
SAS/ACCESS® Interface to the PI System: Reference, Third Edition
Copyright © 2016, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others’ rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government’s rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

9.4-P3:acpi
Contents

Recommended Reading ......................................................... 71
Index ................................................................................ 73
What’s New in SAS/ACCESS Interface to the PI System

Overview

SAS/ACCESS Interface to the PI System was originally released with SAS 9.4M2. This release included support for the Data Archive in the PI System.

The next release for SAS/ACCESS Interface to the PI System became available in February 2016. This release included support for the Asset Framework and event frames.

Asset Framework

Support for the Asset Framework for the PI System was added in the February 2016 release of SAS/ACCESS Interface to the PI System.

Event Frames

Support for event frames was added in the February 2016 release of the SAS/ACCESS Interface to the PI System. Event frames are supported as part of the PI System Asset Framework.

UNIX Platforms

Support for the PI System on UNIX platforms was added in the February 2016 release of SAS/ACCESS Interface to the PI System. Supported UNIX platforms include AIX, HP-UX for Itanium, Linux for x64, Solaris for SPARC, and Solaris for x64. For more information, see “Supported PI System Features” on page 5.

In SAS 9.4M6, support was added for the SAS_PI_WEB_AUTH= environment variable. Use this environment variable to specify the authentication method that should be used when you connect to the PI System from a UNIX platform. Specifying the authentication method can reduce the amount of time it takes to connect to the PI System.
**Picomp_Summary Virtual Table**

Support for the Picomp_Summary virtual table was added for Microsoft Windows environments for SAS 9.4M4. This table enables calculation of summary statistics over a specified time span. This table is part of the PI System Data Archive.

In the SAS 9.4M6 release, support for the Picomp_Summary table was added for UNIX environments. Calculation of median statistics is not supported UNIX environments, because these environments use the PI Web API to access PI System data.

---

**Data Set Options**

**CALCULATION_BASIS= Data Set Option**

Support for the CALCULATION_BASIS= data set option was added for SAS 9.4M4. This option is used with the Picomp_Summary table.

**LABEL= Data Set Option**

In the February 2016 release of SAS/ACCESS Interface to the PI System, the LABEL= data set option is new. This data set option is an alias for the DESCRIPTION= data set option.

**MEDIAN= Data Set Option**

Support for the MEDIAN= data set option was added for SAS 9.4M4. This option is used with the Picomp_Summary virtual table.

*Note:* Calculation of median statistics is not supported when you access the PI System using the PI Web API.

**MIXED= Data Set Option**

Support for the MIXED= data set option was added for SAS 9.4M4. This option is used with the Picomp virtual table.

**SHOWINDEX= Data Set Option**

Support for the SHOWINDEX= data set option was added for SAS 9.4M4. This option is used with the Picomp virtual table.

**TAGLIST_JOIN= Data Set Option**

Support for the TAGLIST_JOIN= data set option was added for SAS 9.4M4. This option is used with the Picomp and Picomp_Summary virtual tables.
Accessibility

For information about the accessibility of this product, see Accessibility Features of the Windowing Environment for SAS 9.4 at support.sas.com.
What's New in SAS/ACCESS Interface to the PI System
Part 1

SAS/ACCESS Interface to the PI System

Chapter 1
Introduction to the PI System ........................................ 3

Chapter 2
PI System Data Archive .................................................. 13

Chapter 3
Sample Code for the Data Archive .................................... 37

Chapter 4
PI System Asset Framework ............................................ 47

Chapter 5
Sample Code for the Asset Framework ............................... 67
Chapter 1
Introduction to the PI System

About the PI System
PI System Data
PI System Terminology
Supported PI System Features
Methods to Access PI System Data
Functional Differences When You Access PI System Data
  from a UNIX Environment
Security for a Microsoft Windows Environment
Security for a UNIX Environment
PROC DATASETS Considerations
Data Types for the PI System
  Supported Data Types
  SAS Data Types
  Data Conversion from the PI System to SAS
  Data Conversion from SAS to the PI System
  Data Issues between the PI System and SAS
Date Considerations
PI System Null Values and SAS Missing Values
Multi-Byte Character Considerations

About the PI System

PI System Data

The PI System is a repository for time series data. The data is collected from sensors that are attached to various types of monitoring and analysis equipment. For example, a wind-generation farm might have generators with attached sensors that report wind speed, temperature, orientation, and kilowatts of electricity. These timed-data values are sent to a server at predefined intervals and are stored in the PI System for later retrieval.

The PI System is designed to continuously collect and store data from a large number of data sources. These data sources are typically sensors that monitor processes,
production, or status within different types of systems. The PI System consists of these functional layers:

Data Archive
is the actual storage of the PI System time series data. The data is organized as a flat namespace of tags, which are also called PI Points. Each tag consists of a list of events, which is a time series record. Each event stores the tag name, the timestamp of the event, the event value, and status information. Within the context of SAS, a tag is a data set and events are observations.

Support for the Data Archive was provided in the initial release of SAS/ACCESS Interface to the PI System, which was part of SAS 9.4M2.

Asset Framework
is a more recent OSIsoft product that sits on top of the Data Archive. It provides a hierarchical view of the flat namespace of tags and other metadata such as attributes and units of measure.

Support for the Asset Framework was added in the February 2016 release of SAS/ACCESS Interface to the PI System. This release also includes support for PI Event Frames. PI Event Frames track important process events and are supported as part of the Asset Framework.

The PI System is not ODBC-based or SQL-based, so it has no pass-through engine. However, after the data is in a SAS data set, you can process the data using PROC SQL. For sample code to read PI System data, see Chapter 3, “Sample Code for the Data Archive,” on page 37.

**PI System Terminology**

The PI System is not a relational database management system, and there are some terms that are unique to this system.

**element**
is a functional part of the Asset Framework. An element can provide hierarchical structure to the Asset Framework or it can contain data. An element can contain other elements, similar to a directory that contains subdirectories. Alternatively, an element can be a tag (flat data file).

For example, suppose your system contains temperature sensors for multiple buildings in several cities. Your data might be organized as TemperatureSystem → Region → City → Building → Floor → Section. The elements at the Region level might be Northeast, Southeast, Midwest, and Southwest. The element Northeast might contain the elements NewYork, Buffalo, and Boston. At the Section level, the elements are flat files (tags) that contain time series data from temperature sensors in a section of a building.

An element that contains data corresponds to a data set in SAS. An element that contains other elements corresponds to a library in SAS.

**event frame**
tracks process events that are important to your business. Events that you might track include power outages, starting and stopping servers, or readings that are out of specified tolerances. The attributes about an event include the start and end times and any other information that you choose to track about that event.

PI Event Frames is part of the PI Asset Framework.
tag
is a flat data file with data from a sensor or other device. The Data Archive consists of data stored entirely in tags. The Asset Framework consists of elements. Some of those elements are tags that contain sensor data.

A tag corresponds to a data set in SAS.

PI point
is another name for a tag. Some virtual table names refer to PI points.

**Supported PI System Features**

Here are the features that SAS/ACCESS Interface to the PI System supports across available platforms. To find out which versions of your DBMS are supported, see your system requirements documentation.

**Table 1.1  Supported Features for SAS/ACCESS and the PI System**

<table>
<thead>
<tr>
<th>Platform</th>
<th>SAS/ACCESS LIBNAME Statement</th>
<th>Push Down To the PI Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>HP-UX for Itanium</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Linux for x64</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Solaris for SPARC</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Solaris for x64</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Windows 64-bit</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

To improve performance, the processing of some data set options is pushed down to the PI server.

SAS/ACCESS Interface to the PI System does not support these functions:
- passing SAS functions
- passing joins

In addition, there is no bulk-load functionality for the PI System. For the Data Archive on a Windows environment, you can create new tags and add data by using SAS DATA step syntax.

For UNIX platforms, there is no support for writing to the PI System, which includes creating tags, writing data, and deleting tags.

**Methods to Access PI System Data**

If you use SAS/ACCESS Interface to the PI System on a Windows environment, including the PI System client, you use the native API to access the PI System data.

In the February 2016 release of SAS/ACCESS Interface to the PI System, support has been added to run SAS/ACCESS Interface to the PI System on a UNIX environment. In this configuration, SAS runs on a UNIX environment, and the PI Server is on a
Windows machine. When SAS runs on a UNIX environment, you access the data using the PI Web API.

Note: Because the default memory allocation for SAS in a UNIX environment is 2G, you might need to increase memory by setting the MEMSIZE system option to 4G.

### Functional Differences When You Access PI System Data from a UNIX Environment

There are some differences in functionality when you access PI System data from a UNIX environment. To access the PI System from a UNIX environment, you must use the PI Web API:

- You can specify the authentication method to use when connecting to the PI System from UNIX with the SAS PI WEB AUTH= environment variable. By specifying the authentication method, you can improve the time it takes to connect to the PI System via the PI Web API. For more information, see “Specify the Authentication Method for Accessing the PI System” on page 7.

- Connection to the PI System on UNIX requires an SSL certificate file. You specify the location of this file with the SSLCALISTLOC= system option. For more information, see your installation and configuration documentation for SAS/ACCESS Interface to the PI System.

The following authentication-related options are not supported from a UNIX environment:

- DOMAIN=
- PASSWORD=
- USER= (Asset Framework) or USERNAME= (Data Archive)

- Because you use the PI Web API server when you connect to the PI System from UNIX, you must specify the HOST= and PORT= data set options.

- On UNIX, you can only read data from the PI System. You cannot create or modify tags on the PI System.

- Some data set options are not available or are restricted when you access PI System data from UNIX:
  - ATTRIBUTE=. The ATTRIBUTE= option is case-sensitive. Although other options are not case-sensitive when you use the PI Web API, the ATTRIBUTE= option is. Ensure that you specify values with the correct case to match values in the PI System.

    This option is used on the Asset Framework only.

  - COUNT=. For the Data Archive, the COUNT= data set option is available only when you access the Picomp_Summary virtual table. For the Asset Framework, the COUNT= data set option is not available when you access the PI System with the PI Web API.

  - MEDIAN=. Calculation of median statistics is not available when you access PI System data with the PI Web API.

  - TESTDATE=. The TESTDATE= data set option is not available. This option is used on the Data Archive only.

- For UNIX environments, date formats that you use must match the Windows system locale on the PI Web API server. For more information, see “Formatting Date Values” on page 11.
If you are accessing the PI System Data Archive for data that uses multi-byte characters, set the system locale for the PI Web API host machine to the desired encoding. For more information, see “Multi-Byte Character Considerations” on page 12.

Security for a Microsoft Windows Environment

When running on a Microsoft Windows platform, OSIsoft strongly recommends using Integrated Windows Authentication. This authentication uses the credentials of the user’s existing Microsoft Windows session to authenticate with the PI System server. You must configure the PI System server to allow Read, Write, or Read-Write access for users and user groups. The PI System allows access using trusts. OSIsoft does not recommend using an explicit user sign-on using a user ID and password, although this SAS/ACCESS engine does provide options to specify these credentials.

Security for a UNIX Environment

Specify the Location of SSL Certificates

When running on a UNIX platform, you must specify an SSL certificate file to connect to the PI System via the PI Web API. You specify the location of this file with the SSLCALISTLOC= system option. For more information, see your installation and configuration documentation for SAS/ACCESS Interface to the PI System.

Specify the Authentication Method for Accessing the PI System

You can specify the authentication method that should be used when you connect to the PI System from a UNIX platform with the SAS_PI_WEB_AUTH= environment variable. By specifying the authentication method, you can reduce the time it takes to connect to the PI System. You specify environment variables in the SAS invocation command or by using the OPTIONS statement with the SET= option. For more information, see “Defining Environment Variables in UNIX Environments” in SAS Companion for UNIX Environments.

Here are the possible values for the SAS_PI_WEB_AUTH= environment variable:

NONE
indicates that no user name or password is required.

KERBEROS
indicates that Kerberos authentication via Generic Security Services (GSS) is used to connect to the PI System.

If you do not set a value for SAS_PI_WEB_AUTH=, then SAS first attempts to connect to the PI System using no authentication (NONE) and then using Kerberos authentication (KERBEROS). SAS attempts all possible methods to authenticate until a successful attempt is made or until all methods have been attempted. When you set SAS_PI_WEB_AUTH=, that means that SAS first attempts to use the method that you specify and then tries any remaining authentication methods.

PROC DATASETS Considerations

Due to the SAS data set name length restriction of 32 bytes, PROC DATASETS might have limitations if tag names are greater than 32 bytes or if tag names contain special
characters. Although the Tag column displays the full tag name, the Name column can contain generated tag names that do not work with PROC DATASETS commands.

In the following PROC DATASETS output, rows 1–5 contain tag names that comply with SAS rules for data set names and you see the names in the Name column. In rows 6–9, PROC DATASETS determined that the tag name is not a valid SAS name. In these rows, the tag name is longer than 32 bytes or the tag names contain periods (.) and spaces. When PROC DATASETS processes a tag that is not a valid SAS name, it writes a generated tag name in the Name column. Rows 6–9 contain generated tag names in the Name column. It is these generated tag names that are not valid as data set names in PROC DATASETS.

When you issue a PROC DATASETS command, such as DELETE, MODIFY, and APPEND, specify the tag name using the same capitalization that is used for the tag name in the PI System. Otherwise, an error occurs. For example, if you want to delete the CDEP158 tag that was generated by SAS, you must match the case exactly by specifying CDEP158 and not cdep158. If you specified cdep158, PROC DATASETS would not find the data set for this tag. Some of these commands are also available using other procedures, such as PROC DELETE and PROC APPEND, and do not have this limitation.

If your PI Server has more than 10,000 tags, a best practice for performance is to limit the tags of interest. To do this, use a LIBNAME command with the PROC_DATASETS_TAGFILTER= option with a wildcard filter string. This filter is pushed down to the PI Server, resulting in a faster response time. For more information, see “PROC_DATASETS_TAGFILTER=filter-string” on page 25.

---

**Data Types for the PI System**

**Supported Data Types**

All data types for the PI System engine are fixed except for the Value column.

A Value column can be one of these data types:

- INT16
- INT32
- FLOAT16
SAS has two data types, CHARACTER and NUMERIC. SAS character variables (columns) are of a fixed length with a maximum of 32,767 characters. SAS numeric variables are signed 8-byte, floating-point numbers. When SAS numeric values are used in conjunction with SAS formats, they can represent a number of data types, including DATE, TIME, and DATETIME. For more information about SAS data types, see *SAS Language Reference: Concepts*.

**Data Conversion from the PI System to SAS**

SAS reads numeric values as a DOUBLE, TIMESTAMP values as a SAS DATETIME value, and all other values as CHARACTER.

This table shows the SAS and PI System data types and the default SAS formats that are assigned to SAS variables when SAS reads PI System data.

<table>
<thead>
<tr>
<th>PI System Data Type</th>
<th>SAS Data Type</th>
<th>Default SAS Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT16</td>
<td>NUMERIC</td>
<td>none</td>
</tr>
<tr>
<td>INT32</td>
<td>NUMERIC</td>
<td>none</td>
</tr>
<tr>
<td>FLOAT16</td>
<td>NUMERIC</td>
<td>none</td>
</tr>
<tr>
<td>FLOAT32</td>
<td>NUMERIC</td>
<td>none</td>
</tr>
<tr>
<td>FLOAT64 **</td>
<td>NUMERIC</td>
<td>none</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>NUMERIC</td>
<td>DATETIME22.3</td>
</tr>
<tr>
<td>STRING</td>
<td>CHARACTER</td>
<td>$n. *</td>
</tr>
<tr>
<td>BLOB</td>
<td>CHARACTER</td>
<td>$n. *</td>
</tr>
<tr>
<td>DIGITAL</td>
<td>CHARACTER</td>
<td>$n. (Enumeration is resolved to a string.) **</td>
</tr>
</tbody>
</table>

* The length of the format is based on the value of the DBMAX_TEXT= LIBNAME option.
** FLOAT64 is equivalent to a DOUBLE.
**Data Conversion from SAS to the PI System**

The following table shows the PI System data types and table properties that are assigned when SAS creates a table for the PI System.

*Note:* You cannot create data in the PI System if you use the PI Web API to access the PI System. UNIX environments are required to use the PI Web API to access the PI System.

**Table 1.3  Data Types When Writing from SAS to the PI System**

<table>
<thead>
<tr>
<th>SAS Data Type</th>
<th>SAS Format</th>
<th>PI System Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARACTER</td>
<td>$n.</td>
<td>STRING</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>DATETIMEw:d</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td></td>
<td>DATEw:</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td></td>
<td>TIMEw:d</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td></td>
<td>other numeric formats</td>
<td>FLOAT64</td>
</tr>
</tbody>
</table>

**Data Issues between the PI System and SAS**

**Text Length**

When reading PI System data, the default maximum string length for the Value and Status columns is 32 for the Data Archive and 256 for the Asset Framework. If a longer string is read from the PI System, SAS issues a warning message. You can use the DBMAX_TEXT= option to increase the string length to a value that is greater than the default.

**Rounding Numeric Values in Output**

When printing output, the last significant digit in fractions, such as milliseconds, might be off by 1 when comparing values in SAS output and in PI System Explorer output. This can be seen because SAS rounds the last significant value in the output, and the PI System Explorer truncates the last significant value. For example, when printing to three significant digits, the value 0.1235 would be printed in SAS output as 0.124. In PI System Explorer output, this value would appear as 0.123.

*Note:* SAS holds the full precision value in memory, and the full value is used in computations. SAS rounds values only when printing to output.

To ensure agreement to a specific precision in the output, provide a format that displays additional digits of precision, so that the values match to the required precision. A SAS numeric format of 10.5 ensures that SAS and PI System Explorer output match to 3 significant digits.
Date Considerations

**Formatting Date Values**

Dates that you specify using the START= and END= data set options are parsed by the PI System using the Windows system locale setting. To ensure correct parsing of dates, specify these dates using the format for your location. For example, July 1, 2014 might be specified as '7/1/2014' in the United States. In France, you would specify the same date as '1.7.2014'.

*Note:* For UNIX platforms, the date format must match the Windows system locale on the PI Web API server.

Ensure that you specify the month and day correctly for your location. Because the parser can adjust date separators, such as '/',' and '.', based on location, dates that are specified incorrectly might not result in an error. For example, if you specify '7/1/2014' in France, the month and the day are interpreted as January 7 and not July 1. If you want the date to be July 1, you would specify '1/7/2014'. You can use the TESTDATE=LIBNAME option to test how a given date is interpreted on your system. For a sample program, see “Use the TESTDATE Option” on page 45.

**Date Limit for Datetime Values**

For the PI System, date values must fall on or after the PI System baseline date and time, which is January 1, 1970 00:00:00.001 GMT. Typically, datetime values in your data are stored in GMT. For this reason, you must ensure that all datetime values specified in SAS (as local time) fall after January 1, 1970 00:00:00.001 GMT. For example, the local time zone in Beijing is eight hours ahead of GMT, or UTC+8. Therefore, all datetime values in Beijing must fall on or after January 1, 1970 08:00:00.001. This ensures that when a local datetime value is converted to GMT, the date and time do not fall before the PI System baseline time.

**Missing Date Values**

In the PI system, a missing value is represented as the PI System zero value of January 1, 1970 00:00:00.000 GMT. SAS reads missing PI System dates as valid datetime values, so a best practice is to treat these values as missing in your data. Also, remember that times are typically converted to local time, so be sure to adjust January 1, 1970 00:00:00.000 GMT to your local time. In general, any datetime value before January 2, 1970 can be treated as missing.

**PI System Null Values and SAS Missing Values**

The PI System might store NULL values as a result of a bad record status. SAS reads these as SAS missing values and displays them as ".". SAS reads missing strings as blank character strings.

PI System missing datetime values are stored as the zero date value of January 1, 1970 00:00:00.000 GMT. SAS reads missing date values as valid datetime values, so treat these values as missing.

When SAS missing values are written to the Value column, SAS writes "No Data" as the value of the Status column.
Multi-Byte Character Considerations

The PI System engine supports National Language characters, such as Asian character sets that are stored as multi-byte sequences. Perform these tasks for this to work properly:

- Set the Windows System Locale to the desired character encoding. The character encoding should match the encoding of the PI System. Setting the Windows System Locale is typically not a requirement for other SAS/ACCESS engines. However, this step is needed for SAS/ACCESS Interface to the PI System because of the internal workings of the PI System.

  When you run SAS/ACCESS Interface to the PI System on a Windows machine, set the Windows System Locale on that machine. When you run SAS/ACCESS Interface to the PI System on a UNIX machine, set the Windows System Locale on the PI Web API host machine.

- If you are using the PI Web API to access the PI System Data Archive, set the System Locale for the PI Web API host machine to the same encoding as that used for the PI System.

- Start SAS with the desired ENCODING and LOCALE command line options. The ENCODING and LOCALE values that you specify should match those that were used for the PI System.

- Set the options VALIDVARNAME=ANY and VALIDMEMNAME=EXTEND before using any commands that use multi-byte characters.

Tag names can contain multi-byte characters, although this is not fully supported by the PI System. There are some limitations, such as multi-byte characters that do not have the high-order bit set in each byte. If you get an error, use a different name. Also remember that tags that are created in SAS for the Data Archive are limited to 32 bytes (not 32 characters). This limitation affects the maximum length of a multi-byte tag name.

The Status column might contain certain pre-defined status strings, such as "OK" or "No Data". Therefore, specifying multi-byte characters for Status has no effect and are read as "No Data".

The tag descriptor ('member label' in SAS) can store any multi-byte character sequence up to 1024 bytes in length with no other restrictions.

Any string-based Value column value can also consist of any multi-byte character sequence up to 1024 bytes in length with no other restrictions.
Chapter 2  
PI System Data Archive

Data Structure for the Data Archive ............................................. 13
Naming Conventions for the Data Archive .................................... 14
SAS and Data Archive Objects ................................................... 14
PI Server Name Requirements .................................................. 14
Reading and Writing with the Data Archive .............................. 15
Connect to the PI System Server ............................................. 15
Requirements for Writing Data to the Data Archive ................. 15
Read Tags and Metadata Using the Pipoint Virtual Table .......... 16
Read Tag Data Using the Picomp Virtual Table ....................... 16
Best Practices for Performance Optimization ......................... 18
Write Data Archive Data ...................................................... 19
Special Virtual Tables for the Data Archive ............................. 19
Overview of Data Archive Virtual Tables ................................. 19
Generating Summary Statistics in Picomp_Summary ................. 20
Controlling the Date Ranges When Reading from Multiple Tags ... 22
Performance Consideration for the Pipoint Table .................... 22
Performance Considerations for the Picomp_Summary Table ...... 23
LIBNAME Statement Specifics for the Data Archive .................. 23
Overview .............................................................................. 23
Required Arguments .......................................................... 23
LIBNAME Options .............................................................. 23
Data Set Options for the Data Archive ...................................... 27

Data Structure for the Data Archive

In SAS, a PI System tag is a 7-column data set unless the HIDEFLAGS=YES data set option is set. When HIDEFLAGS=YES, the last three columns of the data set are omitted.

These are the PI System data set columns:

Tag
is the name of the PI Point (table). In SAS, a table is referred to as a data set.

Timestamp
is the date and the time, including milliseconds, when the tag value was first recorded. The Timestamp value cannot be null or missing.
Value
is the value of the sample, usually numeric. It could also be a character string or a timestamp.

Status
is the status of the sample. The value can be OK, a status that indicates an error, or another meaningful status.

Questionable
is a flag that indicates whether the event has been marked as questionable.

Annotated
is a flag that indicates whether the event has an annotation. SAS does not handle the annotation itself.

Substituted
is a flag that indicates whether the value has been modified after its original recording.

The Annotated and Substituted columns are read-only. They can be changed only by the PI System when a tag has been annotated or has been substituted.

**Naming Conventions for the Data Archive**

**SAS and Data Archive Objects**

SAS and Data Archive objects include data sets and columns. They follow these naming conventions:

- SAS data set names are limited to 32 characters from A-Z, 0–9, and underscore (_).
  In the PI System, tag names can be up to 1024 characters long and can contain almost any character. If a tag name falls within the conventions of a valid SAS data set name, that tag name can be used as if it were a data set name. Otherwise, you must read from Picomp using the TAGLIST= data set option that contains the set of tags that you want to read.

- Tag names are stored and displayed using the case in which they were created. However, when a tag is read, the name can be specified in any case. Therefore, MyTag and MYTAG refer to the same tag. A tag that was originally created as MyTag is displayed as MyTag.

- Tags that are created in SAS are limited to the conventions of valid SAS data set names.

- Columns names are all predefined (for example, Tag, Timestamp, Value, and Status), and fall within the conventions of valid SAS names.

For general information, see “SAS Names and Support for DBMS Names” in *SAS/ACCESS for Relational Databases: Reference*.

**PI Server Name Requirements**

When you specify a PI System server name in the LIBNAME statement, use the name as it is defined by the user in the PI System SDK Utility. Do not use the (possibly fully qualified) host name. For example, if the PI Server is defined as <myServer>, you must use <myServer> in the LIBNAME= statement, and not <myServer.myCompany.com>. Otherwise, the connection fails.
Reading and Writing with the Data Archive

Connect to the PI System Server

To connect to the PI System server, you use the LIBNAME statement using the engine PISYSTEM. You can use the default PI System server or you can specify the name of the server using the SERVER= LIBNAME option.

```
libname pi pisystem;
libname pi pisystem server=myserver;
```

Although options exist to specify user authentication credentials, the best practice is to default to using Integrated Windows Authentication, which does not require explicit LIBNAME authentication options.

You can use these LIBNAME options to configure the server connection:

- `DBMAX_TEXT=n` specifies the maximum length of a character string for the Value and Status columns.
- `DBPROMPT=[YES | NO]` specifies whether to use dialog boxes for authentication credentials.
- `DEFER=[YES | NO]` specifies whether to wait until a tag is opened to connect to the PI System server.
- `HIDEFLAGS=[YES | NO]` specifies whether to hide in output the columns Questionable, Annotated, and Substituted.
- `SHOWFLAGS=[YES | NO]` specifies whether to display in output the columns Questionable, Annotated, and Substituted.
- `TESTDATE= "PI-System-datetime-string"` is a special test-only option. You use it to specify a PI System DATETIME string so that you can see how the value appears as a SAS DATETIME value. 
  
  *Note:* This option is not supported when you use the PI Web API to access PI System data from a UNIX environment.

- `TIMEZONE=[LOCAL | GMT | UTC]` specifies whether SAS reads and writes PI System timestamps in local or GMT time. The default is LOCAL. Use GMT or UTC only when you want to see the time using GMT.

For more information, see “LIBNAME Statement Specifics for the Data Archive” on page 23.

Requirements for Writing Data to the Data Archive

When you create a Data Archive tag (table), it can contain the following columns, and the names that you use must match the following list:

- (Required) Timestamp. Timestamp values must be numeric and preferably formatted as DATETIME values.
- (Required) Value. Values in the Value column can be of any type.
- (Optional) Status. Values in the Status column must be CHARACTER values. If no Status column is provided, a default status value of OK is used.
• (Optional) Questionable. Values in the Questionable column must be NUMERIC. If no Questionable column is provided, a default value of 0 (false) is used.

The order of the columns can vary, but the names must match the preceding list.

A tag that you create can contain the Tag, Annotated, and Substituted columns, but these columns are not written to a tag on output to the Data Archive. By definition, these columns contain read-only values.

Read Tags and Metadata Using the Pipoint Virtual Table

Using the Pipoint data set, you can use the TAGFILTER= data set option and WHERE clauses to acquire the list of tags that you want to process, along with their metadata.

Using the TAGFILTER= option, you specify a wildcard filter string to select the tags. For tag metadata, see your PI System documentation.

Note: For systems that use the PI Web API to access PI System data, using the PI Web Server 1.7.0 or higher results in improved performance when reading Pipoint data.

This example creates the SAS data set Sinus that contains the tags and their metadata for tag names that begin with 'sinus':

```sas
libname pi pisystem;
data sinus;
  set pi.pipoint (tagfilter="sinus*");
run;
```

This example creates the SAS data set Zero that contains only the tag names for tags where the zero attribute has a value of 100:

```sas
libname pi pisystem;
data zero;
  set pi.pipoint (keep=tag zero where=(zero=100));
run;
```

The two examples above can be combined to create a data set that contains tag names that begin with 'sinus' and have a zero-attribute value of 100.

```sas
libname pi pisystem;
data sinus;
  set pi.pipoint (keep=tag zero tagfilter="sinus" where=(zero=100));
run;
```

Regarding performance, the TAGFILTER= option is processed faster by the PI Server than the WHERE clause that is processed within SAS. For better performance, avoid using WHERE=tag-name.

Read Tag Data Using the Picomp Virtual Table

After you have used Pipoint to acquire the tags that you want to process, you use Picomp to process the tags.

If tag names are longer than 32 characters or if they contain special characters other than the underscore (_), use the TAGLIST= data set option to specify a data set that contains the list of tags to process.

You can use other data set options to subset the data values, or records, that you want to process. For more information, see “Data Set Options for the Data Archive” on page 27.
In this example, SAS prints the records for all tags that begin with 'sinus' and whose Status value is OK:

libname pi pisystem;
data sinus;
   set pi.pipoint (tagfilter="sinus*" edx);
run;
proc print data=pi.picomp (taglist=sinus select=ok);
run;

If you read from a single tag and the tag name complies with the rules for valid SAS data set names, the Picomp virtual table with the TAGLIST= data set option is not required. In this case, use the tag name as the data set name:

libname pi pisystem;
proc print data=pi.sys10149;
run;

To process a subset of records, you use data set options to filter the data. These data set options are specific for the Picomp data set:

- **COUNT=** specifies to interpolate the data and evenly distribute the data between the start and end times using a specific number of observations.
  
  *Note*: The COUNT= option is not supported to access the Picomp table when you access data from the PI Web API on a UNIX environment.

- **END=** specifies an ending time.

- **FILTER=** specifies a PI System filter expression that is used to limit the records that are read. For information about filter expressions, refer to the Performance Equations Reference Manual from OSIsoft.

- **INTERVAL=** specifies to interpolate the data using an interval time.

- **MAX=** specifies a maximum value for Value column data.

- **MIN=** specifies a minimum value for the Value column data.

- **SELECT=** specifies to return all data, data with a specific status, or the last snapshot data.

- **START=** specifies a starting time.

When you use any of these data set options, the filtering is performed on the PI System server. For more information, see “Data Set Options for the Data Archive” on page 27.

You can use other data set options as well. If you use data set options that are not specific for the PI System, then processing is performed by SAS. For more information, see SAS Data Set Options: Reference.

These examples print the data for one day using different filtering criteria:

libname pi pisystem;
/* Print the records of the sinusoid tag for one day */
proc print data=pi.sinusoid (start="06/14/2014" end="06/15/2014");
run;

/* For one day, print the sinusoid tag records with errors. */
proc print data=pi.sinusoid (start="06/14/2014" end="06/15/2014" select=bad);
run;

/* For one day, print the sinusoid tag records that have a minimum value of 10 */
/* and a maximum value of 50 */
proc print data=pi.sinusoid (start="06/14/2014" end="06/15/2014" min=10 max=50); run;
/* For one day, print records interpolated for every hour. */
proc print data=pi.sinusoid (start="06/14/2014" end="06/15/2014" interval="01:00:00"); run;

For more examples, see Chapter 3, “Sample Code for the Data Archive,” on page 37.

**Best Practices for Performance Optimization**

Performance is optimized when you can push down processing to the PI Server. Here are some recommendations:

- Do not use the DBMAX_TEXT option with values larger than necessary for reading the data, especially when reading large data volumes. DBMAX_TEXT defaults to 32 for reading from Picomp, which usually is sufficient.

- Reading from Pipoint:
  - Use the TAGFILTER data set option to reduce the number of records returned from the PI Server.
  - Use the KEEP= data set option to reduce the number of columns that are created. Specify only those columns (tag attributes) that you are interested in. Always include the Tag column. This can speed up the response by a factor of 3 compared to reading all 128 columns. For more information, see "KEEP= Data Set Option" in **SAS Data Set Options: Reference**.
  - Increasing memory by setting the MEMSIZE= system option to a larger value, such as MEMSIZE=4G, can prevent a job from ending due to memory constraints.
  - For systems that use the PI Web API to access the PI System, use the PI Web Server 1.7.0 or higher to improve performance when reading data.

- Reading from Picomp or tag-name:
  - Constrain the time interval with the START= and END= data set options to minimize the number of records that are returned from the PI Server. Using a WHERE clause results in poor performance because all tag data records are read and then partially discarded when SAS processes the WHERE clause. The START= and END= options are processed by the PI Server.
  - Use SELECT=, MIN=, and MAX= data set options when possible, which are also processed by the PI Server.
  - For advanced users of the Data Archive, using the FILTER= data set option is preferable to using a SAS WHERE clause. Refer to the OSIsoft documentation for details about how to formulate complex filter expressions. The SELECT=, MIN=, and MAX= options are used to dynamically build a filter expression and are therefore incompatible with the FILTER option.
  - SAS WHERE clause filtering occurs in SAS. Use the WHERE clause when data set options that push filtering to the PI Server are insufficient. You can combine data set option filtering with SAS WHERE clause filtering.
Write Data Archive Data

Writing Data Archive data is allowed only for new tags or for existing tags that are created in SAS, as determined by the PointSource attribute. Tag names, or data sets, must follow the naming rules for SAS data sets. The TAGLIST= option is ignored. You can write to only one tag at a time.

For examples, see “Create a New Tag from a SAS Data Set” on page 41.

Special Virtual Tables for the Data Archive

Overview of Data Archive Virtual Tables

A common practice when reading data in the Data Archive is to read multiple tags at one time using a filter. This enables you to process a subset of events that are located in multiple tags. Tag names can be up to 1024 characters long, whereas the length of a table name in SAS can be a maximum of 32 characters. Tag names can also contain characters that do not comply with rules for SAS data set names. To accommodate processing PI System tags and reading multiple tags at once, you create a data set that contains the tags that you want to read.

SAS uses these virtual tables in the Data Archive to read a directory of tags and to read the data:

- Pipoint represents a 128-column data set that consists of all Data Archive tags and their attributes. Pipoint can be considered a directory of available tags. You use Pipoint to create a table of tags that you want to read.

- Picomp produces a seven-column data set that contains the time series data that is stored in each tag. When you read from Picomp, you must specify the TAGLIST= data set option. This option specifies the data set that contains the tags that you want to read. This is the data set that you created using the Pipoint virtual table. The tag name is in the Tag column. This enables tag names that do not conform to the SAS data set naming rules to be accessed, as well as reading from multiple tags at once.

- Picomp_Summary generates a report of summary statistics per time interval over a specified time range. For example, you might obtain hourly summary statistics over a 24-hour period. Similar to the Picomp virtual table, use the TAGLIST= data set option and other options to specify the time period and intervals that are of interest to you.

When you process data in the Data Archive, you typically filter Pipoint to build a data set that consists of selected tags. You then use this data set as the value of the TAGLIST= option when you process Picomp or Picomp_Summary. For more information, see “Connect to the PI System Server” on page 15.

For examples, see “Read Multiple Tags Simultaneously” on page 42.
Generating Summary Statistics in Picomp_Summary

Summary Statistics That Are Available in Picomp_Summary

Here is the list of columns in the Picomp_Summary table. This list includes the summary statistics that are calculated and stored in the Picomp_Summary virtual table. The summary statistics are based on a time interval that you specify. For example, if you want a list of hourly summary statistics over the duration of one day, the time interval for the summary statistics is one hour.

Table 2.1 Columns in Picomp_Summary

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>String</td>
<td>Tag name</td>
</tr>
<tr>
<td>StartTime</td>
<td>Timestamp</td>
<td>Start time of a summary interval</td>
</tr>
<tr>
<td>EndTime</td>
<td>Timestamp</td>
<td>End time of a summary interval</td>
</tr>
<tr>
<td>Count</td>
<td>Numeric</td>
<td>Sample count for event-weighted statistics 1/(number of seconds) for time-weighted statistics</td>
</tr>
<tr>
<td>Total</td>
<td>Numeric</td>
<td>Total of sample values in an interval for event-weighted statistics Average of sample values in an interval for time-weighted statistics</td>
</tr>
<tr>
<td>Minimum</td>
<td>Numeric</td>
<td>Smallest sample value in an interval</td>
</tr>
<tr>
<td>MinimumTime</td>
<td>Timestamp</td>
<td>Timestamp of the smallest sample value</td>
</tr>
<tr>
<td>Maximum</td>
<td>Numeric</td>
<td>Largest sample value in an interval</td>
</tr>
<tr>
<td>MaximumTime</td>
<td>Timestamp</td>
<td>Timestamp of the largest sample value</td>
</tr>
<tr>
<td>Range</td>
<td>Numeric</td>
<td>Difference of the Maximum and Minimum values for an interval</td>
</tr>
<tr>
<td>Average</td>
<td>Numeric</td>
<td>Mean of the sample values over an interval</td>
</tr>
<tr>
<td>Median*</td>
<td>Numeric</td>
<td>Median of sample values over an interval Note: Median statistics are not available when you use the PI Web API to access the PI System.</td>
</tr>
<tr>
<td>MedianCount*</td>
<td>Numeric</td>
<td>Number of sample values used to determine the Median Note: Median statistics are not available when you use the PI Web API to access the PI System.</td>
</tr>
</tbody>
</table>
The median statistics are optional, but if requested (via the MEDIAN= data set option), they are calculated by reading all relevant data into SAS. This can significantly increase processing time.

**Specifying the Report Duration and Time Intervals for Summary Statistics**

Use the following data set options to specify the duration of a summary report and to determine the time intervals that are used to calculate summary statistics. The resulting records are saved to the Picomp_Summary table.

- **START=** specifies when the report period begins. To generate summary statistics over a 24-hour period that begins at 8:00 A.M. on March 1, 2016, specify \texttt{START=3/1/2016 08:00:00.000 AM}'.

- **END=** specifies when the report period ends. Using the \texttt{START=} value above, to get summary statistics over a 24-hour period, specify \texttt{END=3/2/2016 07:59:59.999 AM}'.

- **COUNT=** specifies the number of sample intervals to evenly divide the report duration into. You specify the duration of the report with the \texttt{START=} and \texttt{END=} values.

  Interaction: If you specify \texttt{COUNT=} and \texttt{INTERVAL=} for a tag, then \texttt{INTERVAL=} takes precedence.

- **INTERVAL=** specifies the length of the time interval to use for summary statistics. If the interval that you specify does not divide evenly into the duration specified by \texttt{START=} and \texttt{END=} , then remaining records are discarded. For example, if you specify a 5-hour interval for a report that spans 24 hours, then four summary records are stored in Picomp_Summary, and data from the remaining four hours is not summarized. For this reason, for a report that spans 24 hours, the recommended intervals are 1, 2, 3, 4, 6, 8, or 12 hours.

  Interaction: If you specify \texttt{INTERVAL=} and \texttt{COUNT=} for a tag, then \texttt{INTERVAL=} takes precedence.

**Calculating Median Values for Picomp_Summary**

There is no median statistic calculated in the PI System. Therefore, before you can calculate median values, SAS/ACCESS Interface to the PI System must read the data into SAS. This might require reading a very large number of records and might significantly increase computation times. For this reason, the Median and MedianCount columns in the Picomp_Summary table are disabled by default.

*Note:* Median statistics are not supported when you use the PI Web API to access the PI System.
Here are the data set options that control calculation of median statistics:

- **MEDIAN=** specifies whether to calculate the median statistics. Use this data set option to control the number of data points to use when calculating median statistics. For more information, see “MEDIAN=NO | YES | count” on page 32.

- **CALCULATION_BASIS=** specifies whether to use observed events or to interpolate data points across a summary interval. For more information, see “CALCULATION_BASIS=method” on page 27.

*Note:* Only values of EventWeighted or TimeWeighted are valid when calculating median statistics.

**Controlling the Date Ranges When Reading from Multiple Tags**

You can specify dates to limit the records that are read from a tag. For an individual tag, you can use the **START=** and **END=** data set options to specify the date range of interest. However, if you are reading from multiple tags, you typically list those tags in a data set that you reference with the **TAGLIST=** data set option. You can also specify start and end dates for each tag in Start and End columns in the **TAGLIST=** data set. Values in the Start and End columns in the **TAGLIST=** data set override any values given in the **START=** or **END=** data set options for the referenced tag. For more information, see “Data Set Options for the Data Archive” on page 27.

Suppose that you need to create Start and End columns for some other purpose in the data set that you reference with the **TAGLIST=** data set option. In this case, you can use **Start_Time** and **End_Time**, respectively, to specify the date range to use when you read from a tag. If there is a **Start** and **Start_Time** column, then the **Start_Time** column takes precedence over the **Start** column. Similarly, the **End_Time** column takes precedence over the **End** column.

You can specify the date value in the Start and End (or Start_Time and End_Time) columns in the following ways:

- as a character string. Use any timestamp string that is acceptable by the PI System. This is the same format that you specify for the **START=** and **END=** data set options. The PI System expects timestamps to be in a localized format, based on the settings for your system. For example, a system in the United States might require a value such as "1/20/2014 00:00:00". However, a system in France might require a value such as "20.1.2014 00:00:00".

- as a SAS date, time, or datetime value with related formats. The value that you supply is used as given. For example, you might supply a datetime value as '01JAN2014 12:00:00'dt or a date value as '01JAN2014'd.

- as a numeric value. This is not recommended and generates a warning in the log file. Numeric values are interpreted as the number of seconds since January 1, 1960 00:00:00, which is the SAS baseline date. (This differs from the PI System baseline date of January 1, 1970 00:00:00.)

For examples, see “Create a **TAGLIST=** Data Set with SAS Datetime Ranges” on page 44 and “Create a **TAGLIST=** Data Set with Text Date Ranges” on page 44.

**Performance Consideration for the Pipoint Table**

The Pipoint table contains 128 columns that include the tag name and additional tag attributes. If you need only the list of tag names in the Pipoint table, it is recommended that you specify the **KEEP=TAG** data set option when you read from Pipoint. Because
the list of tags can be very large, sometimes including 10,000 tags or more, using KEEP=TAG substantially improves performance when you read from this table.

Performance Considerations for the Picomp_Summary Table

The Picomp_Summary table requires possibly extensive calculations in the PI Server. For better performance, use the KEEP= data set option to retain only those columns that you require.

The Median and MedianCount columns are disabled by default because calculation of the median is not performed by the PI System. Therefore, data must be loaded into SAS before the median is calculated. As a best practice, do not enable the Median and MedianCount columns (using the MEDIAN= data set option) unless they are truly required.

Note: Median statistics are not supported when you use the PI Web API to access the PI System.

LIBNAME Statement Specifics for the Data Archive

Overview

This LIBNAME statement associates a libref with the PI System Data Archive and enables you to read and write Data Archive tags.

LIBNAME libref PISYSTEM <SERVER=server-name> <LIBNAME-options>

Required Arguments

libref

specifies any SAS name that serves as an alias to associate SAS with a database, schema, server, or group of data sets and views.

PISYSTEM

specifies the SAS engine name to connect to the PI System Data Archive.

LIBNAME Options

SERVER=<>'server-name'<'>

specifies the PI System server name to connect to. The value server-name is the name of the PI Server as defined in PI System Management Tools. Note that server-name is not necessarily the host name of the machine that is running the PI Server.

<table>
<thead>
<tr>
<th>Aliases</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATASOURCE</td>
<td>the default PI Server specified for the PI System client that is installed on your computer.</td>
</tr>
</tbody>
</table>
Requirement Use quotation marks if `server-name` contains spaces or nonalphanumeric characters or if the server name is an IP address.

**DOMAIN=**`domain-name`<

specifies the domain for authentication. Use this with `USERNAME=` and `PASSWORD=` options.

Default the current domain.

Restriction This option is not supported when you use the PI Web API to access PI System data from a UNIX environment.

Requirement Use quotation marks if `domain-name` contains spaces.

**USERNAME=**`user-name`<

specifies the user name for authentication.

Aliases `UID=`

`USER=`

Default Integrated Windows Authentication

Restriction This option is not supported when you use the web API to access PI System data from a UNIX environment.

Requirement Use quotation marks if `user-name` contains spaces.

**PASSWORD=**`password`

specifies the password for authentication.

Aliases `PASS=`

`PWD=`

Default Integrated Windows Authentication

Restriction This option is not supported when you use the web API to access PI System data from a UNIX environment.

Requirement Enclose the password in quotation marks.

**HOST=**`host-name`

specifies the network host name of the PI Web API server. This server might differ from the server that hosts the PI System database.

UNIX specifics This option is required when you use the PI Web API to access PI System data from a UNIX environment.

**PORT=**`port-ID`

specifies the port ID for the PI Web API server.

Default 443

UNIX specifics This option is applicable only when you use the PI Web API to access PI System data from a UNIX environment.
DBMAX_TEXT=\text{maximum-character-length}

specifies the maximum number of bytes to allocate for character columns. Data with a string value that is longer than the allocated number of bytes is truncated. If a truncation occurs when SAS reads the data, SAS issues a warning and indicates a sufficient value for DBMAX_TEXT=.

The number of bytes that are allocated for all character strings is based on the minimum of these values: the default column size or the specified value of DBMAX_TEXT.

\text{Note: } The length of a GUID column is always 36 bytes.

\begin{tabular}{|l|l|}
\hline
Default & 32 bytes \\
Range & 1–976 \\
Restriction & DBMAX_TEXT= does not apply to the Tag column or to any column when reading from Pipoint. These maximum lengths are determined dynamically at run time. \\
Interaction & The value of the DBMAX_TEXT= data set option overrides the DBMAX_TEXT= LIBNAME option. \\
\hline
\end{tabular}

DBPROMPT=YES | NO

specifies whether to use dialog boxes to enter server, domain, user name, and password information.

YES specifies to prompt the user using dialog boxes for authentication information. When you specify YES, the DEFER= option is set to YES.

NO specifies not to prompt the user for authentication information.

Default NO

DEFER=YES | NO

specifies whether to wait until a tag is opened to connect to the PI System server.

YES specifies to wait until a tag is opened to connect to the PI System server.

NO specifies to not wait to connect to the PI System server.

Defaults NO

\text{If DBPROMPT=YES, the default for DEFER is YES.}

HIDEFLAGS=YES | NO

specifies whether the columns Questionable, Annotated, and Substituted are hidden.

YES Hides the Questionable, Annotated, and Substituted columns.

NO Displays all columns.

Default NO

Interaction The value of the HIDEFLAGS= data set option overrides the HIDEFLAGS= LIBNAME option.

PROC_DATASETS_TAGFILTER=\text{"filter-string"}

specifies a character string, which can include the wildcard characters ‘*’ and ‘?’, that is used to limit the output from the DATASETS procedure.
* represents zero or more of any character.
? represents any single character.

Default "*"

**SHOWFLAGS=**YES | NO

specifies whether the columns Questionable, Annotated, and Substituted display.

YES Displays the Questionable, Annotated, and Substituted columns.
NO Displays only the Tag, Timestamp, Value, and Status columns.

Default YES

Interaction The value of the SHOWFLAGS= data set option overrides the SHOWFLAGS= LIBNAME option.

**TESTDATE=**"PI-System-datetime-string"

in a test environment, specifies a PI System datetime string that is used to test how
the value appears as a SAS datetime value. The PI System allows for calculations in
dates and relative references. For example, "*-1d+2h" is the current time, minus 1
day, plus 2 hours. With this option users can test whether the date strings that are
specified in START, END, or INTERVAL options are what is expected.

Alias TESTTIME=

Restrictions This option does not assign an actual libref and clears any existing
libref of the same name. Ignore the “libref assign failed” error
message when you use this option. Use this option only for testing
SAS commands. Do not use it in production environments.

This option is not supported when you use the PI Web API to access
PI System data from a UNIX environment.

Examples libname pidata pisystem testdate="**";
          libname pidata pisystem testdate="*-1h";
          libname pidata pisystem

**TIMEZONE=**LOCAL | UTC | GMT

specifies whether SAS reads and writes Data Archive timestamps using local time or
GMT.

**Note:** For SAS/ACCESS Interface to the Data Archive, GMT and UTC are
equivalent and can be used as alias values for each other.

The Data Archive internally stores timestamps using GMT. For example, in the Data
Archive, a numeric value of 0 in GMT is 01JAN1970:00:00:00. In Eastern Time, that
GMT value is 31DEC1969:19:00:00.

LOCAL specifies to use the local time to read and write timestamps in
SAS.

GMT | UTC specifies to use GMT to read and write timestamps in SAS.

**CAUTION:**
Data might be sequenced incorrectly for PI tags that are read between 1 A.M. and 2 A.M. when changing from daylight to standard time. This can cause differences when importing and exporting tag data. Timestamps in the range between 1 A.M. and 2 A.M. cannot be properly converted back to UTC, which might result in the data values in that time range being sequenced differently.

<table>
<thead>
<tr>
<th>Alias</th>
<th>TZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>LOCAL</td>
</tr>
<tr>
<td>Interaction</td>
<td>The value of the TIMEZONE= data set option overrides the TIMEZONE= LIBNAME option.</td>
</tr>
</tbody>
</table>

**Data Set Options for the Data Archive**

These data set options are supported by the PI System. Default values are provided where applicable. For other data set options, see *SAS Data Set Options: Reference*.

**CALCULATION_BASIS=method**

specifies the method to determine the number of data points to use when calculating summary statistics. Some methods use the existing data points and some methods require interpolating data points across the summary interval.

Here are the possible values for method:

- **EventWeighted**
  
  specifies that summary statistics evaluate each event (data point) equally. No interpolation of data over the summary interval is done.

  The following rules are implemented with respect to summary interval boundaries:

  - Use events at both boundaries when there is only one summary interval.
  - Include events at the start time of a summary interval when there are multiple summary intervals and the intervals are in ascending time order.
  - Include events at the end time of a summary interval when there are multiple summary intervals and the intervals are in descending time order.

<table>
<thead>
<tr>
<th>Alias</th>
<th>EW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>There must be at least one event within the summary interval to perform a successful calculation. Two events are required to calculate a standard deviation.</td>
</tr>
<tr>
<td>Interaction</td>
<td>When you calculate median statistics, this method is not affected by any count value for the MEDIAN= data set option.</td>
</tr>
</tbody>
</table>

- **EventWeightedExcludeMostRecentEvent**

  specifies that summary calculation behaves as EventWeighted, except in the handling of events at the boundaries of multiple summary intervals. Use this option to prevent events at the interval boundary from being counted in two intervals. With this option value, events at the end time (most recent time) of an interval are not used in that interval.
Aliases EWEMRE

EW_EMRE

EventWeightedExcludeEarliestEvent specifies that summary calculation is similar to EventWeightedExcludeMostRecentEvent, except that events at the start time (earliest time) for an interval are not used in that interval.

Aliases EWEEE

EW_EEE

EventWeightedIncludeBothEnds specifies that events on the boundary at either end of an interval are included in event-weighted calculations.

Aliases EWIBE

EW_IBE

TimeWeighted specifies to weight the values in the calculation by the time over which they apply. Interpolation is based on whether the attribute is stepped. Interpolated events are generated at the interval boundaries if necessary.

For non-median statistics, TimeWeighted specifies that the PI system should interpolate one data point per second over a summary interval.

For median statistics, SAS/ACCESS interpolates one data point per second when MEDIAN=YES. However, SAS/ACCESS does not generate more than 1000 data points over the time span indicated by the INTERVAL= option. When MEDIAN=\texttt{count}, \texttt{count} data points are interpolated.

Alias TW

TimeWeightedContinuous specifies to apply weighting as you do for TimeWeighted, but perform interpolation between values as if they represent continuous data (standard interpolation), whether the value is continuous or discrete.

Aliases TWC

TW_C

TimeWeightedDiscrete specifies to apply weighting as in TimeWeighted, but perform interpolation as if values represent discrete values (stair-step plot), whether the value is continuous or discrete.

Aliases TWD

TW_D

Aliases CALCULATIONBASIS

CALCBASIS
| **COUNT=integer** | for Picomp, specifies the number of evenly spaced observations to interpolate between the START= and END= times. For Picomp_Summary, this option specifies the number of sample intervals to display between the START= and END= times. For example, if START= and END= span 24 hours, then COUNT=24 creates 24 one-hour intervals between START= and END=.

Processing for this option is performed in the PI System. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Defaults</strong></td>
<td>For Picomp, do not interpolate. Display all recorded events between the START= and END= times. For Picomp_Summary, create only one interval that covers the time spanned by START= and END= values.</td>
</tr>
<tr>
<td><strong>Restrictions</strong></td>
<td>The START= and END= options are required when you use this data set option for the Data Archive. This data set option is not valid for reading from Pipoint. When you use the PI Web API to access PI System data from a UNIX environment, you can use the COUNT= option only for the Picomp_Summary table.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>If you specify COUNT=, do not specify INTERVAL=, MIN=, or MAX= data set options.</td>
</tr>
<tr>
<td><strong>See</strong></td>
<td>“INTERVAL='interval-time’” on page 31</td>
</tr>
</tbody>
</table>

| **DBMAX_TEXT=maximum-character-length** | specifies the maximum number of bytes to allocate for character columns. Data with a string value that is longer than the allocated number of bytes is truncated. If a truncation occurs when SAS reads the data, SAS issues a warning and indicates a sufficient value for DBMAX_TEXT=.

The number of bytes that are allocated for all character strings is based on the minimum of these values: the default column size or the specified value of DBMAX_TEXT. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default</strong></td>
<td>32 bytes</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>1–976</td>
</tr>
<tr>
<td><strong>Restriction</strong></td>
<td>This data set option is not valid for reading from Pipoint.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>The value of the DBMAX_TEXT= data set option overrides the DBMAX_TEXT= LIBNAME option.</td>
</tr>
</tbody>
</table>

| **DELAY=number-of-seconds** | specifies the number of seconds to wait before SAS begins to read the data. When data is added to a tag, it often takes up to a second before data is available to read. |
This option facilitates cases where data is added to a tag and is immediately read back.

Default  0

**END=end-time**

specifies the timestamp of the last tag value to read in the form of a PI System timestamp string.

Processing for this option is performed in the PI System.

**Alias**  END_TIME=

**Default**  "12/31/9999 11:59:59 PM"

**Restriction**  This data set option is not valid for reading from Pipoint.

**Notes**  Specifying END="12/23/2014" implies a time of "00:00:00". Specifying a date with no time component for the END= data set option includes records up to and including midnight the day before. To include a given day, specify either the next day with the END data set option or explicitly use END="12/23/2014 11:59:59 PM".

To ensure a correct end time, specify end-time using the same date format as your Windows system locale setting. For example, in the United States, you would specify '7/1/2014' for July 1, 2014. In France, you would specify '1.7.2014'.

**See**  “START=start-time” on page 33

**FILTER=PI-System-filter-expression**

specifies a PI System filter expression that is used to limit the records that are read.

**Restriction**  This data set option is not valid for reading from Pipoint.

**Interaction**  If you specify FILTER=, do not specify MAX=, MIN=, and SELECT= data set options.

**Tip**  To create more complex filters, refer to the Performance Equations Reference Manual from OSIsoft.

**Example**  FILTER="(.' >= 10) AND (.' <= 50)"

**HIDEFLAGS=YES | NO**

specifies whether the columns Questionable, Annotated, and Substituted are hidden.

YES  Hides the Questionable, Annotated, and Substituted columns.

NO  Displays all columns.

**Default**  NO

**Restriction**  This data set option is not valid for reading from Pipoint.

**Interaction**  The value of the HIDEFLAGS= data set option overrides the HIDEFLAGS= LIBNAME option.

**See**  “SHOWFLAGS=YES | NO” on page 33
**INTERVAL='interval-time'**

for Picomp, specifies to interpolate data using the interval *interval-time*. The *interval-time* value that you specify must be a PI System time interval.

For Picomp_Summary, specifies the sample interval for display between the START= and END= times. For example, if START= and END= span 24 hours, then INTERVAL='1:00:00' results in 24 one-hour intervals.

Typically, you use this data set option with the START= and END= data set options. If you do not specify values for START= and END=, the PI System generates results from the earliest recorded record. This could result in a very large number of records.

Processing for this option is performed in the PI System.

### Alias

<table>
<thead>
<tr>
<th>INT=</th>
</tr>
</thead>
</table>

### Defaults

For Picomp, do not interpolate. Display all recorded events between the START= and END= times.

For Picomp_Summary, create only one interval that covers the time spanned by START= and END= values.

### Restriction

This data set option is not valid for reading from Pipoint.

### Interaction

If you specify INTERVAL=, do not specify COUNT=, MIN=, or MAX= data set options.

### Windows specifics

When you access PI System data from a Microsoft Windows environment, the limit for the number of returned records is 1,500,000.

### UNIX specifics

When you use the PI Web API to access PI System data from a UNIX environment, the limit for the number of returned records is 150,000.

### See

“COUNT=integer” on page 29

### Example

'1:00:00' is one hour.

**LABEL='label-string'**

specifies a description of a tag that is created in SAS. This value is used in the descriptor string for a tag when you write to the PI System. Use this option only when you create a new tag or when you modify an existing tag that you created by using SAS. Otherwise, the description is ignored.

### Aliases

<table>
<thead>
<tr>
<th>DESC=</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION=</td>
</tr>
<tr>
<td>DESCRIPTR=</td>
</tr>
</tbody>
</table>

**MAX=value**

sets the maximum value of the Value column to read.

### Alias

| MAXVAL= |

### Default

Do not filter.

### Restriction

This data set option is not valid for reading from Pipoint.
Interaction

If you specify MAX=, do not specify the COUNT=, FILTER, or INTERVAL= data set options.

**MEDIAN=NO | YES | count**

specifies whether to calculate median statistics (Median and MedianCount) in the Picomp_Summary table. Calculation of median statistics requires reading all required data records into SAS/ACCESS, which can significantly increase computation time.

Setting MEDIAN=0 is equivalent to setting MEDIAN=NO. Setting MEDIAN=1 is equivalent to setting MEDIAN=YES.

**NO**

specifies that SAS should not calculate median statistics.

**YES**

specifies to read required data into SAS and calculate median statistics. SAS interpolates 1000 data points over the summary interval or one data point per second over the specified summary interval, whichever is smaller.

**count**

specifies to read required data into SAS and calculate median statistics. SAS interpolates count data points over the summary interval and then calculates the median per interval.

Default **NO**

Restrictions

When you calculate median statistics, the only valid values for CALCULATION_BASIS are EventWeighted or TimeWeighted.

Median statistics are not supported when you use the PI Web API to access the PI System.

**MIN=value**

specifies the minimal value of the Value column to read.

Alias **MINVAL=**

Default **Do not filter.**

Restriction This data set option is not valid for reading from Pipoint.

Interaction If you specify MIN=, do not specify the COUNT=, FILTER=, and INTERVAL= data set options.

**MIXED=YES | NO**

specifies whether to merge data sets with different data types for Value with the Picomp virtual table. The data sets being merged with Picomp are listed in TAGLIST. If the data sets in TAGLIST have different data types for Value, then you must set MIXED=YES. Setting MIXED=YES converts all numeric values into strings if mixed numeric and string values are detected.

Default **NO**

**SELECT=ALL | OK | GOOD | ERROR | BAD | SNAPSHOT | SNAP**

selects which values are displayed.

**ALL** Displays all records.
OK | GOOD Displays only records with a Status column value of OK.

ERROR | BAD Displays only records that do not have a Status column value of OK.

SNAP | SNAPSHOT Displays the last snapshot record.

Default ALL

Restriction This data set option is not valid for reading from Pipoint.

Interaction If you specify SNAP or SNAPSHOT, do not specify the COUNT=, END=, FILTER=, INTERVAL=, MIN=, MAX=, or START= data set options.

SHOWFLAGS=YES | NO specifies whether the columns Questionable, Annotated, and Substituted display.

YES Displays the Questionable, Annotated, and Substituted columns.

NO Displays only the Tag, Timestamp, Value, and Status columns.

Default YES

Restriction This data set option is not valid for reading from Pipoint.

Interaction The value of the SHOWFLAGS= data set option overrides the SHOWFLAGS= LIBNAME option.

See “HIDEFLAGS=YES | NO” on page 30

SHOWINDEX=YES | NO specifies whether to include column _index in the Picomp table that indicates the index corresponding to a timestamp. The index value is 1 when there is only one observation for a given timestamp. The index value increments by one for each additional observation that is recorded with the same timestamp. The _index column enables you to create a unique key for each observation.

Two observations are considered to occur at the same time when their timestamps differ by less than 0.001 seconds (1 ms).

Default NO

START='start-time' specifies the timestamp of the earliest tag value to read in the form of a PI System TIMESTAMP string.

Processing for this option is performed in the PI System.

Alias START_TIME=

Default '1/1/1970 12:00:00.001 AM GMT'

Restrictions If SAS is running on a machine with a positive UTC time zone offset (east of London), do not specify a start time that would convert to a time prior to 1/1/1970 12:00:00.001 AM GMT. As a rule, do not specify a start time before 1/2/1970.
Notes

The earliest start time allowed by the PI System is one millisecond after midnight of 1/1/1970 GMT.

To ensure a correct start time, specify `start-time` using the same date format as your Windows system locale setting. For example, in the United States, you would specify '7/1/2014' for July 1, 2014. In France, you would specify '1.7.2014'.

See "END='end-time'" on page 30

**TAGFILTER="filter-string"**

specifies a character string that is used to filter tags when reading from the Pipoint virtual table. Wildcard characters * and ? are allowed:

* represents zero or more of any character.

? represents any single character.

Default *

Restriction This data option is valid only for reading from Pipoint.

**TAGLIST=data set-name[(data-set-options)]**

specifies a SAS data set name that is used for the selection process when reading from Picomp. The specified data set can contain one or more tag names in the Tag column. If there is no Tag column in the data set specified by the TAGLIST= option, the first column is treated as the Tag column and a warning is printed to the log.

Restriction This data set option is valid for reading from the Picomp data set only.

Requirement `data set-name` must conform to the rules for SAS data set names. For more information, see “Rules for SAS Data Set Names, View Names, and Item Store Names” in *SAS Language Reference: Concepts*.

**TAGLIST_JOIN=YES | NO**

performs a join between the requested table (Picomp or Picomp_Summary) and the table that is listed in TAGLIST. The system joins the regular column layout of the requested table with the data columns from the table in TAGLIST.

Alias TAGLISTJOIN

Default NO

**TIMEZONE=LOCAL | GMT | UTC**

specifies whether SAS reads and writes PI System timestamps using local time or GMT.

Note: For SAS/ACCESS and the PI System, GMT and UTC are equivalent and can be used as alias TIMEZONE= values for each other.

The PI System stores timestamps using GMT. For example, in the PI System, a numeric value of 0 in GMT is 01JAN1970:00:00:00. In Eastern Time, that GMT value is 31DEC1969:19:00:00.
LOCAL specifies to use the local time to read and write timestamps in SAS.

GMT | UTC specifies to use GMT to read and write timestamps in SAS.

**CAUTION:**
Data might be sequenced incorrectly for PI tags that are read between 1 A.M. and 2 A.M. when changing from daylight to standard time. This can cause differences when importing and exporting tag data. Timestamps in the range between 1 A.M. and 2 A.M. cannot be properly converted back to UTC, which might result in the data values in that time range being sequenced differently.

<table>
<thead>
<tr>
<th>Alias</th>
<th>TZ=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>LOCAL</td>
</tr>
<tr>
<td>Interaction</td>
<td>The value of the TIMEZONE= data set option overrides the TIMEZONE= LIBNAME option.</td>
</tr>
</tbody>
</table>
Chapter 3
Sample Code for the Data Archive

Introduction to Sample Code for the Data Archive

References to specific PI tags, such as SINUSOID, SINUSOID1, and SINUSOID2, might apply to data that is not present on your system. In this case, these examples serve as illustrations of the tasks that you can perform with your own data.

Assign the LIBNAME Statement

Use PROC DATASETS and a PROC_DATASETS_TAGFILTER for List Tags
Use PROC CONTENTS to List Tag Metadata
Use Various Options to Print Tag Data
Read and Write Tags
Append Data to a PI System Tag
Create a New Tag from a SAS Data Set
Use PROC SQL
Read Multiple Tags Simultaneously
  Create a TAGLIST= Subset of Tags
  Create a TAGLIST= Data Set with SAS Datetime Ranges
  Create a TAGLIST= Data Set with Text Date Ranges
Use Complex Nested Picomp and Pipoint Virtual Tables
Read PI Data from an Excel Spreadsheet
Use the TESTDATE Option
Print Summaries from the Picomp_Summary Table
Additional Samples
Use PROC DATASETS and a PROC_DATASETS_TAGFILTER for List Tags

```sas
libname db pisystem proc_datasets_tagfilter="sinus*";
proc datasets lib=db;
quit;
```

Use PROC CONTENTS to List Tag Metadata

```sas
libname db pisystem;
proc contents data=db.pipoint varnum;
run;
proc contents data=db.sinusoid varnum;
run;
```
libname db pisystem;

proc print data=db.sinusoid;
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" interval="1:00:00");
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" count=10);
run;
proc print data=db.sinusoid (select=snapshot);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" select=all);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" tz=gmt);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" select=ok);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" select=bad);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" select=ok min=10);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" select=ok max=10);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" select=ok min=10 max=50);
run;
proc print data=db.sinusoid (start="07/08/2014" end="07/09/2014" filter="*.' > 50");
run;

Here is the output when select=snapshot:

<table>
<thead>
<tr>
<th>Obs</th>
<th>Tag</th>
<th>Timestamp</th>
<th>Value</th>
<th>Status</th>
<th>Questionable</th>
<th>Annotated</th>
<th>Substituted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SINUSOID</td>
<td>19JUN2014:17:00:31.000</td>
<td>74.8048</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Read and Write Tags**

When you use the native API, you can use SAS to create tags. You can modify (append or delete) only tags that were originally created by using SAS/ACCESS Interface to the PI System.

When you use the PI Web API, remember that you cannot write to the PI System. Therefore, you cannot create, modify, or delete a tag.

```sas
libname db pisystem;
proc delete data=db.newtag;
run;
data sinusoid;
   set db.sinusoid;
run;

data db.newtag (desc="My new tag");
   set sinusoid;
run;
```

---

**Append Data to a PI System Tag**

When you use the native API, you can modify (append or delete) only tags that were originally created by using SAS/ACCESS Interface to the PI System.
When you use the PI Web API, remember that you cannot write to the PI System. Therefore, you cannot create, modify, or delete a tag.

libname db pisystem;
proc delete data=db.zz;
run;
data db.zz (desc="My new tag");
set db.sinusoid;
run;
proc append base=db.zz data=db.sinusoid;
run;
proc print data=db.zz (obs=100);
run;

Create a New Tag from a SAS Data Set

When you use the PI Web API, remember that you cannot write to the PI System. Therefore, you cannot create, modify, or delete a tag.

/* Use a data type of double. */
/* The PI System does not allow dates that are in the future!!! */
libname db pisystem showflags=no;
proc delete data=db.zz;
run;
data db.zz (desc="Values are double");
  format Timestamp DATETIME22.3;
  Timestamp='01jan2014:10:11:12.123'dt;
  Value=10;
  output;
  Timestamp='02feb2014:11:12:13'dt;
  Value=20;
  output;
  Timestamp='03mar2014:12:13:14'dt;
  Value=30;
  output;
run;
proc print data=db.zz;run;

/* Use a data type of string. */
libname db pisystem showflags=no;
proc delete data=db.zz;
run;
data db.zz (desc="Values are string");
  format Timestamp DATETIME22.3 Value $32.;
  Timestamp = '01jan2014:10:11:12.123'dt;
  Value = "First event";
  output;
  Timestamp = '02feb2014:11:12:13'dt;
  Value = "Second event";
  output;
  Timestamp = '03mar2014:12:13:14'dt;
  Value = "Third event";
  output;
Use PROC SQL

```sas
libname db pisystem;

/* WHERE is processed by SAS */
proc sql;
   select * from db.sinusoid where (Value >= 99);
quit;

/* Same result, but filter processing is pushed to the PI System */
proc sql;
   select * from db.sinusoid (min = 99);
quit;
```

Read Multiple Tags Simultaneously

**Create a TAGLIST= Subset of Tags**

The following code reads all tags that begin with 'sinus', and prints the last entry for each tag (select=snap):

```sas
libname db pisystem;
data list; set db.pipoint (tagfilter="sinus*");
run;

proc print data=db.picomp (taglist=list select=snap);
run;
```
By modifying the LIBNAME statement options, you can hide the three flag columns:

```sas
libname db pisystem showflags=no;
data list; set db.pipoint (tagfilter="sinus*");run;
```

```sas
proc print data=db.picomp (taglist=list select=snap);
run;
```

### The SAS System

<table>
<thead>
<tr>
<th>Obs</th>
<th>Tag</th>
<th>Timestamp</th>
<th>Value</th>
<th>Status</th>
<th>Questionable</th>
<th>Annotated</th>
<th>Substituted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SINUSOID</td>
<td>19JUN2014:17:00:31.000</td>
<td>74.8048</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>sinusoid1</td>
<td>19JUN2014:16:59:59.000</td>
<td>1.7047</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>sinusoid2</td>
<td>19JUN2014:17:00:31.000</td>
<td>74.8048</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>sinusoid3</td>
<td>19JUN2014:17:00:31.000</td>
<td>85.5932</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>sinusoid4</td>
<td>19JUN2014:17:01:29.000</td>
<td>7.3600</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>sinusoid5</td>
<td>19JUN2014:17:01:59.000</td>
<td>34.9829</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>sinusoid6</td>
<td>19JUN2014:16:53:59.000</td>
<td>99.3808</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>sinusoid7</td>
<td>19JUN2014:17:01:59.000</td>
<td>40.0054</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>sinusoid8</td>
<td>19JUN2014:17:01:29.000</td>
<td>5.4636</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>sinusoid9</td>
<td>19JUN2014:17:01:59.000</td>
<td>82.4965</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>SINUSOIDU</td>
<td>19JUN2014:17:00:59.000</td>
<td>0.0018</td>
<td>OK</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Create a TAGLIST= Data Set with SAS Datetime Ranges

```sas
data mytags;
  format Tag $20. Start DATETIME22.3 End DATETIME22.3;
  Tag="SINUSOID"; Start='01JAN2014 12:00:00'dt; End='01JAN2014 18:00:00'dt;
    output;
  Tag="SINUSOID1"; Start='02JAN2014 12:00:00'dt; End='02JAN2014 18:00:00'dt;
    output;
  Tag="SINUSOID2"; Start='03JAN2014 12:00:00'dt; End='03JAN2014 18:00:00'dt;
    output;
run;

proc print data=mytags; run;
```

This code results in two listings. The first listing prints the Mytags data set that you define in the first DATA step. The second listing prints the sensor readings from 12 noon until 6 p.m. on January 1, 2014 through January 3, 2014 for tags SINUSOID, SINUSOID1, and SINUSOID2.

Create a TAGLIST= Data Set with Text Date Ranges

```sas
data mytags;
  format Tag $20.;
  Tag="SINUSOID"; Start="1/1/2014 12:00:00"; End="1/1/2014 18:00:00";
    output;
  Tag="SINUSOID1"; Start="1/2/2014 12:00:00"; End="1/2/2014 18:00:00";
    output;
  Tag="SINUSOID2"; Start="1/3/2014 12:00:00"; End="1/3/2014 18:00:00";
    output;
run;

proc print data=mytags; run;
```

This code results in two listings. The first listing prints the Mytags data set that you define in the first DATA step. The second listing prints the sensor readings from 12 noon to 6 p.m. on January 1, 2014 through January 3, 2014 for the tags SINUSOID, SINUSOID1, and SINUSOID2.

Use Complex Nested Picomp and Pipoint Virtual Tables

```sas
libname db pisystem;
proc compare
  base=db.picomp (taglist=db.pipoint(where=(lower(tag)="sinus")))
  compare=db.picomp (taglist=db.pipoint(tagfilter="sinus"));
run;
```

44 Chapter 3 • Sample Code for the Data Archive
This results in an equal comparison because the WHERE clause and the TAGFILTER= option that is used in the base and compare tables generate the same results.

Read PI Data from an Excel Spreadsheet

```sas
/* Using Excel to provide the list of tags to read */
/* This assumes you have SAS/ACCESS to PCFILES */

libname xls pcfiles path="C:\MyTagList.xlsx";
proc print data=db.picomp (taglist="xls.sheet1$" cab);
run;
```

Use the TESTDATE Option

This output from this example shows how PI System dates are displayed in SAS:

```sas
libname db pisystem
testdate="**"  
testdate="*-1d"
```

Print Summaries from the Picomp_Summary Table

Assume that the List data set was created as shown in “Read Multiple Tags Simultaneously” on page 42. Use the following PRINT statements to print different summary data.

```sas
/* For each tag 'list' dataset, print a single 24-hr event-weighted summary */
proc print data=db.picomp_summary (taglist=list start="8/1/2016" end="8/2/2016" calcbasis=ew);
run;

/* For each tag 'list' dataset, print a single 24-hr time-weighted summary with median */
proc print data=db.picomp_summary (taglist=list start="8/1/2016" end="8/2/2016" calcbasis=tw median=yes);
run;

/* For each tag 'list' dataset, print hourly time-weighted summaries with median*/
proc print data=db.picomp_summary (taglist=list start="8/1/2016" end="8/2/2016" count=24 calcbasis=tw median=yes);
run;

/* For each tag 'list' dataset, print hourly time-weighted summaries with median*/
proc print data=db.picomp_summary (taglist=list start="8/1/2016" end="8/2/2016" interval="1:00:00" calcbasis=tw median=10);
run;
```
Additional Samples

There are additional samples that are delivered with your software. These samples are available in SASHOME/9.4/accesssample/sample.

Note: Because the PI System is read-only under UNIX, there is not a way to deliver additional samples on UNIX systems.
Chapter 4
PI System Asset Framework

LIBNAME Statement Specifics for the Asset Framework
Overview........................................................................ 47
Required Arguments.................................................... 47
LIBNAME Options....................................................... 48
Data Set Options for the Asset Framework............... 50
About the Asset Framework Tables....................... 56
  Element Table......................................................... 56
  Element_Attribute Table........................................... 57
  Eventframe Table.................................................... 58
  Eventframe_Attribute Table........................................ 59
  Eventframe_Tsdata Table........................................... 60
  Member Table.......................................................... 61
  Tsdata_Attribute Table.............................................. 62
  Tsdata_Pipoint Table................................................ 63
  Tsdata_Summary....................................................... 64

LIBNAME Statement Specifics for the Asset Framework
Overview

This section describes the LIBNAME statement options that SAS/ACCESS Interface to
the PI System supports for the Asset Framework.

Here is the LIBNAME statement syntax for accessing the Asset Framework.

LIBNAME libref PIAF <LIBNAME-options>

Required Arguments

libref
  specifies any SAS name that serves as an alias to associate SAS with a database,
schema, or group of tables and views.

PIAF
  specifies the SAS/ACCESS engine name for the PI System Asset Framework
  interface.
LIBNAME Options

Here are the LIBNAME options that you can specify for the PI System Asset Framework interface:

**DATABASE=<'database-name'><'>**

specifies the name of the PI System Asset Framework database. If the database name contains spaces or nonalphanumeric characters, enclose the name in quotation marks.

**Alias**

**SCHEMA=**

Windows default: PI System database server

UNIX default: None. You must supply a value when you use the PI Web API.

**DBMAX_TEXT=number-of-bytes**

specifies the maximum number of bytes to allocate for character columns. Data with a string value that is longer than the allocated number of bytes is truncated.

The number of bytes that is allocated for all character strings is based on the minimum of these values: the default column size or the specified value of DBMAX_TEXT.

*Note:* The length of a GUID column is always 36 bytes.

**Alias**

**TEXTSIZE=**

**Default** 256

**DBPROMPT=YES | NO**

specifies whether to use dialog boxes to prompt for server, domain, user name, and password information.

**Default** NO

**DEFER=YES | NO**

specifies whether to wait until a tag is opened to connect to the PI System Asset Framework server.

**Default** YES

**DOMAIN=<'domain-name'><'>**

specifies the domain for authentication. Use the DOMAIN=, USER=, and PASSWORD= options together, if you specify those options. This option is not required if you use the Integrated Windows Authentication for your system.

**Default** current domain

**Restriction**

Use of the DOMAIN=, USER=, and PASSWORD= options is discouraged by OSIsoft.

**UNIX specifics**

This option does not apply when you access PI System data by using the PI Web API.
HIDEOPTVARS=YES | NO
specifies whether to hide the secondary variables of a data set. To display only the primary variables of a data set, specify YES. To display all of the variables in a data set, specify HIDEOPTVARS=NO.

Default  NO

HOST=host-name
specifies the network host name of the PI Web API server. This server might differ from the server that hosts the PI System database.

UNIX specifics  This option is required when you access PI System data by using the PI Web API.

PASSWORD=<"password">
specifies the password for authentication. Use the DOMAIN, USER, and PASSWORD options together, if you specify those options. This option is not required if you use the Integrated Windows Authentication for your system.

Alias  PASS=

Restriction  Use of the DOMAIN=, USER=, and PASSWORD= options is discouraged by OSIsoft.

UNIX specifics  This option does not apply when you access PI System data by using the PI Web API.

PORT=port-ID
specifies the port ID for the PI Web API server.

Default  443

UNIX specifics  This option is applicable only when you access PI System data by using the PI Web API.

SERVER=<"server-name">
specifies the PI System server name. The value of server-name is the name of the PI server as defined in the PI System Management Tools. If the server name contains spaces or nonalphanumeric characters or if it is an IP address, enclose it in quotation marks.

Windows default  Default PI System server

UNIX default  Value of HOST=

TIMEZONE=LOCAL | GMT | UTC
specifies whether SAS reads and writes PI System timestamps using local time or GMT.

Note:  For SAS/ACCESS Interface to the PI System, GMT and UTC are equivalent and can be used as alias values for each other.

The PI System internally stores timestamps using GMT. For example, in the PI System, a numeric value of 0 in GMT is 01JAN1970:00:00:00. In Eastern Time (in the US), that GMT value is 31DEC1969:19:00:00 (GMT-5 hours).

CAUTION:
Data might be sequenced incorrectly for PI tags that are read between 1 A.M. and 2 A.M. when changing from daylight to standard time. This can
cause differences when importing and exporting tag data. Timestamps in the range between 1 A.M. and 2 A.M. cannot be properly converted back to UTC, which might result in the data values in that time range being sequenced differently.

Default  LOCAL

USER=<user-name!>
specifies the user name for authentication. Use the DOMAIN, USER, and PASSWORD options together, if you specify those options. This option is not required if you use the Integrated Windows Authentication for your system.

Default  none

Restriction  Use of the DOMAIN=, USER=, and PASSWORD= options is discouraged by OSIsoft.

UNIX specifics  This option does not apply when you access PI System data by using the PI Web API.

---

Data Set Options for the Asset Framework

These data set options pertain to the virtual tables in the PI System Asset Framework. For other data set options, see *SAS Data Set Options: Reference*.

**ATTRIBUTE=attribute-name**
specifies the attribute to use for retrieving elements.

Applies to  Element_Attribute, Eventframe_Attribute, Eventframe_Tsdata, Member, Tdata_Attribute, Tdata_Pipoint, Tdata_Summary

**BOUNDARY=INSIDE | OUTSIDE | INTERPOLATED**
specifies the boundary type to use to retrieve PI Point time series data values.

**INSIDE**
specifies to return recorded values from within the requested time period as the first and last values. That is, the first value is on or after the START= time, and the last value is on or before the END= time.

**OUTSIDE**
specifies to return recorded values outside of the requested time period as the first and last values. That is, the first value is on or before the START= time, and the last value is on or after the END= time.

**INTERPOLATED**
specifies to create an interpolated value at the end points of the requested time period if a recorded value does not appear at the START= or END= times.

Applies to  Eventframe_Tsdata, Tdata_Attribute, Tdata_Pipoint

Requirement  The BOUNDARY= option requires the START= and END= data set options.
The BOUNDARY= option is applicable only when MODEL=ARCHIVE.

**CATEGORY=category-name**
specifies the category name to use for retrieving observations with matching element category values.

*Note:* The CATEGORY= data set option pertains to element categories only.
Matching attribute categories are ignored.

** Applies to **
Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Tdata_Attribute, Tdata_Pipoint

**Requirement**
This option is required when you specify SEARCHBY=CATEGORY.

**CHILD=ALL | NO | IMMEDIATE | (LEAVES | LEAF)**
specifies the retrieval behavior when an engine retrieves elements for the target element.

- **ALL** recursively retrieves all child elements. That is, the system retrieves all children, all children of those children, and so on.
- **NO** does not retrieve child elements.
- **IMMEDIATE** retrieves only the immediate child elements.
- **LEAVES | LEAF** retrieves only the children that have no children of their own.

**Default**
ALL

** Applies to **
Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Tdata_Attribute, Tdata_Pipoint, Tdata_Summary

**Interaction**
This option is used only when you also specify SEARCHBY=PATH and the PATH= option.

**COUNT=number**
specifies the number of data values to retrieve.

*Note:* When you specify MODEL=PLOT and a small value for COUNT=, you might receive more records than expected. The PI System supplies enough values to reproduce the shape of the underlying data in a plot. For each interval, the data is examined, and notable values are returned. These notable values include up to five unique values—the first, last, maximum, minimum, and exceptional (bad status or bad digital state) values.

** Applies to **
Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Tdata_Attribute, Tdata_Pipoint, Tdata_Summary

**Interaction**
This option applies when you specify either MODEL=COUNT or MODEL=PLOT.

This option can be used with the MODEL=, START=, END=, and BOUNDARY= options.
UNIX specifics

This option does not apply when you access PI System data by using the PI Web API.

---

**DBMAX_TEXT=number-of-bytes**

specifies the maximum number of bytes to allocate for character columns. Data with a string value that is longer than the allocated number of bytes is truncated.

The number of bytes that are allocated for all character strings is based on the minimum of these values: the default column size or the specified value of DBMAX_TEXT.

*Note:* The length of a GUID column is always 36 bytes.

<table>
<thead>
<tr>
<th>Alias</th>
<th>TEXTSIZE=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>256</td>
</tr>
</tbody>
</table>

**Applies to**

Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Member, Tsdata_Attribute, Tsdata_Pipoint, Tsdata_Summary

**Interaction**

This value overrides the value of the DBMAX_TEXT= LIBNAME option.

---

**END="end-time"**

specifies the end of the time period to use to retrieve PI Point values. Enclose the value in quotation marks.

When you use START= and END= with event frames, SAS returns all of the records from event frames that overlap the START= and END= time period. This means that you are likely to receive records outside of the specified time period.

For more information about time values, see START=.

**Applies to**

Eventframe, Eventframe_Attribute, Tsdata_Summary

---

**GUID=element-ID**

specifies the element ID to use for retrieving an element.

**Applies to**

Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Tsdata_Attribute, Tsdata_Pipoint

**Requirement**

This option is required when you specify SEARCHBY=GUID.

---

**HIDEOPTVARS=YES | NO**

specifies whether to exclude nonessential variables in the data set. To include all of the variables in a data set, specify HIDEOPTVARS=NO. For more information about which variables are hidden when HIDEOPTVARS=YES, see “About the Asset Framework Tables” on page 56.

**Default**

NO

**Applies to**

Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Member, Tsdata_Attribute, Tsdata_Pipoint, Tsdata_Summary
INTERVAL="interval-time"

specifies the time interval to use when retrieving Pipoint values. You can specify
days, hours, minutes, seconds, and milliseconds, and the resulting time intervals are
evenly spaced. The value that you specify must be a PI System time interval.

Typically, you use this data set option with the START= and END= data set options. If you do not specify values for START= and END=, the PI System generates results from the earliest recorded record. This could result in a very large number of records.

When you specify a positive interval, interval calculation begins at the earliest point in the specified time period and intervals are calculated moving forward in time. If you specify a negative interval, then interval calculation begins at the end of the time period and intervals are calculated moving backward in time. The order of returned values still begins at or near the start time and moves forward in time, whether you specify a positive or negative interval.

Default

Do not interpolate and display all recorded events between the START= and END= times.

Applies to

Eventframe, Eventframe_Attribute, Eventframe_Tsdata,
Tdata_Attribute, Tdata_Pipoint, Tdata_Summary

Interaction

This data set option can be used with the MODEL=, START=, END=, and BOUNDARY= options.

Windows specifics

When you access PI System data on a Windows platform, the limit for the number of returned records is 1,500,000.

UNIX specifics

When you use the PI Web API to access PI System data, the limit for the number of returned records is 150,000.

Note

Processing for this option is performed in the PI System.

Example

"1:00:00.000" is one hour.

MODEL=ARCHIVE | COUNT | PLOT | SAMPLED

specifies the data model to use to retrieve PI Point time series data values. This setting applies to the attribute of the PI Point value that is retrieved.

ARCHIVE | ARCHIVE_RANGE

returns a list of compressed values for the requested time range. This value requires use of the START= and END= options. When MODEL=ARCHIVE, do not use the INTERVAL= or COUNT= options.

COUNT | ARCHIVE_COUNT

returns a specified number of compressed values from the time period that you indicate with the START= option. This value requires the START= and COUNT= options. When MODEL=COUNT, do not use the END= or INTERVAL= data set options.

INTERPOLATE | SAMPLED

retrieves interpolated values over the specified time period, based on the sampling interval that you specify. This value requires the START=, END=, and INTERVAL= options. When MODEL=INTERPOLATE, do not use the COUNT= or BOUNDARY= options.

PLOT | PLOT_COUNT

returns the specified number of interpolated values, evenly spaced throughout the requested time period.
This value requires the START=, END=, and COUNT= options. When MODEL=PLOT, do not use the INTERVAL= or BOUNDARY= options.

<table>
<thead>
<tr>
<th>Alias</th>
<th>DATAMODEL=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applies to</td>
<td>Eventframe_Tsdata, Tsdata_Attribute, Tsdata_Pipoint</td>
</tr>
<tr>
<td>Interaction</td>
<td>Use this setting with the BOUNDARY=, START=, END=, COUNT=, and INTERVAL= options.</td>
</tr>
</tbody>
</table>

**NAME= 
<>'element-name<'**

specifies the element name to use for retrieving elements. Enclose the *element-name* value in quotation marks if the name contains characters other than alphanumeric characters or underscores ('_').

You can specify a character pattern for the *element-name* value that includes wildcard characters. You can use the backslash ('\') to escape a wildcard character, or use a double backslash ("\") to match a single backslash character. Character pattern matching follows these rules:

- An empty string matches all element names.
- Characters and numbers must match with an element name exactly.
- The * wildcard character can be placed anywhere in a pattern, and it corresponds to zero or more characters in a pattern.
- The ? wildcard character can be placed anywhere in a pattern, and it corresponds to exactly one character.
- You can specify multiple characters at a single position by placing the possible characters within [ and ] symbols. For example, a[bc] matches either ab or ac. This pattern does not match ad or abd.
- You can exclude multiple characters at a single position by placing them within [ and ] symbols and by preceding them with the ! character. For example, a[!bc] matches ad, but it would not match ab, ac, or abd.

**Applies to** Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Tsdata_Attribute, Tsdata_Pipoint

**Requirement** This option is required when you specify SEARCHBY=NAME.

**PATH="full-element-path"**

specifies the full path to use for retrieving an element and any children.

**Applies to** Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Tsdata_Attribute, Tsdata_Pipoint

**Requirement** This option is required when you specify SEARCHBY=PATH.

**Interaction** When you specify the PATH= option, you can also specify the CHILD= option. The CHILD= option is optional.

**Example** Specify the PATH= option similar to this example:

PATH="\myServer\myDB\MainProcess\myState\Equipment"
SEARCHBY= ATTRIBUTE | CATEGORY | (GUID | ID) | NAME | PATH | 
TEMPLATE 
specifies the search method that locates element objects that are defined in the target 
server and database.

ATTRIBUTE specifies to search for an element based on attributes. This 
value requires that you also specify the ATTRIBUTE= data set 
option.

CATEGORY specifies to search for an element object based on a category. 
This value requires that you also specify the CATEGORY= data 
set option.

GUID | ID specifies to search for an element object by its unique ID. This 
value requires that you also specify the GUID= data set option.

NAME specifies to search for an element object by name. This value 
requires that you also specify the NAME= data set option.

PATH specifies to search for an element object based on its path. This 
value requires that you also specify the PATH= data set option.

TEMPLATE specifies to search for an element object based on a template. 
This value requires that you also specify the TEMPLATE= data 
set option.

Applies to Element, Element_Attribute, Eventframe, Eventframe_Attribute, 
Eventframe_Tsdata, Tsdata_Attribute, Tsdata_Pipoint 

Restriction Do not specify any other search options when you use the 
SEARCHBY= data set option. The only exception is the 
ATTRIBUTE= option, if this is allowed for the virtual table that you 
specify.

Requirement This option is a required option for virtual tables.

START="start-time"
specifies the start of the time period to use to retrieve the PI Point values. Enclose the 
value in quotation marks.

If the START time is before the END time, the resulting values are retrieved in time-
ascending order. Otherwise, values are retrieved in time-descending (reverse) order.

When you use START= and END= with event frames, SAS returns all of the records 
from event frames that overlap the START= and END= time period. This means that 
you are likely to receive records outside of the specified time period.

Time values are interpreted as local time unless a time zone indicator, such as a 
trailing “Z” or “GMT”, indicates a different time zone. For example, you might 
supply any of the following values: "17Oct2014", '10/16/2014 12:00:00', 
"2014-10-17Z", or '2014-10-17 12:00:00gmt'.

You can also specify relative time formats that use “T” (today), “Y” (yesterday), “*” 
(now), “D” (days), or “H” (hours). You can specify the letters in upper or lowercase. 
Here are some examples that use relative time formats:

"T" today (at time 00:00:00.000) 
"Y" yesterday (at time 00:00:00.000) 
"*-7d" Seven days ago at the current time 
"*-1h" One hour before the current time (one hour ago)
Applies to  
Eventframe, Eventframe_Attribute, Tsdata_Summary

Interaction  
You can use this option with the MODEL=, START=, END=, and BOUNDARY= options.

**TEMPLATE=template-name**  
specifies the template to use for retrieving elements.

Applies to  
Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Tsdata_Attribute, Tsdata_Pipoint, Tsdata_Summary

Requirement  
This option is required when you specify SEARCHBY=TEMPLATE.

**TIMEZONE=LOCAL | GMT | UTC**  
specifies the time zone to apply to Datetime values when reading and storing data in SAS. By default, SAS stores Datetime values in local time. This means that a value of 0.0 is interpreted in SAS as 01JAN1960 00:00:00 in SAS. However, the PI System stores Timestamp values in UTC time, which is equivalent to GMT time. In the PI System, a value of 0.0 is interpreted as 01JAN1970 00:00:00 GMT.

When you use LOCAL time, there could be an issue with your data on the night when Daylight Saving Time changes back to Standard Time. On this night, there are two sets of data points that fall between 1 and 2 A.M. This might cause differences when importing and exporting tag data as the timestamps cannot be properly converted back to GMT and might result in the data values being sequenced incorrectly.

Default  
LOCAL

Applies to  
Element, Element_Attribute, Eventframe, Eventframe_Attribute, Eventframe_Tsdata, Member, Tsdata_Attribute, Tsdata_Pipoint, Tsdata_Summary

Interaction  
This value overrides the value of the TIMEZONE= LIBNAME option.

---

### About the Asset Framework Tables

#### Element Table

The Element table is a virtual table in the Asset Framework. This table contains information about elements in the Asset Framework. You can use the following data set options when you work with the Element table:

- CHILD=
- DBMAX_TEXT=
- HIDEOPTVARS=
- SEARCHBY= with related data set options

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE=</td>
<td>NAME=</td>
</tr>
<tr>
<td>CATEGORY=</td>
<td>PATH=</td>
</tr>
<tr>
<td>GUID=</td>
<td>TEMPLATE=</td>
</tr>
</tbody>
</table>
Table 4.1  Columns in the Element Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElementID</td>
<td>GUID</td>
<td>•</td>
</tr>
<tr>
<td>ElementName</td>
<td>Character</td>
<td>•</td>
</tr>
<tr>
<td>Type</td>
<td>Character</td>
<td>•</td>
</tr>
<tr>
<td>Path</td>
<td>Character</td>
<td>•</td>
</tr>
<tr>
<td>HasChildren</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>IsRoot</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Categories</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Template</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Parent</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

**Element_Attribute Table**

The Element_Attribute table is a virtual table that contains information about element attributes. You can use the following data set options when you work with the Element_Attribute table:

- ATTRIBUTE=
- CHILD=
- DBMAX_TEXT=
- HIDEOPTVARS=
- SEARCHBY= with related data set options
  - ATTRIBUTE= NAME=
  - CATEGORY= PATH=
  - GUID= TEMPLATE=
- TIMEZONE=  
### Table 4.2  Columns in the Element_Attribute Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeID</td>
<td>GUID</td>
<td>●</td>
</tr>
<tr>
<td>AttributeName</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>Type</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Value</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Path</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Categories</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Template</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>UnitOfMeasure</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>DataReference</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>ElementID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>ElementPath</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

**Eventframe Table**

The Eventframe virtual table contains a list of event frames and properties for a set of queried event frames. You can specify the following data set options when you work with the Eventframe table:

- CHILD=
- DBMAX_TEXT=
- HIDEOPTVARS=
- SEARCHBY= with related data set options
  - ATTRIBUTE= NAME=
  - CATEGORY= PATH=
  - GUID= TEMPLATE=
- START= and END=
- TIMEZONE=
### Table 4.3  Columns in the Eventframe Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventID</td>
<td>GUID</td>
<td>●</td>
</tr>
<tr>
<td>EventName</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>StartTime</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>EndTime</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>Description</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Categories</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Template</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>PrimaryElementID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>PrimaryElementPath</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

### Eventframe_Attribute Table

The Eventframe_Attribute table contains a list of attributes and properties for a set of queried Event Frames. You can use the following data set options when you work with the Eventframe_Attribute table:

- ATTRIBUTE=
- CHILD=
- DBMAX_TEXT=
- HIDEOPTVARS=
- SEARCHBY= with related data set options
  - ATTRIBUTE= NAME=
  - CATEGORY= PATH=
  - GUID= TEMPLATE=
- START= and END=
- TIMEZONE=

### Table 4.4  Columns of the Eventframe_Attribute Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeID</td>
<td>GUID</td>
<td>●</td>
</tr>
<tr>
<td>AttributeName</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Column</td>
<td>PI System Data Type</td>
<td>Displayed When HIDEOPTVARS=YES</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>AttributePath</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>Type</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Value</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Description</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Categories</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Template</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>PrimaryElementID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>PrimaryElementPath</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>EventID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>EventPath</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

**Eventframe_Tsdata Table**

The Eventframe_Tsdata table contains the time series data for event frames. You can use the following data set options when you work with the Eventframe_Tsdata table:

- ATTRIBUTE=
- CHILD=
- DBMAX_TEXT=
- HIDEOPTVARS=
- MODEL= with related data set options
  - BOUNDARY=
  - INTERVAL=
  - COUNT=
  - START= and END=
- SEARCHBY= with related data set options
  - ATTRIBUTE=
  - NAME=
  - CATEGORY=
  - PATH=
  - GUID=
  - TEMPLATE=
- START= and END=
- TIMEZONE=
**Table 4.5  Columns of the Eventframe_Tsdata Table**

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeID</td>
<td>GUID</td>
<td>●</td>
</tr>
<tr>
<td>AttributeName</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>Value</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>ValueN</td>
<td>Numeric</td>
<td>●</td>
</tr>
<tr>
<td>Status</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Questionable</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Annotated</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Substituted</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>PointClass</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>PointType</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>PIPointID</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>PIPointPath</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>EventID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>EventPath</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

**Member Table**

The Member table contains information about all of the elements in the Asset Framework. The following data set options are valid when you work with the Member table:

- ATTRIBUTE=
- DBMAX_TEXT=
- HIDEOPTVARS=
- TIMEZONE=
Table 4.6  Columns in the Member Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeID</td>
<td>GUID</td>
<td>●</td>
</tr>
<tr>
<td>AttributeName</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>Type</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Value</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Path</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Categories</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Template</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>UnitOfMeasure</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>DataReference</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

**Tsdata_Attribute Table**

The Tsdata_Attribute table is a virtual table that contains time series data for an element attribute. You can use the following data set options when you work with the Tsdata_Attribute table:

- ATTRIBUTE=
- CHILD=
- DBMAX_TEXT=
- HIDEOPTVARS=
- MODEL= with related data set options
  - BOUNDARY=
  - INTERVAL=
  - COUNT=
  - START= and END=
- SEARCHBY= with related data set options
  - ATTRIBUTE=
  - NAME=
  - CATEGORY=
  - PATH=
  - GUID=
  - TEMPLATE=
- TIMEZONE=
### Table 4.7 Columns in the Tsdata_Attribute Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeID</td>
<td>GUID</td>
<td>●</td>
</tr>
<tr>
<td>AttributeName</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>Type</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Value</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Status</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Questionable</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Annotated</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Substituted</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>DataReference</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>PIPointID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>PIPointPath</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>ElementID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>ElementPath</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

### Tsdata_Pipoint Table

The TSDATA_PIPOINT table contains time series data for an element PI point. You can use the following data set options when you work with the TSDATA_PIPOINT table:

- ATTRIBUTE=
- CHILD=
- DBMAX_TEXT=
- HIDEOPTVARS=
- MODEL= with related data set options
  - BOUNDARY=
  - INTERVAL=
  - COUNT=
  - START= and END=
- SEARCHBY= with related data set options
  - ATTRIBUTE=
  - NAME=
  - CATEGORY=
  - PATH=
Table 4.8  Columns in the TSDATA_PIPOINT Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIPointID</td>
<td>Numeric</td>
<td>●</td>
</tr>
<tr>
<td>PIPointPath</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>Value</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>ValueN</td>
<td>Numeric</td>
<td>●</td>
</tr>
<tr>
<td>Status</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Questionable</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Annotated</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Substituted</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>PointClass</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>PointType</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>ElementID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>ElementPath</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>

**Tsddata_Summary**

The TSDATA_SUMMARY virtual table contains a summary of time series data from PI Points in queried elements. You can use the following data set options when you work with the TSDATA_SUMMARY table:

- **ATTRIBUTE=**
- **CHILD=**
- **DBMAX_TEXT=**
- **HIDEOPTVARS=**
- **SEARCHBY=** with related data set options
  
  ATTRIBUTE=  NAME=
  CATEGORY=   PATH=
  GUID=       TEMPLATE=  

Table 4.9  Columns in the TSDATA_SUMMARY Table

<table>
<thead>
<tr>
<th>Column</th>
<th>PI System Data Type</th>
<th>Displayed When HIDEOPTVARS=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeID</td>
<td>GUID</td>
<td>●</td>
</tr>
<tr>
<td>AttributeName</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Statistics</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Value</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Unit</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Status</td>
<td>Character</td>
<td>●</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>●</td>
</tr>
<tr>
<td>PIPointID</td>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>PIPointPath</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>ElementID</td>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>ElementPath</td>
<td>Character</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 5
Sample Code for the Asset Framework

Introduction to Sample Code for the Asset Framework

Because the Asset Framework is read-only, the examples in this chapter work only if the OSIsoft-supplied sample data is installed on your server. References to specific PI tags, such as K_023, might apply to data that is not present on your system. If the referenced PI tags are not present, these examples serve as illustrations of the tasks that you can perform with your own data.

Connect to Asset Framework Server and List Elements

/* Connect to specified server and database */
libname db piaf server=myserver database=mydatabase;

/* List all member elements in the database */
proc datasets lib=db;
run;
quit;

The elements that are listed by the DATASETS procedure show the acceptable SAS element names. Use these names to refer to elements in SAS programs.
List the Attributes for an Element

```
proc print data=db.K_023;
run;
```

The PRINT procedure displays the attributes of the K-023 element.

*Note:* K_023 is the SAS version of the K-023 element name. K_023 is listed when you run the DATASETS procedure to see the element names in a database.

Access Data from an Element and from Virtual Tables

```
proc sql;

* select variables from target member data set;
select Path, AttributeName, Value, Timestamp from db.K_023(hideoptvars=yes);

* find element under specified path;
select ElementName from db.element(searchby=path path="\mymachine\myDB\region" child=all);

* find named attribute and its value under element path;
select Path, Value from db.element_attribute
   (searchby=path path="\mymachine\myDB\region\cityA\System Process" child=leaves)
   where AttributeName = "Power Savings";

* list the recorded time series data of the attributes of elements: H-4*;
select ElementID, AttributeName, Timestamp, Value, DataReference
   from db.tsdata_attribute
   (searchby=name name="H-431" start='y' end='t' datamodel=archive)
   where DataReference = "PI Point";

* list the recorded time series data of the pipoints of element: H-431;
select ElementID, PIPointPath, Timestamp, Value from db.tsdata_pipoint
   (searchby=name name="H-431" start='y' end='t' datamodel=archive);

* list the recorded time series data of the pipoints of attribute: ;
*   Process Feedrate of element: H-431;
select ElementID, PIPointPath, Timestamp, Value from db.tsdata_pipoint
   (searchby=name name="H-431" attribute="Process Feedrate"
       start='y' end='t' datamodel=sampled interval="1d");

* query all templates used in the connect database;
select unique Template from db.element(searchby=name name="*");

* query all categories used in the connect database;
select unique Categories from db.element(searchby=name name="*");```
* list the recorded time series summary data of element "B-235";
  select * from db.tsdata_summary
  {searchby=name name="B-235" start='y' end='t'};

* list the recorded eventframe data for element "B-235";
  select * from db.EventFrame
  {searchby=name name="B-235" start='y' end='t'};

* find event attribute and its value for element "B-235";
  select * from db.EventFrame_Attribute
  {searchby=name name="B-235" start='y' end='t'};

* list the recorded time series event frame data for element "B-235";
  select * from db.EventFrame_Tsdata
  {searchby=name name="B-235" start='y' end='t'};

quit;

libname db clear;

* maximum string length default to 256;
libname db piaf server=user database=user_database dbmax_text=1024;

* to list all member data set (elements) under the connected database;
  * Information for Member Name, Type, Element ID, Element Name, Element Path;
  proc datasets lib=db;
    * to list columns information for B_235 data set;
      contents data=db.B_235 varnum; run;
    quit;

* to list columns information for B_235 data set;
  proc contents data=db.B_235 varnum; run;

* variables selection with extended string length;
  proc contents data=db.B_235(keep=AttributeName Timestamp Value Path dbmax_text=1000)
    varnum; run;
  proc contents data=db.B_235(drop=AttributeID Type UnitOfMeasure) varnum; run;
  proc contents data=db.B_235(hideoptvars=yes) varnum; run;

* use UTC time;
  proc contents data=db.B_235(hideoptvars=yes timezone=gmt) varnum; run;

* data read;
  proc sql;
  select * from db.B_235;
  * select columns;
  select AttributeName, Timestamp, Value, Path from db.B_235;
  * select columns with keep option;
  select * from db.B_235(keep=AttributeName Timestamp Value Path) ;
* set gmt, notice the column name change;
select * from db.B_235(timezone=utc keep=AttributeName Timestamp_gmt Value);
quit;

proc print data=db.B_235; run;
* hide optional columns;
proc print data=db.B_235(hideoptvars=yes); run;
* use utc time;
proc print data=db.B_235(hideoptvars=yes timezone=utc); run;
* use local time - default;
proc print data=db.B_235(hideoptvars=yes timezone=local); run;

* select columns;
proc print data=db.B_235(keep=AttributeName Timestamp Value Path); run;

* refer Timestamp_GMT instead of Timestamp when timezone=gmt;
proc print data=db.B_235(keep=AttributeName Timestamp_GMT Value Path hideoptvars=yes
timezone=gmt); run;

/*Options START= and END= are invalid when used with this virtual table*/
* The START= and END= options affect the snapshot value;
/*This results in a large number of records*/
proc print data=db.Tsdata_attributes(searchby=name name='B-235'
    start="01/01/1970 12:00:00.001 am gmt" end="*" );
run;
proc print data=db.Tsdata_pipoint(searchby=name name='B-235'
    start='1d' end='*'); run;
proc print data=db.Tsdata_pipoint(searchby=name name='B-235'
    start='y' end='t'); run;
proc print data=db.Tsdata_pipoint(searchby=name name='B-235'
    start='y' model=count count=5); run;

---

**Additional Samples**

There are additional code samples that are delivered with your software. These samples are available in `!SASHOME\9.4\accesssample\sample`.

These code samples are read-only and read data that might be available in OSIsoft sample data files. If the sample data files are not available on your system, then the code samples serve as illustrations of code usage.

**Note:** Because the PI System is read-only under UNIX, there is not a way to deliver additional samples on UNIX systems.
Recommended Reading

- SAS/ACCESS for Relational Databases: Reference

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books
Index

A
Asset Framework
element 4
event frames 4
LIBNAME syntax 47
required arguments 47
tag 5
Asset Framework tables 62
Element 56
Element_Attribute 57
Eventframe 58
Eventframe_Attribute 59
Eventframe_Tsdata 60
Member 61
TSDATA_PIPOINT 63
TSDATA_SUMMARY 64
ATTRIBUTE= data set option
Asset Framework 50
authentication
Data Archive 24, 25
authentication, PI System 24

column length 25
columns
required 15
conversion of data
from SAS to the PI System 10
COUNT= data set option
Asset Framework 51
Data Archive 29

D
Data Archive
data set options 27
Picomp table 19
Pipoint table 19
server name requirements 14
tag 5
write data 19
data conversion
issues 10
PI System to SAS 9
SAS to the PI System 10
data set options
ATTRIBUTE= 50
BOUNDARY= 50
CALCULATION_BASIS= 27
CATEGORY= 51
CHILD= 51
COUNT= 29, 51
Data Archive 27
DATAMODEL= 53
DBMAX_TEST= 29
DBMAX_TEXT= 52
DELAY= 29
DESC= 31
DESCRIPTION= 31
DESCRIPTOR= 31
END= 30, 52
FILTER= 30
GUID= 52
HIDEFLAGS= 30
HIDEOPTVARS= 52
INT= 31
INTERVAL= 31, 53

B
best practices 18
BLOB data type
PI System 9
BOUNDARY= data set option
Asset Framework 50

C
CALCBASIS data set option
Data Archive 27
CALCULATION_BASIS= data set option
Data Archive 27
CALCULATIONBASIS data set option
Data Archive 27
CATEGORY= data set option
Asset Framework 51
CHARACTER data type
SAS 9
CHILD= data set option
Asset Framework 51
INDEX

LABEL= 31
MAX= 31
MIN= 32
MINVAL= 32
MODEL= 53
NAME= 54
PATH= 54
SEARCHBY= 55
SELECT= 32
SHOWFLAGS= 33
START_TIME= 33
START= 33, 55
TAGLIST= 34
TIMEZONE= 34, 56
TZ= 34

data types
PI System 8
DATABASE= LIBNAME option
Asset Framework 48
DATASETS procedure
PI System data 7
DBMAX_TEXT= data set option
Asset Framework 52
Data Archive 29
DBMAX_TEXT= LIBNAME option
Asset Framework 48
Data Archive 25
DBPROMPT= LIBNAME option
Asset Framework 48
Data Archive 25
DEFER= LIBNAME option
Asset Framework 48
Data Archive 25
DELAY= data set option
Data Archive 29
DIGITAL data type
PI System 9
DOMAIN= LIBNAME option
Asset Framework 48
Data Archive 24

E
element 4
Element table 56
Element_Attribute table 57
END= data set option
Asset Framework 52
Data Archive 30
environment variables
SAS_PI_WEB_AUTH= 7
event frames 4
Eventframe table 58
Eventframe_Attribute table 59
Eventframe_Tsdata table 60
events 4

F
filter
Data Archive data 25
tags, overview 16
tags, reading from Picomp 34
tags, reading from Pipoint 34
values to display 32
filter Data Archive data 25, 26
FILTER= data set option
Data Archive 30
FLOAT16 data type
PI System 8
FLOAT32 data type
PI System 9
FLOAT64 data type
PI System 9

G
GUID= data set option
Asset Framework 52

H
HIDEFLAGS= data set option
Data Archive 30
HIDEFLAGS= LIBNAME option
Data Archive 25
HIDEOPTVARS= data set option
Asset Framework 52
HIDEOPTVARS= LIBNAME option
Asset Framework 49
HOST= LIBNAME option 49
Data Archive 24

I
INT16 data type
PI System 8
INT32 data type
PI System 8
interpolate data
evenly between start and end time 29
using intervals 31
INTERVAL= data set option
Asset Framework 53
Data Archive 31

L
LABEL= data set option
Data Archive 31
LIBNAME engine 23
LIBNAME options
DATABASE= 48
DBMAX_TEXT= 25, 48
DBPROMPT= 25, 48
DEFER= 25, 48
DOMAIN= 24, 48
HIDEFLAGS= 25
HIDEOPTV ARS= 49
HOST= 24, 49
PASSWORD= 24
PORT= 24, 49
PROC_DATASETS_TAGFILTER= 25
server connection 15
SERVER= 23, 49
SHOWFLAGS= 26
TESTDATE= 26
TESTTIME= 26
TEXTSIZE= 48
TIMEZONE= 26, 49
TZ= 26
USER= 50
USERNAME= 24
LIBNAME statement
PI System Asset Framework 47

M
MAX= data set option
Data Archive 31
maximum Value column length 29
Member table 61
MIN= data set option
Data Archive 32
missing values 11
MODEL= data set option
Asset Framework 53

N
NAME= data set option
Asset Framework 54
naming conventions
PI System 14
National Language support 12
NULL values 11
NUMERIC data type
SAS 9

P
PASS= LIBNAME option
Asset Framework 49
password
Data Archive 24
PASSWORD= LIBNAME option
Asset Framework 49
Data Archive 24
PATH= data set option
Asset Framework 54
performance optimization 18
PI Asset Framework
element 4
PI Event Frames 4
PI Event Frames 4
PI point 5
PI System
asset framework 4
data 3
data archive 4
data conversion from SAS 10
data conversion to SAS 9
data types 8
features 5
naming conventions 14
SAS data set columns 13
security 7
server connection 15
PI System Asset Framework
LIBNAME syntax 47
PI System LIBNAME engine 23
PI System server 23
PI System tag 5
Picomp 19
processing tags 16
Pipoint 19
reading tags 16
PORT= LIBNAME option 24, 49
PROC DATASETS
PI System data 7
PROC_DATASETS_TAGFILTER=
LIBNAME option
Data Archive 25

R
reading tags 16
required LIBNAME arguments
PI System Asset Framework 47
required tag columns 15
rounding vs. truncating
output 10

S
SAS and PI System differences
rounding and truncating 10
SAS data set columns 13
SAS data types 9
SAS to the PI System
data conversion 10
SAS_PI_WEB_AUTH= environment
variable 7
SEARCHBY= data set option
Asset Framework 55
security 7
SELECT= data set option
  Data Archive 32
server
  connecting 25
  name requirements 14
  PI System 23
SERVER= LIBNAME option
  Asset Framework 49
  Data Archive 23
SHOWFLAGS= data set option
  Data Archive 33
SHOWFLAGS= LIBNAME option
  Data Archive 26
SSLCALISTLOC= system option 7
START= data set option
  Asset Framework 55
  Data Archive 33
STRING data type
  PI System 9
system options
  SSLCALISTLOC= 7

T
tag 5
  PI point 5
tag creation
  variable names 15
TAGFILTER= data set option
  Data Archive 34
TAGLIST= data set option
  Data Archive 34
tags
  processing 16
  reading 16
TESTDATE= LIBNAME option
  Data Archive 26
TIMESTAMP data type
  PI System 9
TIMEZONE= data set option
  Asset Framework 56
  Data Archive 34
TIMEZONE= LIBNAME option
  Asset Framework 49
  Data Archive 26
truncating vs. rounding
  output 10
Tsdatta_Attribute table 62
TSDATA_PIPOINT table 63
TSDATA_SUMMARY table 64

U
user name
  Data Archive 24
USER= LIBNAME option
  Asset Framework 50
USERNAME= LIBNAME option
  Data Archive 24

V
variables
  required 15

W
write Data Archive data 19
Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.