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.
# Contents

*Accessibility* ................................................................. vii  
*Recommended Reading* .................................................... ix  

**Chapter 1 • Introduction to the PI System** .................................................. 1  
  Supported SAS Releases ..................................................... 1  
  Supported PI System Features .......................................... 1  
  Security ............................................................................. 2  
  PI System Data .................................................................... 2  
  Special Virtual Tables for the PI System ............................... 3  
  PROC DATASETS Considerations ......................................... 4  
  Date Considerations for Your Location ................................ 5  
  Multi-Byte Character Considerations .................................... 5  

**Chapter 2 • Naming Conventions for the PI System** ................................. 7  
  SAS and PI System Objects .................................................. 7  
  PI Server Name Requirements .............................................. 7  

**Chapter 3 • Data Types for the PI System** .............................................. 9  
  Overview ........................................................................... 9  
  Value Column Data Types .................................................. 9  
  SAS Data Types ............................................................... 10  
  Data Conversion from the PI System to SAS ............................ 10  
  Data Conversion from SAS to the PI System ............................ 11  
  Issues When Converting Data from the PI System to SAS ............ 11  
  PI System Null Values and SAS Missing Values ....................... 11  

**Chapter 4 • Reading and Writing PI System Data** ...................................... 13  
  Connect to the PI System Server ......................................... 13  
  Read Tags and Metadata Using the Pipoint Virtual Table .......... 14  
  Read Tag Data Using the Picomp Virtual Table ......................... 14  
  Best Practices for Performance Optimization ......................... 16  
  Write PI System Data ........................................................ 16  

**Chapter 5 • The LIBNAME Statement for the PI System** ............................ 17  
  Dictionary .......................................................................... 17  

**Chapter 6 • Data Set Options for the PI System** ...................................... 21  
  PI System Data Set Options .................................................. 21  

**Chapter 7 • Sample Code for the PI System** ............................................ 27  
  Assign the LIBNAME Statement ........................................... 27  
  Use PROC DATASETS and a PROC_DATASETS_TAGFILTER for List Tags ....... 27  
  Use PROC CONTENTS to List Tag Metadata ............................. 28  
  Use Various Options to Print Tag Data ................................. 29  
  Read and Write Tags ......................................................... 29  
  Append Data to a PI System Tag ............................................ 30  
  Create a New Tag from a SAS Data Set ................................... 30  
  Use PROC SQL ............................................................... 31  
  Read Multiple Tags Simultaneously ...................................... 31
Use Complex Nested Picomp and Pipoint Virtual Tables .......................... 33
Read PI Data from an Excel Spreadsheet ............................................. 33
Use the TESTDATE Option ................................................................. 33

Index ................................................................................................. 35
For information about the accessibility of this product, see Accessibility Features of the Windowing Environment for SAS 9.4 at support.sas.com.
Recommended Reading

- *SAS/ACCESS for Relational Databases: Reference*

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

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore
Chapter 1
Introduction to the PI System

Supported SAS Releases

The SAS/ACCESS Interface to the PI System is supported for SAS releases beginning with the second maintenance release of SAS 9.4.

Supported PI System Features

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

<table>
<thead>
<tr>
<th>Platform</th>
<th>SAS/ACCESS</th>
<th>Push Down</th>
<th>SQL Pass-</th>
<th>ACCESS</th>
<th>DBLOAD</th>
<th>Bulk-Load</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LIBNAME Statement</td>
<td>To the PI Server</td>
<td>Through Facility</td>
<td>Procedure</td>
<td>Procedure</td>
<td>Support</td>
</tr>
<tr>
<td>Windows 64-bit</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To improve performance, the processing of some data set options that are specific for the PI System is pushed down to the PI server.

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

- SQL or SQL pass-through facility
• passing SAS functions
• passing joins
• bulk loading

In particular, there is no specific bulk-load functionality for the PI System. You can create new tags and add data by using SAS DATA step syntax.

Security

Because all PI System components run on Windows, OSIsoft strongly recommends using Integrated Windows Authentication. This authentication uses the credentials of the user’s existing 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 or 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 engine does provide options to specify these credentials.

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 sensors attached to each generator 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 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.

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.

The SAS/ACCESS Interface to the PI System supports access to only 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 HIDEFLAG=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.

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.

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 7, “Sample Code for the PI
System,” on page 27.

Special Virtual Tables for the PI System

A common practice in processing PI System data is to read multiple tags at one time
using a filter in order 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 provides two virtual PI System tables to read a directory of tags as well as to read
the data:

Pipoint represents a 128-column data set that consists of all PI System 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.

When you process PI System data, 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. For more information, see “Connect to the PI System Server”
on page 13.

For examples, see “Read Multiple Tags Simultaneously” on page 31.
PROC DATASETS Considerations

Due to the SAS data set name length restriction of 32 bytes, PROC DATASETS might have limitations when 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 data set name in the Name column. In rows 6–9, PROC DATASETS determined that the tag name is not a valid SAS name because the tag name length is greater 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.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Member Type</th>
<th>Data Type</th>
<th>Point Class</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CDEP158</td>
<td>DATA</td>
<td>Int32</td>
<td>classic</td>
<td>CDEP158</td>
</tr>
<tr>
<td>2</td>
<td>DataCenterDemoRipple1</td>
<td>DATA</td>
<td>Float32</td>
<td>classic</td>
<td>DataCenterDemoRipple1</td>
</tr>
<tr>
<td>3</td>
<td>DataCenterDemoRipple2</td>
<td>DATA</td>
<td>Float32</td>
<td>classic</td>
<td>DataCenterDemoRipple2</td>
</tr>
<tr>
<td>4</td>
<td>SINUSOID</td>
<td>DATA</td>
<td>Float32</td>
<td>classic</td>
<td>SINUSOID</td>
</tr>
<tr>
<td>5</td>
<td>SINUSOIDU</td>
<td>DATA</td>
<td>Float32</td>
<td>classic</td>
<td>SINUSOIDU</td>
</tr>
<tr>
<td>6</td>
<td>TAG_00008001_USE_TAGLIST_OPTION</td>
<td>DATA</td>
<td>Digital</td>
<td>classic</td>
<td>AMI State 1</td>
</tr>
<tr>
<td>7</td>
<td>TAG_00008002_USE_TAGLIST_OPTION</td>
<td>DATA</td>
<td>Digital</td>
<td>classic</td>
<td>AMI State 2</td>
</tr>
<tr>
<td>8</td>
<td>TAG_00008007_USE_TAGLIST_OPTION</td>
<td>DATA</td>
<td>Float64</td>
<td>classic</td>
<td>Atlanta Data Center Server Rack1.ION 6200 Power Meter 11A</td>
</tr>
<tr>
<td>9</td>
<td>TAG_00008008_USE_TAGLIST_OPTION</td>
<td>DATA</td>
<td>Float64</td>
<td>classic</td>
<td>Atlanta Data Center Server Rack1.ION 6200 Power Meter 11B</td>
</tr>
</tbody>
</table>

When you issue a PROC DATASETS command, such as DELETE, MODIFY, and APPEND, the tag name must be specified in the correct case in which it was stored in the PI System. Otherwise, an error occurs. For example, if the CDEP158 tag is a tag that was generated by SAS that you want to delete, 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 19.
Date Considerations for Your Location

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 '1.7.2014'. 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 33.

Multi-Byte Character Considerations

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

- Set the Windows System Locale to the desired character encoding. This is typically not a requirement for other SAS/ACCESS engines and is needed because of the internal workings of the PI System.
- Start SAS with the desired ENCODING and LOCALE command line options.
- 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 keep in mind that tags that are created in SAS are limited to 32 bytes (not 32 characters), which 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
Naming Conventions for the PI System

SAS and PI System Objects

SAS and PI System 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 and not 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.
Chapter 3
Data Types for the PI System

Overview

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

Value Column Data Types

A Value column can be one of these data types:

- **INT16**
  - specifies a 16-bit signed integer.

- **INT32**
  - specifies a 32-bit signed integer.

- **FLOAT16**
  - specifies a 16-bit floating-point number.

- **FLOAT32**
  - specifies a 32-bit floating-point number.

- **FLOAT64**
  - specifies a 64-bit floating-point number.

- **DIGITAL**
  - specifies an enumeration such as 1="Red", 2="Green", and so on.

- **BLOB**
  - indicates an unspecified sequence of data.
TIMESTAMP
    specifies a timestamp, which includes the date, time, and fractions of a second.

STRING
    specifies a character string of up to 1024 characters.

**SAS Data Types**

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

This table shows the PI System data types and table properties that are assigned when SAS creates a table for 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>other numeric formats</td>
<td></td>
<td>FLOAT64</td>
</tr>
</tbody>
</table>

Issues When Converting Data from the PI System to SAS

When reading PI System data, the default maximum string length for the Value and Status columns is 32, which should cover most cases. 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 greater than 32.

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.

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

When you write to a tag, the value of the Timestamp column might not be a SAS missing value.
Chapter 4
Reading and Writing PI System Data

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 pisistem server=myserver;

Although options exist to specify user authentication credentials, a 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=** specifies the maximum length of a character string for the Value and Status columns.
- **DBPROMPT=** specifies whether to use dialog boxes for authentication credentials.
- **DEFER=** specifies whether to wait until a tag is opened to connect to the PI System server.
- **HIDEFLAGS=** specifies whether to hide in output the columns Questionable, Annotated, and Substituted.
- **SHOWFLAGS=** specifies whether to display in output the columns Questionable, Annotated, and Substituted.
- **TESTDATE=** 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.
• 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” on page 17.

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.

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;
```

To compare the performance between using the TAGFILTER= option and the WHERE clause, the TAGFILTER= option is processed faster by the PI Server than the WHERE clause that is processed by 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 have a length greater than 32 characters or if tag names contain special characters other than the underscore (_), you must use the TAGLIST= data set option to specify a data set name that contains the list of tag names 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 Chapter 6, “Data Set Options for the PI System,” on page 21.
In this example, SAS prints the records for all tags that begin with 'sinus' and whose Status value is OK:

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

You do not need to use the TAGLIST= data set option if you read only from a single tag and the tag name complies with the rules for valid SAS data set names. In this case, use the tag name as the data set name:

```sas
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.
- `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, see the Help that is installed with the PI System at `c:\ProgramFiles (x86)\PIPC\help\PEReference.chm`.
- `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 Chapter 6, “Data Set Options for the PI System,” on page 21.

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:

```sas
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 7, “Sample Code for the PI System,” on page 27.

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.

- 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 PI System, 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 PI System Data

Writing PI System 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 30.
Chapter 5
The LIBNAME Statement for the PI System

Dictionary

LIBNAME Statement
Associates a libref with PI System data and enables you to read and write PI System tags.

Syntax
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.

LIBNAME Options
SERVER=""server-name"
specifies the PI System server name to connect to. server-name is the name of the PI Server as defined in PI System Management Tools. server-name is not necessarily the host name of the machine that is running the PI Server.

Aliases
DATASOURCE
DATASRC

Default
the default PI Server specified for the PI System client that is installed on your computer.

Requirement
Use quotation marks if server-name contains spaces.
**DBMAX_TEXT=maximum-character-length**
sets the maximum length of a character string for the Value and Status columns.

If a truncation occurs when SAS reads the data, SAS issues a warning and indicates a sufficient value for DBMAX_TEXT=.

**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.

**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

**DOMAIN=<'>domain-name'<'>**
specifies the domain for authentication. Use this with USERNAME= and PASSWORD= options.

**Default** the current domain.

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

**USERNAME=<'>user-name'<'>**
specifies the user name for authentication.

**Aliases** UID=

**Default** USER=

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

**PASSWORD='password'**
specifies the password for authentication.

**Aliases** PASS=

**Default** PWD=

**Requirement** Enclose the password in quotation marks.
**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

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=** "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 if the date strings that are specified in START, END, or INTERVAL options are what is expected.

Alias TESTTIME

Restriction 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.
Examples

libname pidata pisystem testdate="*";

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

libname pidata pisystem

**TIMEZONE=**LOCAL | UTC | GMT

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 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, 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.

**Alias**TZ

**Default**LOCAL

**Interaction**The value of the TIMEZONE= data set option overrides the TIMEZONE= LIBNAME option.
## Data Set Options for the PI System

### PI System Data Set Options

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*

#### COUNT=integer

specifies to interpolate data. The results are *integer* observations that are evenly spaced between the START= and END= time.

- **Default**: Do not interpolate and display all recorded events between the START= and END= times.
- **Restrictions**: This data set option is not valid for reading from Pipoint.
- **Note**: Processing for this option is performed in the PI System.
- **See**: “INTERVAL=’interval-time’” on page 23

#### DBMAX_TEXT=maximum-character-length

specifies the maximum length for Value, if it is a string, and Status column strings. If truncation occurs when SAS reads the data, SAS issues a warning and specifies a sufficient value for DBMAX_TEXT=.

- **Default**: 32 bytes
- **Range**: 1–976
- **Restriction**: This data set option is not valid for reading from Pipoint.
- **Interaction**: The value of the DBMAX_TEXT= data set option overrides the DBMAX_TEXT= LIBNAME option.

#### 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

**DESCRIPTION=’description-string’**

specifies a description (label) of a tag that is created in SAS. Use this option only when you create a new tag. Otherwise, the description is ignored.

Aliases DESC=

LABEL=

**END=’end-time’**

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

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 only 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’.

Processing for this option is performed in the PI System.

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

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

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

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

Tip To create more complex filters, use the Help that is installed with the PI System at C:\Program Files (x86)\PIPC\help\PEReference.chm

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.
**INTERVAL**='interval-time'

Specifies to interpolate data using the interval *interval-time*.

**Alias**: INT=

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

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

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

**Requirement**: *interval-time* must be specified using a PI System interval.

**Note**: Processing for this option is performed in the PI System.

**See**: “COUNT=integer” on page 21

**Example**: '1:00:00' is one hour.

**MAX=value**

Sets the maximum value of the Value column to read.

**Alias**: MAXVAL=

**Default**: Do not filter.

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

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

**MIN=value**

Specifies the minimal value of the Value column to read.

**Alias**: MINVAL=

**Default**: Do not filter.

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

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

**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

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

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 22

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

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, that is, east of London, do not specify a start time that would convert to a time prior to 1/1/1970. As a rule, do not specify a start time before 1/2/1970.

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

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'.

Processing for this option is performed in the PI System.

See
“END='end-time'” on page 22
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 set 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.

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

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.

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.

Alias          TZ
Default        LOCAL
Interaction    The value of the TIMEZONE= data set option overrides the TIMEZONE= LIBNAME option.
Chapter 7
Sample Code for the PI System

---

Assign the LIBNAME Statement ................................................... 27
Use PROC DATASETS and a
PROC_DATASETS_TAGFILTER for List Tags .................................. 27
Use PROC CONTENTS to List Tag Metadata .................................. 28
Use Various Options to Print Tag Data ....................................... 29
Read and Write Tags ................................................................. 29
Append Data to a PI System Tag .................................................. 30
Create a New Tag from a SAS Data Set ....................................... 30
Use PROC SQL ............................................................................ 31
Read Multiple Tags Simultaneously .............................................. 31
Use Complex Nested Picomp and Pipoint Virtual Tables ................... 33
Read PI Data from an Excel Spreadsheet ..................................... 33
Use the TESTDATE Option ............................................................ 33

---

Assign the LIBNAME Statement

/* Use the default server. */
libname db pisystem;

/* Specify the server name */
libname db pisystem server=myserver;

---

Use PROC DATASETS and a
PROC_DATASETS_TAGFILTER for List Tags

libname db pisystem proc_datasets_tagfilter="sinus*";
proc datasets lib=db;
quit;
Use PROC CONTENTS to List Tag Metadata

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

The SAS System

The CONTENTS Procedure

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>DB.sinusoid</th>
<th>Observations</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type</td>
<td>DATA</td>
<td>Variables</td>
<td>7</td>
</tr>
<tr>
<td>Engine</td>
<td>PISYSTEM</td>
<td>Indexes</td>
<td>0</td>
</tr>
<tr>
<td>Created</td>
<td>10/14/2013 11:41:30</td>
<td>Observation Length</td>
<td>80</td>
</tr>
<tr>
<td>Last Modified</td>
<td>06/19/2014 17:00:31</td>
<td>Deleted Observations</td>
<td>0</td>
</tr>
<tr>
<td>Protection</td>
<td>Compressed</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Data Set Type</td>
<td>PI POINT</td>
<td>Sorted</td>
<td>NO</td>
</tr>
<tr>
<td>Label</td>
<td>12 Hour Sine Wave</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Representation</td>
<td>Default</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoding</td>
<td>Default</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Variables in Creation Order

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
<th>Informat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tag</td>
<td>Char</td>
<td>8</td>
<td>$.</td>
<td>$.</td>
</tr>
<tr>
<td>2</td>
<td>Timestamp</td>
<td>Num</td>
<td>8</td>
<td>DATETIME22.3</td>
<td>DATETIME22.3</td>
</tr>
<tr>
<td>3</td>
<td>Value</td>
<td>Num</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Status</td>
<td>Char</td>
<td>32</td>
<td>$.</td>
<td>$.</td>
</tr>
<tr>
<td>5</td>
<td>Questionable</td>
<td>Num</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Annotated</td>
<td>Num</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Substituted</td>
<td>Num</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use Various Options to Print Tag Data

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;

The following 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

libname db pisystem;
proc delete data=db.newtag;
run;
data sinusoid;
set db.sinusoid;
run;
data db.newtag (desc="My new tag");
Append Data to a PI System 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

/* 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;
Use PROC SQL

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

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

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:

```
libname db pisystem showflags=no;
```
```
data list; set db.pipoint (tagfilter="sinus*"; run;
```
```
proc print data=db.picomp (taglist=list select=snap); run;
```

<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>82.4985</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>
Use Complex Nested Picomp and Pipoint Virtual Tables

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

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

```
/* 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$");
run;
```

Use the TESTDATE Option

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

```
libname db pisystem
   testdate="*"
   testdate="*-1h"
```
Index

A
authentication, PI System  18

B
best practices  16
BLOB data type
PI System  9

C
CHARACTER data type
SAS  10
column length  18
conversion of data
from SAS to the PI System  11
COUNT= data set option  21

D
data conversion
issues  11
PI System to SAS  10
SAS to the PI System  11
data set options  21
COUNT=  21
DBMAX_TEXT=  21
DELAY=  21
DESCRIPTION=  22
END=  22
FILTER=  22
HIDEFLAGS=  22
INTERVAL=  23
MAX=  23
MIN=  23
SELECT=  23
SHOWFLAGS=  24
START=  24
TAGLIST=  25
TIMEZONE=  25
data types
PI System  9
DBMAX_TEXT= data set option  21
DBMAX_TEXT= LIBNAME option  18
DBPROMPT= LIBNAME option  18
DEFER= LIBNAME option  19
DELAY= data set option  21
DESCRIPTION= data set option  22
DIGITAL data type
PI System  9
DOMAIN= LIBNAME option  18

E
END= data set option  22

F
filter
PI System data  19
tags, overview  14
tags, reading from Picomp  25
tags, reading from Pipoint  25
values to display  23
filter PI System data  19
FILTER= data set option  22
FLOAT16 data type
PI System  9
FLOAT32 data type
PI System  9
FLOAT64 data type
PI System  9

H
HIDEFLAGS= data set option  22
HIDEFLAGS= LIBNAME option  19

I
INT16 data type
PI System  9
INT32 data type
  PI System 9
interpolate data
evenly between start and end time 21
  using intervals 23
INTERVAL= data set option 23

L
LIBNAME engine 17
LIBNAME options
  DBMAX_TEXT= 18
  DBPROMPT= 18
  DEFER= 19
  DOMAIN= 18
  HIDEFLAGS= 19
  PASSWORD= 18
  PROC_DATASETS_TAGFILTER= 19
server connection 13
  SERVER= 17
  SHOWFLAGS= 19
  TESTDATE= 19
  TIMEZONE= 20
  USERNAME= 18
LIBNAME statement
  PI System 17

M
MAX= data set option 23
  maximum Value column length 21
MIN= data set option 23
  missing values 11

N
naming conventions
  PI System 7
National Language support 5
NULL values 11
NUMERIC data type
  SAS 10

P
password, PI System 18
  PASSWORD= LIBNAME option 18
performance optimization 16
  PI System
    asset framework 2
data 2
data archive 2
data conversion from SAS 11
data conversion to SAS 10
data set options 21
data types 9
features 1
LIBNAME statement 17
  naming conventions 7
Picompl virtual table 3
Pipoint virtual table 3
SAS data set columns 2
security 2
server connection 13
  server name requirements 7
supported SAS releases 1
write data 16
PI System LIBNAME engine 17
PI System server 17
Picorn 3
  processing tags 14
Pipoint 3
  reading tags 14
PROC DATASETS, using 4
  PROC_DATASETS_TAGFILTER=
    LIBNAME option 19

R
reading tags 14

S
SAS data set columns 2
SAS data types 10
SAS to the PI System
  data conversion 11
security 2
SELECT= data set option 23
server
  connecting 19
  name requirements 7
    PI System 17
  SHOWFLAGS= data set option 24
  SHOWFLAGS= LIBNAME option 19
  START= data set option 24
STRING data type
  PI System 10

T
TAGFILTER= data set option
  PI System 25
TAGLIST= data set option 25
tags
  processing 14
  reading 14
  TESTDATE= LIBNAME option 19
  TIMESTAMP data type
    PI System 10
  TIMEZONE= data set option 25
TIMEZONE= LIBNAME option 20

USERNAME= LIBNAME option 18

U
user name, PI System 18

W
write PI system data 16
Index