoid` cubed and the value of `cdf144` divided by 10:

```
(( 'sinusoid' ) ^ 3 + 'cdf144' / 10)
```

The expression below evaluates to *true* (non-zero) if the absolute value of the tag `mytag` is at least 14.65.

```
(abs('my:tag') >= 14.65)
```

When comparing digital tags, use the negative of the digital state code number. For digital tags, use the digital state string itself. For example:

```
('my:tag' = "Manual")
```

The following expression evaluates to *True* if the value of the tag `sinusoid` is less than 45 and the square root of the value of the tag `vdf1002` is greater than 2:

```
('sinusoid' < 45 and sqrt('vdf1002') > 2)
```

Filter Expressions

Add a filter expression to filter event values using a mathematical expression, eliminating data for which the expression evaluates as *false*. The filter expression is applied to the raw data from PI (and not values that result from the calculation itself).

For example, the simple filter expression:

```
sinusoid < 70
```

would remove all values over 70 from the calculation. You can also use any valid PI performance equation in the filter expression to build more complex expressions to remove atypical peaks in data values, for example.

Select **mark as filtered** to substitute *Filtered* as a placeholder for each value or block of values filtered from the array based on the filter *expression* (page 74).

Manual Function Entry

PI DataLink provides tools that can be used to build any PI DataLink function graphically. Experienced users of Excel and PI may prefer to enter functions directly in the Excel formula bar.

The following topics apply to PI DataLink functions when entered directly in this manner.

Use Array Functions

When defining array functions follow these steps:

- Select an appropriate output range for the array based on the number of expected values.
- Enter the PI function and its arguments into the Excel formula bar.
- Place the PI function into the selected output cells with the **Ctrl+Shift+Enter** combination of keystrokes.

Refer to the online help for Microsoft Excel for more information on array functions.

Mathematical Functions

In a tag calculation expression, mathematical operators and functions act upon tag names. Supported Mathematical Functions are:

abs — absolute value

atn — arc tangent (angle returned in radians)

cos — cosine (argument in radians)

exp — e (2.71828...) raised to a power

int — integer part of a number

log — natural logarithm

sgn — sign function (-1 if argument is negative, 0 if zero, 1 if positive)

sin — sine (argument in radians)

sqr — square root function

tan — tangent (argument in radians)

For example, the expression below finds the difference between the natural logarithm of the value of the tag `cdep158` and the cosine of the value of the tag `cdep158`.

```
(log('cdep158') - cos('cdep158'))
```

String Arguments

String arguments must be surrounded by double quotes. For example, select a 1 x 2 output array and enter:

```
=PICurrVal("sinusoid", 1, "casaba")
```

to retrieve the snapshot time and value for the `sinusoid` tag from the PI Server `casaba`.

Cell References

You can use cell references for any PI DataLink function arguments. For example, the following function:

```
=PICurrVal(A1, A2, A3)
```

gives the same result as the function definition

```
=PICurrVal("sinusoid", 1, "casaba")
```

if the same three arguments are stored in cells A1, A2, and A3, respectively.

PIServer Argument

The PIServer argument is optional provided the desired PIServer is specified as the default under **PI > Connections**.

For example, to retrieve the current value for the tag `sinusoid` from the default PI Server, enter:

```
=PICurrVal("sinusoid", 0, )
```

into a 1x1 output range.

Outcodes

DataLink function syntax includes integer outcode arguments. The outcodes determine how *appended data* (page 16) are returned and output cells are oriented.

Outcodes are automatically generated by function dialog boxes and task panes. However, if you enter a function manually in Excel's function bar, you may need to calculate and include appropriate outcode values yourself.

A function outcode is a decimal representation of binary bits. In PI DataLink, bits carry the following meanings:

- 1st bit: Show Timestamps bit
- 2nd bit: Orientation bit (columns/row)
- 3rd bit: Show Percent Good bit
- 4th bit: Hide Count bit
- 5th bit: Show Extended Status bit
- 6th bit: Show Annotations bit
- 7th bit: Show Start Time
- 8th bit: Show End Time
- 9th bit: Show Min/Max Time

For example, a 0 outcode in a `PICurrVal()` function results in a current snapshot value in the designated output cell. An outcode of 1 places a timestamp in column one and the

snapshot value in column two of a 1 x 2 array. An outcode of 2 places a timestamp in row one and the snapshot value in row two of a 2 x 1 array.

The formula used to calculate a corresponding outcode is:

$$\begin{aligned} \text{bits set} &= \{ \text{Show Timestamps}, \text{Show PctGood}, \text{Show Start Time}, \text{Show Min/MaxTime} \} \\ &= \{ 1, 3, 7, 9 \} \end{aligned}$$

$$\begin{aligned} \text{outcode} &= \sum_i 2^{x_i-1} = 2^{1-1} + 2^{3-1} + 2^{7-1} + 2^{9-1} \\ &= 2^0 + 2^2 + 2^6 + 2^8 \\ &= 1 + 4 + 64 + 256 \\ &= 325 \end{aligned}$$

Note: Functions have different arguments and outcodes. Those valid for one function are not necessarily valid for another. See the *function reference* (page 87) for outcode specifications by function.

EXAMPLE

Suppose we want a Calculated Data function to display **Percent Good**, **Start Time**, and **Min/Max Time**. Since Start Time and Min/Max Time are displayed, the **Show Timestamps** bit must also be set. The function shown on the Excel function bar would look like:

```
=PICalcDat("sinusoid","y","t","1h","minimum", 1, 325,"piserver")
```

Note: If an outcode indicates a function should display multiple columns or rows, but only one appears, right click and choose **Recalculate/Resize** to update the function.

Write Data to PI

While standard PI DataLink functions retrieve data from a PI server, one function allows you to write a value from a worksheet to a PI server. The Excel macro function *PIPutVal()* (page 112) replaces an existing PI archive event with a matching timestamp with a new value supplied by the user.

PIPutVal() can be run only in an Excel 4.0 macro sheet, or in VBA using the application.run method. See the *putval_code* module in the **piexam32.xls** distributed with PI DataLink, and in particular the tags marked *PutVal* and the VBA module *PutVal_code*, for an example of a VBA module used to enter PI functions in a spreadsheet.

Note that VBA does not recognize PI DataLink functions, but you can formulate a function as a text string, and then set the formula Array property of a range of cells to the text string value. The VBA code can then check the cell values for the PI function results. This yields the same results as manual entry of PI functions in the same range of cells.

Troubleshooting

Consult the sections below if you experience difficulty using PI DataLink. If these topics do not address or solve the problem, see *Technical Support and Resources* (page 115) for details on contacting Technical Support.

PI Menu Not Available

If a PI menu does not appear after you install PI DataLink, you may need to *manually configure* (page 4) the add-in for Excel, or reduce *security settings* (page 85) for add-ins.

Some user-written VBA scripts in Excel spreadsheets (or add-ins) have been known to reset the main Excel menu. In such cases, the PI menu may disappear even though the PI DataLink add-in is still loaded, and embedded PI DataLink array functions may work even though the PI menu itself is not available. Restarting Excel usually fixes this problem.

Array and Cell Limits

PI DataLink is subject to an Excel limitation on the number of elements in a calculation array. The maximum numbers of values that can be returned for a single PI function, per retrieval are:

- 65,536 for Excel XP-2003
- 1,048,576 for Excel 2007

The Excel SDK used by PI DataLink limits strings to 255 characters in length for Excel XP-2003, and input strings for PI DataLink expressions in a referenced cell may not exceed this 255 character limit.

Similarly, string output such as PI tag name strings may be truncated by PI DataLink to 255 characters.

Row Limitations

PI DataLink 4.x can utilize all one million rows in a single Microsoft Excel spreadsheet. Data up to this limit may be retrieved in one or more function arrays.

To retrieve very large amounts of data from PI, users may need to configure the `ArcMaxCollect` parameter on their PI server. The default value is 150 thousand rows.

Note that a query can exceed a maximum limit even though it returns a relatively small number of values. What counts is the real number of events called from PI archives. For example, `sinusoid` tag values vary constantly between 0 and 100. A query for sampled values of `sinusoid` every 30 seconds over a year, with filter expression excluding all values less than 95 may return a relatively small number of values, but many more events are retrieved from the archives as the query is processed.

Security

Excel security features for add-ins, ActiveX controls and macros allow you to determine the types of components that can run inside Excel. Different levels of settings provide a means to disable, enable with prompting, or completely enable these types of objects.

Excel security settings may conflict with certain PI DataLink features:

- DataLink is an Excel add-in. Security settings that disable add-ins prevent PI DataLink from loading in Excel.
- *Trend Controls* (page 55) use a signed ActiveX control marked safe for initialization and require some level of enablement for ActiveX controls.
- *PIPutVal* (page 112) spreadsheets use macros and require some level of enablement for macros.

If you regularly apply Excel security to spreadsheets, or encounter problems using these PI DataLink features, you may need to reduce or eliminate Excel security settings depending on how you plan to use PI DataLink.

For further information on Excel security, consult Microsoft Excel online help and online resources.

Appendix B

PI DataLink Function Reference

PI DataLink *function dialog boxes (XP-2003) or task panes (2007)* (page 13) are commonly used to generate PI DataLink functions, with the resulting function depending on the options and arguments selected. These same functions may also be *entered directly* (page 80) in the Excel formula bar.

The following complete reference to supported PI DataLink functions groups functions by type. Each topic describes complete syntax options, and indicates how arguments are used. In most cases an example is also provided.

Single Value Functions

Single value functions associate a PI point with a specific point in time, and return only one corresponding data value unless an array of tag names are referenced.

PICurrVal()

Retrieves current PI point values from the snapshot, using the following syntax:

```
PICurrVal(tagname, outcode, PIServer)
```

ARGUMENTS

Argument	Value
Tagname (string)	The tag name or names matching the desired PI points
outcode (integer)	An output code to determine results placement, either 0, 1, or 2
PIServer (string)	The target PI server

RETURNS

Outcode	Value	Timestamp
0	output cell	none
1	right column	output cell
2	output cell	row above

EXAMPLE

The following retrieves the current value and corresponding timestamp for the tag `let439` from the PI Server `holden` and places the timestamp in the column to the left of the value:

```
=PICurrVal("let439",1,"holden")
```

RELATED TOPICS

- [Current Value \(page 25\)](#)

PIArcVal()

Retrieves archived PI point values corresponding to specified times, using the following syntax:

```
PIArcVal(tagname, timestamp, outcode, PIServer, mode)
```

ARGUMENTS

Argument	Value
tagname (string)	The tag name or names matching the desired PI points
time stamp (string)	The timestamp corresponding to the archive value
outcode (integer)	An output code to determine results placement, either 0, 1, or 2
PIServer (string)	The target PI server
mode (string)	The mode used to determine which values are retrieved

RETURNS

Outcode	Archive Value	Timestamp
0	output cell	
1	left column	output cell
2	output cell	row above

EXAMPLE

The following retrieves the interpolated value for the tag `cdep158` at 7:20pm on December 11, 1992 from the PI Server `casaba`:

```
=PIArcVal("cdep158","11-dec-92 19:20",0, "casaba","interpolated")
```

RELATED TOPICS

- [Archive Value \(page 26\)](#)

PIExpVal()

Retrieves an archived PI point value based on an expression, using the following syntax:

```
PIExpVal(expression, timestamp, PIServer)
```

ARGUMENTS

Argument	Value
expression (string)	A PI calculation expression
time stamp (string)	The timestamp corresponding to the archive value
PIServer (string)	The target PI server

RETURNS

Outcode	Archive Value	Timestamp
0	output cell	
1	left column	output cell
2	output cell	row above

EXAMPLE

The following calculates the square root of the value of the tag `sinusoid` from the PI server `thevax` at midnight yesterday:

```
=PIExpVal("sqr('sinusoid')","y",0,"thevax")
```

RELATED TOPICS

- *Archive Value* (page 26)

Multiple Value Functions

Multiple value functions associate a PI point with a range of time over which there can be one or many corresponding values at different points in time.

PINCompDat()

Retrieves a number of PI point values corresponding to a specified start time, using the following syntax:

```
PINCompDat(tagname, stime, numvals, outcode, PIServer, mode)
```

ARGUMENTS

Argument	Value
tagname (string)	The tag name or names matching the desired PI points
stime (string)	The start time for the range of retrieved values (in PI time format)

Argument	Value
numvals (integer)	The number of desired values (negative if counting backward in time)
outcode (integer)	An output code
PIServer (string)	The target PI server
mode (string)	The method of handling values at boundaries of the time range

RETURNS

Results display is determined by an *output code* (page 82) value.

EXAMPLE

The following retrieves 10 compressed data points and corresponding time stamps for the tag `cdf144` starting from 1:00 a.m. this morning from the default PI server with inside as boundary type:

```
=PINCompDat("cdf144", "1:00", 10, 1, , "inside")
```

RELATED TOPICS

- *Compressed Data* (page 27)

PINCompFilDat()

Retrieves a number of filtered PI point values corresponding to a specified start time, using the following syntax:

```
PINCompFilDat(tagname, stime, numvals, filtxp, filtcode, outcode,  
PIServer, mode)
```

ARGUMENTS

Argument	Value
tagname (string)	The tag name or names matching the desired PI points
stime (string)	The start time for the range of retrieved values (in PI time format)
numvals (integer)	The number of desired values (negative if counting backward in time)
filtxp (string)	A filter expression used to filter out results
filtcode (integer)	A filter code, either 1 to denote <i>Filtered</i> status or 0 to skip
outcode (integer)	An output code
PIServer (string)	The PI Server
mode (string)	The method of handling values at boundaries of the time range

RETURNS

Results display is determined by an *output code* (page 82) value.

EXAMPLE

The following retrieves 10 compressed data points and the corresponding times, starting at 2:00 a.m., for the tag `cdf144` while the tag `cdep158` is greater than 38:

```
=PINCompFilDat("cdf144","2:00",10,"'cdep158'>38",1,1,,)
```

The status **Filtered** is displayed for times when the filter condition is false. Data are retrieved from the default PI Server with `inside` as boundary type.

RELATED TOPICS

- *Compressed Data* (page 27)
- *filter expression* (page 74)

PICompDat()

Retrieves PI point values corresponding to a specified time range, using the following syntax:

```
PICompDat(tagname, stime, etime, outcode, PIServer, mode)
```

ARGUMENTS

Argument	Value
tagname (string)	The tag name or names matching the desired PI points
stime (string)	The start time for the range of retrieved values (in PI time format)
etime (string)	The end time for the range (in PI time format)
outcode (integer)	An output code
PIServer (string)	The target PI server
mode (string)	The method of handling values at boundaries of the time range

RETURNS

Results display is determined by an *output code* (page 82) value.

EXAMPLE

The following retrieves compressed data points and corresponding time stamps for the tag `cdf144` starting from 1:00 a.m. to 3:00 a.m. this morning from the default PI server with `inside` as boundary type:

```
=PICompDat("cdf144","1:00","3:00",1,, "inside")
```

RELATED TOPICS

- *Compressed Data* (page 27)

PICompFilDat()

Retrieves filtered PI point values corresponding to a specified time range, using the following syntax:

```
PICompFilDat(tagname, stime, etime, filtexp, filtcode, outcode,  
PIServer, mode)
```

ARGUMENTS

Argument	Value
tagname (string)	The tag name or names matching the desired PI points
stime (string)	The start time for the range of retrieved values (in PI time format)
etime (string)	The end time for the range (in PI time format)
filtexp (string)	The filter expression used to filter results
filtcode (integer)	A filter code, either 1 to return <i>Filtered</i> status or 0 to skip
outcode (integer)	An output code
PIServer (string)	The target PI server
mode (string)	The method of handling values at boundaries of the time range

RETURNS

Results display is determined by an *output code* (page 82) value.

EXAMPLE

The following retrieves compressed data points and the corresponding times, starting from 2:00 a.m. and ending at 10:00 a.m., for the tag `cdf144` when the tag `cdep158` is greater than 38:

```
=PICompFilDat("cdf144","2:00","10:00","'cdep158'>38",1,1,,)
```

The status *Filtered* is displayed for times when the filter condition is false. Data are retrieved from the default PI server with inside as boundary type.

RELATED TOPICS

- *Compressed Data* (page 27)
- *filter expression* (page 74)

PI SampDat()

Retrieves sampled PI point values corresponding to a specified time range, using the following syntax:

```
PI SampDat(tagname, stime, etime, interval, outcode, PIServer)
```

ARGUMENTS

Argument	Value
tagname (string)	The tag name or names matching the desired PI points

Argument	Value
stime (string)	The start time for the range of sampling (in PI time format)
etime (string)	The end time for the range (in PI time format)
interval (string)	The interval between sampled values (in PI time format)
outcode (integer)	An output code, either 0, 1, 2, or 3
PIServer (string)	The target PI server

RETURNS

Outcode	Sampled Value	Timestamp
0	output cell	none
1	right column	output cell
2	output cell	none
3	row below	output cell

An *output code* (page 82) of 2 transposes the output array that results from an output code of 0. An output code of 3 transposes the output array that results from an output code of 1.

EXAMPLE

The following retrieves sampled data for the tag located in cell B1, from midnight yesterday to midnight today at 3-hour intervals:

```
=PISampDat (b1, "y", "t", "3h", 1, )
```

Timestamps are also returned.

RELATED TOPICS

- *Sampled Data* (page 30)

PISampFilDat()

Retrieves filtered, sampled PI point values corresponding to a specified time range, using the following syntax:

```
PISampFilDat(tagname, stime, etime, interval, filtxp, filtcode, outcode, PI Server)
```

ARGUMENTS

Argument	Value
tagname (string)	The tag name or names matching the desired PI points
stime (string)	The start time for the range of sampling (in PI time format)
etime (string)	The end time in PI time format
interval (string)	The interval between sampled values (in PI time format)
filtxp (string)	The filter expression used to filter results

Argument	Value
<code>filtcode</code> (integer)	A filter code, either 1 to return <i>Filtered</i> status or 0 to skip
<code>outcode</code> (integer)	An output code, either 0, 1, 2, or 3
<code>PIServer</code> (string)	The target PI server

RETURNS

Outcode	Sampled Value	Timestamp
0	output cell	none
1	right column	output cell
2	output cell	none
3	row below	output cell

An *output code* (page 82) of 2 transposes the output array that results from an output code of 0 and an output code of 3 transposes the output array that results from an output code of 1.

EXAMPLE

The following formula retrieves sampled data for `sinusoid` at 11-Jan-97, 11-Jan-97 1:00AM, 11-Jan-97 2:00AM, and 11-Jan-97 3:00AM:

```
=PISampFilDat("sinusoid", "11-Jan-97", "+3h", "1h", A1, 1, 1, )
```

Timestamps are shown in column1 and values in column2. A *Filtered* status is returned for values do not satisfy the filter condition in cell A1.

RELATED TOPICS

- *Sampled Data* (page 30)
- *filter expression* (page 74)

PIExpDat()

Retrieves PI point values based on an expression, using the following syntax:

```
PIExpDat(expression, stime, etime, interval, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
<code>expression</code> (string)	A PI calculation expression
<code>stime</code> (string)	The start time for the range of sampled values (in PI time format)
<code>etime</code> (string)	The end time (in PI time format)
<code>interval</code> (string)	The interval between sampled values (in PI time format)
<code>outcode</code> (integer)	An output code to determine results placement, either 0, 1, 2, or 3
<code>PIServer</code> (string)	The target PI server

RETURNS

Outcode	Sampled Value	Timestamp
0	output cell	none
1	right column	output cell
2	output cell	none
3	row below	output cell

An *output code* (page 82) of 2 transposes the output array that results from an output code of 0 while an output code of 3 transposes the output array that results from an output code of 1.

EXAMPLE

The following calculates the square root of the value of the tag `sinusoid` from the PI server `thevax` at one-hour intervals, starting from midnight yesterday to midnight today:

```
=PIExpDat("sqr('sinusoid')", "y", "t", "1h", 1, "thevax")
```

It also displays the timestamps.

RELATED TOPICS

- *Sampled Data* (page 30)
- *PI calculation expression* (page 74)

PITimeDat()

Retrieves sampled PI point values corresponding to a specified array of timestamps, using the following syntax:

```
PITimeDat(tagname, timestamps, PIServer, mode)
```

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name or names matching the desired PI points
timestamps (reference)	A reference to an array of timestamps in PI time format
PIServer (string)	The target PI server
mode (string)	The mode used to determine which values to retrieve

RETURNS

Sampled data. If the timestamps are in a single column, the data are returned along a single column. If the timestamps are in a single row, the data are returned along a single row.

EXAMPLE

The following retrieves interpolated timed data for the times located in cells B1 through B12 for the tag `mytag` from the default PI Server:

```
=PITimeDat("mytag", b1:b12, , "interpolated")
```

RELATED TOPICS

- *Timed Data* (page 31)

PITimeExpDat()

Retrieves PI point values based on an expression, and corresponding to a specified array of timestamps, using the following syntax:

```
PITimeExpDat(expression, timestamps, PIServer)
```

ARGUMENTS

Argument	Explanation
expression (string)	A PI calculation expression
timestamps (reference)	A reference to an array of timestamps in PI time format (monotonically increasing or decreasing for PI 2 servers)
PIServer (string)	The target PI server

RETURNS

Sampled data. If the timestamps are in a single column, the data are returned along a single column. If the timestamps are in a single row, the data are returned along a single row.

EXAMPLE

The following calculates the square root of the value of the tag `sinusoid` for the times located in cells B1 through B12 from the default PI server:

```
=PITimeExpDat("sqr('sinusoid')",b1:b12,)
```

RELATED TOPICS

- *Timed Data* (page 31)
- *PI calculation expression* (page 74)

Calculation Functions

Calculation functions compute new values from PI point values corresponding to a specific time range.

PICalcVal()

Retrieves a calculated PI point value using the following syntax:

```
PICalcVal(tagname, stime, etime, mode, cfactor, outcode, PIServer)
```

Note: `PICalcVal()` is generated only by the Calculated Data dialog box in *XP-2003* versions. *2007* versions use `PIAdvCalcVal()`.

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name or names matching the desired PI points for calculation
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
mode (string)	The type of calculation to be used to compute point values
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, 0-7, and not 3
PIServer (string)	The target PI server

RETURNS

The *output code* (page 82) argument affects the output as follows:

- If the outcode is 0, only the calculated value is returned.
- If the outcode is 1, the following applies:

Mode	Column 1	Column 2	Column 3
Total	Total value		
Minimum	Time of min. Value	Minimum value	
Maximum	Time of max. Value	Maximum value	
Stdev	Stdev. Value		
Range	Time of min. Value	Time of max. value	Range value
Average	Average value		
Count	Count value		
Mean	Mean value		

- An output code of 2 transposes the output array that results from an output code of 1, and results are returned in rows instead of in columns.
- Outcode of 3 is not used.
- If the outcode is 4, the percent good is returned to the right of the value cell.
- If the outcode is 5, the following applies:

Mode	Column 1	Column 2	Column 3	Column 4
Total	Total value	Percent good		
Minimum	Time of min. value	Minimum value	Percent good	
Maximum	Time of max. value	Maximum value	Percent good	
Stdev	Stdev. value	Percent good		
Range	Time of min. value	Time of max. value	Range value	Percent good
Average	Average value	Percent good		
Count	Count value	Percent good		

Mode	Column 1	Column 2	Column 3	Column 4
Mean	Mean value	Percent good		

- An output code of 6 transposes the output array that results from an output code of 4, and results are returned in rows instead of in columns.
- An output code of 7 transposes the output array that results from an output code of 5, and results are returned in rows instead of in columns.

EXAMPLE

The following calculates a time-weighted total for the tag `cdf144` from yesterday to today from the default PI Server:

```
=PICalcVal("cdf144", "y", "t", "total", 1440, 4, )
```

The result is multiplied by 1440. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)

PIAdvCalcVal()

Retrieves a calculated PI point value using the following syntax:

```
PIAdvCalcVal(tagname, stime, etime, mode, calcbasis, minpctgood, cfactor, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name matching the desired PI point for calculation
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
mode (string)	The type of calculation to be used to compute point values
calcbasis (string)	The weighting method used to calculate values
minpctgood (number)	The minimum percentage of good data required to calculate and return a value
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
PIServer (string)	The target PI server

RETURNS

See specific information for *PICalcVal()* outcodes (page 96).

EXAMPLE

The following calculates a time-weighted total for the tag `cdf144` from yesterday to today from the default PI Server:

```
=PIAdvCalcVal("cdf144","y","t","total","time-weighted",50,1,4,)
```

The result is multiplied by 1. The `minpctgood` is 50 percent so result will only be displayed if percent good is 50 percent or greater. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)

PIAdvCalcFilVal()

Retrieves a filtered, calculated PI point value using the following syntax:

```
PIAdvCalcFilVal(tagname, stime, etime, filtextp, mode, calcbasis,  
sampmode, sampfreq, minpctgood, cfactor, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name or names matching the desired PI points for calculation
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
filtextp (string)	The filter expression used to filter results
mode (string)	The type of calculation to be used to compute point values
calcbasis (string)	The weighting method used to calculate values
sampmode (string)	The sampling method used for calculation expressions
sampfreq (string)	The frequency used for <i>interpolated</i> sampling
minpctgood (number)	The minimum percentage of good data required to calculate and return a value
Cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
PIServer (string)	The target PI server

RETURNS

See specific information for *PICalcVal()* *outcodes* (page 96).

EXAMPLE

The following calculates a time-weighted total for the tag `cdf144` from yesterday to today during the time periods when tag `cdm158` is in Manual from the default PI Server:

```
=PIAdvCalcFilVal("cdf144","y","t","'cdm158'=""Manual""", "total", "time-weighted", "pt. compressed", "10m", 50, 1, 4,)
```

The result is multiplied by 1. Since point compressed is the specified sampling mode, the sampling frequency is ignored and the filter expression will be sampled at the compressed events of `cdf144`. The `minpctgood` is 50 percent so result will only be displayed if percent good is 50 percent or greater. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)
- *filter expression* (page 74)

PIAdvCalcExpVal()

Retrieves a calculated PI point value based on an expression, using the following syntax:

```
PIAdvCalcExpVal(expression, stime, etime, mode, calcbasis, sampmode, sampfreq, minpctgood, cfactor, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
<code>expression (string)</code>	A PI calculation expression
<code>stime (string)</code>	The start time for the range of calculation (in PI time format)
<code>etime (string)</code>	The end time (in PI time format)
<code>mode (string)</code>	The type of calculation to be used to compute point values
<code>calcbasis (string)</code>	The weighting method used to calculate values
<code>sampmode (string)</code>	The sampling method used for calculation expressions
<code>sampfreq (string)</code>	The frequency used for <i>interpolated</i> sampling
<code>minpctgood (number)</code>	The minimum percentage of good data required to calculate and return a value
<code>cfactor (number)</code>	The time unit of flow for the source tag, required for <i>Total</i> calculations
<code>outcode (integer)</code>	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
<code>PIServer (string)</code>	The target PI server

RETURNS

See specific information for *PICalcVal()* *outcodes* (page 96).

EXAMPLE

The following calculates a time-weighted total for the expression 'cdf144'+ 'cdt158' from yesterday to today from the default PI Server:

```
=PIAdvCalcExpVal("'cdf144'+ 'cdt158'", "y", "t", "total", "time-weighted", "compressed", "10m", 50, 1, 4,)
```

The result is multiplied by 1. Since compressed is the specified sampling mode, the sampling frequency is ignored and the expression will be sampled at the combined compressed events of cdf144 and cdt158. The minpctgood is 50 percent so result will only be displayed if percent good is 50 percent or greater. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)
- *PI calculation expression* (page 74)

PIAdvCalcExpFilVal()

Retrieves a filtered, calculated PI point value based on an expression, using the following syntax:

```
PIAdvCalcExpFilVal(expression, stime, etime, filtxp, mode,
  calcbasis, sampmode, sampfreq, minpctgood, cfactor, outcode,
  PIServer)
```

ARGUMENTS

Argument	Explanation
expression (string)	A PI calculation expression
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
filtxp (string)	The filter expression used to filter results
mode (string)	The type of calculation to be used to compute point values
calcbasis (string)	The weighting method used to calculate values
sampmode (string)	The sampling method used for calculation expressions
sampfreq (string)	The frequency used for <i>interpolated</i> sampling
minpctgood (number)	The minimum percentage of good data required to calculate and return a value
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
PIServer (string)	The target PI server

RETURNS

See specific information for *PICalcVal()* *outcodes* (page 96).

EXAMPLE

The following calculates a time-weighted total for the expression 'cdf144'+ 'cdt158' when 'productid' = "Product2" from yesterday to today from the default PI Server:

```
=PIAdvCalcExpFilVal("'cdf144'+ 'cdt158'", "y", "t", "'productid'='Product2'", "total", "time-weighted", "compressed", "10m", 50, 1, 4, )
```

The result is multiplied by 1. Since compressed is the specified sampling mode, the sampling frequency is ignored and the expression will be sampled at the combined compressed events of cdf144 and cdt158. The minpctgood is 50 percent so result will only be displayed if percent good is 50 percent or greater. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)
- *PI calculation expression* (page 74)
- *filter expression* (page 74)

PICalcDat()

Retrieves calculated PI point values using the following syntax:

```
PICalcDat(tagname, stime, etime, interval, mode, cfactor, outcode, PIServer)
```

Note: PICalcDat() is generated only by the Calculated Data dialog box in *XP-2003* versions. *2007* versions use PIAdvCalcDat().

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name or names matching the desired PI points for calculation
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
interval (string)	The interval between calculated values (in PI time format)
mode (string)	The type of calculation to be used to compute point values
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, 0-7
PIServer (string)	The target PI server

RETURNS

The *output code* (page 82) argument affects the output as follows:

- 0 produces calculated values along a column
- 1 produces timestamps and values along 2 columns
- 2 produces calculated values along a row

- 3 produces timestamps and values along 2 rows
- 4 produces values and percent good along 2 columns
- 5 produces timestamps, values, and percent good along 3 columns
- 6 produces values and percent good along 2 rows
- 7 produces timestamps, values, and percent good along 3 rows

EXAMPLE

The following retrieves totals for the tag located in cell B1, from midnight yesterday to midnight today at 3-hour intervals from the default PI Server:

```
=PICalcDat(b1,"y","t","3h","total",24,1,)
```

Timestamps are also returned. The values are multiplied by 24 before they are displayed in the spreadsheet.

RELATED TOPICS

- *Calculated Data* (page 33)

PIAdvCalcDat()

Retrieves calculated PI point values using the following syntax:

```
PIAdvCalcDat(tagname, stime, etime, interval, mode, calcbasis,  
minpctgood, cfactor, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name or names matching the desired PI points for calculation
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
interval (string)	The interval between calculated values (in PI time format)
mode (string)	The type of calculation to be used to compute point values
calcbasis (string)	The weighting method used to calculate values
minpctgood (number)	The minimum percentage of good data required to calculate and return a value
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, 0-7
PIServer (string)	The target PI server

RETURNS

See specific information for *PICalcDat()* *outcodes* (page 102).

EXAMPLE

The following retrieves totals for the tag `cdf144`, from midnight yesterday to midnight today at three-hour intervals from the default PI Server:

```
=PIAdvCalcDat("cdf144","y","t","3h","total","time-weighted",50,1,4,)
```

The result is multiplied by 1. The `minpctgood` is 50 percent so result will only be displayed if percent good is 50 percent or greater. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)

PIAdvCalcFilDat()

Retrieves filtered, calculated PI point values using the following syntax:

```
PIAdvCalcFilDat(tagname, stime, etime, interval, filtxp, mode, calcbasis, sampmode, sampfreq, minpctgood, cfactor, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name or names matching the desired PI points for calculation
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
interval (string)	The interval between calculated values (in PI time format)
filtxp (string)	The filter expression used to filter results
mode (string)	The type of calculation to be used to compute point values
calcbasis (string)	The weighting method used to calculate values
sampmode (string)	The sampling method used for calculation expressions
sampfreq (string)	The frequency used for <i>interpolated</i> sampling
minpctgood (number)	The minimum percentage of good data required to calculate and return a value
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, 0-7
PIServer (string)	The target PI server

RETURNS

See specific information for *PICalcDat()* *outcodes* (page 102).

EXAMPLE

The following retrieves totals for the tag `cdf144`, from midnight yesterday to midnight today at 3-hour intervals during the time periods when tag `cdf144` is in Manual from the default PI Server:

```
=PIAdvCalcFilDat("cdf144","y","t","3h","'cdm158'=""Manual""","total",
"time-weighted","pt. compressed","10m",50,1,4,)
```

The result is multiplied by 1. Since point compressed is the specified sampling mode, the sampling frequency is ignored and the filter expression will be sampled at the compressed events of `cdf144`. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)
- *filter expression* (page 74)

PIAdvCalcExpDat()

Retrieves calculated PI point values based on an expression, using the following syntax:

```
PIAdvCalcExpDat(expression, stime, etime, interval, mode,
calcbasis, sampmode, sampfreq, minpctgood, cfactor, outcode,
PIServer)
```

ARGUMENTS

Argument	Explanation
expression (string)	A PI calculation expression
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
interval (string)	The interval between calculated values (in PI time format)
mode (string)	The type of calculation to be used to compute point values
calcbasis (string)	The weighting method used to calculate values
sampmode (string)	The sampling method used for calculation expressions
sampfreq (string)	The frequency used for <i>interpolated</i> sampling
minpctgood (number)	The minimum percentage of good data required to calculate and return a value
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
PIServer (string)	The target PI server

RETURNS

See specific information for *PICalcDat()* *outcodes* (page 102).

EXAMPLE

The following retrieves totals for the expression 'cdf144'+ 'cdt158', from midnight yesterday to midnight today at 3-hour intervals from the default PI Server:

```
=PIAdvCalcExpDat("'cdf144'+ 'cdt158'", "y", "t", "3h", "total", "time-weighted", "compressed", "10m", 50, 1, 4,)
```

The result is multiplied by 1. Since compressed is the specified sampling mode, the sampling frequency is ignored and the expression will be sampled at the combined compressed events of cdf144 and cdt158. The minpctgood is 50 percent so result will only be displayed if percent good is 50 percent or greater. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)
- *PI calculation expression* (page 74)

PIAdvCalcExpFilDat()

Retrieves filtered, calculated PI point values based on an expression, using the following syntax:

```
PIAdvCalcExpFilDat(expression, stime, etime, interval, filtxp, mode, calcbasis, sampmode, sampfreq, minpctgood, cfactor, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
expression (string)	A PI calculation expression
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
interval (string)	The interval between calculated values (in PI time format)
filtxp (string)	The filter expression used to filter results
mode (string)	The type of calculation to be used to compute point values
calcbasis (string)	The weighting method used to calculate values
sampmode (string)	The sampling method used for calculation expressions
sampfreq (string)	The frequency used for <i>interpolated</i> sampling
minpctgood (number)	The minimum percentage of good data required to calculate and return a value
cfactor (number)	The time unit of flow for the source tag, required for <i>Total</i> calculations
outcode (integer)	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
PIServer (string)	The target PI server

RETURNS

See specific information for *PICalcDat()* *outcodes* (page 102).

EXAMPLE

The following calculates a time-weighted total for the expression 'cdf144'+ ' cdt158' when 'productid' = "Product2" at 3-hour intervals from yesterday to today from the default PI Server:

```
=PIAdvCalcExpFilDat("'cdf144'+ ' cdt158'", "y", "t", "3h", "'productid'='Product2'", "total", "time-weighted", "compressed", "10m", 50, 1, 4, )
```

The result is multiplied by 1. Since compressed is the specified sampling mode, the sampling frequency is ignored and the expression will be sampled at the combined compressed events of cdf144 and cdt158. The minpctgood is 50 percent so result will only be displayed if percent good is 50 percent or greater. The percent good is returned next to the total.

RELATED TOPICS

- *Calculated Data* (page 33)
- *PI calculation expression* (page 74)
- *filter expression* (page 74)

PITimeFilter()

Returns the amount of time over specified intervals for which a PI point expression evaluates as *true*, using the following syntax:

```
PITimeFilter(expression, stime, etime, interval, timeunit, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
expression (string)	A PI calculation expression
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
interval (string)	The interval between evaluations (in PI time format)
timeunit (string)	Time unit of the calculated result (in PI time format)
outcode (integer)	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
PIServer (string)	The target PI server

RETURNS

Outcode	Column 1	Column 2
0	Calculated data	
1	Timestamps	Calculated data

An *output code* (page 82) of 2 transposes the output array that results from an output code of 0 and an output code of 3 transposes the output array that results from an output code of 1.

EXAMPLE

The following calculates the amount of time that the expression from cell `$A$1` is true from the PI Server named `thevax` at one-hour intervals, starting from midnight yesterday to midnight today:

```
=PITimeFilter($A$1,"y","t","1h","seconds",1,"thevax")
```

It also displays the timestamps for the start time of each calculation interval. The calculation result is expressed in seconds.

RELATED TOPICS

- *Time Filtered* (page 37)
- *PI calculation expression* (page 74)

PITimeFilterVal()

Returns the amount of time over which a PI point expression evaluates as *true* for a specified time range, using the following syntax:

```
PITimeFilterVal(expression, stime, etime, timeunit, outcode,  
PIServer)
```

ARGUMENTS

Argument	Explanation
expression (string)	A PI calculation expression
stime (string)	The start time for the range of calculation (in PI time format)
etime (string)	The end time (in PI time format)
timeunit (string)	Time unit of the calculated result (in PI time format)
outcode (integer)	An output code, either 0, 1, 2, 4, 5, 6, or 7, and not 3
PIServer (string)	The target PI server

RETURNS

- If the *output code* (page 82) is 0, 1, or 2 only the calculated value is returned.
- Outcode of 3 is not used.
- If the outcode is 4 or 5, the percent good is returned to the right of the value cell.
- If the outcode is 6 or 7, the percent good is returned below the value cell.

EXAMPLE

The following calculates the amount of time that the expression from cell `$A$1` is true from the PI Server named `thevax`, starting from midnight yesterday to midnight today:

```
=PITimeFilterVal($A$1,"y","t","seconds",1,"thevax")
```

The calculation result is expressed in seconds.

RELATED TOPICS

- *Time Filtered* (page 37)
- *PI calculation expression* (page 74)

Tag Functions

PIPointIDToTag()

Retrieves a tag name corresponding to a PI point ID, using the following syntax:

```
PIPointIDToTag(pointid, PIServer)
```

ARGUMENTS

Argument	Explanation
pointid (string)	A PI point ID, which is a unique internal ID not reused upon deletion of a point
PIServer (string)	The target PI server

RETURNS

Tagname

EXAMPLE

The following returns the tagname of the tag with point ID of 123 from server named thevax.

```
=PIPointIDToTag("123","thevax")
```

RELATED TOPICS

- *Point ID to Tag* (page 41)

PIAttributeMaskToTag ()

Retrieves a tag name corresponding to one or more point attributes, using the following syntax:

```
PIAttributeMaskToTag(tagname, descriptor, pointsource, pointtype, pointclass, engunits, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
tagname (string)	Tagname mask
descriptor (string)	Descriptor mask
pointsource (string)	Point Source mask
pointtype (string)	Point Type mask

Argument	Explanation
pointclass (string)	Point Class mask
engunits (string)	Engineering Units mask
outcode (integer)	An output code, either 0 or 2
PIServer (string)	The target PI server

RETURNS

Outcode	Column 1	Column 2
0	Tagname	

An *output code* (page 82) of 2 transposes the output array that results from an output code of 1.

EXAMPLE

The following returns the tagname of the tags which start with "sin" from server named thevax.

```
=PIAttributeMaskToTag("sin*","*","*","*","*","*",0,"thevax")
```

RELATED TOPICS

- *Attribute Mask to Tag* (page 42)

PITagAtt()

Retrieves the value of a tag attribute based on a tag name, using the following syntax:

```
PITagAtt(tagname, attribute, PIServer)
```

ARGUMENTS

Argument	Explanation
tagname (string)	The tag name or names matching the desired attributes
attribute (string)	The desired PI point attribute
PIServer (string)	The target PI server

RETURNS

Tag attribute value

EXAMPLE

The following retrieves from the default PI Server the engineering units for the tag located in cell D1:

```
=PITagAtt(d1,"units",)
```

RELATED TOPICS

- *Tag Attributes* (page 42)

Module Database Functions

PIAliasToTag()

Retrieves a tag name based on a PI point alias, using the following syntax:

```
PIAliasToTag(alias, modulepath, querydate, outcode, PIServer)
```

ARGUMENTS

Argument	Explanation
alias (string)	The module database alias corresponding to the desired PI point
modulepath (string)	The full module path to the alias
querydate (integer)	A date limit such that only aliases created at or after the date are returned
Outcode (integer)	An output code, either 0, 1, or 2
PIServer (string)	The target PI server

RETURNS

Outcode	Column 1	Column 2
0	Tagname	
1	Tagname	Server of tag

An *output code* (page 82) of 2 transposes the output array that results from an output code of 1.

EXAMPLE

The following returns the tagname corresponding to the *Temperature* alias of module *Reactor1* which is a submodule of *Unit1*. *Unit1* is at the root of the module database for a PI server named *thevax*. The module database is queried at current time.

```
=PIAliasToTag("Temperature", "\Unit1\Reactor1", "*", 0, "thevax")
```

RELATED TOPICS

- *Alias to Tag* (page 47)

PIPropertyToValue()

Retrieves the value of a specified PI point property, using the following syntax:

```
PIPropertyToValue(property, modulepath, querydate, PIServer, outcode)
```

ARGUMENTS

Argument	Explanation
Property (string)	The module database property corresponding to the desired PI point
modulepath (string)	The full module path to the property

Argument	Explanation
querydate (integer)	A date limit such that only properties created at or after the date are returned
PIServer (string)	The target PI server
Outcode (integer)	An output code, either 0, 1, or 2

RETURNS

- An *output code* (page 82) of 0 displays the value of the property in a column.
- An output code of 2 displays the value of the property in a row.

EXAMPLE

The following returns the value corresponding to the *Implementer* property of module *Reactor1* which is a submodule of *Unit1*. *Unit1* is at the root of the module database for a PI server named *thevax*. The module database is queried at current time.

```
=PIPropertyToValue("Implementer", "\Unit1\Reactor1", "*", "thevax")
```

RELATED TOPICS

- *Property to Value* (page 48)

Input Functions

PIPutVal()

```
PIPutVal(tagname, value, time stamp, PIServer, outcell)
```

PIPutVal supports all PI 3 data types (including string input to string tags), as well as sub-second data. Timestamps are interpreted using client machine time zone information for PI 3 unless the **Use PI Server Time Zone** is enabled.

ARGUMENTS

Argument	Explanation
tagname (string)	The tag for which you to write the value to PI
value (reference)	The reference to a value to be written, either a string (for digital states) or a number
time stamp (string)	A timestamp in PI time format
PIServer (string)	The target PI server
outcell (reference)	A reference to a cell in which the return value of this macro function is displayed

RETURNS

Returns the string value *False* if the macro function fails and writes an error message to the outcell. If the write to the PI Server succeeds, the string *True* is returned and the status is written to the outcell.

Note: `PIPutVal()` does not verify the user-specified range for a tag before writing. The range is defined from `zero` to `zero + span`, where `zero` and `span` are tag attributes specified during point (tag) creation. `DataLink` returns a value even though *Over Range* or *Under Range* may be entered in PI.

EXAMPLE

The following puts the value located in the cell A5 into the PI Server of the default PI Server for midnight today for the tag `testtag`:

```
=PIPutVal("testtag",a5,"t",,e5)
```

The result of the macro function is placed into cell E5 on the macro sheet.

Appendix C

Technical Support and Resources

You can read complete information about technical support options, and access all of the following resources at the OS/2 Technical Support Web site:

<http://techsupport.osisoft.com>

Before You Call or Write for Help

When you contact OS/2 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 OS/2 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 Contact Us > **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**.

OSIsoft Support Engineers may remotely access your server in order to provide hands-on troubleshooting and assistance. See the Remote Access 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 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.

Index

A

- Add-in, Excel • 1, 9
- Advanced Calculated Data function • 32
- Alias to Tag function • 47
- annotations, show • 19
- apostrophe, use in Excel • 14
- Archive Value function • 26
- arguments, PI Datalink function • 14
- array size • 53
- array, function • 50, 51
- Attribute Mask to Tag function • 42
- average • 32

B

- backwards in time • 27
- boundary type • 27
- browse • 47

C

- Calculated Data function • 32
- calculation basis • 32
- calculation expressions • 74
- calculation mode • 32
- cell references • 14
- cells, spreadsheet
 - function values • 51
 - references • 14
- Compressed Data function • 27
- Connections • 12
- conversion factor • 32
- Current Value function • 25

D

- data access • 70

E

- errors • 83
- expression sampling frequency • 32
- expression sampling mode • 32
- expressions
 - filter • 80
 - PI calculation • 74
 - syntax • 73

F

- filter expressions • 80
- formats, time and number • 16, 19
- functions • 25, 47, 49
 - define • 8, 11, 13
 - Dialog Boxes • 87
 - Entering into Excel • 80
 - Excel • 13
 - function dialog boxes • 11
 - function task panes • 8
 - modify • 52
 - Refreshing PI data in a spreadsheet • 49

M

- Microsoft Excel • 4
- minimum percent good • 32
- Module Database, browse • 45
- move function arrays • 51

N

- notifications, PI system • 63
- number formats • 16, 19, 21

O

- operators • 73
- orientation controls • 19
- OSIsoft Technical Support • 113
- outcode argument • 80, 82
- Output Fields • 14

P

- performance equations • 74
- PI calculation expressions • 74
- PI DataLink • 1
 - for Excel Services • 3
 - interface • 7
 - setup • 69
- PI menu • 8, 10
- PI Module Database • 45
 - browse • 47
 - functions • 47
- PI Notifications • 63
- PI SDK • 2
- PI server connections • 12

- PI time strings • 71
- PIAdvCalcDat • 102
- PIAdvCalcExpDat • 104
- PIAdvCalcExpFilDat • 105
- PIAdvCalcExpFilVal • 100
- PIAdvCalcExpVal • 99
- PIAdvCalcFilDat • 103
- PIAdvCalcFilVal • 98
- PIAdvCalcVal • 98
- PIAliasToTag • 110
- PIArcVal • 88
- PICalcDat • 101
- PICalcVal • 96
- PICompDat • 91
- PICompFilDat • 91
- PICurrVal • 87
- PIExpDat • 94
- PIExpVal • 88
- PINCompDat • 89
- PINCompFilDat • 90
- PIPropertyValue • 111
- PIPutVal • 112
- PISampDat • 92
- PISampFilDat • 93
- PITagAtt • 110
- PITimeDat • 95
- PITimeExpDat • 95
- PITimeFilter • 106
- PITimeFilterVal • 107
- Point Access • 70
- Point ID to Tag function • 41
- Property to Value function • 48

R

- refresh, data • 50
 - automatic • 50
 - manual • 50
- retrieval mode • 31

S

- Sampled Data function • 29
- Settings • 21
- setup • 69
 - configure Excel • 4
 - silent installation • 69
- show annotations • 19
- show percent good • 18
- show timestamp • 18
- show value attributes • 19
- single quote • 14

- spreadsheets • 49
- standard deviation • 32

T

- Tag Attributes function • 42
- tag functions • 41
- Tag Search • 39
- time arguments • 16
- Time Filtered function • 37
- time formats • 16, 71
- time interval • 26
- Time Interval • 29
- Timed Data function • 31
- timestamps • 17, 18
- Totals • 32
- trend • 55
- troubleshooting • 83

V

- value attributes, show • 19