SAS® IT Resource Management 2.7

User's Guide

The Power to Know®

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS® IT Resource Management 2.7: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® IT Resource Management 2.7: User's Guide

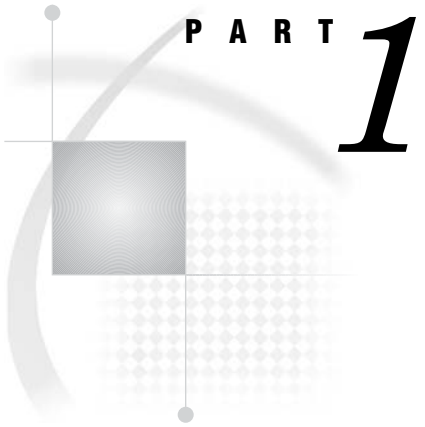
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, July 2004

2nd printing, May 2006

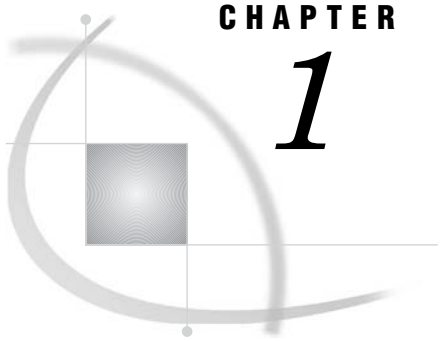
SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/pubs** or call 1-800-727-3228.

Contents



PART 1

General



CHAPTER

1

Introduction to SAS IT Resource Management

Introduction to SAS IT Resource Management

Overview of SAS IT Resource Management

SAS IT Resource Management is a performance evaluation solution that enables you to analyze your IT resources more effectively, by enabling you to access, manage, and

analyze large quantities of performance data. SAS IT Resource Management can assist you in areas such as troubleshooting, performance measurement and tuning, planning for growth, benchmarking, and discovery. SAS IT Resource Management enables you to summarize large volumes of data into a common format and then quickly and easily analyze and report on the data.

The SAS IT Resource Management documents are available in several locations.

- The Web site http://support.sas.com/documentation/online/itsv has all the documentation except the course notes. Documents are published to this Web site at SAS IT Resource Management releases and can be published between releases if there is a significant update. You can view and/or print the documents from this Web site.

  Note: In the SAS IT Resource Management Server Setup Guide, you can print a file's worth at a time. In the others, you can print the whole document (or any part of it) at one time. △

- The course notes are available on the from the SAS Bookstore's Web site: http://www.sas.com/apps/pubscat/welcome.jsp. Select **`Complete Catalog`** and then, under **`Browse Titles by Category`**, select **`Course Notes`**. Scroll to the course titles for SAS IT Resource Management. Course notes are published approximately every other release. You can order the course notes from this Web site.

  Note: When you enroll in the courses, you do not need to order the course notes separately. You will get the course notes in class (along with what's new in the most recent release, if the course notes are not for the most recent release). △

- Some of the documents are also accessible in other locations. For example, the SAS IT Resource Management User's Guide and Macro Reference documents are available in SAS System Help. You can access SAS System Help by displaying a SAS window, such as the SAS Program Editor window, and selecting Help.

  Within the SAS IT Resource Management client GUI, another way to access SAS System Help and these documents is to select **OnlineHelp ▶ SAS IT Resource Management Help ▶ (on UNIX only) All Topics ▶ SAS Products ▶ IT Resource Management**

  The copies of the documents in SAS System Help are not typically updated between releases.

Note: SAS IT Resource Management is the new name for IT Service Vision. You might see references in our older documentation to its former name. △

Using SAS IT Resource Management, you can read and process data from almost any data source. This is because SAS IT Resource Management not only supports many popular network and systems management tools and data sources on z/OS, UNIX, and Microsoft Windows, but also enables you to read and analyze data from any time-based data source.

You can also customize many aspects of the SAS IT Resource Management software. You can use or modify tables and reports that are supplied with this solution, or you can create new (custom) tables and reports.

SAS IT Resource Management also provides interactive and batch facilities through which you can access, manage, analyze, and present your performance data. The SAS IT Resource Management batch macros enable you to perform in batch mode the tasks that you can perform within the SAS IT Resource Management graphical user interfaces (GUIs). You can either code a macro call outside the GUI or, in many cases, select a task within the GUI and save its macro call in a file. You can then submit the macro calls in a batch job, file, or script and submitting the job/file/script to a batch processing

facility on your system. For information about how to perform tasks in batch mode with macro calls, see the SAS IT Resource Management macro reference documentation .

For addition introductory information, see the following topics:

- "What Is the Work Flow?" on page 5
- "What Are the Interfaces?" on page 6
- "What Is a PDB?" on page 7
- "Locating Help" on page 11.

What Is the Work Flow?

The routine tasks that you perform to process data and view reports can be summarized as follows:

1. Set up your data collector to work with SAS IT Resource Management. You can use the QuickStart Wizard to create your PDB and create batch jobs that can be used to process data, reduce data, and create reports based on your data. If you use the QuickStart Wizard, the process, reduce, and reporting tasks that are mentioned in the following steps can be performed through daily jobs that are created when you run the QuickStart Wizard.

   Based on the source of the data you select in the wizard, the wizard also provides an instruction file to assist you in setting up your collector with SAS IT Resource Management. For information about customizing and running the QuickStart jobs, follow the instructions that are provided with the wizard or refer to "Using the QuickStart Wizard" in the *Getting Started with SAS IT Resource Management* document.

   For more information about setup, refer to "General-Purpose Server Setup Documentation" on page 12 or "Overview of Setup" on page 37.

2. Stage the IT data that you want to analyze. You can stage data by using supplied staging code or by using custom staging code that you write for a specific data source.

3. Process raw data into the detail level of the tables in your PDB. (For information about how to process data, see "Processing Data" on page 413.) You can create a daily job (which typically runs each night) to read and process the data, or you can use the process-and-reduce job (which typically runs each night) that is created when you run the QuickStart Wizard. When detail-level data reaches an age limit that you specify, it is automatically deleted from the PDB. For more information about age limits, see "How Long Is Data Stored in the PDB?" on page 10.

4. Reduce the detail level data into one or more summary levels in the PDB: day, week, month, or year. You can set up a daily job (which typically runs each night) to reduce the data, or you can use the job (which typically runs each night) that is created when you run the QuickStart Wizard. (For more information about how to reduce data, see "Reducing Data" on page 416.)

5. Run the supplied report definitions or create and/or run your own custom report definitions to help analyze your data and identify performance issues. You can generate reports that use data from any level of the PDB. (For information about how to run supplied report definitions and how to create and run custom report definitions, see "Using the Manage Report Definitions Tool" on page 468. For other information about supplied report definitions, see "Exploring Supplied Report Definitions (UNIX, Windows)" on page 488.

What Are the Interfaces?

SAS IT Resource Management has one batch interface, several GUIs, and a Web interface.

Typically, the batch interface is used for production work and the GUIs are used for ad hoc work. The production work can be prepared directly for the batch interface by using an editor, or it can be prepared and tested in the GUIs and then saved in the format that is used for production.

The Web interface is an optional, additional way to make available the reports that are generated in batch mode or in interactive mode.

Batch Interface

To use the batch interface, you create and run one or more jobs (z/OS), scripts (UNIX), or batch files (Windows) in batch mode. The jobs/scripts/files invoke SAS software. Within SAS software, the jobs/scripts/files set the SAS IT Resource Management global macro variables and invoke the SAS IT Resource Management macros. (You can also include SAS DATA step code and SAS procedures in the jobs/scripts/files.)

The batch interface is described in the document *SAS IT Resource Management: Macro Reference*.

Server Interfaces (GUIs)

There are two server interfaces: one for z/OS, and one for UNIX and Windows. The primary focus of the server interfaces is administration. (The client interface has some interactive functionality for administration, but the full interactive functionality for administration is in the server interfaces.) Typically the server interface is located on a different host from the client interfaces. But on Windows and UNIX, it is possible for the server host to have both a server license and a client license, and in that case the functionality is additive.

The server interfaces are described in "Part 2: Setup" and "Part 3: Administration" of the document *SAS IT Resource Management: User's Guide*. In cases where additional information is provided in the macro reference documentation, there are pointers in the User's Guide to the relevant sections of the Macro Reference documentation.

Client Interface (GUI)

The primary focus of the client interface is reporting. (The server interfaces have some interactive functionality for reporting, but the full interactive functionality for reporting is in the client interface.) Typically, the client interface is located on a host that does not have a server license. But on Windows and UNIX, it is possible for the server host to have both a server license and a client license, and in that case the functionality is additive.

The client interface is described in "Part 4: Reporting" of the document *SAS IT Resource Management: User's Guide*. In cases where additional information is provided in the macro reference documentation, there are pointers in the User's Guide to the relevant sections of the Macro Reference documentation.

Web Interface (Galleries)

The primary focus of the Web interface is reporting. The Web interface is optional. It consists of one or more galleries that display Web-enabled reports that typically were generated by the client interface and/or the batch interface. The reports that use Java applets or ActiveX components generate additional reports interactively as you drill down. You use a Web browser to view and navigate through the galleries.

The Web interface is described in "Part 4: Reporting" of the document *SAS IT Resource Management: User's Guide*. Most of the information that is Web related is in

- the topic "Directing a Report to the Web" in the chapter "Working with Report Definitions"
- the chapter "Working with Galleries."

In cases where additional information is provided in the macro reference documentation, there are pointers in the User's Guide to the relevant sections of the Macro Reference documentation.

What Is a PDB?

A performance data database (also referred to as a performance data warehouse or a PDB) is a group of SAS libraries that contains the data that SAS IT Resource Management uses. You use a collector to record data from your IT systems, and then extract, load, and transform that data and store it in a PDB.

The PDB consists of an integrated set of nine libraries that contain data, metadata, workspace, and so on.

Your site can have one or more PDBs, depending on how you want to collect and store your data. You can use one PDB for each data source, such as Windows servers, UNIX servers, and phone switches. You might want to do this because the data uses different variables, the PDBs might be managed by different groups, or for other reasons.

You can also choose to combine data from multiple data sources in the same PDB. Or, if you want to investigate specific performance problems or use a specific PDB for testing, then you might want to create a test PDB. You can organize your data and PDBs to meet your site's needs.

SAS IT Resource Management provides many supplied table and variable definitions. These definitions are the default specifications of how to create that table or variable. When you create a PDB at your site, you can

- copy the supplied table and variable definitions and use them in your new PDB
- copy the supplied table and variable definitions to your new PDB and modify them for your specific needs
- create new table and variable definitions in your PDB.

For example, if you add or copy a supplied table or variable definition to your PDB, you might want to change defaults such as the age limits of data in each data level, the settings that specify which data is kept at the detail level, and the settings that specify which statistics are calculated at each summary level. You might also want to create new variables to calculate specific statistics or delete variables that you do not want to use.

The software that you use to gather observations about your IT services is referred to as a collector. SAS IT Resource Management supports many popular collectors and has tables in which to store the input data. However, you can also create your own tables, and SAS IT Resource Management can process that data into your tables.

SAS IT Resource Management reads raw data (metrics) in the format in which they are logged by your collector or systems management tool. SAS IT Resource Management can minimize the volume of data that you keep by summarizing the detailed performance data into smaller groups or levels in the PDB. You can accept the default statistics or you can select the statistics that you want to be calculated at each summary level. After the statistics are calculated and the data is reduced into the summary levels in the PDB, you can analyze information over different time periods by producing reports on summarized data.

You can also customize many aspects of SAS IT Resource Management. You can use or modify tables and reports that are supplied with this solution, or you can create new tables and reports.

The SAS IT Resource Management administrator is also responsible for managing future modifications to the PDB and making backup copies of the PDB. For more information about setting up your PDB, refer to "General-Purpose Server Setup Documentation" on page 12, or see your SAS IT Resource Management administrator.

Tables, Variables, and Definitions in a PDB

A *table* is a group of data and its metadata that are stored in a PDB. Each table contains data that is related to a specific performance area, such as a specific type of network activity. The metadata includes the table definition and the information about the status of the data in the table.

A *table's definition* includes properties such as

- the table's name and description
- the name of the collector that is used to gather the data and the name of the tool that is used to stage the data
- the type of table: interval or event
- the age limits of the data in each data level.

A table's status information includes

- the number of observations at each level
- the minimum and maximum datetime stamps on the observations at each level.

A variable's definition includes properties such as

- the name and description of the variable
- the data type, which defines whether the variable is numeric or character
- the statistics to be calculated for the variable at each summary level in the PDB
- the variable's format, for external display
- the variable's maximum length, for internal storage
- the interpretation type, which is used to determine the default statistics, default format, and default length.

The table and the variables within the table are created when you use a GUI to add the table or when you create the table definition in batch mode by using the %CPDDUTL macro.

You can also create *formula variables* and *derived variables*. Formula variables are variables whose values are calculated from the data that you collect. These variables are calculated dynamically and displayed when you access the data; however, the values of these variables are not stored in the PDB. Derived variables are variables that do not exist in your raw data, but are computed based on your data. After they are computed, the values of the derived variables are stored with your data in the PDB.

Within a table, data is stored in five physical libraries (also known as levels within a table): DETAIL, DAY, WEEK, MONTH, and YEAR. The data in the detail level is very similar to the data as it was recorded by your collector. The other four levels, known as summary levels or reduction levels, contain your summarized (reduced) data. The day level contains data that is summarized by the day, the week level contains data that is summarized by week, and so on. For example, a table for TCP protocol statistics might contain summarized TCP protocol data in the detail level and also summarized data in each of the four summary levels (day, week, month, and year).

When you display the data from one level of a table, this "picture" of the data is referred to as a *view*. A view is named by using a combination of the level name and

the table name in the form of *level.table*. For example, the view for the detail level in a table named HN2IFT would be DETAIL.HN2IFT. The view for the day level of the same table would be DAY.HN2IFT. When you report on your data, the reports are based on the views.

In the UNIX/Windows GUI, you can view a list of tables in a specific PDB, by selecting **`Manage Tables`** from the **`Administration`** tab in the main window. To see a list of variables for one of the tables, select a table from the list and select **Locals ▶ List Variables**

In the z/OS GUI, you can view the list of tables by selecting **PDB Admin ▶ Config Active PDB Dictionary**

in the main menu. To see a list of variables for one of the tables, select a table from the list and select **ItemActions ▶ List Variables**

Summarizing Data into Libraries in the PDB

Within a table, data is stored physically in five data libraries that are known logically as *levels*. When your data is first copied into the PDB from your raw data file, this is known as processing your data into the PDB, or the *process* task. This processed data is stored in the *DETAIL* library. The data in the detail level is very similar to the original data that you collected, with only simple transformations (when applicable), such as converting continuously ascending counters into rates per second or adjusting a counter that has exceeded its maximum and restarted at zero.

After the data has been processed into the detail level of the PDB, you then summarize, or *reduce*, the data into the other levels of the PDB. These other four levels are called *summary, or reduction,* levels, and they are day, week, month, and year.

DETAIL library
: The DETAIL library contains your data after you process it into the PDB. Each observation in a data set represents an event or an interval. The data in the detail level is very similar to your data records in the raw data file.

DAY library
: The DAY library contains daily statistics that are summarized by grouping (class) variables, such as MACHINE, HOUR, and DAY. These statistics are calculated from the data in the detail level. For example, this library might contain statistics for a given machine for a given hour of the day for a given day.

WEEK library
: The WEEK library contains weekly statistics summarized by grouping (class) variables. The statistics are calculated from the table's data in the detail level. For example, each observation in a data set in this library might contain the statistics for a given machine for a given hour of the day for a given week.

MONTH library
: The MONTH library contains monthly statistics that are summarized by grouping (class) variables. The statistics are calculated from the table's data in the detail level. For example, each observation in a data set in this library might contain the statistics for a given machine for a given hour of the day for a given month.

YEAR library
: The YEAR library contains yearly statistics that are summarized by grouping (class) variables. The statistics are calculated from the table's data in the detail level. For example, each observation in a data set in this library might contain the statistics for a given machine for a given hour of the day for a given year.

When numeric data is summarized (also known as "reduced"), it is read from the detail level, included in a calculation of one or more statistics, and then the updated value of the statistics are written to (or updated in) an observation in the summary

level. The observation in the summary level contains the requested statistics for all the data in the detail level that has the same values of the grouping variables. Character data is not summarized, but it can be in the summary levels (as data for the grouping variables). You select which statistics you want to be calculated for each summary level. You can use the default statistics that are provided with a new PDB, or you can customize the list of statistics.

In addition to the data libraries that are described above, the PDB contains several other libraries that are related to the data in a specific PDB. These libraries must be present in a PDB, even if they are empty.

ADMIN library
: The ADMIN library can be made available to all users, and therefore, it can be a convenient place to store graph and text reports, report palettes, logs, and so on, within a PDB. If you specify that data is to be checked for duplicates during processing, then this library might also contain files that are used for duplicate data checking.

COLLECT library
: The COLLECT library is used for temporary data storage as your data is processed into the PDB. For example, you can use the COLLECT library to store any staging data sets that you create for use by the generic collector software. This library might also contain information or libraries that are used with MXG software.

DICTLIB library
: Each PDB contains a *data dictionary*. The data dictionary is stored in the *DICTLIB* library. The PDB's data dictionary contains general PDB properties as well as information about the data that you stored in each level of the PDB. For example, the DICTLIB library contains specifications about which data to keep in the PDB and a count of the number of observations in each level of the PDB.

PDBWORK library
: The PDBWORK library is used by the reduction step for intermediate data and also can be used by the processing step. The PDBWORK library (which is permanent) is used instead of the SASWORK library (which is temporary) so that reduction can be restarted from a checkpoint if necessary.

Depending on your host, additional items can be stored in other libraries:

- On UNIX and Windows hosts, additional software can be stored in an auxiliary library named CPSYSLIB.
- If your site uses MXG, items such as MXG formats can be stored in an auxiliary library, which is named MXG.MXG.FORMATS in the MXG documentation.
- If more than one release level of SAS IT Resource Management is installed, you will have a separate version of the PGMLIB library for each release.
- If you choose to Archive data, additional archive libraries are created. Their names begin with *Z*.
- When you run the QuickStart Wizard, programs are stored in additional libraries whose default location is "under" the PDB that is created by the wizard.

How Long Is Data Stored in the PDB?

The DETAIL library can easily become very large, because this library typically contains the data as it is logged by your collector or system management software. Additionally, as you process new data into the PDB and reduce the data into the summary levels, you will need to decide how long to keep data at each level in the PDB.

To reduce the space that is required for the DETAIL library and the other libraries, your site can set *age limits* on the data at each level. When data is reduced from the

detail level into the summary levels, you might not need to keep the detail-level data, or you might want to keep only a few days of data at that level. For the other summary levels, you can choose a time limit, such as 3 days, 10 days, 1 month, and so on, and data will be deleted from each level when the level's age limit is reached. Using each of the summary levels, you can still analyze and report on long-term trends for your IT systems, but you do so by using summarized data instead of large quantities of detail data.

When you create a new table, the following default age limits are assigned to the levels in the table:

- detail - 10 days
- day - 45 days
- week - 15 weeks
- month - 18 months
- year - 5 years.

In the UNIX/Windows GUI, you can view the current age limits (and change the age limits if you have write access to the PDB) by following this path from the **`Administration`** tab in the SAS IT Resource Management main window: **Manage Tables ▶ select a table ▶ File ▶ Properties**

From the Table Properties window, select the **`Table Status`** tab to review or edit the age limit settings. In the z/OS GUI, you can view the current age limits (and change the age limits if you have write access to the PDB) by following this path from the main menu: **PDB Admin ▶ Config Active PDB Dictionary**

Select the table whose age limits you want to change. Then select **ItemActions ▶ Edit Definition or Browse Definition**

The age limits are displayed in the lower-right corner of the window.

Locating Help

The following resources and documentation are available to assist you in setting up and using SAS IT Resource Management. Select one of the following items to find out how to access documentation or assistance for the selected topic:

□ “SAS Technical Support” on page 16.

Getting Started with SAS IT Resource Management

The document *Getting Started with SAS IT Resource Management* provides step-by-step instructions for using the QuickStart Wizard (to create the PDB) and the Process/Reduce Wizard (to process and reduce data into the PDB). The documentation also provides an overview of the SAS IT Resource Management report tools. This document can be ordered from SAS book sales at 1-800-727-3228.

It is also available on the Web. See “SAS IT Resource Management Web Site” on page 13.

QuickStart Wizard

The QuickStart Wizard provides a quick and easy way for you to start using SAS IT Resource Management. The QuickStart Wizard is available within the server GUIs for SAS IT Resource Management.

Information about the wizard is available in the document *Getting Started with SAS IT Resource Management* document. Information is also available in the document *SAS IT Resource Management: User’s Guide*, in “Using the QuickStart Wizard” on page 50, and in “Overview of Setup Case 1” on page 53. For information about customizing the QuickStart reports, see topics in the “Overview of Galleries” on page 551.

General-Purpose Server Setup Documentation

The general-purpose documentation for setting up a SAS IT Resource Management server is in the process of moving from one location to another. First look here: “Setting Up the Server, with Setup Cases” on page 38.

If you do not find what you are looking for, then look in the *SAS IT Resource Management Server Setup Guide*. You can access the *SAS IT Resource Management Server Setup Guide* from the SAS IT Resource Management client GUI or the SAS IT Resource Management server GUI for UNIX and Windows. Follow this path from the main window: **OnlineHelp ▶ Other ITRM Documentation ▶ Server Setup Guide**

Or, you can look in the general-purpose setup documentation on the Web. See “SAS IT Resource Management Web Site” on page 13.

Collector-Specific Setup Documentation

The general-purpose documentation for setting up a SAS IT Resource Management server refers you to specific information in the *SAS IT Resource Management Server Setup Guide*. You can access the *SAS IT Resource Management Server Setup Guide* from the SAS IT Resource Management client GUI or the SAS IT Resource Management server GUI for UNIX and Windows. Follow this path from the main window: **OnlineHelp ▶ Other ITRM Documentation ▶ Server Setup Guide**

Or, you can look in the collector-specific setup documentation on the Web. See “SAS IT Resource Management Web Site” on page 13.

For information about what you have to customize in MXG and SAS IT Resource Management in order for MXG-based tables to work, see “Using MXG-Based Tables (CMAPP2): Introduction” on page 133.

Client Setup Documentation

For information about how to set up the SAS IT Resource Management client, see “Setting Up the Client” on page 43.

Window Help

The window-specific help is available from the window's **Help** button or, in some cases, by selecting **OnlineHelp ▶ Help for this Window**

Usage Documentation and Reference Documentation

You can access the user's guide and the macro reference document on the Web and in SAS IT Resource Management GUIs. See "Overview of SAS IT Resource Management" on page 3.

The GENERATE SOURCE Control Statement

The GENERATE SOURCE control statement enables you to make a preliminary draft of the table and variable definitions. You would typically use this statement for data from a collector for which SAS IT Resource Management does not supply table and variable definitions. The GENERATE SOURCE control statement is used with the %CPDDUTL macro.

For more information about the GENERATE SOURCE control statement, see "Using the GENERATE SOURCE Control Statement" on page 48. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the macro reference documentation.

The Generic Collector Facility

The Generic Collector Facility is available in the %CMPROCES, %CPPROCES, %CSPROCES, and %CWPROCES macros. The Generic Collector Facility enables you to process data into tables that you created (typically with the GENERATE SOURCE control statement).

For more information about the Generic Collector Facility, see "Using the Generic Collector Facility" on page 158.

For more information about the context of using the Generic Collector Facility, see "Overview of Setup Case 3" on page 81 and "Overview of Setup Case 4" on page 99.

Demonstration PDBs, Sample Code, and QuickStart Examples

Here is information about sample code, QuickStart examples and demonstration PDBs.

- The *MISC* directory (UNIX), *SASMISC* directory (Windows), and *CPMISC* PDS (z/OS) are installed with SAS IT Resource Management. These directories and data sets contain many examples of programs and scripts that will assist you in setting up and running SAS IT Resource Management.
- Demonstration PDBs are available within SAS IT Resource Management.
- To access the QuickStart Examples, follow this path from the main window in the SAS IT Resource Management GUI for UNIX and Windows: **OnlineHelp ▶ Other ITRM Documentation ▶ QuickStart Examples**

  If the examples are zipped, unzip them.

SAS IT Resource Management Web Site

Information that is related to SAS IT Resource Management is available on the Web page at http://support.sas.com/documentation/onlinedoc/itsv.

SAS IT Resource Management Listserv

A listserv is available for customers to exchange information and participate in discussions about SAS IT Resource Management and other solutions in the SAS IT Management Solutions family of solutions. To subscribe to the listserv, send e-mail to *LISTSERV@VM.SAS.COM* with the message

```
SUBSCRIBE ITMS-L yourfirstname yourlastname
```

To unsubscribe from the listserv, send e-mail to LISTSERV@VM.SAS.COM with the message

```
UNSUBSCRIBE ITMS-L yourfirstname yourlastname
```

To post messages to the listserv, send e-mail to *ITMS-L@VM.SAS.COM*. You can send messages to the listserv only if you have subscribed to the listserv.

SAS IT Security Management Documentation

If your site has purchased IT Security Management, see the document *Getting Started with SAS IT Security Management*. You can access the document on the following Web site: http://support.sas.com/documentation/onlinedoc/itsecurity. Also, help is available within SAS IT Security Management by selecting **Help** in a window.

SAS IT Charge Management Documentation

If your site has purchased SAS IT Charge Management, which is used with SAS IT Resource Management, then see the document *SAS IT Charge Management: User's Guide*. You can access the document on the following Web site: http://support.sas.com/documentation/onlinedoc/chargemanager. Also, help is available within SAS IT Charge Management by selecting **Help** in a window.

SAS IT Service Level Management Documentation

If your site has purchased SAS IT Service Level Management, which is used with SAS IT Resource Management, then see the documents *Getting Started with SAS IT Service Level Management* and *SAS IT Service Level Management: User's Guide*. You can access the documents on the following Web site: http://support.sas.com/documentation/onlinedoc/itslm. Also, help is available within SAS IT Service Level Management by selecting **Help** in a window.

Additional SAS Documentation

The SAS System includes many software products and solutions in addition to SAS IT Resource Management, SAS IT Security Management, SAS IT Service Level Management, and SAS IT Charge Management. Some SAS documentation is shipped with your software, and other books can be ordered from SAS Book Sales.

Additional information about SAS documentation is available in the SAS Publications Catalog or by contacting Book Sales (1-800-727-3228).

Also, you can access help on SAS from the Help menu in any SAS window. The information in these help topics is broad. It includes help on SAS windows, other SAS products and solutions, and many SAS procedures and language statements that are used by SAS IT Resource Management.

Suggested Reading

If you are new to performance evaluation and want to do some background reading, we suggest the following books:

- *Operating System Concepts, Sixth Edition*, by Abraham Silberschatz et al, Wiley Text Books, 2002, ISBN 0471250600. Or *Operating Systems Concepts with Java, Sixth Edition*, by Abraham Silberschatz et al, Wiley Text Books, 2003, ISBN 0471489050.

  These books are a good introduction to the purposes and internals of operating systems in general.

- *Merrill's Expanded Guide to Computer Performance Evaluation Using the SAS System.* By H. W. "Barry" Merrill. SAS Institute Inc. 1984. ISBN 0-917382-54-4.

  This book focuses on concepts, relationships, and analysis techniques. See Chapters 1, 2, and 7–30.

You might also want to read the following papers. They are platform specific but general enough to show some of the concepts in use. They are in materials from the following organization:

```
Computer Measurement Group (CMG), Inc.
http://www.cmg.org
```

- "Distributed System Analysis Case Study." By Barbara H. Pendergrass and Robert N. Bonham. CMG 95 Proceedings. Volume 2.
- "Getting Started in Performance Tuning and Capacity Planning." By Neil Ervin. CMG 94 Proceedings. Volume 2.
- "Introduction to Performance Management: What to Measure, What to Report." By Stephen Samson. CMG 94 Proceedings. Volume 1.
- "An Introduction to Time Series Forecasting for CPE." By Brian Bennett. CMG Transactions. Summer 1994.
- "Network Analysis Case Study." By Barbara Pendergrass and Robert N. Bonham. CMG 94 Proceedings. Volume 2.

If you want to learn more about the SAS software and language, read these books:

- *The Little SAS Book: A Primer*, by Lora D. Delwiche and Susan J. Slaughter, Cary, NC: SAS Institute Inc., 1995, ISBN 1-55544-215-3.
- *SAS Language and Procedures: Introduction, Version 6, First Edition*, Cary, NC: SAS Publishing, 1990, ISBN 1-55544-410-5.

The following works provide additional information about the SAS data sources output by MXG Software.

- Chapters 32–33 of *Merrill's Expanded Guide to Computer Performance Evaluation Using the SAS System*. By H. W. "Barry" Merrill. 1984. SAS Institute Inc. ISBN 0-917382-54-4.
- *Merrill's Expanded Guide Supplement*. By H. W. "Barry" Merrill. 1987. SAS Institute Inc. ISBN 1-55544-067-3.

For more information about performance tuning and evaluation of the z/OS operating system, read the following IBM subsystem manuals and the manuals for z/OS itself. (Also consult your source of IBM manuals to locate other manuals of this type.)

- *z/OS Initialization and Tuning Guide.*
- *z/OS Initialization and Tuning Reference.*

For advanced MVS applications and ideas, including resource management and tuning, read *Cheryl Watson's TUNING Letter*. You can subscribe to the letter from

```
Watson & Walker Inc.
www.watsonwalker.com
```

For more information about performance tuning and evaluation of UNIX and Windows operating systems, consult the documentation of your collector software for terminology and other suggested reading.

SAS Technical Support

Technical Support is available to all sites that license software from SAS Institute. A printed "Overview of Technical Support Services" is provided with your software shipment. For information about how to contact Technical Support, see http://support.sas.com/techsup/contact/

CHAPTER

2

Getting Started

Getting Started with SAS IT Resource Management

Overview of Getting Started

To get started with the batch interface, see this topic:

- "Working with the Interface for Batch Mode" on page 18.

To get started with the server interface on z/OS, see these topics:

- "Starting the z/OS GUI" on page 19
- "Using the z/OS GUI" on page 21
- "Exiting from the z/OS GUI" on page 21.

To get started with the server interface and client interface on UNIX or Windows, see these topics:

- "Starting the UNIX and Windows GUI" on page 21
- "Using the UNIX and Windows GUI" on page 22
- "Exiting from the UNIX and Windows GUI" on page 23.

To get started with the Web interface, see this topic:

- "Working with the Web Galleries" on page 23.

For additional information about getting started, see the following documentation:

- *Getting Started with SAS IT Resource Management*
- *Getting Started with SAS IT Security Management*.

Note: You can access the interfaces after SAS IT Resource Management is installed. But you cannot do much real work until SAS IT Resource Management is set up. For more information about setup, see "Overview of Setup" on page 37. △

Working with the Interface for Batch Mode

Working with the Interface for Batch Mode

You can submit a SAS program or SAS code

- through the SAS GUI (by using the SAS Enhanced Editor window or the SAS Program Editor window)
- in batch mode.

Submitting a SAS Program through the SAS GUI

How you submit the SAS program (or code) depends on what the program contains and what else is running.

- If you want to submit a SAS program that calls SAS IT Resource Management macros and if the SAS IT Resource Management GUI is running, then *exit from the SAS IT Resource Management GUI* to the SAS GUI before you submit the program.

  1 To exit from the SAS IT Resource Management GUI to its underlying SAS GUI:

- In the SAS IT Resource Management GUI for UNIX and Windows, select **Exit ► Exit SAS IT Resource Management Only ► OK**
- In the SAS IT Resource Management GUI for z/OS, select **Exit ► Exit SAS IT Resource Management Only**

2 In the SAS (Enhanced) Editor window or the SAS Program Editor window in the SAS GUI, you can enter a new SAS program by typing it into the body of the window. Or you can open an existing SAS program by selecting **File ► Open**

To submit the program, select **Run ► Submit**

Note: For more information about using either of these windows, select the window and then select **Help ► Using This Window** △

3 When the SAS program finishes, you can invoke the SAS IT Resource Management GUI again by submitting this SAS program:

```
%CPSTART();
```

This is a call to the macro that starts SAS IT Resource Management.

- If you want to submit a SAS program that does not call SAS IT Resource Management macros and if the SAS IT Resource Management GUI is running, then you can submit the program (through the SAS Enhanced Editor window or SAS Program Editor window) without first exiting from the SAS IT Resource Management GUI.
- If the SAS IT Resource Management GUI is not running, then you can submit a SAS program (through the SAS Enhanced Editor or SAS Program Editor window) whether or not the program calls SAS IT Resource Management macros.

Submitting a SAS Program in Batch Mode

- How to submit the program (or code) in batch mode depends on the operating system. For information and examples relevant to your operating system (UNIX, Windows, or z/OS), see “Chapter 1: Introduction” of the SAS IT Resource Management macro reference documentation.

Note: For more information about batch mode, see the SAS IT Resource Management macro reference documentation . △

Working with the z/OS GUI

Working with the z/OS GUI

Starting the z/OS GUI

SAS IT Resource Management runs in a SAS session. You can start the SAS IT Resource Management GUI for z/OS

- by using a CLIST.

  At the TSO READY prompt, invoke the CMCPE CLIST. The CMCPE CLIST invokes SAS and, from within that SAS session, invokes SAS IT Resource Management.

- by doing the invocations yourself.

  At the TSO READY prompt, invoke SAS. Then, from within that SAS session, invoke SAS IT Resource Management. To invoke SAS IT Resource Management

  - if you have a command line, type *ITRM* or *ITSV*at the command prompt and then press ENTER or RETURN.
  - if you have pull-down menus, select **Globals ▶ Command ▶ Command**

    Type *ITRM* in the command field and then select **OK**.

To invoke SAS IT Resource Management with write access to the active site library, use the following instructions.

- If the command line is not displayed, follow this path from the pull-down menus: **Globals ▶ Command ▶ Command**
- Type *itrm siteacc=old* at the command prompt and then press ENTER or RETURN.

SAS IT Resource Management will now be restarted with write access to the active site library.

Note: If you are the only user of the SAS IT Resource Management server or if you have your own CLIST, then you can set the SITEACC=OLD parameter in the %CPSTART macro in the CMAUTOEX member of the CPMISC PDS. You can use a text editor (for instance, ISPF Edit) to edit this member. Add the SITEACC=OLD parameter at any place in the list of %CPSTART parameters, checking for commas and closing parentheses.

If, in other tasks, you want to invoke SAS IT Resource Management with parameters, then use the PDB= parameter to specify the PDB name, the DISP=OLD parameter for write access to the PDB, and the SITEACC=OLD parameter for write access to the active site library. Here is an example:

- By issuing the following command you can invoke the SAS IT Resource Management server with *your.test.pdb* as the active PDB, read access to the PDB, and write access to the active site library:

```
itrm pdb='your.test.pdb' siteacc=old
```

- By issuing the following command you can invoke the SAS IT Resource Management server with *your.test.pdb* as the active PDB, write access to the PDB, and read access to the active site library:

```
itrm pdb='your.test.pdb' disp=old siteacc=shr
```

- By issuing the following command you can invoke the SAS IT Resource Management server with *your.test.pdb* as the active PDB, write access to the PDB, and write access to the active site library:

```
 itrm pdb='your.test.pdb' disp=old siteacc=old
```

- If this command does not fit on your command line, then you can obtain the same results by modifying the CMAUTOEX member of the CPMISC PDS (and changing it back later if necessary). Or you can invoke the SAS IT Resource Management server by typing these lines, starting at the first line in the body of the SAS Program Editor window:

```
%cpstart(pdb='your.test.pdb',
disp=old,
siteacc=old);
```

  Then, from your SAS Program Editor window, select **Run ▶ Submit**

△

Using the z/OS GUI

The SAS IT Resource Management window (which is a SAS window) overlays the other SAS windows.

Because SAS IT Resource Management runs within a SAS session, you have access to both SAS IT Resource Management windows and SAS windows. You can cycle among the windows by issuing the NEXT command, which is assigned to the F8 key or the SHIFT+F8 key combination by default. For more information about SAS windows, see

- "Using SAS Windows" on page 24
- "Using Menus" on page 25
- "Using Commands" on page 26
- the SAS Companion documentation for your current version of SAS.

The main menu of SAS IT Resource Management has buttons. The top row of buttons corresponds to administrative tasks. The middle row of buttons corresponds to reporting tasks. The bottom row of buttons corresponds to tasks that are useful for both administration and reporting. For example, the bottom row contains a **Help** button.

Exiting from the z/OS GUI

If you want to exit from SAS IT Resource Management and then from SAS, on the main menu select **Exit ▶ Exit SAS and SAS IT Resource Management** You return to the TSO READY prompt.

If you want to exit from SAS IT Resource Management in order to work in SAS (and perhaps to reinvoke SAS IT Resource Management later within the same SAS session), then on the main menu select **Exit ▶ Exit SAS IT Resource Management only** After you work in the SAS windows,

- if you want to return to SAS IT Resource Management from SAS, then do one of the following steps:
  - in the body of the SAS Program Editor window, type

    ```
    %CPSTART();
    ```

    and select **Run ▶ Submit**
  - at the SAS command line, type *ITRM* and press RETURN or ENTER.
- if you want to exit from SAS, do one of the following steps:
  - in the body of the SAS Program Editor window, type

    ```
    endsas ;
    ```

    Then select **Submit** (from the menu), or type *SUBMIT* (on the command line) and press ENTER or RETURN.
  - at the SAS command line, type *BYE* and press RETURN or ENTER.

Working with the UNIX and Windows GUI

Working with the UNIX and Windows GUI

Starting the UNIX and Windows GUI

SAS IT Resource Management runs in a SAS session. You can start the SAS IT Resource Management GUI for UNIX and Windows

- by running a script or batch file. The script/file is provided with the installation of SAS IT Resource Management.
  - On UNIX, the installation puts the script named *itsv.sh* in the SASMISC directory. Then, you can customize that script, and you can save it to a location that is appropriate at your site.
  - On Windows, the installation puts the batch file *itsv.bat* in the SASMISC folder. Then, you must customize that script, and you can save it to a location that is appropriate at your site.
- by doing the invocations yourself. Invoke SAS and then, from within that SAS session, invoke SAS IT Resource Management by typing *itrm* or *itsv* in the SAS command field and pressing ENTER or RETURN.

To invoke SAS with write access to the active site library, use the following instructions:

- From the SAS IT Resource Management main window, select **Administration tab ▶ Site Options ▶ General tab ▶ Sitelib Access ▶ WRITE ▶ OK**

  Note: Both the ITSV command and the %CPSTART macro invoke SAS IT Resource Management. △

If, in other tasks, you want to invoke SAS IT Resource Management with parameters, then create a file that contains the %CPSTART macro with the desired parameters and start SAS by pointing to the file as follows (see SAS System documentation for more information about using the -AUTOEXEC option):

```
sas -autoexec mypgm.sas
```

Or start SAS and type the statements in the SAS Program Editor window. Examples of these statements follow:

- This statement invokes the SAS IT Resource Management server with a PDB named pdb-test as the active PDB, read access to the PDB, and write access to the active site library:

  ```
  %CPSTART( pdb='/tmp/pdb-test', siteacc=write );
  ```

- This statement invokes the SAS IT Resource Management server with a PDB named pdb-test as the active PDB, write access to the PDB, and read access to the active site library:

  ```
  %CPSTART( pdb='/tmp/pdb-test', access=write,siteacc=readonly );
  ```

- This statement invokes the SAS IT Resource Management server with a PDB named pdb-test as the active PDB, write access to the PDB, and write access to the active site library:

  ```
  %CPSTART( pdb='/tmp/pdb-test', access=write, siteacc=write );
  ```

- If this command does not fit on your command line, then you can obtain the same results by typing these lines, starting at the first line in the body of the SAS Program Editor window:

  ```
  %CPSTART( pdb='/tmp/pdb-test',
  access=write,
  siteacc=write );
  ```

Then, from your SAS Program Editor window, select **Run ▶ Submit**

Using the UNIX and Windows GUI

Typically, the SAS IT Resource Management window (which is a SAS window) is open and the other SAS windows are iconized.

For more information about SAS windows, see

- "Using SAS Windows" on page 24
- the SAS Companion documentation for your current version of SAS software
- "Using Menus" on page 25
- "Using Commands" on page 26
- "Using the Toolbar and Command Line" on page 27
- "Resizing and Moving Windows" on page 28
- "Manipulating Text and Objects" on page 28
- "Selecting Options in Dialog Boxes" on page 29.

The main window of SAS IT Resource Management has several tabs. The **Administration** tab corresponds to administrative tasks (tasks that typically involve write access and relate to creating and working with your PDB, tables, and variables). The **Reporting** tab corresponds to reporting tasks (tasks that typically involve read access and relate to creating and working with report definitions, palette definitions, and rule definitions). The **Applications** tab corresponds to other user-, PDB-, and site-specific applications, if you have any. For more information, see

- "The Administration Tab" on page 30
- "The Applications Tab" on page 32
- "The Reporting Tab" on page 30.

Exiting from the UNIX and Windows GUI

To exit from SAS IT Resource Management, follow these steps:

1. Select the **Exit** icon on the toolbar or select this menu path from the SAS IT Resource Management main window: **File ▶ End** The Exit window opens.
2. Select **Exit SAS IT Resource Management Only** or **Exit SAS IT Resource Management and SAS**.
3. If you are currently connected to a remote server, then you have the option to automatically sign off the remote server. To do so, select the server sign-off check box.
4. Select **OK** to perform the selected task or tasks.

Working with the Web Galleries

Working with the Web Galleries

Point your Web browser to the welcome.htm file in the directory where the report galleries are located.

Typically, Web galleries are created by running a batch job (for example, an xreport.sas job that is created by using the QuickStart Wizard). For information about modifying the structure and contents of the Web galleries, see "Overview of Galleries" on page 551.

Appendixes to Getting Started

Using SAS Windows

Depending on the way that you invoke SAS and SAS IT Resource Management, your default SAS windows can be one or more of the windows in this list:

Enhanced Editor
is an ASCII editor that uses visual aids such as color coding and code sections to help you write and debug your SAS programs while retaining some familiar SAS Program Editor features.

Explorer
provides a central access point to data such as catalogs, tables, libraries, and host files. It enables you to perform basic SAS tasks such as:

creating new libraries, new library members, catalog entries, and file shortcuts

opening and editing SAS files.

Note: For details, see the SAS Online Help by selecting the **Help** button in a SAS window. △

Graph
displays the output for programs that generate graphics output, when you specify the Graph window as your graphics output device. To close this window, select **File ▶ End** By specifying the appropriate device driver, you can direct output to the Graph window, a SAS catalog GRSEG entry, various printing devices, or an external graphics stream file. You can also use the GDEVICE procedure to list the available device drivers for your system.

For more information about PROC GDEVICE or device drivers, refer to SAS/GRAPH software reference documentation for your current version of SAS.

Log
contains a log of your SAS IT Resource Management session and also displays anything that you explicitly print to the Log window, such as table and variable definitions. This window records notes, warnings, and error messages about your job. If you are running a job in batch, then by default on UNIX and Windows the log file is saved as *jobname*.log, and on z/OS the log file is printed with the rest of the output from the job (unless the job's output is held, in which case look in the job's output for the section with the log's DDname, which is SASLOG by default).

Output
is the default destination for any text output or reports that you generate by using SAS IT Resource Management. This window cannot display graphics. If you are running a job in batch, then by default on UNIX and Windows, the Output file is saved as *jobname*.lst.

Program Editor [or Editor]
lets you perform the following tasks:

entering and editing text (including program statements)

submitting program statements to the SAS System for execution

saving and recalling source files.

To submit code from this window, type the code in the SAS Program Editor (or SAS Editor) window and select **Run ▶ Submit**

Note: For more details about how to write and submit SAS programs, you can also view the SAS Online documentation. △

Results
lists the procedure output that is produced by the SAS System. You can use this window to locate and manage SAS output. Open this window by using the ODSRESULTS command or by selecting **View ► Results**

Note: For more detailed descriptions of these and other SAS windows, see the SAS Companion for the current version of your operating system and release of SAS software. △

Using Menus

The menus differ depending on the operating system that you are using. The menus also differ depending on the window that you are viewing.

Using Menus (z/OS)

The following menus frequently appear on the menu bar in the SAS IT Resource Management GUI for z/OS:

File	contains actions that relate to copying or retrieving the contents of the window to or from an external medium like disk or paper, such as Print, Save, Copy, or Open. This menu usually contains the End or Cancel actions if they are appropriate to the window.
Edit	contains window-specific tasks such as Undo, Clear Text, and Find.
ItemActions	lists actions that can be performed on items on the list that is displayed in the primary portion of the window. If you choose one of the actions on the **`ItemActions`** list, then the action is applied to all selected (highlighted) items. This menu includes actions such as Run (for example, to run one or more report definitions that you have selected), Delete, and Browse.
Actions	lists actions that are not directly related to a specific selection in the list. This menu includes actions such as Set Collector, Set PDB Options, and Set Rawdata.
Globals	lists global SAS items such as Log (to move to the Log window) or Output (to move to the Output window) that relate to the SAS System as a whole.
Help	provides help on using the window.

Typically, a window has either pull-down menus or a command line. If your window has a command line and you would prefer to use pull-down menus, then issue the PMENU command, which is associated with F9 or SHIFT+F9 by default.
To return to a command line, again select F9 or SHIFT+F9.

Using Menus (UNIX, Windows)

The following menus frequently appear on the menu bar in the SAS IT Resource Management GUI for UNIX and Windows:

File	contains actions that relate to copying or retrieving the contents of the window to or from an external medium like disk or paper, such as Print, Save, Copy, or Open. This menu usually contains the End or Cancel actions if they are appropriate to the window.

Edit	contains window-specific tasks such as Undo, Clear text, and Find.
Locals	contains window-specific tasks, such as Select Palette or Select Table.
Tools	contains tools that apply to the active window, such as filter items in a list.
Globals	lists global SAS items such as Log (to move to the LOG window) or Output (to move to the OUTPUT window) that relate to the SAS System as a whole.
Window	lists the active windows. Select a window name from this menu to activate that window.
OnlineHelp	enables you to access the SAS IT Resource Management online Help system and the SAS System help. For more information, see "Locating Help" on page 11.

Using Pop-up Menus (z/OS, UNIX, Windows)

Frequently used menu items (for example, Cut, Copy, Delete, and other tasks specific to the active window) are available on the main menus in each window and might also be available on the pop-up menu. To display the pop-up menu follow these steps.

1. Position your mouse pointer in the window where you want to perform a task or at the location in a window where you want the action to occur.
2. Click the right mouse button.
3. Choose a command from the list.

The commands on the pop-up menu vary based on your active window and the task that you perform.

Using Commands

Below are a few basic commands that you can use in SAS windows to move around and search for information in SAS windows. You can also use the horizontal and vertical scroll bars or the PAGE UP and PAGE DOWN keys to scroll through the information that is displayed in the window.

BACKWARD	scrolls backward in the window.
BOTTOM	moves to the bottom of the window.
CLEAR TEXT	clears the text in the body of the window.
END	closes the active window. The END command cannot be used to close the LOG and OUTPUT windows.
FIND or RFIND	finds a text string or repeats the previously specified find.
FORWARD	scrolls forward in the window.
KEYS	views or changes the settings of function keys. To change a key setting, type over the command that is listed beside the key name.
NEXT	moves to the next SAS window.
PMENU or COMMAND	toggles between pull-down menus and command lines. PMENU turns off pull-down menus for all windows in the system. COMMAND turns off pull-down menus only for the window where

	you issue the command. Use the same command to restore pull-down menus.
RECALL	recalls the last program in the SAS Program Editor window.

Most SAS commands are the same regardless of the operating system that you are using. The location for issuing the command differs depending on the operating system that you are using. If the SAS command line is turned on or if you have access to the command field, then you can use commands to perform tasks.

Using Commands (z/OS)

Typically, a window has either pull-down menus or a command line. If you have pull-down menus and would prefer to issue commands, issue the PMENU command, which is associated with F9 or SHIFT+F9 by default.

To return to pull-down menus, again select F9 or SHIFT+F9.

Using Commands (UNIX, Windows)

On UNIX and Windows, you can issue commands through the command field. To issue a command, type it in the command field and then press ENTER or RETURN.

Using the Toolbar and Command Line

The toolbar or toolbox is a group of buttons that you can select or click with your mouse to execute commands. *The tasks available from the toolbar are also available from the menus on the menu bar.* If you point to an item on the toolbar but do not click the item with the mouse pointer, then a small window called a ToolTip opens. A ToolTip describes the task that is performed in your active window when you select that button.

The tasks available from the toolbar are specific to the active window; therefore, the available buttons change as you select tasks from the SAS IT Resource Management main window and open new windows or dialog boxes. To activate the appropriate toolbar for your active window, click inside the window. The toolbar changes to display tasks that can be performed in the active window. Some windows or dialog boxes contain buttons for all tasks that can be performed in that window or dialog box. These windows do not have specific SAS IT Resource Management toolbar items or menus and will display the default SAS toolbar and menu structure. For information about using SAS windows or the SAS GUI, refer to the SAS Companion documentation for your current version of SAS software.

The **`Command`** text box is available in the toolbox or to the left of the toolbar, depending on your platform. To execute commands from the command text box, type the command in the box and press ENTER.

Note: If you are using the SAS IT Resource Management server on a UNIX platform and if you are accessing that platform through an X-windows emulator such as Exceed, then the SAS toolbar frequently steals the cursor focus from other SAS windows. If you are not accessing UNIX using an emulator, then you will not see this behavior. To avoid this behavior, you can iconize the SAS toolbar or change the focus policy for your mouse.

To change the focus policy, start Exceed and select the “running” icon with your right mouse button. Then make the following selections: **Tools ▶ Configuration ▶ Window Mode**

Change the **`Focus Policy`** from **`Click`** to **`Pointer`**. For more information, refer to the documentation for your emulation software. △

Resizing and Moving Windows

To resize a window, place your mouse pointer on the side or corner of the window until the pointer changes to a double arrow. Click on the window frame and drag the frame to resize the window.

To move a window, place your mouse pointer on the title bar of the window and, while selecting the title bar, drag the window to the new location.

If you change the font that is used to display your SAS windows, then this change affects your SAS IT Resource Management windows, and you might experience problems with displaying them. For example, if you receive a message similar to

```
WARNING: Client could not be resized to fit
client area; client will be hidden.
```

then specifying a larger font for your SAS windows can correct the problem.

To display the active font and/or select a new font, use the DLGFONT command in the SAS toolbox or on the command line.

Manipulating Text and Objects

The following terms describe common methods of manipulating text and windows within SAS IT Resource Management. For more information about standard SAS commands for working within a SAS window, see “Using SAS Windows” on page 24 or the SAS Companion for your operating environment. (See “Additional SAS Documentation” on page 14.)

Activate
: Select an item and make it the active item, such as bringing a window to the foreground, displaying a menu, or making a PDB active. For example, the PDB that you are currently using is the active PDB, or when you are working with multiple windows, the window that you are currently working in is the active window.

Click
: Place the tip of the mouse pointer on an item and press the left mouse button once.

Deactivate
: Un-select a document, window, or menu, by pressing the ESC key or clicking outside the selected area.

Double-click
: Place the tip of the mouse pointer on an item and press the left mouse button twice.

Drag
: Place the tip of the mouse pointer on an item. Hold down the left mouse button as you move the mouse pointer over the text or object. Dragging selects one or more items, text characters, or graphics so that you can move an item or simultaneously perform an action on multiple items.

Select
: Highlight or click on a section of text, graphic, menu name, command, or other object with the keyboard or mouse. Selecting something performs a task such as opening a menu from the menu bar or activating a window so that you can use the window.

Selecting Options in Dialog Boxes

Dialog boxes enable you to set options for many tasks, such as report options, site options, printing options, and more. Dialog boxes open when you select a menu item that is followed by ellipsis points (...), such as the **File ▶ Open...** menu item, or when you select certain buttons on the toolbar, such as Print.

You can set options in dialog boxes by performing actions such as clicking a button, selecting an option from a list, or typing information in a box. When several items or selections are related, they are grouped together in a group box within the dialog box. You may be enabled to select multiple items from a group box or you may be required to select only a single item, depending on the option you are setting. The methods for setting options in dialog boxes are described here:

Check Box — a square box that is toggled on or off by clicking the box. When several check boxes are grouped together, you can select multiple check boxes in a group.

Option Button — a button or box that turns an option on or off. In most dialog boxes, you can select only one option button in a group. Selecting one value un-selects all other values. In dialog boxes for some standard SAS windows, each selection is independent of the others so that you can select multiple options at the same time.

Command — buttons that perform an action in a dialog box or a command that you type in the command box (in a SAS window).

- Help - provides information about how to complete the dialog box.
- OK - closes a dialog box and saves or executes your selections.
- CANCEL - closes the dialog box without saving or executing your selections.
- RUN - executes the program in the active window.
- SAVE - saves the text, code, or options set in the window or dialog box.
- RESULTS - enables you to look at the source or the output of this program.
- CUSTOMIZE - displays windows that aid in changing descriptive information such as titles, footnotes, and keywords associated with the active program.
- CANCEL or GOBACK - ignores changes made to this window and returns to the window that invoked this window.
- ADVANCED - displays advanced options for your active task or selection.

List Box — a box that contains a list of items or selections. You can select an item or option from the list to activate the option or to perform an action on the item, such as modifying, viewing, deleting an item.

Text Box — a box in which you type information. In some cases, the box also provides arrows, which you can use to increase or decrease values, such as date or time values.

Icon Button — a push button that displays a list of possible values when you select the button. You can select one or more items from the list.

SAS IT Resource Management Tabs

The Administration Tab

The **Administration** tab in the main window of the SAS IT Resource Management GUI for UNIX and Windows provides the following icons (tasks)

- QuickStart Wizard
- Create PDB Wizard
- Process/Reduce Wizard
- Manage PDBs
- Manage Tables
- Examine PDB Data
- Site Options
- Define Work Shifts
- Define Holidays
- Explore Tables/Variables.

In the main menu of the SAS IT Resource Management GUI for z/OS, the buttons approximately correspond to the icons on the **Administration** tab.

Many of these tasks enable you to process and reduce your data and to set up and manage your PDB and its components on the SAS IT Resource Management server. These tasks require you to have write access to a PDB. (For more information, see "Actions and Access Rights Required" on page 44.) You must perform many of these tasks on the host where the SAS IT Resource Management server license is installed.

If you are accessing the PDB from a client host, you can access the PDB, view the tables and variables in a PDB, and create and manage the remote profiles for your PDBs.

All tasks available on the **Administration** tab can also be performed in batch mode by using SAS IT Resource Management macros. (For more information about the use of macros, see the SAS IT Resource Management macro reference documentation .)

For more information about the task that is associated with each icon, select the icon and refer to the Help available in the window that opens. Also, information about these tasks is available in Part 3 of the *SAS IT Resource Management User's Guide*.

The Reporting Tab

The **Reporting** tab in the main window of the SAS IT Resource Management GUI for UNIX and Windows and the SAS IT Resource Manager client GUI contains several unique tools that you can use to explore and analyze your IT resources. In the main menu of the SAS IT Resource Management GUI for z/OS, the buttons approximately correspond to the icons on the **Administration** tab.

There are report tools that enable you to perform ad hoc analysis and to drill down, create high-quality graphs that use patterns, colors, and symbols to represent different variables and trends in your data, and create reports that you can run, save, and rerun against your production data, interactively or in batch mode.

You can create basic analysis reports with one or more variables, reports that group your analysis variables based on a specific variable, spectrum reports that use color to display trends in your data, and exception reports that identify anomalies in your data based on exception criteria.

To access the reporting tools, start the SAS IT Resource Management client and select the **Reporting** tab from the SAS IT Resource Management main window. The **Reporting** tab displays a list of tasks that include the following:

Manage Report Definitions
: opens the Manage Report Definitions window, from which you can create, modify, run, and save the interactive form of report definitions. You can also save report definitions as macro calls that you can run in batch mode. For more information about report definitions, see "Using the Manage Report Definitions Tool" on page 468.

Explore Supplied Reports
: opens the Explore Supplied Reports window, from which you can browse a list of supplied report definitions. (For more information about supplied report definitions, see "Exploring Supplied Report Definitions (UNIX, Windows)" on page 488.)

Perform Exception Analysis
: opens the Exception Reporting window, from which you can create exception rules and run them against your data. (For more information about rule definitions, see "Using the Perform Exception Analysis Tool" on page 468.) You can identify exception conditions in your performance data without visually scanning many graphs or long tabular reports. For example, you can create an exception rule that will automatically identify any disk that is more than 99% full. Exception reporting applies a set of binary (true/false) rules to specific data. Users can modify the sample rules and can add new rules for site-specific conditions.

Graph Data
: opens the Graph Data (Graph-N-Go) window, from which you can do simple ad hoc examination of the data in your PDB by using graphs, charts, and drill-down techniques. For more information about graph data, see "Using the Graph Data Tool" on page 467.

Data Visualization
: opens the Data Visualization system, which enables you to interactively explore your data through a variety of sophisticated plots and charts. For more information about data visualization, see "Using the Data Visualization Tool" on page 466.

Use the following table to match your reporting needs with the tool that will best match them.

If you need...	**Consider using...**
Broad, ad hoc data exploration that uses a variety of existing report styles	Graph Data or Data Visualization
Production reports that you can run interactively or in batch mode	Explore Supplied Reports or Manage Report Definitions
Custom Reports that you can design, run, and save for use interactively or in batch mode	
Reports that you can distribute via the Web	
Automated production reports that identify error or exception conditions	Perform Exception Analysis

Note: For more information, see “Overview of Report Concepts and Tools” on page 465. △

The Applications Tab

The **Applications** tab in the main window of the SAS IT Resource Management GUI for UNIX and Windows and the SAS IT Resource Management client GUI lists the names of three work areas. You can use these work areas to store custom applications:

- User Applications - applications that you create for your own use
- PDB Applications - applications that you and others create for use with the active PDB
- Site Applications - applications that you or others at your site create for use by anyone at your site.

If you select one of these items on the **Applications** tab, then a window or work area opens. Initially these custom work areas are empty. However, you can create custom applications by adding a command icon, assigning a command to the icon, and saving the icon in one of the work areas.

The supplied application icons are stored in the toolbox. To open the toolbox, select the toolbox icon on the toolbar. The supplied icons in the toolbox work area include an editor, viewer, and a SAS/AF template.

Using the **Viewer** template, you can create a command to view a specific table in a PDB. Using the **Editor** template, you can create a command to start a specific editor that you frequently use. Using the **AF Template**, you can run any SAS command line command, including any arbitrary SCL code. Examples for using the templates are available in Application Tab Examples, later in this document.

Items that you or others store in the work areas on the **Applications** tab are saved in the locations listed below. Items that are stored in these areas are not removed or altered with future updates to SAS IT Resource Management.

- User Applications - your SASUSER directory.

  On UNIX, the SASUSER library is associated with your user ID or login.

  On z/OS, by default, the TSO CLIST that is supplied by SAS allocates SASUSER to a library that is unique to each user ID.

  On a Windows host, the SASUSER directory is not individual to the user, but is used for anyone who accesses SAS or SAS IT Resource Management on that host. On Windows, SASUSER is by default a directory that is stored in the location where SAS is installed.
- PDB Applications - the ADMIN library in the active PDB. You must have write or update access to the PDB in order to save or modify items in this work area. (For more information about access to the active PDB, see “Actions and Access Rights Required” on page 44 and “Viewing/Editing Your Active PDB’s Access Mode” on page 325.)
- Site Applications - the active site library. To save or modify items in this work area, you must have write or update access to SITELIB, which is the active site library. (For more information about access to the active site library, see “Actions and Access Rights Required” on page 44 and “Viewing/Editing Your Active Site Library’s Access Mode” on page 281.)

To create a new command button follow these steps:

1. Select the **Applications** tab in the SAS IT Resource Management main window.
2. Select the toolbox icon on the toolbar to open the SAS IT Resource Management toolbox.

3. Open the work area where you want to store the new command: **User Applications**, **PDB Applications**, **Site Applications**.
4. Copy one of the items/templates from the toolbox to your work area. To copy an item, simultaneously press the CTRL key and select from the toolbox the item to copy. For example, you might select the **AF Command Template** item from the toolbox. While selecting the item, drag it to your work area.
5. In your work area (not the toolbox), select the new item with your right mouse button, and from the pop-up menu select **Attributes**.
6. In the Attributes window, type the command that you want to execute when you select this item, and type a label or name for the item.
7. Select **Save** to save the items in the new workspace and close the workspace folder.
8. Select the new item in your work area to execute the command as necessary.

Application Tab Examples

The templates on the **Applications** tab enable you to create custom templates to quickly execute commands or SAS source code. The **Editor** and **Viewer** templates can be used to quickly open an editor or viewer that you frequently use. These are just a few examples of using the templates to run SAS commands.

The **AF Template** enables you to run any SAS command-line command, including SCL code. You can directly assign a command to the template/icon, such as

```
runeis appl=command
```

which will run the specified report object, or

```
submit "%cprunrpt(report,folder=libref.myrpts);"
```

which will run the specified report in the specified report folder.

Note: To invoke an AF application from the **AF Template**, you must use the AFA command rather than the AF command, since there is already an AF application in progress (namely, SAS IT Resource Management). △

To run a report definition from the AF command template, create an ASCII file that contains the macro and the report that you want to run. For example, you could create a file that contains

```
%cprunrpt(report,folder=libref.myrpts);
```

Open your copy of the AF template in your work area and type

```
pgm;include "c:\temp\example.sas";submit;
```

To use the AF template to query a table or to combine tables, copy the AF template to your user work area. Open the template and in the command box type

```
Query
```

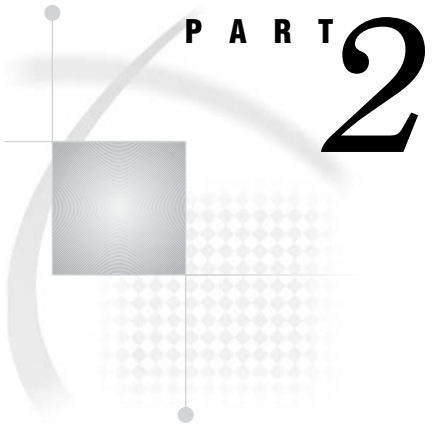
Save the template, and when you access this template in the future it will execute this command to open the SQL Query Tables window. For more information about SQL, select **Help** in the SQL window.

To use the SAS procedure GREPLAY, which replays or displays existing graphics output in a specific catalog, copy the AF template to your user work area. Open the template in your work area and type

```
proc greplay igout=my.catalog; run;
```

To start the GREPLAY procedure without pointing to a specific folder, select the **AF** icon and type

```
proc greplay; run;
```



PART 2

Setup

CHAPTER

3

Setup: Introduction

Setup: Overview

Overview of Setup

Note: You must install SAS IT Resource Management before you set it up to work with data from your collector(s). For more information about installation, in the Installation Guide for SAS see the section on SAS IT Resource Management. △

If you are setting up for the first time, here are the setup steps:

1. For each collector that you want to work with, find which setup case includes that collector and then follow the instructions for that setup case. After you finish setting up the server, adjust the access privileges to the objects that you created so that you and other users have the appropriate access.
2. Then, set up one or more clients.
3. Finally, you and every other user need to set up your default Web browser.

For additional information about setup, see the following topics:

- “Setting Up the Server, with Setup Cases” on page 38
- “Setting Up the Client” on page 43
- “Setting Up Your Default Web Browser” on page 43.

Note: After you set up SAS IT Resource Management to work with your collectors, occasionally the instructions accompanying a new release of SAS IT Resource

Management recommend a change to your setup. For example, a collector might have new fields for which it now collects data, and you might want to use that new data in SAS IT Resource Management. In that case, the installation instructions for the release will describe how to modify your setup to match the new release of SAS IT Resource Management. Similarly, a new release of SAS might require a change to SAS IT Resource Management. For example, the structure of SAS libraries changed in SAS Version 8. In that case, too, the installation instructions for the release describe how to migrate your setup to the new SAS release.

For information about the change instructions that were used in the past, see "Migrating Your SAS IT Resource Management Environment from SAS Version 6 to SAS Version 8" on page 243. △

Setting Up the Server

Setting Up the Server, with Setup Cases

The main goals for server setup are as follows:

- Create a PDB in which the data will be located. Add or create tables for the data.

  Also customize the PDB and the site library, as necessary.
- Create a program to process and reduce the day's data into the tables.

  Also customize the program, as necessary, and test.
- Create a report structure for displaying Web-accessible reports about the data.

  If the report structure is based on PDSs, also create and run a program that FTPs the report structure to directories.
- Create a program that produces Web-accessible reports about the data and writes them to the report structure.

  If the report structure is based on PDSs, also run the FTP program in order to FTP the reports to the directories.

  Additionally, customize the program(s), as necessary, and test.
- Schedule the process-and-reduce program and report generation program to run, typically one time each night.

  If the report structure is based on PDSs, also schedule the FTP-reports program to run, typically one time each night.

There are five setup cases. The setup cases contain instructions for setting up SAS IT Resource Management servers.

Note: The information in the setup cases takes precedence over the information in the *SAS IT Resource Management Server Setup Guide*. However, the setup cases can refer you to some collector-specific information that is found only in the *SAS IT Resource Management Server Setup Guide*. △

You can tell which setup case applies to your data by consulting the following list. Consider the setup cases in order, and take the first setup case that applies to your data. After you know which setup case applies to your data, you can see the instructions for that setup case at the end of the list.

Note: In setup cases 1, 2, and 5, the term "data collected on" refers to the data that can be collected on at least one of the operating systems that SAS IT Resource Management runs on. For example, "data collected on UNIX" means that the data can

be collected on at least one of the UNIX operating systems that SAS IT Resource Management runs on, such as HP-UX. △

- Setup Case 1: This case covers setup for data for which SAS IT Resource Management supplies one or more table definitions and has a path in the QuickStart Wizard.

  Here are the data collectors and data sources to which this case applies:

  - For a PDB on UNIX:
    - HP OpenView Performance Agent (formerly known as HP VantagePoint Performance Agent, HP MeasureWare, and HP Performance Collection Software) server data collected on UNIX or Windows
    - HP OpenView Reporter data from an Oracle database
    - Enhanced support NTSMF exchange data and/or server data collected on Windows (for NTSMF objects for which SAS IT Resource Management supplies table definitions; otherwise, see Setup Case 5)
    - SMF server data, storage data, and/or workload data collected on z/OS
    - SAP R/3 workload data collected by the SAS IT Management Adapter for SAP
    - SAP R/3 statistics file's workload data collected on UNIX or Windows
    - SAP R/3 API's workload data collected on UNIX or Windows
    - **`sar`** command's data collected on UNIX
    - Web log data in common log format (CLF) collected on UNIX or elsewhere
    - Web log data in extended log format (ELF) collected on UNIX or elsewhere
    - Web log data in Microsoft IIS original format collected on UNIX or elsewhere
    - Web log data in user-defined format collected on UNIX or elsewhere.
  - For a PDB on Windows:
    - HP OpenView Performance Agent (formerly known as HP VantagePoint Performance Agent, HP MeasureWare, and HP Performance Collection Software) server data collected on UNIX or Windows or by HP OpenView Reporter
    - HP OpenView Reporter data from an Oracle database
    - Enhanced support NTSMF exchange data and/or server data collected on Windows (for NTSMF objects for which SAS IT Resource Management supplies table definitions; otherwise, see Setup Case 5)
    - SMF server data, storage data, and/or workload data collected on z/OS
    - SAP R/3 workload data collected by the SAS IT Management Adapter for SAP
    - SAP R/3 statistics file's workload data collected on Windows or UNIX
    - SAP R/3 API's workload data collected on Windows or UNIX
    - sar command's data collected on UNIX
    - Web log data in common log format (CLF) collected on Windows or elsewhere
    - Web log data in extended log format (ELF) collected on Windows or elsewhere
    - Web log data in Microsoft IIS original format collected on Windows or elsewhere

- Web log data in user-defined format collected on Windows or elsewhere.

- For a PDB on z/OS:
  - HP OpenView Reporter data from an Oracle database
  - Enhanced support NTSMF exchange data and/or server data collected on Windows (for NTSMF objects for which SAS IT Resource Management supplies table definitions; otherwise, see Setup Case 5)
  - SAP R/3 workload data collected by the SAS IT Management Adapter for SAP
  - SAP R/3 statistics file's workload data collected on UNIX or Windows
  - SAP R/3 API's workload data collected on UNIX or Windows
  - **`sar`** command's data collected on UNIX
  - SMF server data, storage data, and/or workload data collected on z/OS
  - Web log data in common log format (CLF) collected on z/OS or elsewhere
  - Web log data in extended log format (ELF) collected on z/OS or elsewhere
  - Web log data in Microsoft IIS original format collected on z/OS or elsewhere
  - Web log data in user-defined format collected on z/OS or elsewhere.

- Setup Case 2: This case covers setup for data for which SAS IT Resource Management supplies one or more table definitions but for which SAS IT Resource Management does not have a path in the QuickStart Wizard.

  Here are the data collectors and data sources to which this case applies:

  - For a PDB on UNIX:
    - accton command's data collected on UNIX
    - Aprisma (formerly Cabletron) SPECTRUM data collected on UNIX (for one or more model types for which SAS IT Resource Management supplies table definitions; otherwise, see setup case 5)
    - BMC Software PATROL data collected on UNIX or Windows (in one or more classes for which SAS IT Resource Management supplies table definitions; otherwise, see setup case 5)
    - BMC Software Visualizer workload data collected on UNIX or Windows
    - Cisco NetFlow FlowCollector network data collected on UNIX
    - HP OpenView Network Node Manager data collected on UNIX or Windows (in one or more MIBs for which SAS IT Resource Management supplies table definitions; otherwise, see setup case 5)
    - Landmark Systems Performance Works data about Oracle collected on UNIX
    - PROBE/Net data collected on UNIX
    - ROLM PBX (CBX) data collected on UNIX or elsewhere
    - SiteScope data collected on UNIX and Windows
    - SunNet Manager or Sun Enterprise Manager data collected on UNIX (from one or more schema files for which SAS IT Resource Management supplies table definitions; otherwise, see setup case 5)
    - Concord TRAKKER data collected on UNIX.

  - For a PDB on Windows:
    - accton command's data collected on UNIX

- BMC Software PATROL data collected on Windows or UNIX (in one or more classes for which SAS IT Resource Management supplies table definitions; otherwise, see setup case 5)
- BMC Software Visualizer workload data collected on UNIX or Windows
- Cisco NetFlow FlowCollector network data collected on UNIX
- HP OpenView Network Node Manager data collected on Windows or UNIX (in one or more MIBs for which SAS IT Resource Management supplies table definitions; otherwise see setup case 5)
- Landmark Systems Performance Works data about Oracle collected on Windows
- ROLM PBX (CBX) data collected on Windows or elsewhere
- SiteScope data collected on Windows and UNIX.

- For a PDB on z/OS:
  - accton command's data collected on UNIX
  - BMC Software PATROL data collected on UNIX or Windows (in one or more classes for which SAS IT Resource Management supplies table definitions; otherwise, see setup case 5)
  - BMC Software Visualizer workload data collected on UNIX or Windows
  - Cisco NetFlow FlowCollector network data collected on UNIX
  - IBM DCOLLECT data collected on z/OS
  - IBM EREP data collected on z/OS
  - Candle ETEWATCH data collected on z/OS
  - Boole and Babbage (BMC Software) IMS Measurement Facility (IMF) data collected on z/OS
  - ROLM PBX (CBX) data collected on z/OS or elsewhere
  - IBM SMF data collected on z/OS
  - IBM Transaction Processing Facility data collected on z/OS
  - The Monitor for CICS from Landmark version 8.0 data collected on z/OS;
  - The Monitor for CICS from Landmark version 1.3 data collected on z/OS
  - The Monitor for CICS from Landmark version 2 or later data collected on z/OS
  - The Monitor for DB2 from Landmark data collected on z/OS
  - Computer Associates Tape Management System data collected on z/OS
  - IBM VM Monitor data collected on z/OS

- Setup Case 3: This case covers setup for data, in a character-delimited file, for which SAS IT Resource Management does not supply a table definition or have a path in the QuickStart Wizard.

  Additional criteria for this case are as follows:

  - Your data collector or data source writes raw data to a character-delimited file. Or you or a consultant can reformat the raw data as a character-delimited file.
  - The format of the raw data fits the requirements that are described in "Appendix 9: Using Character-Delimited Data," which is in "Using the Generic Collector Facility" on page 158.
  - The format of the raw data fits (or can be temporarily modified to fit) the requirements that are described in "Appendix 13: Character-Delimited

Support in GENERATE SOURCE," which is in "Using the Generic Collector Facility" on page 158.

For supplementary information for this case, see "Using the Generic Collector Facility" on page 158.

- Setup Case 4: This case covers setup for data, in a SAS data set, for which SAS IT Resource Management does not supply a table definition or have a path in the QuickStart Wizard.

  The additional criterion for this case is as follows:

  - You have one or more SAS programs (provided with MXG, written by you, or written by consultants) that read the raw data, optionally clean and transform the data in some way, and write the data to one or more SAS data sets or views.

  Among others, the AS/400 data falls into this setup case.

  For supplementary information for this case, see "Using the Generic Collector Facility" on page 158.

- Setup Case 5: This case covers setup for data in any of the following categories, for which SAS IT Resource Management does not supply a table definition:

  - a new OpenView/Tivoli NetView MIB
  - a new NTSMF object
  - a new DSI class
  - a new PATROL class
  - a new Aprisma SPECTRUM model type
  - a new SunNet Manager or Enterprise Manager schema file.

  The criteria for this case are as follows:

  - You used setup case 1 or 2 for the data collector or data source, and some of the data in the data collector or data source was not covered by setup case 1 or 2.
  - The *SAS IT Resource Management Server Setup Guide* covers the data collector or data source. For more information about the Server Setup Guide, see "Collector-Specific Setup Documentation" on page 12.

  Here are the data collectors or data sources to which this case applies:

  - For a PDB on UNIX:
    - Aprisma (formerly Cabletron) SPECTRUM data collected on UNIX (for a model type for which table definitions are not supplied; otherwise, see setup case 2)
    - Enhanced support NTSMF data collected on Windows (in an object for which a table definition is not supplied; otherwise, see setup case 1)
    - HP OpenView Network Node Manager data collected on UNIX or Windows (in a MIB for which a table definition is not supplied; otherwise, see setup case 2)
    - HP OpenView Performance Agent data collected on UNIX or Windows by means of DSI (in a class for which a table definition is not supplied; otherwise, see setup case 1)
    - BMC Software PATROL data collected on UNIX or Windows (in a class for which a table definition is not supplied; otherwise, see setup case 2)
    - SunNet Manager or Enterprise Manager data collected on UNIX (from a schema file for which a table definition is not supplied; otherwise, see setup case 2).

- For a PDB on Windows:
  - Enhanced support NTSMF data collected on Windows (in an object for which a table definition is not supplied; otherwise, see setup case 1)
  - HP OpenView Network Node Manager data collected on Windows or UNIX (in a MIB for which a table definition is not supplied; otherwise, see setup case 2)
  - HP OpenView Performance Agent data collected on Windows or UNIX by means of DSI (in a class for which a table definition is not supplied; otherwise, see setup case 1)
  - BMC Software PATROL data collected on Windows or UNIX (in a class for which a table definition is not supplied; otherwise, see setup case 2).
- For a PDB on z/OS:
  - Enhanced support NTSMF data collected on Windows (in an object for which a table definition is not supplied; otherwise, see setup case 1)
  - BMC Software PATROL data collected on Windows or UNIX (in a class for which a table definition is not supplied; otherwise, see setup case 2).

For the instructions for each setup case, see

- "Overview of Setup Case 1" on page 53
- "Overview of Setup Case 2" on page 65
- "Overview of Setup Case 3" on page 81
- "Overview of Setup Case 4" on page 99
- "Overview of Setup Case 5" on page 119.

Setting Up the Client

Setting Up the Client

After you have installed and set up the server(s) and installed your SAS IT Resource Management client software, you can set up the client by following these steps.

1. Start SAS IT Resource Management on your client platform. For more information, see "Starting the z/OS GUI" on page 19 or "Starting the UNIX and Windows GUI" on page 21.
2. Set up a remote server profile for each remote server that you will use to access a remote PDB. (For more information, see "Overview of Remote Server Profiles" on page 571.)
3. If you did not select a default Web browser during installation, you should specify the Web browser that you want to use by default with SAS IT Resource Management. For more information, see "Setting Up Your Default Web Browser" on page 43.

Setting Up the Web Browser

Setting Up Your Default Web Browser

The SAS System and SAS IT Resource Management provide information and enable you to create reports that can be displayed by using a Web browser.

Your current default Web browser might not be the one that you prefer.
On Windows, to change the default Web browser you can follow these steps.

1. Right-click the Internet Explorer desktop icon and select **Internet Options**.
2. On the **Programs** tab, check the box at the bottom of the window that is labeled "Internet Explorer should check to see whether it is default browser."
3. When you open your next web site, you will be asked this question and you can change your default browser.

On UNIX, to change the default Web browser you can specify the value of the X resource *SAS.webBrowser*. You can specify this in your Xresources file or by using the *-xrm* option when you start SAS. For example, to point to a locally installed copy of Netscape, you could issue your SAS command with the following option:

```
-xrm 'SAS.webBrowser:/usr/local/bin/netscape'
```

Setup: Related Topics

Actions and Access Rights Required

Administration

Logon host: where the PDB resides or will reside

License: the logon host has a server license

- *PDB Administration*

  Read/write required: PDB

  Read-only OK: SITELIB and PGMLIB

  - create a PDB
  - add/edit/create tables and regular variables
  - create/edit PDB-specific formats
  - create/edit PDB-specific formula variables
  - edit PDB-specific work shift
  - run process
  - run process with user exits
  - run process with data-duplication macros
  - run reduce
  - run reduce with external "detail"
  - save/modify PDB-specific report definitions in ADMIN
  - save PDB-specific graph or text reports in ADMIN
  - write PDB-specific data to downloadable file
  - maintain the PDB's data dictionary from the master data dictionary.

- *SITELIB Administration*

  Read/write required: SITELIB

  Read-only OK: PDB and PGMLIB

  - copy the shipped SITELIB to make a writable SITELIB

- □ create/edit site-specific work shift and holiday schedules
- □ create/edit site-specific formats
- □ create/edit site-specific report definitions.

□ *PGMLIB Administration*

Read/write required: PGMLIB

Read-only OK: Neither PDB nor SITELIB requires read access

- □ initial install of SAS IT Resource Management
- □ install of SAS IT Resource Management updates (maintenance and release)
- □ install of table definitions from the PDB's data dictionary to the master data dictionary.

Users (and Administrator)

Logon host: the host on which the PDB of interest is accessible by means of the local file system

License: the logon host must have a client license and/or server license (data visualization and DeskTop Reporter require at least a client license)

or

Logon host: the host on which the PDB of interest is accessible by means of a remote server

License: the logon host must have a client license (and can also have a server license, though that is not required for these tasks)

□ *General User: Use data and supplied, PDB-specific, and site-specific report and rule definitions*

- □ *Read/write required*: -NONE-
- □ *Read-only OK*: PDB and SITELIB and PGMLIB
  - □ display and perhaps print PDB-specific and site-specific graph and text reports by using the interactive interface
  - □ perform data visualization
  - □ run supplied and PDB-specific and site-specific report definitions, and display and perhaps print graph and text reports by means of the interactive interface.
  - □ put PDB-specific graph or text reports onto a Web

Note: These do not require read access to PDB or SITELIB or PGMLIB: △

- □ browse PDB-specific and site-specific graph and text reports by using a Web browser (this does require read access to the Web-enabled reports)

□ *Power User*: Modify/Create user-specific report definitions

Read/write required

Read-only OK: PDB and SITELIB and PGMLIB

- □ display and perhaps print tables, variables, and statistics
- □ modify and run supplied and PDB-specific and site-specific report definitions, and display and perhaps print graph and text reports by means of the interactive interface

- create and run user-specific definitions, and display and perhaps print graph and text reports in the interactive interface
- modify (but do not save)/use PDB-specific and site-specific formats
- save/edit report definitions in SASUSER
- save graph and text reports in SASUSER
- create/save user-specific formats in SASUSER.

Note: Read/write access to SASUSER is assumed in all cases, other than batch operations on z/OS (where it is optional). △

Note: Using your site's security system, verify (and adjust if necessary) access rights for SAS IT Resource Management structures:

Table 3.1 Access Rights Table

Structures	— Access Rights —	
	Administrator	User
MXG.MXG.FORMATS	READ	READ
MXG.MXG.SOURCLIB	READ	READ
MXG.USERID.SOURCLIB	READ/WRITE	READ
On UNIX: misc/cpe	READ	
On WNT: cpe\sasmisc	READ	

△

Grouping Reports in a Web Gallery

1 Report groupings usually make a particular report easier to find. Also, report groupings usually make a particular report easier to interpret, by comparing it with related reports in the same group.

Before you decide how to group your own reports, look at how reports are grouped in the QuickStart Examples.

a From the main window in in the SAS IT Resource Management server GUI for UNIX and Windows environments or in the SAS IT Resource Management client GUI, select **OnlineHelp ▶ Other ITRM Documentation ▶ QuickStart Examples**
If the examples are zipped, unzip them.

b To see the groupings for **`UNIX Server Reports`**, for example, select **`UNIX Server Reports`**. There are five level 0 groupings: **`Overview`**, **`Utilization`**, **`Service`**, **`Forecasts`**, and **`Exceptions`**.

c To the left of **`Overview`** is a folder with no symbol on it. This icon indicates that you have reached the end of a path. Click on **`Overview`**. You can access all of the report groups at this location by selecting the down arrow next to the **`Report Group`** field, and selecting one of the report groups from the list. In this case, there is only one report group, so to the right you are seeing all the reports at that location.

d To the left of **`Utilization`** is a folder with a plus sign (+) on it. This icon indicates that you have not reached the end of a path. Click on **`Utilization`**. There are two level 1 groupings: **`Top Systems`** and **`Critical Systems`**.

To the left of **Top Systems** is a folder with a plus sign (+) on it. You have still not reached the end of a path. Click on **Top Systems**. There are four level 2 groupings: **Processor**, **Memory**, **Storage**, and **Network**.

To the left of **Network** is a folder with no symbol on it. You have reached the end of a path. Select **Network**. You can access all of the report groups at this location by selecting the down arrow next to the **Report Group** field. (You may need to drag frame borders aside to get to the field and its down arrow). A list displays with this location's three report groups: **All Systems Network I/O**, **All Systems NFS Call Rate**, and **Top Systems Network I/O Analysis**.

Select **Top Systems Network I/O Analysis**. The reports that are displayed to the right are the ones that are in this report group.

e Select **Return** to return to the highest-level menu.

2 Decide on report groupings for your data source.

When you plan the groupings of supplied reports for your data source, there are several rules to keep in mind:

- Paths can be of any length, but 10 levels is a sensible limit and fewer is usually better.
- Reports are available only at the end of a path. At that location, there can be one or more report groups.
- A report group can be used for graphic reports or text reports but not both. If the report group is used for graphic reports, it can have any number of reports. If the report group is used for text reports, it can have only one report.

  Note: To see a sample text report in the QuickStart Examples, from the highest-level menu follow this path: **SAP R/3 Workload Reports ▶ Utilization ▶ Tabular Reports ▶ CPU**

  The output is one report. The width of the report varies depending on the width of the values on that page. △

You do not need to create any entities at this point. They will be created for you later.

3 Decide on names for the report groupings and their locations. You do not need to create any entities at this point. They will be created for you later.

- For each of your report groups: Choose a name (40 characters or less). This name will appear in the list that displays when you or another user selects the down arrow on the **Report Group** field. You will use this name later, in the %CPRUNRPT macro.

  In the QuickStart Examples, **Top Systems network I/O Analysis** is the name of a report group.
- For each end-of-path: Choose a name for its physical location (a directory for a PDB that is located in a UNIX or Windows environment; a PDS for a PDB that is located in a z/OS environment). The reports for all of its report groups will be in this location. You will use this name later, in the %CPHTREE macro.

  In the QuickStart Examples, the reports for the **All Systems Network I/O** report group, the **All Systems NFS Call Rate** report group, and the **Top Systems Network I/O Analysis** report group are in a directory named *pservtn*.
- Choose the level names for your paths. The names can be of any length. Shorter is usually better so that the user does not need to move one or more frame boundaries to see the right ends of the names.

In the QuickStart Examples, the level names on the paths are **`Overview`**, **`Utilization`**, **`Top Systems`**, **`Processor`**, **`Memory`**, **`Storage`**, **`Network`**, **`Critical Systems`**, and so on.

- For the data source: Choose a name for its physical location (a directory for a PDB that is located in a UNIX or Windows environment; a PDS for a PDB that is located in a z/OS environment). All the end-of-path locations will reside "under" this location. This is called the "sub" location.

  In the QuickStart Examples, the "sub" location for the UNIX server reports is the *pserv* directory.

- For the data source: Choose a menu name.

  In the QuickStart Examples, the menu name for the UNIX server reports is **`UNIX Server Reports (from HPOVPA Data)`**.

- Choose the name for a "super" location "under" which all the report groups for any data sources will reside. (You have only one data source now, but might have additional ones later.)
  - For a UNIX or Windows PDB: You need to decide on the full path and name for a directory. For example, you might name the location *\abc\itsvrpts* (for Windows) or */abc/itsvrpts* (for UNIX).
  - For a z/OS PDB: You need to decide on the fully qualified name for a PDS. For example, you might name the location *ABC.ITSVRPTS*.

  In the QuickStart Examples, as installed, the "super" location is *.../cpe/itsvdocs/qs/web* or *...\cpe\itsvdocs\qs\web*.

4 Read about how report structures are described to SAS IT Resource Management.

See "The Contents of the Catalog Entry" and "The Layout of a Line in the Outline" in the information about the CAT= parameter in the %CPHTREE macro, which can be found in the SAS IT Resource Management macro reference documentation.

Note: In your outline, you will ignore the fourth and fifth fields. △

Troubleshooting Batch Jobs That Fail

For information about troubleshooting batch jobs, see the topic "Troubleshooting Batch Jobs That Fail" in "Appendix 1: Recovery Procedures" in the Macro Reference.

Using the GENERATE SOURCE Control Statement

The GENERATE SOURCE control statement enables you to make a preliminary draft of the table and variable definitions for data from a collector for which SAS IT Resource Management does not supply table and variable definitions. The GENERATE SOURCE control statement is used with the %CPDDUTL macro.

For more information about the GENERATE SOURCE control statement, see the topic GENERATE SOURCE in "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements," in the SAS IT Resource Management macro reference documentation.

You can create your own table and variable definitions from the following sources of information:

- from an existing SAS data set

- from an existing table in a PDB
- from a recognized file format (for example, a Class Specification File for HP MeasureWare's Data Source Integration).

Use the GENERATE SOURCE statement of the %CPDDUTL macro to build a first draft of the CREATE TABLE, UPDATE TABLE, and CREATE VARIABLE control statements needed to define the table. The generated source statements can be stored in either a SAS catalog member of type .SOURCE or in an external flat file.

Typically, you would use GENERATE SOURCE when setting up to use the Generic Collector Facility, in order to aid in defining the table in the PDB. As such, the data set used for staging the data for the Generic Collector Facility would serve as the DATASET= parameter for the GENERATE SOURCE statement.

If you are required to use a staged data set as input to %C*x*PROCES (where *x*=M, P, S, or W) you must ensure that it conforms to the following rules:

- A variable called *DATETIME* must exist in the input data set. For type Event tables, DATETIME represents the date and time when the event occurred. For type Interval tables, datetime is usually the start of an interval. If you want to use an alternative datetime variable that is not named *DATETIME*, then use the DATETIME= parameter to identify that variable.
- If the table being created is a type Interval table, a variable called *DURATION* must exist in the input data set. DURATION is the total time (length) of the interval. If you want to use an alternative duration variable that is not named *DURATION*, then use the DURATION= parameter to identify that variable.

Review the generated source statements before executing them. While GENERATE SOURCE is as accurate as possible, it is not infallible. Examine the generated definitions to verify (and modify if necessary) the following things in particular:

- On the CREATE TABLE statement, the default age limit for each of the levels should be reviewed and, if necessary, modified.
- A CREATE VARIABLE control statement is built for each variable in the table. For examples of CREATE VARIABLE control statements, see "Examples" in the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation. Examine each CREATE VARIABLE statement to ensure that
  - the correct interpretation type is being set by the INTERPRET= parameter. The GENERATE SOURCE control statement builds all of its CREATE VARIABLE control statements in such a way that the statistics that are specified when a table is installed in the PDB (as indicated by the ISTATS= parameter) are set according to the interpretation type that is chosen for the variable. To see how the GENERATE SOURCE control statement determines what is specified as the interpretation type in the preliminary draft, refer to "Appendix 8: Algorithm Used by GENERATE SOURCE" on page 188. For more information about selecting appropriate variable interpretation types, see "Appendix 7: Variable Interpretations and Default Statistics" on page 187.
  - the name of the variable is easy to understand. If not, use a more meaningful variable name. For more information about the CREATE VARIABLE control statement, see the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation.

For more information about the context for using the GENERATE SOURCE control statement, see the following topics:

- "Overview of Setup Case 3" on page 81
- "Overview of Setup Case 4" on page 99.

Using the QuickStart Wizard

The QuickStart Wizard is available on the **Administration** tab in the SAS IT Resource Management GUI for UNIX and Windows, and provides a quick and easy way for you to get started using SAS IT Resource Management. The Wizard assists you in setting up your data with SAS IT Resource Management by guiding you through the process of creating a new performance data base (PDB), identifying the type of data that you want to store in the PDB, creating a structure in which your Web-enabled reports will be located, and creating SAS jobs that you can run when you exit the Wizard. For more information, see setup case 1 in "Setting Up the Server, with Setup Cases" on page 38.

Using the QuickStart Wizard you can

- create a new PDB
- specify the collector that you are using to gather data
- specify which type of that collector's data you will store in the PDB
- create the report structure in which your Web-enabled reports will be located. The QuickStart Wizard runs the job that creates the report structure and also saves that job so that you can see it if you want to modify the structure later.
- create SAS jobs that you can use to add tables to your PDB, process your performance data, reduce the data, and create reports based on that data.
- if you are running on z/OS and not using the z/OS UNIX File System area for your report structure, the QuickStart Wizard will also create a batch job that you can use to FTP your report files from the report structure to a UNIX or Windows system for viewing by using a Web browser.

If your data source is not currently supported in the QuickStart Wizard, you might want to run the Wizard with any data source and then use, as examples, the programs and reports that are created by the Wizard. By reviewing those programs and the documentation for the macros that are used in those programs, you can modify or create your own programs that perform the same tasks as the QuickStart Wizard process and reduction program and report program. For more information, see setup cases 2, 3, 4, and 5 in "Setting Up the Server, with Setup Cases" on page 38.

The **Getting Started with SAS IT Resource Management** book, which is shipped with SAS IT Resource Management, contains additional information about using the QuickStart Wizard. Also, as the Wizard guides you through the process, help is provided for each window. The following resources provide additional information about using the Wizard, running the jobs that are created by the Wizard, and modifying the reports that are created by the Wizard:

- Sample output from QuickStart report jobs is available by following this path from the main window of the SAS IT Resource Management client GUI: **OnlineHelp ▶ Other ITRM Documentation ▶ QuickStart Examples**
- The SAS IT Resource Management macro reference documentation lists the SAS IT Resource Management macros, some of which are used with the QuickStart Wizard. This documentation can assist you in creating your own QuickStart type of programs or in modifying the QuickStart programs. For more information about the macros, see " SAS IT Resource Management Macros Grouped by Task" in "Chapter 1: Using SAS IT Resource Management Macros" in the SAS IT Resource Management macro reference documentation.
- You can also work with galleries and their contents to change the regular reports, exception reports, help, notes, and so on that are created by your QuickStart report program. For more information, see "Overview of Galleries" on page 551.
- When you run the QuickStart Wizard, information and help files for your data source and reports are created within the *qs* directory (on UNIX or Windows or z/

OS) or *.QS* PDS (on z/OS) in the location where SAS IT Resource Management is installed. The information in these files will assist you in setting up your data collector or data source with SAS IT Resource Management.

CHAPTER 4

Setup Case 1

Setup Case 1

Overview of Setup Case 1

For general information about setting up the server, see “Setting Up the Server, with Setup Cases” on page 38. That topic also identifies which setup case applies to your situation.

Below are the instructions for setup case 1. For each step, there is an overview of the step and, after that, the actual directions for the step. Typically, UNIX and Windows have one or two sets of directions and z/OS has one or two sets of directions. The steps are in the following order.

1. “The QuickStart Wizard” on page 54
2. “The Raw Data” on page 55
3. “The PDB and Site Library” on page 55
4. “The Tables and Variables” on page 55
5. “The Process-and-Reduce Program” on page 56
6. “The Report Structure” on page 56
7. “The Report Program(s)” on page 57
8. “The Job Schedule” on page 58
9. “Your Notes” on page 59
10. “Optional Customization” on page 60
11. “Additional Data” on page 63.

The QuickStart Wizard

Run the QuickStart Wizard for your data collector or data source.

The wizard creates a PDB, creates a program that processes and reduces the data, creates a report structure (and saves the program with which it created the structure, and also creates (if necessary) a program for FTPing the structure and its reports), and creates a program that reports on the data.

- In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **`Administration`** tab, select **`QuickStart Wizard`**. Follow your data collector or data source's path through the wizard.

  The path creates a PDB (without any tables at this point) and a report structure.

  Also, the path creates a *qs* directory and, under that, creates a *cntl* subdirectory that has a *readme.txt* file, an *xrptstr.sas* file, an *xprocess.sas* file, and an *xreport.sas* file, where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other directories and files.

  Note: In the path through the wizard, you are asked to supply the location of the file that contains the raw data for the data collector or data source. If you do not already have the file, you will create the file in the next step. In the wizard, supply the filename of the existing file or the filename that you intend to use. (If you decide later to use a different name, you can edit the filename in the *xprocess.sas* file.) △

- In the SAS IT Resource Management server GUI for z/OS: In the main menu, select the **`PDB Admin`** button and, from the menu it opens, select **`QuickStart Wizard`**. Follow your data collector or data source's path through the wizard.

  The path creates a PDB (without any tables at this point).

  The path also creates a report structure. The location of the report structure depends on how you specified the parent location for the Web reports.

  - If you specified a directory in the z/OS UNIX File System as the parent location, the report structure consists of directories in the z/OS UNIX File System. This choice is called *z/OS with a directory-based report structure*.
  - If you specified a PDS as the parent location, the report structure consists of PDSs. This choice is called *z/OS with a PDS-based report structure*. In order to be able to view the reports with a Web browser, you will also create a corresponding directory-based report structure on a UNIX or Windows system. After you do that, this choice is called *z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system)*.

  Also, the path creates a *.QS.CNTL* PDS that has a *$$README* member, a *$xINSTR* member, an *xRPTSTR* member, an *xPROCESS* member, and an *xREPORT* member (and depending on your selections, can create an *xFTPHTML* member), where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other members, PDSs, and data sets. For example, the path can create other PDSs with *.QS* as a high-level qualifier and/or directories with */qs* in the directory path.

  Note: In the path through the wizard, you are asked to supply the location of the data set that contains the raw data for the data collector or data source. If you do not already have the data set, you will create that data set in the next step. So in the wizard, supply the name of the existing data set or the data set that you

intend to use. (If you decide later to use a different name, you can edit the data set name in the *xPROCESS* member.) △

Note: All new PDSs are created in a location that is based on the SAS system option named FILEDEV. Check with the person who installed SAS at your site to determine whether FILEDEV points to a set of temporary volumes. If FILEDEV does point to a set of temporary volumes, then you must move the PDSs to a permanent location. △

For more information about the QuickStart Wizard, see "QuickStart Wizard" on page 12.

The Raw Data

Start to collect raw data, if you do not already collect raw data.

Several steps ahead, you will need at least a day's worth (preferably a few days' worth) of raw data from your data collector or data source. By starting to collect data now, you will not need to wait for it later.

- For UNIX or Windows: To obtain raw data from the data collector or data source, follow the instructions in the *readme.txt* file that the QuickStart Wizard generated. (Direct the raw data to the file whose name you supplied to the wizard, above.)
- For z/OS: To obtain raw data from the data collector or data source, follow the instructions in the *$$README* member that the QuickStart Wizard generated. (Direct the raw data to the data set whose name you supplied to the wizard, above.)

For more information about your data, see the documentation from the vendor of your data collector or data source.

The PDB and Site Library

The QuickStart Wizard already created the PDB. The site library was created at the time that SAS IT Resource Management was installed. Customize the PDB and site library, as necessary.

Depending on your operating system, the QuickStart Wizard created a PDB that is located in a UNIX, Windows, or z/OS environment.

- For UNIX or Windows: In the *readme.txt* file that the QuickStart Wizard generated, follow the instructions for the PDB and site library.
- For z/OS: In the *$$README* member and *$xINSTR* member that the QuickStart Wizard generated, follow the instructions for the PDB and site library.

For more information about PDB properties, see "Overview of Whole PDBs" on page 296. For more information about site library properties, see "Overview of Site Libraries" on page 270.

The Tables and Variables

Your PDB does not have tables yet. The tables will be added automatically in the next step, when you run the process-and-reduce job that the QuickStart Wizard created.

For more information about tables, see "Overview of Tables" on page 353. For more information about variables, see "Overview of Variables" on page 378.

The Process-and-Reduce Program

The QuickStart Wizard already created a program to process and reduce data. Customize the process-and-reduce program, as necessary, and run it.

1 Customize the process-and-reduce program.
   - For UNIX or Windows: Follow the instructions in the comments in the *xprocess.sas* file that the QuickStart Wizard generated.
   - For z/OS: Follow the instructions in the comments in the *xPROCESS* member that the QuickStart Wizard generated.

2 To the program, add a step that backs up the whole PDB.

   In the long run, it does not matter much whether the backup step runs before or after the process and reduce steps. For now, however, put the backup step after the process and reduce steps, so that there will be some data in the PDB by the time that the PDB is backed up.

   For information, recommendations, and examples related to PDB backup, see "Backing Up a PDB" on page 303.

   Note: Backing up the whole PDB is not the same as archiving incoming data. You need to back up the whole PDB regardless of the decision that you made about archiving. For more information about archiving, see "Archiving Incoming Data" on page 411. △

3 When at least a day's worth of raw data has been collected, run the program that processes, reduces, and backs up the data.
   - For UNIX or Windows: In batch mode (because that is how you will run the nightly jobs), submit the *xprocess.sas* file (which is a SAS program) to SAS.
   - For z/OS: Create a program that invokes SAS and passes to SAS the *xPROCESS* member (which is a SAS program). Then, submit that job in batch mode (because that is how you will run the nightly jobs).

   The first time that you run the program (that is, this time), the PDB has no tables and so the program adds the appropriate table definitions (and their variable definitions) to the PDB before processing and reducing the data. In the future, when you run the program, the PDB already has the table (and variable) definitions, so the program immediately processes and reduces the data into those tables.

   For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18. For more information about processing data, see "Processing Data" on page 413. For more information about reducing data, see "Reducing Data" on page 416.

4 The data in the PDB corresponds to the raw data.

   If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.

5 The PDB backup has been created.

The Report Structure

The QuickStart Wizard already created a report structure. For z/OS with a PDS-based report structure, FTP the report structure to a UNIX or Windows system.

- For UNIX or Windows: The report structure that the QuickStart Wizard created is the only one that you need.

- For z/OS with a directory-based report structure (in the z/OS UNIX File System): The report structure that the QuickStart Wizard created is the only one that you need.
- For z/OS with a PDS-based report structure: You will need an additional report structure on a UNIX or Windows system. By FTPing the PDS-based report structure to a UNIX or Windows system, you will create a corresponding directory-based report structure.

  If you decide to FTP the report structure now, the primary benefit is having the work already taken care of instead of leaving it for later. If you decide to FTP the report structure in a later step, the primary benefit is a one-time saving of the time it takes for the program to construct FTP commands.

  The QuickStart Wizard already created a program to FTP the report structure to another system. If you want to FTP the report structure now, look at the *xFTPHTML* member that the QuickStart Wizard created. It contains calls to the %CPSTART macro and the %CMFTPSND macro. Follow the instructions in the comments in the program and then run the program.

  For more information about the %CMFTPSND macro, see the %CMFTPSND topic in the SAS IT Resource Management macro reference documentation. The %CMFTPSND macro can be used, as it is being used here, to FTP a PDS-based report structure in order to create a directory-based report structure. (The %CMFTPSND macro can also be used, as it will be used after the reports have been generated, to FTP anything not already FTP'd, such as the reports, to the directory-based report structure.)

  For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

The Report Program(s)

The QuickStart Wizard already created the program for generating reports on the data. Customize the reporting program, as necessary, and test.

1. Customize the program.
   - For UNIX or Windows: Follow the instructions in the comments of the *xreport.sas* file that the QuickStart Wizard generated.
   - For z/OS: Follow the instructions in the comments of the *xREPORT* member that the QuickStart Wizard generated.

   For more information about the macros in the reporting programs, see the "Macros Used for Analyzing Data" topic in Chapter 3, "Report Macros," in the macro reference documentation.
2. Run the program in order to generate reports.
   - For UNIX or Windows: In batch mode (because that is how you will run the nightly programs), submit the *xreport.sas* file (which is a SAS program) to SAS.

     Reports are generated to the directory-based report structure that was created by the QuickStart Wizard.

     To view the reports, point your Web browser to the *welcome.htm* file in the subdirectory named *web* in the location in which you created the report structure.
   - For z/OS with a directory-based report structure: In batch mode (because that is how you will run the nightly programs), submit the job in the *xREPORT* member (which contains a SAS program).

Reports are generated to the directory-based report structure that was created by the QuickStart Wizard.

To view the reports, point your Web browser to the *welcome.htm* file in the subdirectory named *web* in the location in which you created the directory-based report structure.

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): In batch mode (because that is how you will run the nightly programs), submit the job in the *xREPORT* member (which contains a SAS program). Then, follow the instructions in the *xFTPHTML* member if you did not do the earlier FTP step, and submit the customized job in the *xFTPHTML* member. This time, %CMFTPSND will FTP the reports to the additional report structure.

  As a result, reports are generated to the PDS-based report structure that was created by the QuickStart Wizard and then are transferred to the directory-based report structure that was created by the FTP program.

  Note: If you omitted the earlier FTP step, this step will FTP the report structure as well as the reports.) △

  To view the reports, point your Web browser to the *welcome.htm* file in the subdirectory named *web* in the location to which you FTP'd the report structure.

For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

3 The data in the reports corresponds to the data in the PDB.

When you navigate through the report structure in the left frame, one or more reports are displayed in the right frame when you reach the end of a navigational path.

Graph reports are displayed as "thumbnail" sketches. To view a full-size version of a graph report, click on its thumbnail sketch. To return the report to thumbnail size, select **`Shrink`**.

For more information about examining the data in the PDB, see "Viewing Data Values" on page 420.

4 Confirm that "read" access to the reports (or execute access, if appropriate to the report mechanisms) is available to you and to anyone else who needs to view them.

For more information about access rights, see "Actions and Access Rights Required" on page 44.

The Job Schedule

Schedule the nightly jobs. Typically the jobs run one time a day, at night. However, you can schedule them to run more or less often than one time a day, and at whatever time(s) work best for your site.

- For UNIX: Schedule *xprocess.sas* and *xreport.sas* to run (in that order) nightly.

  For more information, refer to your *cron* and *crontab* man pages.

- For Windows: Schedule *xprocess.sas* and *xreport.sas* to run (in that order) nightly.

  For more information, refer to the documentation for your task scheduler.

- For z/OS with a directory-based report structure: Schedule *xPROCESS* and *xREPORT* to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Schedule *xPROCESS*, *xREPORT*, and *xFTPHTML* to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.

Your Notes

Make notes in your own schedule.

- The first day after scheduling, confirm that the programs ran as scheduled.
- The first day after scheduling, confirm that the backup of the PDB was created and that it is identical to (or can be used to regenerate a structure that is identical to) the PDB.

  Unless you have a strong preference for leaving the backup step after the process and reduce steps, in the process, reduce, and backup job move the backup step in front of the process and reduce steps.
- Every day, check the logs from the nightly jobs for error messages, if any. You might also want to check for warning messages.

  If you need to restart a nightly job:

  - For UNIX or Windows: See the *restart.txt* file that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
  - For z/OS: See the *$RESTART* member that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
- In a step above, you updated the site library's list of holidays so that it is correct for your site through some date in the future. When that date is reached, SAS IT Resource Management will write a reminder to the log during the process-and-reduce run. But in case you miss seeing that reminder, you might want to add an item to your own schedule to update the list again before that last date.

  For more information about the holiday list, see "Viewing/Editing Your Active Site Library's List of Holidays" on page 286.
- In a step above, you set the workshift schedule in the PDB. If your site has seasonal shift changes (for example, summer hours), you might want to add an item to your own schedule to update the workshift schedule when the season changes.

  For more information about the workshift schedule, see "Viewing/Editing Your Active PDB's Work Shift Schedule" on page 334.
- You might want to schedule a periodic check of table and variable usage. If data is being kept for any variables or tables that are not being used for reporting, you might want to set the variables' or tables' Kept status to *No* so that you can save space in the PDB and time during processing and reduction.
- You might want to schedule a periodic check of space usage.

  For more information about space usage, see "Space Usage at a Level in a Table" on page 349.
- If you have any formats that are based on changeable information (for example, formats related to network configuration or storage devices), you might want to schedule a periodic review of the formats.

  For more information about formats, see SAS online help.

- At some sites, copies of the demonstration PDBs are used for practice or testing: for example, testing what a new report would look like, before a production PDB has enough data for a report.

  At some sites, a demonstration PDB is used when the active PDB must be some other PDB than the one that is to be used. (Some operations cannot run on the active PDB.)

  At other sites, one or more of the demonstration PDBs are never used.

  You can delete one or more of the demonstration PDBs if you have no need for them. Remember, though, that at least one PDB (demonstration PDB or site-created PDB) is required to start SAS IT Resource Management. For more information, see "Deleting a PDB" on page 315.

Optional Customization

After enough data has been collected to fully populate most of the reports (that is, they have data over the full time-span of the report), you might or might not want to make other customizations for your site.

- You might want to adjust the report definitions and rule definitions.

  For example, you might want to add, change, or delete definitions.

  For more information about report definitions, see "Using the Manage Report Definitions Tool" on page 468. For more information about rule definitions, see "Using the Perform Exception Analysis Tool" on page 468.

- You might want to adjust static files in the report structure(s).

  A report gallery's static files are the files that are *not* cleared when the reports in the gallery are cleared by %CPWEBINI, %CPMANRPT, or the WEBCLR= parameter.

  For information about customizing these files, see "Customizing a Report Gallery's Static Files" on page 562.

- You might want to age out old reports instead of deleting them.

  Where you want to age out old reports, replace the calls to the %CPWEBINI macro with calls to the %CPMANRPT macro, and make the corresponding changes to the OUTLOC= parameter values in the macro calls that generate reports.

  For more information about %CPWEBINI, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation. For more information about %CPMANRPT, see the topic %CPMANRPT in the SAS IT Resource Management macro reference documentation.

- You might want to customize the report structure.

  Typically, the report structure is not customized. However,

  - you can add new paths in the report structure. If you add new paths, be very careful to specify end-of-path locations that either do not currently exist or, if they exist, are empty and not used for any other purpose.
  - where you use %CPWEBINI to clear reports, you can remove or revise the existing paths in the report structure.
  - where you use %CPMANRPT to age out reports, you can adjust a few aspects of the report structure: change the titles, and change the text (but not directory names) that is displayed on existing paths. You cannot remove or revise existing paths in the report structure (unless you are willing to risk losing all the reports on those paths).

To customize the report structure:

- For UNIX or Windows: Edit the *xrptstr.sas* file, which contains the program that the QuickStart Wizard used to generate the report structure. You might also need to edit the catalog entry/ies that the program calls. (For more information about editing catalog entries, see the Note below.)

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the *xrptstr.sas* program.

  Finally, revise the *xreport.sas* program to generate reports to the appropriate destinations in the revised report structure, as necessary.
- For z/OS with a directory-based report structure: Edit the *xRPTSTR* job, which contains the program that the QuickStart Wizard used to generate the report structure. You might also need to edit the catalog entry/ies that the program calls. (For more information about editing catalog entries, see the Note below.)

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the *xRPTSTR* job.

  Finally, revise the *xREPORT* job to generate reports to the appropriate destinations in the directory-based report structure, as necessary.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Edit the *xRPTSTR* job, which contains the program that the QuickStart Wizard used to generate the report structure. You might also need to edit the catalog entry/ies that the program calls. (For more information about editing catalog entries, see the Note below.) Then, make the corresponding edits in the *xFTPHTML* job, as necessary.)

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structures if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structures. Submit the *xRPTSTR* job and then the *xFTPHTML* job.

  Finally, revise the *xREPORT* job to generate reports to the appropriate destinations in the PDS-based revised report structure, as necessary. (The nightly *xFTPHTML* job will FTP the reports to the directory-based report structure.)

Note: To edit an entry in a SAS catalog, you can use the SAS Program Editor window. For example, suppose that you want to copy the entry in *PGMLIB.JSWIZCAT.JNEXNGTR.SOURCE*, edit it, and then save it to *ADMIN.JSWIZCAT.JNEXNGTR.SOURCE*, which is in the ADMIN library in your active PDB. Here are the steps:

1. Open the SAS Program Editor window.
2. In the command field, type the following command and press ENTER:

   ```
   copy pgmlib.jswizcat.jnexngtr.source
   ```

3. The entry appears. Edit the entry.
4. It is recommended that you not save the program back to PGMLIB, even if you have write access to PGMLIB, because your new version will disappear when PGMLIB is replaced at the next release.

   You can save the entry to any library that exists and has a libref that is defined with write access. (If the library exists and a libref is not currently

defined or is defined with read-only access, you can define a libref by using the SAS LIBNAME statement.)

The library (in the active PDB) that the ADMIN libref points to is a convenient one, and the libref is already defined if SAS IT Resource Management has been invoked in the current SAS session.

In order to write to ADMIN, if the PDB is activated with read-only access, then activate it with write access.

In the command field, type the following command and press ENTER:

```
save admin.jswizcat.jnexngtr.source
```

If the PDB was activated initially with read-only access, activate it again with read-only access.

5 Close or iconize the SAS Program Editor window.

6 Save the original report structure program under a different name.

7 Then, edit the report structure program to refer to the entry in ADMIN instead of the entry in PGMLIB.

△

For more information about grouping reports in the tree, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide. For more information about the %CPHTREE macro, which is called in *xrptstr.sas* and *xRPTSTR*, see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. For more information about the %CPENTCPY macro, which is called in *xrptstr.sas* and *xRPTSTR*, see the %CPENTCPY topic in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

- You might want to adjust the table definitions and the variable definitions.

  For more information about tables, see "Overview of Tables" on page 353. For more information about variables, see "Overview of Variables" on page 378.

- You might want to adjust the PDB properties and site library properties.

  For more information about PDBs, see "Overview of Whole PDBs" on page 296. For more information about site libraries, see "Overview of Site Libraries" on page 270.

- You might want to review how checking for duplicate data is handled.

  For more information about checking for duplicate data, see "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation.

- If you want to change some aspect of the process step and you do not see any supplied way of making that change, you might want to read about process exits.

  For more information about process exits, see "Overview of Using Process Exits" on page 442.

- You might want to combine or split PDBs.

  For information about combining PDBs, see "Combining PDBs" on page 304. For information about splitting PDBs, see "Splitting One PDB into Two PDBs" on page 320.

- If your users will use the SAS IT Resource Management client GUI to report on data in a PDB on a SAS IT Resource Management server host, make a list of the answers that the users will need in order to create a remote server profile for that host.

For more information, see “Creating a Remote Server Profile” on page 574.

Additional Data

If you have additional data in an object that is not covered by setup case 1, see “Overview of Setup Case 5” on page 119.

CHAPTER 5

Setup Case 2

Setup Case 2

Overview of Setup Case 2

For general information about setting up the server, see “Setting Up the Server, with Setup Cases” on page 38. That topic also identifies which setup case applies to your situation.

Following are the instructions for setup case 2. For each step, there is an overview of the step and, after that, the specific directions for the step. Typically, UNIX and Windows have one or two sets of directions and z/OS has one or two sets of directions. The steps are in the following order.

1. “The QuickStart Wizard” on page 66
2. “The Raw Data” on page 67
3. “The PDB and Site Library” on page 67
4. “The Tables and Variables” on page 68
5. “The Process-and-Reduce Program” on page 69
6. “The Report Structure” on page 71
7. “The Report Program(s)” on page 74
8. “The Job Schedule” on page 76
9. “Your Notes” on page 76
10. “Optional Customization” on page 78
11. “Additional Data and Cleanup” on page 80.

The QuickStart Wizard

Run the QuickStart Wizard.

The QuickStart Wizard does not have a path for your data collector or data source. However, the QuickStart Wizard generates objects that contain, among other things, documentation. In this setup case, you run the QuickStart Wizard *only* to acquire that documentation, which will be useful when you work with your own data collector or data source.

To select the path to use in the wizard, follow these instructions:

1. Access information about any process macro; for example, the %CPPROCES macro. See the topic %CPPROCES in the SAS IT Resource Management macro reference documentation.
2. In the description of the %CPPROCES macro's COLLECTR= parameter, see the Collector/Toolname table and find the line for your data collector or data source. On that line, under the collector name, find which %C*x*PROCES macro(s) is available for use by data from your data collector or data source, where *x* = M, P, S, or W.
3. If the %CPPROCES macro is available for your data collector or data source, take the NTSMF path through the QuickStart Wizard. If the %CPPROCES macro is not available, and if the %CSPROCES macro or %CWPROCES macro is available, take the HP OVPA path through the QuickStart Wizard. If the %CPPROCES macro is not available, and if the %CMPROCES macro is available, take the SMF path through the QuickStart Wizard.

The wizard creates a PDB, creates a program that processes and reduces the data, creates a report structure (and saves the program with which it created the structure, and also creates (if necessary) a program for FTPing the structure and its reports), and creates a program that reports on the data. In later steps in this setup case, you will generate similar (but not identical) structures and programs for your own data collector or data source.

- In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **`Administration`** tab, select **`QuickStart Wizard`**. Follow the path of the data collector or data source that you selected earlier in this step (by looking in the Collector/Toolname table). As the wizard asks for names and locations, invent names and locations that are appropriate for that path's data collector or data source, not your data collector or data source. (For example, when you are asked to supply the location of the file that contains the raw data for the path's data collector or data source, type a location that would be legitimate on your operating system. The file does not need to exist.)

  The path creates a PDB (without any tables at this point) and a report structure.

  Also, the path creates a *qs* directory and, under that, creates a *cntl* subdirectory that has a *readme.txt* file, an *xrptstr.sas* file, an *xprocess.sas* file, and an *xreport.sas* file, where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other directories and files.
- In the SAS IT Resource Management server GUI for z/OS: In the main menu, select the **`PDB Admin`** button and, from the menu it opens, select **`QuickStart Wizard`**. Follow the path of the data collector or data source that you selected earlier in this step (by looking in the Collector/Toolname table). As the wizard asks for names and locations, with one exception invent names and locations that are appropriate for that path's data collector or data source, not your data collector or data source. The exception is that, at one point in the path, the wizard asks where you want the parent location for the Web reports to be. Answer in the way that you want to set up for *your* data collector or data source.

The path creates a PDB (without any tables at this point).

The path also creates a report structure. The location of the report structure depends on how you specified the parent location for the Web reports.

- If you specified a directory in the z/OS UNIX File System as the parent location, the report structure consists of directories in the z/OS UNIX File System. This choice is called *z/OS with a directory-based report structure*.
- If you specified a PDS as the parent location, the report structure consists of PDSs. This choice is called *z/OS with a PDS-based report structure*. In order to be able to view the reports with a Web browser, you will also create a corresponding directory-based report structure on a UNIX or Windows system. After you do that, this choice is called *z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system)*.

Also, the path creates a *.QS.CNTL* PDS that has a *$$README* member, a *$xINSTR* member, an *xRPTSTR* member, an *xPROCESS* member, an *xREPORT* member, and an *xFTPHTML* member, where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other members, PDSs, and data sets. For example, the path can create other PDSs with *.QS* as a high-level qualifier and/or directories with */qs* in the directory path.

Note: All new PDSs are created in a location that is based on the SAS system option named FILEDEV. Check with the person who installed SAS at your site to determine whether FILEDEV points to a set of temporary volumes. If FILEDEV does point to a set of temporary volumes, then you must move the PDSs to a permanent location. △

The Raw Data

Start to collect raw data, if you do not already collect raw data.

Several steps ahead, you will need at least a day's worth (preferably a few days' worth) of raw data from your data collector or data source. By starting to collect data now, you will not need to wait for it later.

For more information, see the instructions about the data collector or data source in the *SAS IT Resource Management Server Setup Guide*. (For more information about the Server Setup Guide, see "Collector-Specific Setup Documentation" on page 12.) Also, see the documentation from the vendor of the data collector or data source.

The PDB and Site Library

Create a PDB for your data collector or data source. The site library was created at the time that SAS IT Resource Management was installed. Customize the PDB and site library, as necessary.

1 Create a PDB.

- In the SAS IT Resource Management GUI for UNIX or Windows: On the main window's **`Administration`** tab, select **`Create PDB Wizard`**. Follow the path through this wizard, including selection of the supplied table definitions that you want to be added to the new PDB. When the wizard asks if you want the new PDB to become the active PDB, answer *Yes*.

  The result is a PDB with table definitions. Also, the new PDB becomes the active PDB.

- In z/OS: You can use interactive mode only, or you can use a combination of batch mode and interactive mode.
  - Using interactive mode only: In the SAS IT Resource Management server GUI for z/OS, on the main menu select **PDB Admin ▶ Define New PDB**

    Follow the path through this wizard, including allocation of space and selection of the table definitions that you want to be added to the new PDB. (For good values to use for space allocation, see the *ALLOC* job step in the *CMQSTART* member in the SAS IT Resource Management *CPMISC* PDS.)

    The result is a PDB with table definitions.
  - Using a combination of batch mode and interactive mode: In batch mode, make a copy of the *CMQSTART* member that is in the SAS IT Resource Management *CPMISC* PDS. In the copy of CMQSTART, delete the call to the %CPSTART macro and replace it with a copy of the call to %CPSTART that is in the *CMAUTOEX* member in *CPMISC* and, in its PDB= parameter, specify the PDB that you created for your data collector or data source. In the copy of CMQSTART, also delete everything after the *ALLOC* job step. Follow the instructions in the comments that apply to the job steps that remain. Submit the copy as a batch job. At this point, the result is a PDB without table definitions. For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

    Then, in the SAS IT Resource Management server GUI for z/OS, from the main menu select **PDB Admin ▶ Config Active PDB Dictionary ▶ Collector List ▶ Add Table**

    Select the value that is appropriate for your data collector or data source, if it is not already selected, and select **OK**. The Select Tables window opens and displays a list of the supplied table definitions that are available for your data collector or data source. Select the table definitions that you want to use, and then select **OK ▶ File ▶ End ▶ OK**

    The table definitions are added to the PDB, and SAS IT Resource Management returns you to the main menu. The result is a PDB with table definitions.

For more information about PDBs, see "Overview of Whole PDBs" on page 296. For more information about tables, see "Overview of Tables" on page 353.

2 Customize the PDB and site library, as necessary.
- For UNIX or Windows: In the *readme.txt* file that the QuickStart Wizard generated, follow the instructions for the PDB and site library.
- For z/OS: In the *$$README* member and *$xINSTR* member that the QuickStart Wizard generated, follow the instructions for the PDB and site library.

For more information about PDB properties, see "Overview of Whole PDBs" on page 296. For more information about site library properties, see "Overview of Site Libraries" on page 270.

The Tables and Variables

You already added tables to the PDB.

If you want to see the tables:

- In the SAS IT Resource Management GUI for UNIX or Windows: From the main window's **Administration** tab, select Manage Tables. The Manage Tables window opens and lists the tables in the PDB.
- In the SAS IT Resource Management GUI for z/OS: From the main menu, select **PDB Admin ▶ Config Active PDB Dictionary**

  The Dictionary Tables window opens and lists the tables in the PDB.

The Process-and-Reduce Program

Create a program to process and reduce data.

1 After you have at least one day's worth of raw data from your data collector or data source, process and reduce the data.

   - In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **Administration** tab, select **Process/Reduce Wizard**.

     The first part of the wizard's path enables you to process the data interactively. (Also, in the first part of the wizard's path, before processing the data, when you see the **Save** button in the process part of the wizard, select the button, specify a location, and save. The result is a file that contains the batch equivalent of processing the data interactively. You will use this file in a later step.)

     Then, continuing in the wizard, the second part of the wizard's path enables you to reduce the data interactively. (Also, in the second part of the wizard's path, before reducing the data, when you see the **Save** button in the reduce part of the wizard, select the button, specify the same location that you specified in the process part of the wizard, save, and append. The result is that the above file now also contains the batch equivalent of reducing the data interactively. You will use this file in a later step.)
   - In z/OS:

     a Make a copy of the *CMQSTART* member that is in the SAS IT Resource Management *CPMISC* PDS. (This is an additional copy, if you already made a copy in a previous step.)

     b In the copy of CMQSTART, delete the call to the %CPSTART macro and replace it with a copy of the call to the %CPSTART macro that is in the *CMAUTOEX* member in *CPMISC*.

     c In the copy of CMQSTART, also delete the *CLEANUP* job step, delete the *ALLOC* job step, and, in the *LOAD* job step, delete everything after the %PUT statement that is associated with the call to the %CPREDUCE macro.

     d Access information about any process macro; for example, the %CPPROCES macro. See the topic %CPPROCES in the SAS IT Resource Management macro reference documentation.

     e In the description of the %CPPROCES macro's COLLECTR= parameter, see the Collector/Toolname table and find the line for your data collector or data source. On that line, under the collector name, find which %C*x*PROCES macro(s) are available for use by data from your data collector or data source, where *x* = M, P, S, or W.

     f If the %CMPROCES macro is available for your data collector or data source, replace the existing table list with a list of the tables that you selected for your data source, and replace the existing values for the COLLECTR= parameter and TOOLNM= parameter with the values, from the Collector/Toolname table, for your data collector or data source.

If the %CMPROCES macro is not available for your data collector or data source, replace "%CMPROCES(," with "%CPPROCES(", replace the existing table list with a list of the tables that you selected for your data collector or data source, replace the existing values for the COLLECTR= parameter and TOOLNM= parameter with the values, from the Collector/Toolname table, for your data collector or data source, and delete "WKCMPRES=NO,".

g Follow the instructions in the comments that apply to the job steps that remain.

h Submit this copy as a batch job (because the nightly jobs will run in batch mode).

For more information about processing data, see "Processing Data" on page 413. For more information about reducing data, see "Reducing Data" on page 416. For more information about the %CMPROCES macro, see the topic %CMPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPREDUCE macro, see the topic %CPREDUCE in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

Note: Theoretically, it is possible to process and reduce the data interactively. In practice, if you process a large amount of data interactively, your TSO transaction is so long that its priority drops severely during the process step. Usually, it is much faster to process and reduce the data in batch mode. △

2 The data in the supplied tables corresponds to the raw data.

If you want to examine the data in the PDB, see "Viewing Data Values" on page 420.

3 Create the program that runs nightly to process and reduce the data.

This SAS program calls (in this order) the %CPSTART macro (to invoke SAS IT Resource Management), one of the %C*x*PROCES macros (to process data), and the %CPREDUCE macro (to reduce data).

For more information about the %CPSTART macro, see the %CPSTART topic in the SAS IT Resource Management macro reference documentation. For more information about the %CMPROCES macro, see the %CMPROCES topic in the SAS IT Resource Management macro reference documentation. For more information about the %CPPROCES macro, see the %CPPROCES topic in the SAS IT Resource Management macro reference documentation. For more information about the %CSPROCES macro, see the %CSPROCES topic in the SAS IT Resource Management macro reference documentation. For more information about the %CWPROCES macro, see the %CWPROCES topic in the SAS IT Resource Management macro reference documentation. For more information about the %CPREDUCE macro, see the %CPREDUCE macro in the SAS IT Resource Management macro reference documentation.

□ For UNIX or Windows: Above, you created a file by using the Save button during processing and reduction in the Process/Reduce Wizard. You can use this file as the nightly process-and-reduce program.

You might want to rename the file to indicate its data collector or data source and role (to process and reduce the data). For example, if the file is for ACCTON command data, you might rename the file *acctprrd.sas*.

Note: This file corresponds to the *xprocess.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to

look at that program and see if there are any other comments or features that you want to use in your program. △

- For z/OS: Above, you created a job that processed and reduced data. You can use this job as the nightly process-and-reduce program.

  You might want to rename the job to indicate its data collector or data source and role (to process and reduce the data). For example, if the job is for ACCTON command data, you might rename the copy *ACCTPRRD*.

  Note: This job corresponds to the *xPROCESS* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job to see if there are any other comments or features that you want to use in your job. △

4 To accompany the nightly process-and-reduce program, write a program that backs up the whole PDB.

  For information, recommendations, and examples about PDB backup, see "Backing Up a PDB" on page 303.

  Note: Backing up the whole PDB is not the same as archiving incoming data. You need to back up the whole PDB regardless of the decision that you made about archiving. For more information about archiving, see "Archiving Incoming Data" on page 411. △

5 Run the backup program.

6 The PDB backup was created and is identical to (or can be used to regenerate a structure that is identical to) the PDB.

7 Add the backup step to the nightly process-and-reduce program.

  Unless you have a strong preference for putting the backup step after the process and reduce steps, put the backup step in front of the process and reduce steps.

  Note: You are building in the backup step as part of the nightly process-and-reduce program, rather than leaving it as a separate program, so that the backup step is sure to run nightly. △

The Report Structure

Note: If there are supplied report definitions, follow the instructions in this step; otherwise, follow the instructions for reporting in "6. The Report Structure" in setup case 4. △

Create a report structure. In one case (with a PDS-based report structure), also FTP the report structure to another system. Check the report structure(s).

1 Run the supplied report definitions for your data collector or data source.

- In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **`Reporting`** tab select **`Manage Report Definitions`**. In the Folders list, select the folder *PGMLIB.ITSVRPT* if it is not already selected. *Current PDB Reports* displays a list of the supplied report definitions that refer to the tables in your PDB. Select the name of one of the supplied report definitions, and then select **`Run`**.

  You might want to print the report for your convenience in checking and for your use in a later step.

  Repeat for some or all of the other supplied report definitions for your data collector or data source.

- In the SAS IT Resource Management server GUI for z/OS: From the main menu select **Reports ▶ Supplied Reports**

  The Supplied Reports for Datasets in PDB window opens and displays a list of the supplied report definitions that refer to the tables in your PDB. Select the name of one of the supplied report definitions for your data collector or data source, and then select **ItemActions ▶ Run Report**

  You might want to print the report for your convenience in checking and for your use in a later step.

  Repeat for some or all of the other supplied report definitions for your data collector or data source.

For information about report definitions in the SAS IT Resource Management client GUI, see "Using the Manage Report Definitions Tool" on page 468.

Note: In a later step, you will need the names of the supplied report definitions that you want to use. (You can use some or all of the supplied report definitions that are available for your data collector or data source.) The names are available in two places: in the list in the GUI, and in the lower-left corner of each report. △

2 Decide on report groupings for the supplied reports for your data collector or data source.

Report groupings usually make a particular report easier to find. Also, report groupings usually make a particular report easier to interpret, by making it easy to compare it with related reports in the same group.

For more information about report groupings, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide.

3 Create a SAS program.

The program creates a report structure that implements the paths and end-of-path directories that you decided on for your report groupings. The program should call the %CPSTART macro (to invoke SAS IT Resource Management), the %CPCAT macro (to read the description of the report structure), the %CPCAT macro again (to store a description of the report structure temporarily), and the %CPHTREE macro (to read the stored description and create the structure).

For more information about the %CPCAT macro, see the topic %CPCAT in the SAS IT Resource Management macro reference documentation.

For more information about the %CPHTREE macro, see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. In the %CPHTREE topic, focus especially on the outline and on Example 6. In the outline, as described in the documentation of the CAT= parameter, you can ignore the fourth and fifth fields for now. In Example 6, you can ignore sections 4, 5, 6, and 7 for now.

- For UNIX: In the outline, specify directory locations in the local UNIX file system. The directories do not need to exist. In a later step, you will run the program and it will create the directories.

  Note: This file corresponds to the *xrptstr.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program to see if there are any other comments or features that you want to make use of in your program. △

- For Windows: In the outline, specify directory locations on a disk that is attached to your PC or on a disk that is available to your PC through a service such as Network Neighborhood, NFS, AFS, or DFS. The directories do

not need to exist. In a later step, you will run the program and it will create the directories.

Note: This file corresponds to the *xrptstr.sas* file that the QuickStart Wizard created for the other data collector or source. You might want to look at that program to see if there are any other comments or features that you want to make use of in your program. △

- For z/OS with a directory-based report structure: In the outline, specify directory locations in a local z/OS UNIX File System area. The directories do not need to exist. In a later step, you will run the program and it will create the directories.

  Note: This program corresponds to the *xRPTSTR* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job to see if there are any other comments or features that you want to use in your job. △

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): In the outline, specify PDS locations. The PDSs do not need to exist. In a later step, you will run the program and it will create PDSs.

  Note: This program corresponds to the *xRPTSTR* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

You might want to name the program to indicate its data collector or data source and its role (to produce a report structure). For example, if you are working with data from the ACCTON command, you might call the program *acctrpts.sas* (for UNIX or Windows) or *ACCTRPTS* (for z/OS).

4 For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): create an additional SAS program.

You will use this program to FTP the report structure from its PDS-based location on z/OS to a directory-based location on a UNIX or Windows system. This program should call the %CPSTART macro (to invoke SAS IT Resource Management) and the %CMFTPSND macro (to FTP the report structure to the other system).

For more information about the %CMFTPSND macro, see the %CMFTPSND topic in the SAS IT Resource Management macro reference documentation.

For more information about %CMFTPSND, also see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. In the %CPHTREE topic, focus on Example 6. You can ignore sections 2, 3, 4, 5, and 7 for now. In section 6, you can ignore the NEXNG PDSs for now.

- Create the program. The directories do not need to exist. In the next step, you will run the program and it will create the directories.

  Note: This program corresponds to the *xFTPHTML* member that the QuickStart Wizard created for the other data source. You might want to look at that job to see if there are any other comments or features that you want to use in your job. △

You might want to name the program to indicate its data source and its role (to FTP the report structure). For example, if you are working with data from the ACCTON command, you might call the program *ACCTFTP*.

5 Run the SAS program(s).

- For UNIX or Windows: Run the program that creates the directory-based report structure.
- For z/OS with a directory-based report structure: Run the program that creates the directory-based report structure.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the program that creates the PDS-based report structure. Then run the program that FTPs the report structure to a UNIX or Windows system.

For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

6 Check the report structure(s).

Notice that the report structure is, physically, relatively flat.

- For UNIX or Windows: Check that the directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the welcome.htm file in the "super" directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)
- For z/OS with a directory-based report structure: Check that the directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the welcome.htm file in the "super" directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Check that the PDSs have been created in the location and with the names that you specified. Also check that the corresponding directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the welcome.htm file in the "super" directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)

The Report Program(s)

Create the program that runs nightly to generate reports to the report structure, and test the program.

1 Create the program.

This SAS program should contain a call to the %CPSTART macro (to invoke SAS IT Resource Management). Also, for each end-of-path location in the report structure (that is, for each location that contains report groups), the program should contain one call to the %CPWEBINI macro (to clear old reports) and one or more calls to the %CPRUNRPT macro (to generate the reports for the one or more report groups at that location). Here is an example:

```
%CPWEBINI (cat=work.grweb_gr,
   dir=end-of-path-1) ;
%CPRUNRPT (srd1 srd2 srd3 srd4,
   folder=pgmlib.itsvrpt,
   outloc=work.grweb_gr,
   outmode=web,
   outdesc=Report Group A,
   htmldir=end-of-path-1,
```

```
   webstyle=gallery2) ;
%CPRUNRPT (srd5 srd6,
   folder=pgmlib.itsvrpt,
   outloc=work.grweb_gr,
   outmode=web,
   outdesc=Report Group B,
   htmldir=end-of-path-1,
   webstyle=gallery2) ;
```

This code clears reports from the catalog *WORK.GRWEB_GR* and its corresponding directory *end-of-path-1*. Next, the code runs supplied report definitions *srd1*, *srd2*, *srd3*, and *srd4* from the folder *PGMLIB.ITSVRPT*, and generates their reports first to the catalog *WORK.GRWEB_GR* and from there to the directory *end-of-path-1*, as reports in *Report Group A*. Then, the code runs supplied report definitions *srd5* and *srd6* from the folder *PGMLIB.ITSVRPT*, and generates their reports first to the catalog *WORK.GRWEB_GR* and from there to the directory *end-of-path-1*, as reports in *Report Group B*.

For more information about the %CPWEBINI macro, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation.

For more information about the %CPRUNRPT macro, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation. For additional information about %CPRUNRPT, see the topic %CPHTREE in the SAS IT Resource Management macro reference documentation. Focus especially on Example 7. In Example 7, for now you can ignore section 3, section 4's GLBWHERE= parameter, and section 5.

- For UNIX or Windows: For each report group, specify that the destination of the reports is the appropriate one of the end-of-path directories that you created in your directory-based report structure.

  Note: This program corresponds to the *xreport.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program and see if there are any other comments or features that you want to use in your program. △

- For z/OS with a directory-based report structure: For each report group, specify that the destination of the reports is the appropriate one of the end-of-path directories that you created in your z/OS UNIX File System based report structure.

  Note: This program corresponds to the *xreport.sas* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): For each report group, specify that the destination of the reports is the appropriate one of the end-of-path PDSs that you created in your PDS-based report structure.

  Note: This program corresponds to the *xREPORT* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

2 For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): You do not need to create another FTP program to FTP reports from the PDS-based report structure to the directory-based report structure.

You can use the same program that you created to FTP the report structure.

3 Run the program(s).
   - For UNIX or Windows: Run the program that generates the reports (and some dynamic files) to the report structure.
   - For z/OS with a directory-based report structure: Run the program that generates the reports (and some dynamic files) to the z/OS UNIX File System based report structure.
   - For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the program that generates the reports (and some dynamic members) to the PDS-based report structure.

     Then run the FTP program that you created earlier. The %CMFTPSND macro reuses directories and files that already exist and creates the directories and files that did not already exist.

4 Check the reports.

   Point your Web browser to the welcome.htm file in the "super" directory. Check that the report groups are at the end-of-path locations that you intended, and that the reports are in the report groups that you intended.

   Graphic reports display as "thumbnail" sketches. To view a full-size version of a report, click on its thumbnail sketch. To return the report to thumbnail size, select **`Shrink`**.

5 Set permissions so that read-only access to the directory-based report structure (or execute access, if appropriate to the report mechanisms) is available to you and to anyone else who needs to view the reports.

   For more information about access rights, see "Actions and Access Rights Required" on page 44.

The Job Schedule

Schedule the nightly jobs. Typically the jobs run once a day, at night. However, you can schedule them to run more or less often than once a day, and at whatever time(s) works best for your site.

- For UNIX: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to the *cron* and *crontab* manpage in your UNIX documentation.
- For Windows: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to the documentation for your task scheduler.
- For z/OS with a directory-based report structure: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Schedule the process-and-reduce program, the report-generation program, and the FTP program to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.

Your Notes

Make notes in your own schedule.

- The first day after scheduling, confirm that the programs ran as scheduled.
- The first day after scheduling, confirm that the backup of the PDB was created and that it is identical to (or can be used to regenerate a structure that is identical to) the PDB.

  Unless you have a strong preference for leaving the backup step after the process and reduce steps, in the process, reduce, and backup job move the backup step to precede the process and reduce steps.
- Every day, check the logs from the nightly jobs for error messages, if any. You might also want to check for warning messages.

  If you need to restart a nightly job,

  - For UNIX or Windows: See the *restart.txt* file that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
  - For z/OS: See the *$RESTART* member that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
- In a step above, you updated the site library's list of holidays so that it is correct for your site through some date in the future. When that date is reached, SAS IT Resource Management will write a reminder to the log during the process-and-reduce run. But in case you miss seeing that reminder, you might want to add an item to your own schedule to update the list again before that last date.

  For more information about the holiday list, see "Viewing/Editing Your Active Site Library's List of Holidays" on page 286.
- In a step above, you set the workshift schedule in the PDB. If your site has seasonal shift changes (for example, summer hours), you might want to add an item to your own schedule to update the workshift schedule when the season changes.

  For more information about the workshift schedule, see "Viewing/Editing Your Active PDB's Work Shift Schedule" on page 334.
- You might want to schedule a periodic check of table and variable usage. If data is being kept for any variables or tables that are not being used for reporting, you might want to set the variables' or tables' Kept status to *No* so that you can save space in the PDB and time during processing and reduction.
- You might want to schedule a periodic check of space usage.

  For more information about space usage, see "Space Usage at a Level in a Table" on page 349.
- If you have any formats that are based on changeable information (for example, formats related to network configuration or storage devices), you might want to schedule a periodic review of the formats.

  For more information about formats, see SAS online help.
- At some sites, copies of the demonstration PDBs are used for practice or testing: for example, testing what a new report would look like, before a production PDB has enough data to generate a report.

  At some sites, a demonstration PDB is used when the active PDB must be some other PDB than the one that is to be used. (Some operations cannot run on the active PDB.)

  At other sites, one or more of the demonstration PDBs are never used.

  You can delete one or more of the demonstration PDBs if you have no need for them. Remember, though, that at least one PDB (demonstration PDB or

site-created PDB) is required to start SAS IT Resource Management. For more information, see "Deleting a PDB" on page 315.

Optional Customization

After enough data has been collected that most of the reports are fully populated (that is, they have data over the time span of the report), you might or might not want to make other customizations for your site.

- You might want to adjust the report definitions and rule definitions.

  For example, you might want to add, change, or delete definitions.

  For more information about report definitions, see "Using the Manage Report Definitions Tool" on page 468. For more information about rule definitions, see "Using the Perform Exception Analysis Tool" on page 468.

- You might want to adjust static files in the report structure(s).

  A report gallery's static files are the files that are *not* cleared when the reports in the gallery are cleared by %CPWEBINI, %CPMANRPT, or the WEBCLR= parameter.

  For information about customizing these files, see "Customizing a Report Gallery's Static Files" on page 562.

- You might want to age out old reports instead of deleting old reports.

  Where you want to age out old reports, replace the calls to the %CPWEBINI macro with calls to the %CPMANRPT macro, and make the corresponding changes to the OUTLOC= parameter values in the macro calls that generate reports.

  For more information about %CPWEBINI, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation. For more information about %CPMANRPT, see the topic %CPMANRPT in the SAS IT Resource Management macro reference documentation.

- You might want to customize the report structure.

  Typically, the report structure is not customized. However,

  - you can add new paths in the report structure. If you add new paths, be very careful to specify end-of-path locations that either do not currently exist or, if they exist, are empty and not used for any other purpose.
  - where you use %CPWEBINI to clear reports, you can remove or revise the existing paths in the report structure.
  - where you use %CPMANRPT to age out reports, you can adjust a few aspects of the report structure: change the titles, and change the text (but not directory names) that is displayed on existing paths. You cannot remove or revise existing paths in the report structure (unless you are willing to risk losing all the reports on those paths).

  To customize the report structure:

  - For UNIX or Windows: Edit the program that you used to create the report structure.

    Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the revised structure-creation program to SAS.

    Finally, revise the report-generation program to generate reports to the appropriate destinations in the report structure, as necessary.

- For z/OS with a directory-based report structure: Edit the job that you used to create the report structure based on the z/OS UNIX File System.

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the revised structure-creation job.

  Finally, revise the report-generation program to generate reports to the appropriate destinations in the revised report structure, as necessary.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Edit the job that you used to create the PDS-based report structure. Also, edit the FTP job, as necessary.

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structures if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structures. Submit the revised structure-creation job. Then, run the revised FTP job.

  Finally, revise the report-generation program to generate reports to the appropriate destinations in the PDS-based report structure, as necessary. (The nightly FTP job will make the corresponding changes to the directory-based revised report structure.)

Note: To edit an entry in a SAS catalog, you can use the SAS Program Editor window. For example, suppose you want to copy the entry in ADMIN.*catalog-name.entry-name.entry-type*, edit it, and then save it to ADMIN.*catalog-name.entry-name.entry-type*, which is in the ADMIN library in your active PDB. Here are the steps:

1. Open the SAS Program Editor window.
2. In the command field, type the following command and press ENTER:

   ```
   copy ADMIN.catalog-name.entry-name.entry-type
   ```

3. The entry appears.
4. Save the original version under a different name, in case you want to refer to it later.

   You can save the entry to any library that exists and has a libref that is defined with write access. (If the library exists and a libref is not currently defined or is defined with read-only access, you can define a libref by using the SAS LIBNAME statement.)

   In order to write to ADMIN, if the PDB is activated with readonly access, then activate it with write access.

   In the command field, type a command like the following one and press ENTER:

   ```
   save ADMIN.catalog-name.another-entry-name.entry-type
   ```

   where *another-entry-name* is an existing entry or an entry that is to be created by this command.
5. Edit the entry.
6. Save the edited entry.

   In the command field, type the following command and press ENTER:

   ```
   save ADMIN.catalog-name.entry-name.entry-type
   ```

   If the PDB was activated initially with read-only access, activate it again with read-only access.

7 Close or iconize the SAS Program Editor window.

△

For more information about grouping reports in the tree, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide. For more information about the %CPHTREE macro, see the topic %CPHTREE in the SAS IT Resource Management macro reference documentation. For more information about the %CMFTPSND macro, see the topic %CMFTPSND in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

- You might want to adjust the table definitions and the variable definitions.

  For more information about tables, see "Overview of Tables" on page 353. For more information about variables, see "Overview of Variables" on page 378.

- You might want to adjust the PDB properties and site library properties.

  For more information about PDBs, see "Overview of Whole PDBs" on page 296. For more information about site libraries, see "Overview of Site Libraries" on page 270.

- You might want to review how checking for duplicate data is handled.

  For more information about checking for duplicate data, see "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation.

- If you want to change some aspect of the process step and you do not see any supplied way of making that change, you might want to read about process exits.

  For more information about process exits, see "Overview of Using Process Exits" on page 442.

- You might want to combine or split PDBs.

  For information about combining PDBs, see "Combining PDBs" on page 304. For information about splitting PDBs, see "Splitting One PDB into Two PDBs" on page 320.

- If your users will use the SAS IT Resource Management client GUI to report on data in a PDB on a SAS IT Resource Management server host, make a list of the information that the users will need in order to create a remote server profile for that host.

  For more information, see "Creating a Remote Server Profile" on page 574.

Additional Data and Cleanup

If you have data in a MIB, class, model type, or schema file that is not covered by setup case 2, see "Overview of Setup Case 5" on page 119. Otherwise, delete the PDB and related structures that the QuickStart Wizard created.

- For the QuickStart Wizard PDB located in a UNIX or Windows environment: Delete the QuickStart Wizard's PDB, including the *qs* directory (and its subdirectories) that is related to the QuickStart Wizard's PDB.
- For the QuickStart Wizard PDB located in a z/OS environment: Delete the QuickStart Wizard's PDB, including any PDSs with *.QS* as a high-level qualifier and any directories with */qs* in the directory path that are related to the QuickStart Wizard's PDB.

CHAPTER 6

Setup Case 3

Setup Case 3

Overview of Setup Case 3

For general information about setting up the server, see "Setting Up the Server, with Setup Cases" on page 38. That topic also identifies which setup case applies to your situation.

Following are the instructions for setup case 3. For each step, there is an overview of the step and, after that, the specific directions for the step. Typically, UNIX and Windows have one or two sets of directions and z/OS has one or two sets of directions. The steps are in the following order.

1. "The QuickStart Wizard" on page 82
2. "The Raw Data" on page 83
3. "The PDB and Site Library" on page 83
4. "The Tables and Variables" on page 84
5. "The Process-and-Reduce Program" on page 87
6. "The Report Structure" on page 89
7. "The Report Program(s)" on page 92
8. "The Job Schedule" on page 94
9. "Your Notes" on page 94
10. "Optional Customization" on page 95

11 "Cleanup" on page 98

12 "Optional Package" on page 98.

The QuickStart Wizard

Run the QuickStart Wizard and follow its path for NTSMF data.

The QuickStart Wizard does not have a path for your data collector or data source. However, the QuickStart Wizard generates objects that contain, among other things, documentation. In this setup case, you run the QuickStart Wizard *only* to acquire that documentation, which will be useful when you work with your own data collector or data source.

The wizard creates a PDB, creates a program that processes and reduces the data, creates a report structure (and saves the program with which it created the structure, and also creates (if necessary) a program for FTPing the structure and its reports), and creates a program that reports on the data. In later steps in this setup case, you will generate similar (but not identical) structures and programs for your own data collector or data source.

- In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **Administration** tab, select **QuickStart Wizard**. Follow the NTSMF path through the wizard. As the wizard asks for names and locations, invent names and locations that are appropriate for the path's data collector or data source, not your data collector or data source. (For example, when you are asked to supply the location of the file that contains the raw data for the path's data collector or data source, type a location that would be legitimate on your operating system. The file does not need to exist.)

  The path creates a PDB (without any tables at this point) and a report structure.

  Also, the path creates a *qs* directory and, under that, creates a *cntl* subdirectory that has a *readme.txt* file, an *xrptstr.sas* file, an *xprocess.sas* file, and an *xreport.sas* file, where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other directories and files.

- In the SAS IT Resource Management server GUI for z/OS: In the main menu, select the **PDB Admin** button and, from the menu it opens, select **QuickStart Wizard**. Follow the NTSMF path through the wizard. As the wizard asks for names and locations, with one exception invent names and locations that are appropriate for the path's data collector or data source, not your data collector or data source. (For example, when you are asked to supply the location of the data set that contains the raw data for the path's data collector or data source, type a location that would be legitimate on your operating system. The data set does not need to exist.) The exception is that, at one point in the path, the wizard asks where you want the parent location for the Web reports to be. Answer in the way that you want to set up for *your* data collector or data source.

  The path creates a PDB (without any tables at this point).

  The path also creates a report structure. The location of the report structure depends on how you specified the parent location for the Web reports.

  - If you specified a directory in the z/OS UNIX File System as the parent location, the report structure consists of directories in the z/OS UNIX File System. This choice is called *z/OS with a directory-based report structure*.
  - If you specified a PDS as the parent location, the report structure consists of PDSs. This choice is called *z/OS with a PDS-based report structure*. In order to be able to view the reports with a Web browser, you will also create a corresponding directory-based report structure on a UNIX or Windows

system. After you do that, this choice is called *z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system)*.

Also, the path creates a *.QS* PDS and creates a *.QS.CNTL* PDS that has a *$$README* member, a *$xINSTR* member, an *xRPTSTR* member, an *xPROCESS* member, an *xREPORT* member, and an *xFTPHTML* member, where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other members, PDSs, and data sets. For example, the path can create other PDSs with *.QS* as a high-level qualifier and/or directories with */qs* in the directory path.

Note: All new PDSs are created in a location that is based on the SAS system option FILEDEV. Check with the person who installed SAS at your site to determine whether FILEDEV points to a set of temporary volumes. If FILEDEV does point to a set of temporary volumes, then you must move the PDSs to a permanent location. △

The Raw Data

Start to collect raw data, if you do not already collect raw data.

Several steps ahead, you will need at least one day's worth (preferably a few days' worth) of raw data from the data collector or data source. By starting to collect data now, you will not need to wait for it later.

For more information, see the documentation from the vendor of the data collector or data source.

The PDB and Site Library

Create a PDB for your data collector or data source. The site library was created at the time that SAS IT Resource Management was installed. Customize the PDB and site library, as necessary.

1 Create a PDB.

- In the SAS IT Resource Management GUI for UNIX or Windows: On the main window's **`Administration`** tab, select **`Create PDB Wizard`**. Follow the path through this wizard, but ignore table selection. When the wizard asks if you want the new PDB to become the active PDB, answer *Yes*.

  The result is a PDB without table definitions. Also, the new PDB becomes the active PDB.

- In z/OS: You can use interactive mode, or you can use batch mode.

  - Using interactive mode: In the SAS IT Resource Management server GUI for z/OS, from the main menu select **PDB Admin ▶ Define New PDB**

    Follow the path through this wizard, including allocation of space, but ignore table selection. (For good values to use for space allocation, see the *ALLOC* job step in the *CMQSTART* member in the SAS IT Resource Management *CPMISC* PDS.)

    The result is a PDB without table definitions.

  - Using batch mode: In batch mode, make a copy of the *CMQSTART* member that is in the SAS IT Resource Management *CPMISC* PDS. In the copy of CMQSTART, delete the call to the %CPSTART macro and replace it with a copy of the call to %CPSTART that is in the

CMAUTOEX member in *CPMISC* and, in its PDB= parameter, specify the PDB that you created for your data collector or data source. In the copy of CMQSTART, also delete everything after the *ALLOC* job step. Follow the instructions in the comments that apply to the job steps that remain. Submit the copy as a batch job. The result is a PDB without table definitions.

For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

For more information about PDBs, see "Overview of Whole PDBs" on page 296.

2 Customize the PDB and site library, as necessary.

- For UNIX or Windows: In the *readme.txt* file that the QuickStart Wizard generated, follow the instructions for the PDB and site library.
- For z/OS: In the *$$README* member and *$xINSTR* member that the QuickStart Wizard generated, follow the instructions for the PDB and site library.

For more information about PDB properties, see "Overview of Whole PDBs" on page 296. For more information about site library properties, see "Overview of Site Libraries" on page 270.

The Tables and Variables

In the PDB for your data collector or data source, create a table for the data from the character-delimited file, and check the table and variable definitions.

1 Create a program.

This SAS program calls (in this order) the %CPSTART macro (to invoke SAS IT Resource Management), the %CPCAT macro (to read a control statement), the %CPCAT macro again (to store the control statement temporarily), and the %CPDDUTL macro (to run the stored control statement).

- For the call to the %CPSTART macro, specify the following code:
  - For UNIX or Windows: Copy the call to the %CPSTART macro from the *xprocess.sas* program that the QuickStart Wizard generated.

    In the PDB= parameter, replace the reference to the wizard's PDB with a reference to the PDB that you created for your data collector or data source.
  - For z/OS: Copy the call to the %CPSTART macro from the *xPROCESS* member that the QuickStart Wizard generated.

    In the PDB= parameter, replace the reference to the wizard's PDB with a reference to the PDB that you created for your data collector or data source.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation.
- For the first call to the %CPCAT macro, specify the following code:

```
%cpcat;
cards4;
generate source
   infile='rawdata-location'
```

```
    intype=cdc
    delim='delim-chars'
    filename='output-location'
    name=utable-name
    type=table-type ;
;;;;
```

where you specify appropriate values for the INFILE=, DELIM=, FILENAME=, NAME=, and TYPE= parameters, and the ';;;;' must start at the beginning of the line.

For more information about the GENERATE SOURCE control statement, see the topic GENERATE SOURCE in the SAS IT Resource Management macro reference documentation. For more information about the %CPCAT macro, see the topic %CPCAT in the SAS IT Resource Management macro reference documentation.

- For the second call to the %CPCAT macro, specify the following code:

```
%cpcat(cat=work.temp.temp.source) ;
```

- For the call to the %CPDDUTL macro, specify the following code:

```
%cpddutl(entrynam=work.temp.temp.source) ;
```

For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation.

For an example of this program, see "Appendix 13: Character-Delimited Support in GENERATE SOURCE" in "Using the Generic Collector Facility" on page 158.

2 Run the program.

Submit the program to SAS. Based on the information in the header records in the character-delimited file, the program will write to the output file a block of control statements to create the definitions for the corresponding table and its variables.

For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

3 In the location that was specified by the FILENAME= parameter in the GENERATE SOURCE control statement, review the table definition and its variable definitions.

Then, in a text editor, modify the original statements, as necessary, and add any CREATE DERIVED or CREATE FORMULA control statements, as necessary. In particular, here are some changes to consider making:

- In the CREATE TABLE control statement, you might want to change the table's label and/or description. Also review the other parameters.

  For more information about the CREATE TABLE control statement, see the topic CREATE TABLE in the SAS IT Resource Management macro reference documentation.

- In the CREATE VARIABLE control statements, you might want to change the variables' names, labels, and/or descriptions. Also review the other parameters.

  For more information about the CREATE VARIABLE control statement, see the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation.

- After the CREATE VARIABLE control statements, add a CREATE DERIVED control statement for each derived variable that you want to have in the

table, if any. The CREATE DERIVED control statements can be in any order, but alphabetical order is convenient.

For more information about the CREATE DERIVED control statement, see the topic CREATE DERIVED in the SAS IT Resource Management macro reference documentation.

- In the UPDATE TABLE control statement, you might want to revise the table's age limits. Also review the other parameters.

  Note: If you are considering revisions to the BY and CLASS lists for this data collector or data source, see "Restrictions on Modifications of BY Variables Lists" on page 348. △

  For more information about the UPDATE TABLE control statement, see the topic UPDATE TABLE in the SAS IT Resource Management macro reference documentation.

- After the UPDATE TABLE control statement, add a CREATE FORMULA control statement for each formula variable that you want to associate with the table. The CREATE FORMULA control statements can be in any order, but alphabetical order is convenient.

  For more information about the CREATE FORMULA control statement, see the topic CREATE FORMULA in the SAS IT Resource Management macro reference documentation.

Note: If a derived variable depends on another derived variable and/or if a formula variable depends on another formula variable, be sure to read the information about dependencies in the documentation for the CREATE DERIVED and/or CREATE FORMULA control statements. △

4 Create another program.

This SAS program calls (in this order) the %CPSTART macro (to invoke SAS IT Resource Management), and the %CPDDUTL macro (to run the control statements that define the table and its variables).

- For the call to the %CPSTART macro, copy the call that you used earlier in this step.
- For the call to the %CPDDUTL macro, specify the following code:

```
%cpddutl(filename='location-of-control-statements') ;
```

where the value of the FILENAME= parameter is the name of the file that contains the table and variable control statements that you reviewed (and perhaps edited).

For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation.

5 Run the program.

Submit the program to SAS.

For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

6 The table and its variables now exist in the active PDB.

If you want to browse them, you can. For more information about viewing a table definition, see "Viewing/Editing a Table" on page 364. For more information about viewing the definition of a regular variable, see "Creating/Editing/Viewing a Regular Variable" on page 394. For more information about viewing the definition of a derived variable, see "Creating/Editing/Viewing a Derived Variable" on page

381. For more information about viewing the definition of a formula variable, see "Creating/Editing/Viewing a Formula Variable" on page 388.

The Process-and-Reduce Program

Create a program to process and reduce data, and test the program.

1 Create a program.

This SAS program calls the %CPSTART macro (to invoke SAS IT Resource Management), sets macro variables as appropriate, calls the %CPPROCES macro (to process the data), and calls the %CPREDUCE macro (to reduce the data).

- For UNIX or Windows: Use a copy of the %CPSTART call that you used above.

  For the appropriate macro variables and how to set them, see "Appendix 9: Using Character-Delimited Data" in "Using the Generic Collector Facility" on page 158.

  For the call to the %CPPROCES macro, use the following code:

```
%cpproces(RAWDATA=rawdata-location,
   COLLECTR=GENERIC,
   TOOLNM=CHARDELIM,
   DELIM='delim_chars' );
```

  where the values of the RAWDATA= and DELIM= parameters must be appropriate for your data collector or data source. (With these settings, the %CPPROCES macro can find supplied staging code for staging data from a character-delimited file, and will run the staging code before continuing with the rest of the process step.)

  For the call to the %CPREDUCE macro, use the following code:

```
%cpreduce ;
```

- For z/OS: Make a copy of the *CMQSTART* member that is in the SAS IT Resource Management *CPMISC* PDS.

  In this copy of CMQSTART, delete the call to the %CPSTART macro and replace it with a copy of the call to the %CPSTART macro that is in the CMAUTOEX member in *CPMISC*. In this copy of CMQSTART, delete the code after the call to %CPSTART and replace that other code with a call to the %CPPROCES macro and a call to the %CPREDUCE macro.

  For the appropriate macro variables and how to set them, see "Appendix 9: Using Character-Delimited Data" in "Using the Generic Collector Facility" on page 158.

  For the call to the %CPPROCES macro, use the following code:

```
%cpproces( RAWDATA=rawdata-location,
   COLLECTR=GENERIC,
   TOOLNM=CHARDELIM
   DELIM='delim_chars' );
```

  where the values of the RAWDATA= and DELIM= parameters must be appropriate for your data collector or data source. (With these settings, the %CPPROCES macro can find supplied staging code for staging data from the character-delimited file, and will run the staging code before continuing with the rest of the process step.)

  For the call to the %CPREDUCE macro, use the following code:

```
%cpreduce ;
```

For an example of this program, see “Generic Collector Appendix 2: Using Character-Delimited Data” in “Setting Up the Server, with Setup Cases” on page 38.

For more information about processing data, see “Processing Data” on page 413. For more information about reducing data, see “Reducing Data” on page 416. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPREDUCE macro, see the topic %CPREDUCE in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see “Working with the Interface for Batch Mode” on page 18.

2 To the program, add a step that backs up the PDB.

In the long run, it does not matter much whether the backup step runs before or after the process and reduce steps. For now, however, put the backup step after the process and reduce steps, so that there will be some data in the PDB by the time that the PDB is backed up.

For information, recommendations, and examples about PDB backup, see “Backing Up a PDB” on page 303.

Note: Backing up the whole PDB is not the same as archiving incoming data. You need to back up the whole PDB regardless of the decision that you made about archiving. For more information about archiving, see “Archiving Incoming Data” on page 411. △

3 When you have at least a day’s worth of raw data from your data collector or data source, run the program as a batch job.

For more information about submitting a SAS program, see “Working with the Interface for Batch Mode” on page 18.

4 The data in the new table corresponds to the raw data.

If you want to browse the data in the PDB, see “Viewing Data Values” on page 420.

5 The PDB backup was created and is identical to (or can be used to regenerate a structure that is identical to) the PDB.

6 Unless you have a strong preference for leaving the backup step after the process and reduce steps, move the backup step to precede the process and reduce steps.

7 Create the program that runs nightly to process and reduce the data.

This SAS program (in this order) calls the backup step (to back up the PDB), calls the %CPSTART macro (to invoke SAS IT Resource Management), sets the appropriate macro variables, calls the %CPPROCES macro (to process data), and calls the %CPREDUCE macro (to reduce data).

For more information about the backup step, see “Backing Up a PDB” on page 303. For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the macro variables, see “Appendix 9: Using Character-Delimited Data” in “Using the Generic Collector Facility” on page 158. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPREDUCE macro, see the topic %CPREDUCE in the SAS IT Resource Management macro reference documentation.

- For UNIX or Windows: Above, you created a file that backed up the PDB and processed and reduced data. You can use this program as the nightly process-and-reduce program.

You might want to rename the file to indicate its data collector or data source and its role (to process and reduce the data). For example, if the file is for xxxx data, you might rename the file *xxxxprrd.sas*.

Note: This file corresponds to the *xprocess.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program and see if there are any other comments or features that you want to use in your program. △

□ For z/OS: Above, you created a job that backed up the PDB and processed and reduced data. You can use this job as the nightly process-and-reduce program.

You might want to rename the job to indicate its data collector or data source and its role (to process and reduce the data). For example, if the job is for xxxx data, you might rename the copy *xxxxPRRD*.

Note: This job corresponds to the *xPROCESS* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

The Report Structure

Create a report structure. For z/OS with a PDS-based report structure, also FTP the report structure to another system. Check the report structure(s).

1 Create, run, and save report definitions for your data collector or data source.

□ In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **`Reporting`** tab select **Manage Report Definitions ▶ File ▶ New Report Definition**

Using the right side of the window, create a report definition for the data in the table that you created above. Select **`Run`** to generate a report that is based on the report definition. In the report that is generated, the data corresponds to the data in the PDB. (For more information about creating and running a report definition, see "Using the Manage Report Definitions Tool" on page 468. If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.)

Next, select **File ▶ Save Report Definition** and complete that path to save the report definition in its interactive form.

Optionally, create and save other report definitions for the table that you created above.

□ In the SAS IT Resource Management server GUI for z/OS: From the main menu, select **Report ▶ DesignGraphRpt**

From the menu that opens, select the type of report that you want to create (at this point, select only from the six types at the top of the list). The type's main report definition window opens. Select the level that you want to use, select the icon in the upper-left corner to select the table that you want to use, and specify all other parameters that you want to use. Then select **Actions ▶ Run Report** In the report that is generated, the data corresponds to the data in the PDB. (For more information about creating and running a report definition, in the type's main report definition window, select **`Help`**. If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.)

Next, in the main window for the report definition select **File ▶ Save Definition** and complete the path to save the report definition in its interactive form. Then select **File ▶ Quit** to return to the list of custom report definitions.

Optionally, create and save other report definitions for the data in the created table.

For information about report definitions in the SAS IT Resource Management client GUI, see "Using the Manage Report Definitions Tool" on page 468.

Note: In a later step, you will need the names of the supplied report definitions that you want to use. (You can use some or all of the supplied report definitions for your data collector or data source.) Notice that the names are available in two places: in the list in the GUI, and in the lower-right corner of each report. △

2 Decide on report groupings for the custom reports for your data collector or data source.

Report groupings usually make a particular report easier to find. Also, report groupings usually make a particular report easier to interpret, by making it easy to compare it with related reports in the same group.

For more information about report groupings, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide.

3 Create a SAS program.

The program creates a report structure that implements the paths and end-of-path directories that you decided on for your report groupings. The program should call the %CPSTART macro (to invoke SAS IT Resource Management), the %CPCAT macro (to read the description of the report structure), the %CPCAT macro again (to store a description of the report structure temporarily), and the %CPHTREE macro (to read the stored description and create the structure).

For more information about the %CPCAT macro, see the topic %CPCAT in the SAS IT Resource Management macro reference documentation.

For more information about the %CPHTREE macro, see the topic %CPHTREE in the SAS IT Resource Management macro reference documentation. In the %CPHTREE topic, focus especially on the outline and on Example 6. In the outline as described in the documentation of the CAT= parameter, you can ignore the fourth and fifth fields for now. In Example 6, you can ignore sections 4, 5, 6, and 7 for now.

- For UNIX: In the outline, specify directory locations in the local UNIX file system. The directories do not need to exist. In a later step, you will run the program and it will create the directories.

  Note: This file corresponds to the *xrptstr.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program to see if there are any other comments or features that you want to use in your program. △

- For Windows: In the outline, specify directory locations on a disk that is attached to your PC or on a disk that is available to your PC through a service such as Network Neighborhood, NFS, AFS, or DFS. The directories do not need to exist. In a later step, you will run the program and it will create the directories.

  Note: This file corresponds to the *xrptstr.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program to see if there are any other comments or features that you want to use in your program. △

- For z/OS with a directory-based report structure: In the outline, specify directory locations in a local z/OS UNIX File System area. The directories do

not need to exist. In a later step, you will run the program and it will create the directories.

Note: This program corresponds to the *xRPTSTR* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job to see if there are any other comments or features that you want to use in your job. △

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): In the outline, specify PDS locations. The PDSs do not need to exist. In a later step, you will run the program and it will create PDSs.

  Note: This program corresponds to the *xRPTSTR* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

You might want to name the program to indicate its data collector or data source and its role (to produce a report structure). For example, if you are working with data from *xxxx*, you might call the program *xxxxrpts.sas* (for UNIX or Windows) or *xxxxRPTS* (for z/OS).

4 For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Create an additional SAS program.

You will use this program to FTP the report structure from its PDS-based location on z/OS to a directory-based location on a UNIX or Windows system. This program should call the %CPSTART macro (to invoke SAS IT Resource Management) and the %CMFTPSND macro (to FTP the report structure to the other system).

For more information about the %CMFTPSND macro, see the %CMFTPSND topic in the SAS IT Resource Management macro reference documentation.

For more information about %CMFTPSND, also see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. In the %CPHTREE topic, focus on Example 6. You can ignore sections 2, 3, 4, 5, and 7 for now. In section 6, you can ignore the NEXNG PDSs for now.

- Create the program. The directories do not need to exist. In the next step, you will run the program and it will create the directories.

  Note: This program corresponds to the *xFTPHTML* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job to see if there are any other comments or features that you want to use in your job. △

You might want to name the program to indicate its data collector or data source and its role (to FTP the report structure). For example, if you are working with data from *xxxx*, you might call the program *xxxxFTP*.

5 Run the SAS program(s).

- For UNIX or Windows: Run the program that creates the directory-based report structure.
- For z/OS with a directory-based report structure: Run the program that creates the directory-based report structure.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the program that creates the PDS-based report structure. Then run the program that FTPs the report structure to a UNIX or Windows system.

For more information about submitting a SAS program, see “Working with the Interface for Batch Mode” on page 18.

6 Check the report structure(s).

Notice that the report structure is, physically, relatively flat.

- For UNIX or Windows: Check that the directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)

- For z/OS with a directory-based report structure: Check that the directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no reports are visible in the right frame.)

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Check that the PDSs have been created in the location and with the names that you specified. Also check that the corresponding directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)

The Report Program(s)

Create the program that runs nightly to generate reports to the report structure, and test the program.

1 Create the program.

This SAS program should contain a call to the %CPSTART macro (to invoke SAS IT Resource Management). Also, for each end-of-path location in the report structure (that is, for each location that contains report groups), the program should contain one call to the %CPWEBINI macro (to clear old reports) and one or more calls to the %CPRUNRPT macro (to generate the reports for the one or more report groups at that location). Here is an example:

```
%CPWEBINI (cat=work.grweb_gr,
   dir=end-of-path-1) ;
%CPRUNRPT (srd1 srd2 srd3 srd4,
   folder=pgmlib.itsvrpt,
   outloc=work.grweb_gr,
   outmode=web,
   outdesc=Report Group A,
   htmldir=end-of-path-1,
   webstyle=gallery2) ;
%CPRUNRPT (srd5 srd6,
   folder=pgmlib.itsvrpt,
   outloc=work.grweb_gr,
   outmode=web,
   outdesc=Report Group B,
   htmldir=end-of-path-1,
   webstyle=gallery2) ;
```

This code clears reports from the catalog *WORK.GRWEB_GR* and its corresponding directory *end-of-path-1*. Next, the code runs supplied report definitions *SRD1*, *SRD2*, *SRD3*, and *SRD4* from the folder *PGMLIB.ITSVRPT*, and generates their reports first to the catalog *WORK.GRWEB_GR* and from there to the directory *end-of-path-1*, as reports for *Report Group A*. Then, the code runs supplied report definitions *SRD5* and *SRD6* from the folder *PGMLIB.ITSVRPT*, and generates their reports first to the catalog *WORK.GRWEB_GR* and from there to the directory *end-of-path-1*, as reports for *Report Group B*.

For more information about the %CPWEBINI macro, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation.

For more information about the %CPRUNRPT macro, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation. For additional information about %CPRUNRPT, also see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. In the %CPHTREE topic, focus especially on Example 7. In Example 7, for now you can ignore section 3, section 4's GLBWHERE= parameter, and section 5.

- For UNIX or Windows: For each report group, specify that the destination of the reports is the appropriate one of the end-of-path directories that you created in your directory-based report structure.

  Note: This program corresponds to the file *xreport.sas* that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program and see if there are any other comments or features that you want to use in your program. △

- For z/OS with a directory-based report structure: For each report group, specify that the destination of the reports is the appropriate one of the end-of-path directories that you created in your z/OS UNIX File System based report structure.

  Note: This program corresponds to the member *xREPORT* that the QuickStart Wizard created for the other data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): For each report group, specify that the destination of the reports is the appropriate one of the end-of-path PDSs that you created in your PDS-based report structure.

  Note: This program corresponds to the member *xREPORT* that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

2 For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): You do not need to create another FTP program to FTP reports from the PDS-based report structure to the directory-based report structure. You can use the same program that you created to FTP the report structure.

3 Run the program(s).

- For UNIX or Windows: Run the program that generates the reports (and some dynamic files) to the report structure.
- For z/OS with a directory-based report structure: Run the program that generates the reports (and some dynamic files) to the report structure that is based on a z/OS UNIX File System.

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the program that generates the reports (and some dynamic members) to the PDS-based report structure.

  Then run the FTP program that you created earlier. The %CMFTPSND macro reuses directories and files that already exist and creates the directories and files that did not already exist.

4 Check the reports.

  Point your Web browser to the file *welcome.htm* in the "super" directory. Check that the report groups are at the end-of-paths that you intended, and that the reports are in the report groups that you intended.

  Graphic reports display as "thumbnail" sketches. To view a full-size version of a report, click on its thumbnail sketch. To return the report to thumbnail size, select **`Shrink`**.

5 Set permissions so that "readonly" access to the directory-based report structure (or execute access, if appropriate to the report mechanisms) is available to you and to anyone else who needs to view the reports.

  For more information about access rights, see "Actions and Access Rights Required" on page 44.

The Job Schedule

Schedule the nightly jobs. Typically the jobs run once a day, at night. However, you can schedule them to run more or less often than once a day, and at whatever time(s) work best for your site.

- For UNIX: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to your *cron* and *crontab* man pages.

- For Windows: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to the documentation for your task scheduler.

- For z/OS with a directory-based report structure: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Schedule the process-and-reduce program, the report-generation program, and the FTP program to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.

Your Notes

Make notes in your own schedule.

- The first day after scheduling, confirm that the programs ran as scheduled.
- The first day after scheduling, confirm that the backup of the PDB was created and that it is identical to (or can be used to regenerate a structure that is identical to) the PDB.

  Unless you have a strong preference for leaving the backup step after the process and reduce steps, in the process, reduce, and backup job move the backup step to precede the process and reduce steps.

- Every day, check the logs from the nightly jobs for error messages, if any. You might also want to check for warning messages.

  If you need to restart a nightly job:

  - For UNIX or Windows: See the *restart.txt* file that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
  - For z/OS: See the *$RESTART* member that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.

- In a step above, you updated the site library's list of holidays so that it is correct for your site through some date in the future. When that date is reached, SAS IT Resource Management will write a reminder to the log during the process-and-reduce run. But in case you miss seeing that reminder, you might want to add an item to your own schedule to update the list again before that last date.

  For more information about the holiday list, see "Viewing/Editing Your Active Site Library's List of Holidays" on page 286.

- In a step above, you set the workshift schedule in the PDB. If your site has seasonal shift changes (for example, summer hours), you might want to add an item to your own schedule to update the workshift schedule at the seasonal changes.

  For more information about the workshift schedule, see "Viewing/Editing Your Active PDB's Work Shift Schedule" on page 334.

- You might want to schedule a periodic check of table and variable usage. If data is being kept for any variables or tables that are not being used for reporting, you might want to set the variables' or tables' Kept status to *No* so that you can save space in the PDB and time during processing and reduction.

- You might want to schedule a periodic check of space usage.

  For more information about space usage, see "Space Usage at a Level in a Table" on page 349.

- If you have any formats that are based on changeable information (for example, formats that relate to network configuration or storage devices), you might want to schedule a periodic review of the formats.

  For more information about formats, see SAS online help.

- At some sites, copies of the demonstration PDBs are used for practice or testing: for example, testing what a new report would look like, before a production PDB has enough data to generate a report.

  At some sites, a demonstration PDB is used when the active PDB must be some other PDB than the one that is to be used. (Some operations cannot run on the active PDB.)

  At other sites, one or more of the demonstration PDBs are never used.

  You can delete one or more of the demonstration PDBs if you have no need for them. Remember, though, that at least one PDB (demonstration PDB or site-created PDB) is required to start SAS IT Resource Management. For more information, see "Deleting a PDB" on page 315.

Optional Customization

After enough data has been collected that most of the reports are fully populated (that is, they have data over the time span of the report), you might or might not want to make other customizations for your site.

- You might want to adjust the report definitions and rule definitions.

  For example, you might want to add, change, or delete definitions.

  For more information about report definitions, see "Using the Manage Report Definitions Tool" on page 468. For more information about rule definitions, see "Using the Perform Exception Analysis Tool" on page 468.

- You might want to adjust static files in the report structure(s).

  A report gallery's static files are the files that are *not* cleared when the reports in the gallery are cleared by %CPWEBINI, %CPMANRPT, or the WEBCLR= parameter.

  For information about customizing these files, see "Customizing a Report Gallery's Static Files" on page 562.

- You might want to age out old reports instead of deleting old reports.

  Where you want to age out old reports, replace the calls to the %CPWEBINI macro with calls to the %CPMANRPT macro, and make the corresponding changes to the OUTLOC= parameter values in the macro calls that generate reports.

  For more information about %CPWEBINI, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation. For more information about %CPMANRPT, see the topic %CPMANRPT in the SAS IT Resource Management macro reference documentation.

- You might want to modify the report structure.

  Typically, the report structure is not customized. However,

  - you can add new paths in the report structure. If you add new paths, be very careful to specify end-of-path locations that either do not currently exist or, if they exist, are empty and not used for any other purpose.
  - where you use %CPWEBINI to clear reports, you can remove or revise the existing paths in the report structure.
  - where you use %CPMANRPT to age out reports, you can adjust a few aspects of the report structure: change the titles, and change the text (but not directory names) that is displayed on existing paths. You cannot remove or revise existing paths in the report structure (unless you are willing to risk losing all the reports on those paths).

  To customize the report structure:

  - For UNIX or Windows: Edit the program that you used to create the report structure.

    Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the revised structure-creation program to SAS.

    Finally, revise the report-generation program to generate reports to the appropriate destinations in the report structure, as necessary.

  - For z/OS with a directory-based report structure: Edit the job that you used to create the report structure that is based on a z/OS UNIX File System.

    Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structures. Submit the revised structure-creation job.

    Finally, revise the report-generation program to generate reports to the appropriate destinations in the revised report structure, as necessary.

□ For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Edit the job that you used to create the PDS-based report structure. Also, edit the FTP job, as necessary.

Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structures if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the revised structure-creation job. Then, run the revised FTP job.

Finally, revise the report-generation program to generate reports to the appropriate destinations in the PDS-based report structure, as necessary. (The nightly FTP job will make the corresponding changes to the directory-based revised report structure.)

Note: To edit an entry in a SAS catalog, you can use the SAS Program Editor window. For example, suppose that you want to copy the entry in ADMIN.*catalog-name.entry-name.entry-type*, edit it, and then save it to ADMIN.*catalog-name.entry-name.entry-type*, which is in the ADMIN library in your active PDB. Here are the steps:

1. Open the SAS Program Editor window.
2. In the command field, type the following command and press ENTER:

   ```
   copy ADMIN.catalog-name.entry-name.entry-type
   ```

3. The entry appears.
4. Save the original version under a different name, in case you want to refer to it later.

   You can save the entry to any library that exists and has a libref that is defined with write access. (If the library exists and a libref is not currently defined or is defined with read-only access, you can define a libref by using the SAS LIBNAME statement.)

   In order to write to ADMIN, if the PDB is activated with read-only access, then activate it with write access.

   In the command field, type a command like the following one and press ENTER:

   ```
   save ADMIN.catalog-name.another-entry-name.entry-type
   ```

   where *another-entry-name* is an existing entry or an entry that is to be created by this command.
5. Edit the entry.
6. Save the edited entry.

   In the command field, type the following command and press ENTER:

   ```
   save ADMIN.catalog-name.entry-name.entry-type
   ```

   If the PDB was activated initially with read-only access, activate it again with read-only access.
7. Close or iconize the SAS Program Editor window.

△

For more information about grouping reports in the tree, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide. For more information about the %CPHTREE macro, see the %CPHTREE topic in the SAS IT Resource

Management macro reference documentation. For more information about the %CMFTPSND macro, see the topic %CMFTPSND in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

- You might want to adjust the table definitions and the variable definitions.

  For more information about tables, see "Overview of Tables" on page 353. For more information about variables, see "Overview of Variables" on page 378.

- You might want to adjust the PDB properties and site library properties.

  For more information about PDBs, see "Overview of Whole PDBs" on page 296. For more information about site libraries, see "Overview of Site Libraries" on page 270.

- You might want to review how checking for duplicate data is handled.

  For more information about checking for duplicate data, see "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation.

- If you want to change some aspect of the process step and you do not see any supplied way of making that change, you might want to read about process exits.

  For more information about process exits, see "Overview of Using Process Exits" on page 442.

- You might want to combine or split PDBs.

  For information about combining PDBs, see "Combining PDBs" on page 304. For information about splitting PDBs, see "Splitting One PDB into Two PDBs" on page 320.

- If your users will use the SAS IT Resource Management client GUI to report on data in a PDB on a SAS IT Resource Management server host, make a list of the information that the users will need in order to create a remote server profile for that host.

  For more information, see "Creating a Remote Server Profile" on page 574.

Cleanup

Delete the PDB and related structures that the QuickStart Wizard created.

- For the QuickStart Wizard PDB located in a UNIX or Windows environment: Delete the QuickStart Wizard's PDB, including the *qs* directory (and its subdirectories) that is related to the QuickStart Wizard's PDB.
- For the QuickStart Wizard PDB based on z/OS: Delete the QuickStart Wizard's PDB, including any PDSs with *.QS* as a high-level qualifier and any directories with */qs* in the directory path that are related to the QuickStart Wizard's PDB.

Optional Package

If you want to package and install the collector-support that you have created, see "Creating and Installing a Collector Support Package" on page 426 and these macros: the topic %CPPKGCOL in the SAS IT Resource Management macro reference documentation, the topic %CPRPTPKG in the SAS IT Resource Management macro reference documentation, and the topic %CPINSPKG in the SAS IT Resource Management macro reference documentation.

CHAPTER 7

Setup Case 4

Setup Case 4

Overview of Setup Case 4

For general information about setting up the server, see “Setting Up the Server, with Setup Cases” on page 38. That topic also identifies which setup case applies to your situation.

Below are the instructions for setup case 4. For each step, there is an overview of the step and, after that, the specific directions for the step. Typically, UNIX and Windows have one or two sets of directions and z/OS has one or two sets of directions. The steps are in the following order.

1 “The QuickStart Wizard” on page 100
2 “The Raw Data” on page 101
3 “The PDB and Site Library” on page 101
4 “The Tables and Variables” on page 102
5 “Reduce Program” on page 105
6 “The Report Structure” on page 107
7 “The Report Program(s)” on page 111
8 “The Job Schedule” on page 113
9 “Your Notes” on page 113
10 “Optional Customization” on page 114

11 “Cleanup” on page 117

12 “Optional Package” on page 117.

The QuickStart Wizard

Run the QuickStart Wizard and follow its path for NTSMF data.

The QuickStart Wizard does not have a path for your data collector or data source. However, the QuickStart Wizard generates objects that contain, among other things, documentation. In this setup case, you run the QuickStart Wizard *only* to acquire that documentation, which will be useful when you work with your own data collector or data source.

The wizard creates a PDB, creates a program that processes and reduces the data, creates a report structure (and saves the program with which it created the structure, and also creates (if necessary) a program for FTPing the structure and its reports), and creates a program that reports on the data. In later steps in this setup case, you will generate similar (but not identical) structures and programs for your own data collector or data source.

- In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window’s **`Administration`** tab, select **`QuickStart Wizard`**. Follow the NTSMF path through the wizard. As you are asked for names and locations, invent names and locations that are appropriate for the wizard’s data collector or data source, not your data collector or data source. (For example, when you are asked to supply the location of the file that contains the raw data for the path’s data collector or data source, type a location that would be legitimate on your operating system. The file does not need to exist.)

  The path creates a PDB (without any tables at this point) and a report structure.

  Also, the path creates a *qs* directory and, under that, creates a *cntl* subdirectory that has a *readme.txt* file, an *xrptstr.sas* file, an *xprocess.sas* file, and an *xreport.sas* file, where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other directories and files.

- In the SAS IT Resource Management server GUI for z/OS: From the main menu, select the **`PDB Admin`** button and, from the menu it opens, select **`QuickStart Wizard`**. Follow the NTSMF path through the wizard. As you are asked for names and locations, with one exception invent names and locations that are appropriate for the wizard’s data collector or data source, not your data collector or data source. (For example, when you are asked to supply the location of the data set that contains the raw data for the path’s data collector or data source, type a location that would be legitimate on your operating system. The data set does not need to exist.) The exception is that, at one point in the path, the wizard asks where you want to have the parent location for the Web reports. Answer in the way that you want to set up for *your* data collector or data source.

  The path also creates a report structure. The location of the report structure depends on how you specified the parent location for the Web reports.

  - If you specified a directory in the z/OS UNIX File System as the parent location, the report structure consists of directories in the z/OS UNIX File System. This choice is called *z/OS with a directory-based report structure*.
  - If you specified a PDS as the parent location, the report structure consists of PDSs. This choice is called *z/OS with a PDS-based report structure*. In order to be able to view the reports with a Web browser, you will also create a corresponding directory-based report structure on a UNIX or Windows system. After you do that, this choice is called *z/OS with a PDS-based report*

structure (and a corresponding directory-based report structure on a UNIX or Windows system).

The path creates a PDB (without any tables at this point).

Also, the path creates a *.QS* PDS and creates a *.QS.CNTL* PDS that has a *$$README* member, a *$xINSTR* member, an *xRPTSTR* member, an *xPROCESS* member, an *xREPORT* member, and an *xFTPHTML* member, where the value of *x* indicates the data collector or data source for which the programs were generated. Additionally, the path generates other instructions and other members, PDSs, and data sets. For example, the path can create other PDSs with *.QS* as a high-level qualifier and/or directories with */qs* in the directory path.

Note: All new PDSs are created in a location that is based on the SAS system option FILEDEV. Check with the person who installed SAS at your site to determine whether FILEDEV points to a set of temporary volumes. If FILEDEV does point to a set of temporary volumes, then you must move the PDSs to a permanent location. △

The Raw Data

Start to collect raw data, if you do not already collect raw data.

Several steps ahead, you will need at least one day's worth (preferably a few days' worth) of raw data from the data collector or data source. By starting to collect data now, you will not need to wait for it later.

For more information, see the documentation from the vendor of your data collector or data source.

The PDB and Site Library

Create a PDB for your data collector or data source. The site library was created at the time that SAS IT Resource Management was installed. Customize the PDB and site library, as necessary.

1 Create a PDB.

- In the SAS IT Resource Management GUI for UNIX or Windows: On the main window's **`Administration`** tab, select **`Create PDB Wizard`**. Follow the path through this wizard, but ignore table selection. When the wizard asks if you want the new PDB to become the active PDB, answer *Yes*.

  The result is a PDB without table definitions. Also, the new PDB becomes the active PDB.

- In z/OS: You can use interactive mode, or you can use batch mode.
  - Using interactive mode: In the SAS IT Resource Management server GUI for z/OS: From the main menu select **PDB Admin ▶ Define New PDB**

    Follow the path through this wizard, including allocation of space, but ignore table selection. (For good values to use for space allocation, see the *ALLOC* job step in the *CMQSTART* member in the SAS IT Resource Management *CPMISC* PDS.)

    The result is a PDB without table definitions.
  - Using batch mode: In batch mode, make a copy of the *CMQSTART* member that is in the SAS IT Resource Management *CPMISC* PDS. In the copy of CMQSTART, delete the call to the %CPSTART macro and replace it with a copy of the call to %CPSTART that is in the

CMAUTOEX member in *CPMISC* and, in its PDB= parameter, specify the PDB that you created for your data collector or data source. In the copy of CMQSTART, also delete everything after the *ALLOC* job step. Follow the instructions in the comments that apply to the job steps that remain. Submit the copy as a batch job. The result is a PDB without table definitions.

For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

For more information about PDBs, see "Overview of Whole PDBs" on page 296.

2 Customize the PDB and site library, as necessary.

- For UNIX or Windows: In the *readme.txt* file that the QuickStart Wizard generated, follow the instructions for the PDB and site library.
- For z/OS: In the *$$README* member and *$xINSTR* member that the QuickStart Wizard generated, follow the instructions for the PDB and site library.

For more information about PDB properties, see "Overview of Whole PDBs" on page 296. For more information about site library properties, see "Overview of Site Libraries" on page 270.

The Tables and Variables

In the PDB for your data collector or data source, create one table for each SAS data set of incoming data, and check the table and variable definitions.

1 Run the MXG program(s) or your program(s) that reads the raw data and write it to one or more SAS data sets.

Confirm that the SAS data sets are now created.

Note: The amount of raw data does not matter at this point. The following step is based on the data set and column attributes, not on values in the columns. △

2 Create a program.

This SAS program calls (in this order) the %CPSTART macro (to invoke SAS IT Resource Management), the %CPCAT macro (to read a control statement), the %CPCAT macro (to store the control statement temporarily), and the %CPDDUTL macro (to run the stored control statement).

- For the call to the %CPSTART macro, specify the following code:
  - For UNIX or Windows: Copy the call to the %CPSTART macro from the *xprocess.sas* program that the QuickStart Wizard generated.

    In the PDB= parameter, replace the reference to the wizard's PDB with a reference to the PDB that you created for your data collector or data source.
  - For z/OS: Copy the call to the %CPSTART macro from the *xPROCESS* member that the QuickStart Wizard generated.

    In the PDB= parameter, replace the reference to the wizard's PDB with a reference to the PDB that you created for your data collector or data source.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation.

□ For the first call to the %CPCAT macro, specify the following code:

```
%cpcat;
cards4;
generate source
   dataset='libref-1.name-1'
   filename='output-location' REPLACE
   name=utable-name-1
   type=table-type-1 ;
generate source
   dataset='libref-2.name-2'
   filename='output-location' APPEND
   name=utable-name-2
   type=table-type-2 ;
.
.
.
generate source
   dataset='libref-n.name-n'
   filename='output-location' APPEND
   name=utable-name-n
   type=table-type-n ;
;;;;
```

In this code each GENERATE SOURCE control statement corresponds to one of the SAS data sets that is created by a program that reads the raw data, and the ';;;;' must start at the beginning of the line. Within each GENERATE SOURCE control statement, you must specify a consistent set of values for the DATASET=, FILENAME=, NAME=, and TYPE= parameters. That is, for the SAS data set that is specified in the DATASET= parameter, a table definition is created (in the location that is specified by the FILENAME= parameter) for the table that is specified in the NAME= parameter, and that table's type will be as specified in the TYPE= parameter.

Preceding the first call to %CPCAT, put one or more LIBNAME statements that define the libref(s) that is used in the %CPCAT call.

For more information about the GENERATE SOURCE control statement, see the topic GENERATE SOURCE in the SAS IT Resource Management macro reference documentation. For more information about the %CPCAT macro, see the topic %CPCAT in the SAS IT Resource Management macro reference documentation. For more information about the SAS LIBNAME statement, see SAS help.

□ For the second call to the %CPCAT macro, specify the following code:

```
%cpcat(cat=work.temp.temp.source) ;
```

□ For the call to the %CPDDUTL macro, specify the following code:

```
%cpddutl(entrynam=work.temp.temp.source) ;
```

For more information about the %CPDDUTL macro, see the %CPDDUTL macro in the SAS IT Resource Management macro reference documentation.

3 Run the program.

Submit the program to SAS. For each data set and its columns, the program will write to the output file a block of definitions for the corresponding table and its variables.

For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

4 In the location that was specified by the FILENAME= parameter on the GENERATE SOURCE control statements, review the blocks of definitions. Each GENERATE SOURCE control statement generated a block of definitions for a SAS data set. A block of definitions corresponds to a table. In each block there is a CREATE TABLE control statement, CREATE VARIABLE control statements, and an UPDATE TABLE control statement.

In a text editor, modify the control statements in the blocks, as necessary, and add any CREATE DERIVED or CREATE FORMULA control statements to the remaining blocks, as necessary. In any given block, consider the following:

- In the CREATE TABLE control statement, you might want to change the table's name, label, and/or description. Also review the other parameters.

  For more information about the CREATE TABLE control statement, see the topic CREATE TABLE in the SAS IT Resource Management macro reference documentation.
- In the CREATE VARIABLE control statements, you might want to change the variables' names, labels, and/or descriptions. Also review the other parameters.

  For more information about the CREATE VARIABLE control statement, see the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation.
- After the CREATE VARIABLE control statements, add a CREATE DERIVED control statement for each derived variable that you want to have in the table, if any. The CREATE DERIVED control statements can be in any order, but alphabetical order is convenient.

  For more information about the CREATE DERIVED control statement, see the topic CREATE DERIVED in the SAS IT Resource Management macro reference documentation.
- In the UPDATE TABLE control statement, change the table's name (if you changed the name in the CREATE TABLE control statement). You might want to revise the table's age limits. Also review other parameters.

  Note: If you are considering revisions to the table's BY variables list, see "Restrictions on Modifications of BY Variables Lists" on page 348. △

  For more information about the UPDATE TABLE control statement, see the topic UPDATE TABLE in the SAS IT Resource Management macro reference documentation.
- After the UPDATE TABLE control statement, add a CREATE FORMULA control statement for each formula variable that you want to associate with the table. The CREATE FORMULA control statements can be in any order, but alphabetical order is convenient.

  For more information about the CREATE FORMULA control statement, see the topic CREATE FORMULA in the SAS IT Resource Management macro reference documentation.

Note: If a derived variable depends on another derived variable and/or if a formula variable depends on another formula variable, be sure to read the information about dependencies in the documentation for the CREATE DERIVED and/or CREATE FORMULA control statements. △

5 Create another program.

This SAS program calls (in this order) the %CPSTART macro (to invoke SAS IT Resource Management) and the %CPDDUTL macro (to run the control statements that define the tables and their variables).

- For the call to the %CPSTART macro, copy the call that you used earlier in this step.
- For the call to the %CPDDUTL macro, specify the following code:

```
%cpddutl(filename='location-of-control-statements') ;
```

where the value of the FILENAME= parameter is the path and name of the file that contains the table and variable control statements that you reviewed (and perhaps edited).

For more information about the %CPDDUTL macro, see the topic %CPDDUTL in the SAS IT Resource Management macro reference documentation.

6 Run the program.

Submit the program to SAS. The program creates the tables and their variables in the active PDB.

For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

7 The PDB's new table definitions and the tables' variable definitions have been created.

If you want to browse them, you can. For more information about viewing table definitions, see "Viewing/Editing a Table" on page 364. For more information about viewing the definition for a regular variable, see "Creating/Editing/Viewing a Regular Variable" on page 394. For more information about viewing the definition for a derived variable, see "Creating/Editing/Viewing a Derived Variable" on page 381. For more information about viewing the definition for a formula variable, see "Creating/Editing/Viewing a Formula Variable" on page 388.

Reduce Program

Create a program to process and reduce data, and test the program.

1 Create a program.

This SAS program calls the %CPSTART macro (to invoke SAS IT Resource Management), runs the code that creates the SAS data set(s), calls the %CPPROCES or %CMPROCES macro (to process the data), and calls the %CPREDUCE macro (to reduce the data). Use the %CMPROCES macro if an MXG program writes the data to the SAS data set(s). If any other program writes the data to the SAS data set(s), use the %CPPROCES macro.

- For UNIX or Windows: Use a copy of the %CPSTART call that you used above.

The code that creates the SAS data set(s) can run before the call to %CPSTART, but in any case must run before the call to %CPPROCES.

For the call to the %CPPROCES macro, use the following code:

```
%cpproces (GENLIB=libref,
   COLLECTR=GENERIC,
   TOOLNM=SASDS );
```

where the values of the GENLIB=, RAWDATA=, and DELIM= parameters must be appropriate for your data collector or data source.

For the call to the %CPREDUCE macro, use the following code:

```
%cpreduce ;
```

- For z/OS: Make a copy of the *CMQSTART* member that is in the SAS IT Resource Management *CPMISC* PDS.

  In this copy of CMQSTART, delete the call to the %CPSTART macro and replace the call with a copy of the call to the %CPSTART macro that is in the *CMAUTOEX* member in *CPMISC*.

  The code that creates the SAS data set(s) can run before the call to %CPSTART, but in any case must run before the call to %CPPROCES.

  For the call to the %CPPROCES or %CMPROCES macro, use the following code:

```
%cpproces (GENLIB=libref,
   COLLECTR=GENERIC,
   TOOLNM=SASDS ) ;

or

%cmproces (GENLIB=libref,
   COLLECTR=GENERIC,
   TOOLNM=SASDS ) ;
```

  where the values of the GENLIB=, RAWDATA=, and DELIM= parameters must be appropriate for your data collector or data source.

  For the call to the %CPREDUCE macro, use the following code:

```
%cpreduce ;
```

For an example of this program, see “Appendix 6: Fax: Sample Daily Process-and-Reduce Job” on page 185.

For more information about processing data, see “Processing Data” on page 413. For more information about reducing data, see “Reducing Data” on page 416. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CMPROCES macro, see the topic %CMPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPREDUCE macro, see the topic %CPREDUCE in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see “Working with the Interface for Batch Mode” on page 18.

2 To the program, add a step that backs up the PDB.

  In the long term, it does not matter much whether the backup step runs before or after the process and reduce steps. For now, however, put the backup step after the process and reduce steps, so that there will be some data in the PDB by the time that the PDB is backed up.

  For information, recommendations, and examples about PDB backup, see “Backing Up a PDB” on page 303.

  Note: Backing up the whole PDB is not the same as archiving incoming data. You need to back up the whole PDB regardless of your decision about archiving. (For more information about archiving, see “Archiving Incoming Data” on page 411.) △

3 When you have at least one day’s worth of raw data from your data collector or data source, run the program as a batch job.

  For more information about submitting a SAS program, see “Working with the Interface for Batch Mode” on page 18.

4 The data in the new tables corresponds to the data in the SAS data sets.

If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.

5 The PDB backup was created and is identical to (or can be used to regenerate a structure that is identical to) the PDB.

6 Unless you have a strong preference for leaving the backup step after the process and reduce steps, in the program move the backup step to precede the process and reduce steps.

7 Create the program that runs nightly to process and reduce the data.

This SAS program (in this order) calls the backup step (to back up the PDB), the %CPSTART macro (to invoke SAS IT Resource Management), the code that creates the SAS data set(s), and the %CPPROCES or %CMPROCES macro (to process data), and it calls the %CPREDUCE macro (to reduce data).

For more information about the backup step, see "Backing Up a PDB" on page 303. For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CMPROCES macro, see the topic %CMPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPREDUCE macro, see the topic %CPREDUCE in the SAS IT Resource Management macro reference documentation.

□ For UNIX or Windows: Above, you created a file that backed up the PDB and processed and reduced data. You can use this program as the nightly process-and-reduce program.

You might want to rename the file to indicate its data collector or data source and its role (to process and reduce data). For example, if the file is for *xxxx* data, you might rename the file *xxxxprrd.sas*.

Note: This file corresponds to the *xprocess.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program and see if there are any other comments or features that you want to use in your program. △

□ For z/OS: Above, you created a job that backed up the PDB and processed and reduced data. You can use this job as the nightly process-and-reduce program.

You might want to rename the job to indicate its data collector or data source and its role (to process and reduce data). For example, if the job is for *xxxx* data, you might rename the job *xxxxPRRD*.

Note: This job corresponds to the *xPROCESS* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

The Report Structure

Create a report structure. For z/OS with a PDS-based report structure, also FTP the report structure to another system. Check the report structure(s).

1 Create, run, and save report definitions for your data collector or data source.

□ In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **`Reporting`** tab, select **Manage Report Definitions ▶ File ▶ New Report Definition**

Using the right side of the window, create a report definition for the data in the table that you created above. Select **`Run`** to generate a report based on

the report definition. In the report, the data corresponds to the data in the PDB. (For more information about creating and running a report definition, see "Using the Manage Report Definitions Tool" on page 468. If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.)

Next, select **File ▶ Save Report Definition** and complete that path to save the report definition in its interactive form.

Optionally, create and save other report definitions for the data in the table(s) that you created above.

□ In the SAS IT Resource Management GUI for z/OS: From the main menu, select **Reports ▶ DesignGraphRpt**

Select the type of report that you want to create (at this point, select only from the six types at the top of the list). The type's main report definition window opens. Select the level that you want to use, select the icon in the upper-left corner to select the table that you want to use, and specify all other parameters that you want to use. Then, select **Actions ▶ Run Report**

In the report that is generated, the data corresponds to the data in the PDB. (For more information about creating and running a report definition, in the type's main report-definition window, select **`Help`**. If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.)

Next, from the main window for the report definition, select **File ▶ Save Definition** and complete the path to save the report definition in its interactive form. Then select **File ▶ Quit** to return to the list of custom report definitions.

Optionally, create and save other report definitions for the data in the created table.

For more information about report definitions, see "Using the Manage Report Definitions Tool" on page 468.

Note: In a later step, you will need the names of the supplied report definitions that you want to use. (You can use some or all of the supplied report definitions for your data collector or data source.) Notice that the names are available in two places: in the list in the GUI, and in the lower-right corner of each report. △

2 Decide on report groupings for the custom reports for your data collector or data source.

Report groupings usually make a particular report easier to find. Also, report groupings usually make a particular report easier to interpret, by making it easy to compare with related reports in the same group.

For more information about report groupings, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide.

3 Create a SAS program.

The program creates a report structure that implements the paths and end-of-path directories that you decided on for your report groupings. The program should call the %CPSTART macro (to invoke SAS IT Resource Management), the %CPCAT macro (to read a description of the report structure), the %CPCAT macro again (to store the description of the report structure temporarily), and the %CPHTREE macro (to read the stored description and create the structure).

For more information about the %CPCAT macro, see the topic %CPCAT in the SAS IT Resource Management macro reference documentation.

For more information about the %CPHTREE macro, see the topic %CPHTREE in the SAS IT Resource Management macro reference documentation. In the

%CPHTREE topic, focus especially on the outline and on Example 6. In the outline as described in the documentation of the CAT= parameter, you can ignore the fourth and fifth fields for now. In Example 6, you can ignore sections 4, 5, 6, and 7 for now.

- For UNIX: In the outline, specify directory locations in the local UNIX file system. The directories do not need to exist. In a later step, you will run the program and it will create the directories (and will create some static files in the directories).

  Note: This file corresponds to the file *xrptstr.sas* that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program to see if there are any other comments or features that you want to use in your program. △

- For Windows: In the outline, specify directory locations on a disk that is attached to your PC or on a disk that is available to your PC through a service such as Network Neighborhood, NFS, AFS, or DFS. The directories do not need to exist. In a later step, you will run the program and it will create the directories (and will create some static files in the directories).

  Note: This file corresponds to the file *xrptstr.sas* that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program to see if there are any other comments or features that you want to use in your program. △

- For z/OS with a directory-based report structure: In the outline, specify directory locations in a local z/OS UNIX File System area. The directories do not need to exist. In a later step, you will run the program and it will create the directories (and will create some static files in the directories).

  Note: This program corresponds to the member *xRPTSTR* that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job to see if there are any other comments or features that you want to use in your job. △

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): In the outline, specify PDS locations. The PDSs do not need to exist. In a later step, you will run the program and it will create PDSs (and will create some static members in the PDSs).

  Note: This program corresponds to the member *xRPTSTR* that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

You might want to name the program to indicate its data collector or data source and its role (to produce a report structure). For example, if you are working with data from *xxxx*, you might call the program *xxxxrpts.sas* (for UNIX or Windows) or *xxxxRPTS* (for z/OS).

4 For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Create an additional SAS program.

You will use this program to FTP the report structure from its PDS-based location on z/OS to a directory-based location on a UNIX or Windows system. This program should call the %CPSTART macro (to invoke SAS IT Resource Management) and the %CMFTPSND macro (to FTP the report structure to the other system).

For more information about the %CMFTPSND macro, see the %CMFTPSND topic in the SAS IT Resource Management macro reference documentation.

For more information about %CMFTPSND, also see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. In the %CPHTREE topic, focus on Example 6. You can ignore sections 2, 3, 4, 5, and 7 for now. In section 6, you can ignore the NEXNG PDSs for now.

- Create the program. The directories do not need to exist. In the next step, you will run the program and it will create the directories (and will create some static files in the directories).

  Note: This program corresponds to the *xFTPHTML* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job to see if there are any other comments or features that you want to use in your job. △

You might want to name the program to indicate its data collector or data source and its role (to FTP the report structure). For example, if you are working with data from xxxx, you might call the program *xxxxFTP*.

5 Run the SAS program(s).

- For UNIX or Windows: Run the program that creates the directory-based report structure.
- For z/OS with a directory-based report structure: Run the program that creates the directory-based report structure.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the program that creates the PDS-based report structure. Then run the program that FTPs the report structure to UNIX or Windows.

For more information about submitting a SAS program, see “Working with the Interface for Batch Mode” on page 18.

6 Check the report structure(s).

Notice that the report structure is, physically, relatively flat.

- For UNIX or Windows: Check that the directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)
- For z/OS with a directory-based report structure: Check that the directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Check that the PDSs have been created in the location and with the names that you specified.

  Also check that the corresponding directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)

The Report Program(s)

Create the program that runs nightly to generate reports to the report structure, and test the program.

1 Create the program.

This SAS program should contain a call to the %CPSTART macro (to invoke SAS IT Resource Management). Also, for each end-of-path location in the report structure (that is, for each location that contains report groups), the program should contain one call to the %CPWEBINI macro (to clear old reports) and one or more calls to the %CPRUNRPT macro (to generate the reports for the one or more report groups at that location). Here is an example:

```
%CPWEBINI (cat=work.grweb_gr,
   dir=end-of-path-1) ;
%CPRUNRPT (srd1 srd2 srd3 srd4,
   folder=pgmlib.itsvrpt,
   outloc=work.grweb_gr,
   outmode=web,
   outdesc=Report Group A,
   htmldir=end-of-path-1,
   webstyle=gallery2) ;
%CPRUNRPT (srd5 srd6,
   folder=pgmlib.itsvrpt,
   outloc=work.grweb_gr,
   outmode=web,
   outdesc=Report Group B,
   htmldir=end-of-path-1,
   webstyle=gallery2) ;
```

This code clears reports from the catalog *WORK.GRWEB_GR* and its corresponding directory *end-of-path-1*. Next, the code runs supplied report definitions *SRD1*, *SRD2*, *SRD3*, and *SRD4* from the folder *PGMLIB.ITSVRPT*, and generates their reports first to the catalog *WORK.GRWEB_GR* and from there to the directory *end-of-path-1*, as reports for *Report Group A*. Then the code runs supplied report definitions *SRD5* and *SRD6* from the folder *PGMLIB.ITSVRPT*, and generates their reports first to the catalog *WORK.GRWEB_GR* and from there to the directory *end-of-path-1*, as reports for *Report Group B*.

For more information about the %CPWEBINI macro, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation.

For more information about the %CPRUNRPT macro, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation. For additional information about %CPRUNRPT, see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. Focus especially on Example 7. In Example 7, for now you can ignore section 3, section 4's GLBWHERE= parameter, and section 5.

- For UNIX or Windows: For each report group, specify that the destination of the reports is the appropriate one of the end-of-path directories that you created in your directory-based report structure.

  Note: This program corresponds to the *xreport.sas* file that the QuickStart Wizard created for the other data collector or data source. You might want to look at that program and see if there are any other comments or features that you want to use in your program. △

- For z/OS with a directory-based report structure: For each report group, specify that the destination of the reports is the appropriate one of the end-of-path directories that you created in your report structure based on a z/OS UNIX File System.

  Note: This program corresponds to the *xREPORT* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): For each report group, specify that the destination of the reports is the appropriate one of the end-of-path PDSs that you created in your PDS-based report structure.

  Note: This program corresponds to the *xREPORT* member that the QuickStart Wizard created for the other data collector or data source. You might want to look at that job and see if there are any other comments or features that you want to use in your job. △

2 For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): You do not need to create another FTP program to FTP reports from the PDS-based report structure to the directory-based report structure. You can use the same program that you created to FTP the report structure.

3 Run the program(s).

- For UNIX of Windows: Run the program that generates the reports (and some dynamic files) to the report structure.
- For z/OS with a directory-based report structure: Run the program that generates the reports (and some dynamic files) to the report structure based on a z/OS UNIX File System.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the program that generates the reports (and some dynamic members) to the PDS-based report structure.

  Then run the FTP program that you created earlier. The %CMFTPSND macro reuses directories and files that already exist and adds the directories and files that did not already exist.

4 Check the reports.

Point your Web browser to the welcome.htm file in the "super" directory. Check that the report groups are at the end-of-path locations that you intended, and that the reports are in the report groups that you intended.

Graphic reports display as "thumbnail" sketches. To view a full-size version of a report, click on its thumbnail sketch. To return the report to thumbnail size, select **`Shrink`**.

5 Set permissions so that "readonly" access to the directory-based report structure (or execute access, if appropriate for the report mechanisms) is available to you and to anyone else who needs to view the reports.

For more information about access rights, see "Actions and Access Rights Required" on page 44.

The Job Schedule

Schedule the nightly jobs. Typically the jobs run one time a day, at night. However, you can schedule them to run more or less often than one time a day, and at whatever time(s) work best for your site.

- For UNIX: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to your *cron* and *crontab* man pages.
- For Windows: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to the documentation for your task scheduler.
- For z/OS with a directory-based report structure: Schedule the process-and-reduce program and the report-generation program to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Schedule the process-and-reduce program, the report-generation program, and the FTP program to run (in that order) nightly.

  For more information, refer to your scheduling system's documentation.

Your Notes

Make notes in your own schedule.

- The first day after scheduling, confirm that the programs ran as scheduled.
- The first day after scheduling, confirm that the backup of the PDB was created and that it is identical to (or can be used to regenerate a structure that is identical to) the PDB.

  Unless you have a strong preference for leaving the backup step after the process and reduce steps, in the process, reduce, and backup job move the backup step to precede the process and reduce steps.
- Every day, check the logs from the nightly jobs for error messages, if any. You might also want to check for warning messages.

  If you need to restart a nightly job,
  - For UNIX or Windows: See the *restart.txt* file that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
  - For z/OS: See the *$RESTART* member that the QuickStart Wizard generated. Also see "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
- In a step above, you updated the site library's list of holidays so that it is correct for your site through some date in the future. When that date is reached, SAS IT Resource Management will write a reminder to the log during the process-and-reduce run. But in case you miss seeing that reminder, you might want to add an item to your own schedule to update the list again before that last date.

  For more information about the holiday list, see "Viewing/Editing Your Active Site Library's List of Holidays" on page 286.
- In a step above, you set the workshift schedule in the PDB. If your site has seasonal shift changes (for example, summer hours), you might want to add an

item to your own schedule to update the workshift schedule at the seasonal changes.

For more information about the workshift schedule, see "Viewing/Editing Your Active PDB's Work Shift Schedule" on page 334.

- You might want to schedule a periodic check of table and variable usage. If data is being kept for any variables or tables that are not being used for reporting, you might want to set the variables' or tables' Kept status to *No* so that you can save space in the PDB and time during processing and reduction.
- You might want to schedule a periodic check of space usage.

  For more information about space usage, see "Space Usage at a Level in a Table" on page 349.
- If you have any formats that are based on changeable information (for example, formats related to network configuration or storage devices), you might want to schedule a periodic review of the formats.

  For more information about SAS formats, see SAS help.
- At some sites, copies of the demonstration PDBs are used for practice or testing — for example, testing what a new report would look like, before a production PDB has enough data to generate a report.

  At some sites, a demonstration PDB is used when the active PDB must be some other PDB than the one that is to be used. (Some operations cannot run on the active PDB.)

  At other sites, one or more of the demonstration PDBs are never used.

  You can delete one or more of the demonstration PDBs if you have no need for them. Remember, though, that at least one PDB (demonstration PDB or site-created PDB) is required to start SAS IT Resource Management. For more information, see "Deleting a PDB" on page 315.

Optional Customization

After enough data has been collected that most of the reports are fully populated (that is, they have data over the time span of the report), you might or might not want to make other customizations for your site.

- You might want to adjust the report definitions and rule definitions.

  For example, you might want to add, change, or delete definitions.

  For more information about report definitions, see "Using the Manage Report Definitions Tool" on page 468. For more information about rule definitions, see "Using the Perform Exception Analysis Tool" on page 468.
- You might want to adjust static files in the report structure(s).

  A report gallery's static files are the files that are *not* cleared when the reports in the gallery are cleared by %CPWEBINI, %CPMANRPT, or the WEBCLR= parameter.

  For information about customizing these files, see "Customizing a Report Gallery's Static Files" on page 562.
- You might want to age out old reports instead of deleting old reports.

  Where you want to age out old reports, replace the calls to the %CPWEBINI macro with calls to the %CPMANRPT macro, and make the corresponding changes to the OUTLOC= parameter values in the macro calls that generate reports.

  For more information about %CPWEBINI, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation. For more

information about %CPMANRPT, see the topic %CPMANRPT in the SAS IT Resource Management macro reference documentation.

- You might want to modify the report structure.

  Typically, the report structure is not customized. However,

  - you can add new paths in the report structure. If you add new paths, be very careful to specify end-of-path locations that either do not currently exist or, if they exist, are empty and not used for any other purpose.
  - where you use %CPWEBINI to clear reports, you can remove or revise the existing paths in the report structure.
  - where you use %CPMANRPT to age out reports, you can adjust a few aspects of the report structure: change the titles, and change the text (but not directory names) that is displayed on existing paths. You cannot remove or revise existing paths in the report structure (unless you are willing to risk losing all the reports on those paths).

  To customize the report structure:

  - For UNIX or Windows: Edit the program that you used to create the report structure.

    Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the revised structure-creation program to SAS.

    Finally, revise the report-generation program to generate reports to the appropriate destinations in the report structure, as necessary.
  - For z/OS with a directory-based report structure: Edit the job that you used to create the report structure based on a z/OS UNIX File System.

    Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the revised structure-creation job.

    Finally, revise the report-generation program to generate reports to the appropriate destinations in the revised report structure, as necessary.
  - For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Edit the job that you used to create the PDS-based report structure. Also, edit the FTP job, as necessary.

    Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structures if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structures. Submit the revised structure-creation job. Then, run the revised FTP job.

    Finally, revise the report-generation program to generate reports to the appropriate destinations in the PDS-based report structure, as necessary. (The nightly FTP job will make the corresponding changes to the directory-based revised report structure.)

  Note: To edit an entry in a SAS catalog, you can use the SAS Program Editor window. For example, suppose that you want to copy the entry in ADMIN.*catalog-name.entry-name.entry-type*, edit it, and then save it to ADMIN.*catalog-name.entry-name.entry-type*, which is in the ADMIN library in your active PDB. Here are the steps:

1 Open the SAS Program Editor window.
2 In the command field, type the following command and press ENTER:

```
copy ADMIN.catalog-name.entry-name.entry-type
```

3 The entry appears.
4 Save the original version under a different name, in case you want to refer to it later.

You can save the entry to any library that exists and has a libref that is defined with write access. (If the library exists and a libref is not currently defined or is defined with readonly access, you can define a libref by using the SAS LIBNAME statement.)

In order to write to ADMIN, if the PDB is activated with read-only access, then activate it with write access.

In the command field, type a command like the following one and press ENTER:

```
save ADMIN.catalog-name.another-entry-name.entry-type
```

where *another-entry-name* is an existing entry or an entry that is to be created by this command.

5 Edit the entry.
6 Save the edited entry.

In the command field, type the following command and press ENTER:

```
save ADMIN.catalog-name.entry-name.entry-type
```

If the PDB was activated initially with read-only access, activate it again with read-only access.

7 Close or iconize the SAS Program Editor window.

△

For more information about grouping reports in the tree, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide. For more information about the %CPHTREE macro, see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. For more information about the %CMFTPSND macro, see the topic %CMFTPSND in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

- You might want to adjust the table definitions and the variable definitions.

  For more information about tables, see "Overview of Tables" on page 353. For more information about variables, see "Overview of Variables" on page 378.

- You might want to adjust the PDB properties and site library properties.

  For more information about PDBs, see "Overview of Whole PDBs" on page 296. For more information about site libraries, see "Overview of Site Libraries" on page 270.

- You might want to review how checking for duplicate data is handled.

  For more information about checking for duplicate data, see "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation.

- If you want to change some aspect of the process step and you do not see any supplied way of making that change, you might want to read about process exits.

For more information about process exits, see “Overview of Using Process Exits” on page 442.

- You might want to combine or split PDBs.

  For information about combining PDBs, see “Combining PDBs” on page 304. For information about splitting PDBs, see “Splitting One PDB into Two PDBs” on page 320.

- If your users will use the SAS IT Resource Management client GUI to report on data in a PDB on a SAS IT Resource Management server host, make a list of the information that the users will need in order to create a remote server profile for that host.

  For more information, see “Creating a Remote Server Profile” on page 574.

Cleanup

Delete the PDB and related structures that the QuickStart Wizard created.

- For the UNIX or Windows QuickStart Wizard PDB: Delete the QuickStart Wizard’s PDB, including the *qs* directory (and its subdirectories) that is related to the QuickStart Wizard’s PDB.
- For the QuickStart Wizard PDB that is located in a z/OS environment: Delete the QuickStart Wizard’s PDB, including any PDSs with *.QS* as a high-level qualifier and any directories with */qs* in the directory path that are related to the QuickStart Wizard’s PDB.

Optional Package

If you want to package and install the collector-support that you have created, see “Creating and Installing a Collector Support Package” on page 426. Also see the topic %CPPKGCOL in the SAS IT Resource Management macro reference documentation, the topic %CPRPTPKG in the SAS IT Resource Management macro reference documentation, and the topic %CPINSPKG in the SAS IT Resource Management macro reference documentation.

CHAPTER 8

Setup Case 5

Setup Case 5

Overview of Setup Case 5

For general information about setting up the server, see “Setting Up the Server, with Setup Cases” on page 38. That topic also identifies which setup case applies to your situation.

Below are the instructions for setup case 5. For each step, there is an overview of the step and, after that, the specific directions for the step. Typically, UNIX and Windows have one or two sets of directions and z/OS has one or two sets of directions. The steps are in the following order:

1. “The QuickStart Wizard” on page 120
2. “The Raw Data” on page 120
3. “The PDB and Site Library” on page 120
4. “The Tables and Variables” on page 120
5. “The Process-and-Reduce Program” on page 122
6. “The Report Structure” on page 123
7. “The Report Program(s)” on page 125
8. “The Job Schedule” on page 126
9. “Your Notes” on page 126
10. “Optional Customization” on page 126
11. “Cleanup” on page 128.

The QuickStart Wizard

It is assumed that you already worked through setup case 1 or 2 for the part of the data to which setup case 1 or 2 is applicable.

In setup case 5, you will modify the structures and programs from setup case 1 or 2 so they also cover the remaining data.

The Raw Data

Start to collect raw data, if you do not already collect raw data.

If you are not already collecting raw data from the additional MIB, object, class, model type, or schema file, start collecting the data now.

For more information, see “Collector-Specific Setup Documentation” on page 12. Also, see the documentation from the vendor of your data collector or data source.

The PDB and Site Library

You already have a PDB from setup case 1 or 2. In setup case 1 or 2, you already customized the PDB and site library.

The Tables and Variables

In the PDB from setup case 1 or 2, create one or more additional tables for the data from the additional MIB, object, class, model type, or schema file, and check the definitions.

1 Generate a file that contains table and variable definitions.

To generate the file, see the following instructions in the *SAS IT Resource Management Server Setup Guide*. (For more information about the Server Setup Guide, see “Collector-Specific Setup Documentation” on page 12.)

- For Aprisma SPECTRUM: In the *SAS IT Resource Management Server Setup Guide*, see “Aprisma SPECTRUM Appendix 4: Defining SAS IT Resource Management Tables for SPECTRUM Data."
- For DSI: In the *SAS IT Resource Management Server Setup Guide*, see Section 1, Task 3: Action 1 “Create Your Test PDB," and select **Supplied ▶ UNIX ▶ MWA DSI**
- For HP OpenView Network Node Manager: See “Using the MIB to Dictionary Compiler” on page 129.
- For NTSMF: In the *SAS IT Resource Management Server Setup Guide*, see “NTSMF Appendix 2: Using GENERATE SOURCE to Construct PATROL and NTSMF Tables."
- For PATROL: In the *SAS IT Resource Management Server Setup Guide*, see “PATROL Appendix 2: Using GENERATE SOURCE to Construct PATROL and NTSMF Tables."
- For SunNet Manager or Enterprise Manager: In the *SAS IT Resource Management Server Setup Guide*, see “SunNet Manager and Enterprise Manager Appendix 2: Defining Tables from Your Own Schema Files."

2 In the generated file, review and adjust the one or more blocks of definitions.

A block of definitions corresponds to a table. In each block, there is a CREATE TABLE control statement, CREATE VARIABLE control statements, and an UPDATE TABLE control statement.

If the block corresponds to an existing table, delete the block. (If the block's CREATE TABLE control statement has a value for its EXTNAME= parameter that matches the value for the External Name of a table in your PDB, the block corresponds to that table. For more information about viewing the value of External Name in a table in your PDB, see "Viewing/Editing a Table" on page 364.)

If the block does not correspond to an existing table, review the control statements. In a text editor modify the control statements as necessary, and add any CREATE DERIVED or CREATE FORMULA control statements to the block, as necessary. In any given block, consider the following:

- In the CREATE TABLE control statement, you might want to change the table's name, label, and/or description. Also review the other parameters.

  For more information about the CREATE TABLE control statement, see the topic CREATE TABLE in the SAS IT Resource Management macro reference documentation.
- In the CREATE VARIABLE control statements, you might want to change the variables' names, labels, and/or descriptions. Also review the other parameters.

  For more information about the CREATE VARIABLE control statement, see the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation.
- After the CREATE VARIABLE control statements, add a CREATE DERIVED control statement for each derived variable that you want to have in the table, if any. The CREATE DERIVED control statements can be in any order, but alphabetical order is convenient.

  For more information about the CREATE DERIVED control statement, see the topic CREATE DERIVED in the SAS IT Resource Management macro reference documentation.
- In the UPDATE TABLE control statement, change the table's name (if you changed it in the CREATE TABLE control statement). You might want to revise the table's age limits. Also review other parameters.

  For more information about the UPDATE TABLE control statement, see the topic UPDATE TABLE in the SAS IT Resource Management macro reference documentation.

  Note: If you are considering revisions to the BY and CLASS lists for this data collector or data source, first see "Restrictions on Modifications of BY Variables Lists" on page 348. △
- After the UPDATE TABLE control statement, add a CREATE FORMULA control statement for each formula variable that you want to associate with the table. The CREATE FORMULA control statements can be in any order, but alphabetical order is convenient.

  For more information about the CREATE FORMULA control statement, see the topic CREATE FORMULA in the SAS IT Resource Management macro reference documentation.

Note: If a derived variable depends on another derived variable and/or if a formula variable depends on another formula variable, be sure to read the information about dependencies in the documentation for the CREATE DERIVED and/or CREATE FORMULA control statements. △

3 Create a program.

This SAS program calls (in this order) the %CPSTART macro (to invoke SAS IT Resource Management) and the %CPDDUTL macro (to run the control statements that define the tables and their variables.)

- Copy the call to the %CPSTART macro from one of the production programs that you scheduled in setup case 1 or 2.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation.

- In the call to the %CPDDUTL macro,

  ```
  %cpddutl(filename='location-of-control-statements');
  ```

  specify the name of the file containing the table and variable control statements. %CPDDUTL applies the control statements in the file to the active PDB.

  For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation.

4 Run the program.

  Submit the program to SAS.

  For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

5 The PDB's new table and variable definitions now exist.

  If you want to browse them, you can. For more information about viewing the definition of a table, see "Viewing/Editing a Table" on page 364. For more information about viewing the definition of a regular variable, see "Creating/Editing/Viewing a Regular Variable" on page 394. For more information about viewing the definition of a derived variable, see "Creating/Editing/Viewing a Derived Variable" on page 381. For more information about viewing the definition of a formula variable, see "Creating/Editing/Viewing a Formula Variable" on page 388.

The Process-and-Reduce Program

Modify the nightly process-and-reduce program that you created in setup case 1 or 2, and test the program.

1 Modify the program.

  In the table list in the call to the %C*x*PROCES macro (where *x* = M, P, S, or W), comment out the names of the supplied tables from setup case 1 or 2 and add the names of the tables that you created in the step above. (To comment something out, insert comment delimiters, /* before it and */ after it.)

2 When you have collected at least one day's worth of raw data for the tables that you created in this setup case, run the program.

3 The data in the created tables corresponds to the raw data for this setup case.

  If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.

4 Modify the program again.

  Remove the comment delimiters from the names of the supplied tables from setup case 2. As a result, the table list should have the names of the supplied tables from setup case 1 or 2 and should have the names of the tables that were created in this setup case.

The Report Structure

Modify the report structure that you created in setup case 1 or 2.

1 Create new report definitions for the data in the table(s) that you created in this setup case, and run those report definitions.

   □ In the SAS IT Resource Management server GUI for UNIX or Windows: On the main window's **Reporting** tab, select **Manage Report Definitions ▶ File ▶ New Report Definition**

   On the right side of the window, create a report definition for the data in the table(s) that you created above. Select **Run** to generate a report based on the report definition. In the report, the data corresponds to the data in the PDB. (For more information about creating and running a report definition, see "Using the Manage Report Definitions Tool" on page 468. If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.)

   Next, select **File ▶ Save Report Definition** and complete that path to save the report definition in its interactive form.

   Optionally, create and save other report definitions for the data in the table that you created.

   □ For z/OS: In the SAS IT Resource Management server GUI for z/OS, from the main menu select **Reports ▶ Custom Rpts ▶ Design Repts**

   Select the type of report that you want to create (at this point, select only from the seven types at the top of the list). The type's main report definition window opens. Select the level that you want to use, select the icon in the upper-left corner to select the table that you want to use, and specify all other parameters that you want to use. Then select **Actions ▶ Run Report**

   In the report that is generated, the data corresponds to the data in the PDB. (For more information about creating and running a report definition, in the type's main report-definition window, select **Help**. If you want to browse the data in the PDB, see "Viewing Data Values" on page 420.)

   Next, in the main window for the report definition, select **File ▶ Save Definition** and complete the path to save the report definition in its interactive form.

   Optionally, create and save other report definitions for the data in the created table(s).

   For information about report definitions in the SAS IT Resource Management client GUI, see "Using the Manage Report Definitions Tool" on page 468.

   In a later step, you will need the names of the report definitions that you want to use. Notice that the names are available in two places: in the list in the GUI, and in the lower-left corner of each report.

2 Decide on report groupings for the new reports.

   Report groupings usually make a particular report easier to find. Also, report groupings usually make a particular report easier to interpret, by making it easy to compare it with related reports in the same group.

   For more information about report groupings, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide.

   Note: You can decide to generate the new reports to the existing report groupings that are provided from setup case 1 or 2 or to new report groupings, and to the existing end-of-path directories from setup case 1 or 2 or to new end-of-path directories. △

3 Unless you decided to use only the existing end-of-path directories, revise the program that created the report structure in setup case 1 or 2. (At the end of setup case 1 or 2, you might have modified this program. Start from the most recent version of the program.)

- For UNIX or Windows: Edit the file to add any new paths that you decided to use for this setup case. Be careful not to change the end-of-path directories for existing paths. Also be careful, for any new end-of-path locations, to specify directories that do not exist or directories that exist but are not in use for any other purpose.
- For z/OS with a directory-based report structure: Edit the job to add any new paths that you decided to use for this setup case. Be careful not to change the end-of-path directories for existing paths. Also be careful, for any new end-of-path locations, to specify directories that do not exist or directories that exist but are not in use for any other purpose.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Edit the job to add any new paths that you decided to use for this setup case. Be careful not to change the end-of-path PDSs for existing paths. Also be careful, for any new end-of-path locations, to specify PDSs that do not exist or PDSs that exist but are not in use for any other purpose.

  Make the corresponding edits, if any, to the FTP job.

The file or job calls the %CPHTREE macro and %CPENTCPY macro, and can call the %CMFTPSND macro. For more information about the %CPHTREE macro, see the topic %CPHTREE in the SAS IT Resource Management macro reference documentation. For more information about the %CPENTCPY macro, see the topic %CPENTCPY in the SAS IT Resource Management macro reference documentation. For more information about the %CMFTPSND macro, see the topic %CMFTPSND in the SAS IT Resource Management macro reference documentation.

4 Run the program.

- For UNIX or Windows: Run the program that creates the directory-based report structure.
- For z/OS with a directory-based report structure: Run the program that creates the directory-based report structure.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the program that creates the PDS-based report structure. Then, optionally run the program that FTPs the report structure to a UNIX or Windows system.

For more information about submitting a SAS program, see “Working with the Interface for Batch Mode” on page 18.

5 Check the report structure.

Notice that the report structure is, physically, relatively flat.

- For UNIX or Windows: Check that the directories have been created or added in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no new reports are available in the right frame.)
- For z/OS with a directory-based report structure: Check that the directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the “super” directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Check that the PDSs have been created in the location and with the names that you specified.

  Also check that the corresponding directories have been created in the location and with the names that you specified.

  Also, point your Web browser to the file *welcome.htm* in the "super" directory, and check that the left frame works as you intended. (At this point, no reports are available in the right frame.)

The Report Program(s)

Revise the nightly report-generation program from setup case 1 or 2, and test the program.

1 Modify the program.

Use calls to the %CPRUNRPT macro to generate the new reports to the appropriate report groups and end-of-path locations in the revised report structure.

Remember to call the %CPWEBINI macro (if the program uses %CPWEBINI) and/or the %CPMANRPT macro (if the program uses %CPMANRPT) to clear the reports at a given end-of-path location before making one or more calls to %CPRUNRPT to generate reports for report groups at that end-of-path location.

For more information about the %CPWEBINI macro, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation. For more information about the %CPMANRPT macro, see the topic %CPMANRPT in the SAS IT Resource Management macro reference documentation. For more information about the %CPRUNRPT macro, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation.

2 Run the program.

- For UNIX or Windows: Run the modified report-generation program.

  This program generates reports (and some dynamic files) to the directory-based report structure.

- For z/OS with a directory-based report structure: Run the modified report-generation program.

  This program generates reports (and some dynamic files) to the report structure based on a z/OS UNIX File System.

- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Run the modified report-generation program.

  This program generates reports (and some dynamic files) to the PDS-based report structure.

  Then run the modified FTP program.

  This program FTPs the reports from the PDS-based report structure on z/OS to a directory-based report structure on a UNIX or Windows system.

3 Check the reports.

Point your Web browser to the file *welcome.htm* in the "super" directory. Check that the reports from setup case 1 or 2 and the reports from this setup case are in the report groups and the end-of-path locations that you intended.

Graphic reports display as "thumbnail" sketches. To view a full-size version of a report, click on its thumbnail sketch. To return the report to thumbnail size, select **`Shrink`**.

4 Set permissions so that "readonly" access to the modified directory-based report structure (or execute access, if appropriate for the report mechanisms) is available to you and to anyone else who needs to view the reports.

   For more information about access rights, see "Actions and Access Rights Required" on page 44.

The Job Schedule

The nightly programs are already scheduled. (Typically the jobs run once a day, at night. However, you can schedule them to run more or less often than once a day, and at whatever time(s) work best for your site.) Now the nightly programs have been modified to cover both setup case 1 or 2, and this setup case.

Your Notes

Make notes in your own schedule.

- The first day after the revisions, confirm that the scheduled programs ran as expected.

Optional Customization

After enough data has been collected that most of the reports are fully populated (that is, they have data over the time span of the report), you might or might not want to make other customizations for your site.

- You might want to adjust the report definitions and rule definitions.

  For example, you might want to add, change, or delete definitions.

  For more information about report definitions, see "Using the Manage Report Definitions Tool" on page 468. For more information about rule definitions, see "Using the Perform Exception Analysis Tool" on page 468.

- You might want to adjust static files in the report structure(s).

  A report gallery's static files are the files that are *not* cleared when the reports in the gallery are cleared by %CPWEBINI, %CPMANRPT, or the WEBCLR= parameter.

  For information about customizing these files, see "Customizing a Report Gallery's Static Files" on page 562.

- You might want to age out old reports instead of deleting old reports.

  Where you want to age out old reports, replace the calls to the %CPWEBINI macro with calls to the %CPMANRPT macro, and make the corresponding changes to the OUTLOC= parameter values in the macro calls that generate reports.

  For more information about %CPWEBINI, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation. For more information about %CPMANRPT, see the topic %CPMANRPT in the SAS IT Resource Management macro reference documentation.

- You might want to modify the report structure.

  Typically, the report structure is not customized. However,

  - you can add new paths in the report structure. If you add new paths, be very careful to specify end-of-path locations that either do not currently exist or, if they exist, are empty and not used for any other purpose.

- where you use %CPWEBINI to clear reports, you can remove or revise the existing paths in the report structure.
- where you use %CPMANRPT to age out reports, you can adjust a few aspects of the report structure: change the titles, and change the text (but not directory names) that is displayed on existing paths. You cannot remove or revise existing paths in the report structure (unless you are willing to risk losing all the reports on those paths).

To customize the report structure:

- For UNIX or Windows: Edit the most recent version of the program that creates the report structure.

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the revised structure-creation program to SAS.

  Finally, revise the report-generation program to generate reports to the appropriate destinations in the revised report structure, as necessary.
- For z/OS with a directory-based report structure: Edit the most recent version of the program that creates the report structure.

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structure if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structure. Submit the *xRPTSTR* job in order to create the new report structure.

  Finally, revise the *xREPORT* job to generate reports to the appropriate destinations in the revised report structure based on a z/OS UNIX File System or based on a PDS, as necessary.
- For z/OS with a PDS-based report structure (and a corresponding directory-based report structure on a UNIX or Windows system): Edit the most recent version of the program that creates the report structure. Then make the corresponding edits in the FTP program, as necessary.

  Next, where you are using %CPWEBINI to clear reports, you can delete those paths in the report structures if you want to remove the paths or if you intend to revise them; otherwise, do not delete the existing paths in the report structures. Submit the *xRPTSTR* job and then the *xFTPHTML* job, in order to create the new report structures.

  Finally, revise the *xREPORT* job to generate reports to the appropriate destinations in the PDS-based revised report structure, as necessary. (The nightly *xFTPHTML* job will FTP the reports to the directory-based report structure.)

Note: To edit an entry in a SAS catalog, you can use the SAS Program Editor window. For example, suppose you want to copy the entry in ADMIN.*catalog-name.entry-name.entry-type*, edit it, and then save it to ADMIN.*catalog-name.entry-name.entry-type*, which is in the ADMIN library in your active PDB. Here are the steps:

1. Open the SAS Program Editor window.
2. In the command field, type the following command and press ENTER:

   ```
   copy ADMIN.catalog-name.entry-name.entry-type
   ```

3. The entry appears.

4 Save the original version under a different name, in case you want to refer to it later.

   You can save the entry to any library that exists and has a libref that is defined with write access. (If the library exists and a libref is not currently defined or is defined with read-only access, you can define a libref by using the SAS LIBNAME statement.)

   In order to write to ADMIN, if the PDB is activated with read-only access, then activate it with write access.

   In the command field, type a command like the following one and press ENTER:

   ```
   save ADMIN.catalog-name.another-entry-name.entry-type
   ```

   where *another-entry-name* is an existing entry or an entry that is to be created by this command.

5 Edit the entry.

6 Save the edited entry.

   In the command field, type the following command and press ENTER:

   ```
   save ADMIN.catalog-name.entry-name.entry-type
   ```

   If the PDB was activated initially with read-only access, activate it again with read-only access.

7 Close or iconize the SAS Program Editor window.

△

For more information about grouping reports in the tree, see the section "Grouping Reports in a Web Gallery" in the chapter "Setup: Introduction" in the SAS IT Resource Management User's Guide. For more information about the %CPHTREE macro, see the %CPHTREE topic in the SAS IT Resource Management macro reference documentation. For more information about submitting a SAS program, see "Working with the Interface for Batch Mode" on page 18.

- You might want to adjust the table definitions and the variable definitions.

  For more information about tables, see "Overview of Tables" on page 353. For more information about variables, see "Overview of Variables" on page 378.

- You might want to adjust the PDB properties and site library properties.

  For more information about PDBs, see "Overview of Whole PDBs" on page 296. For more information about site libraries, see "Overview of Site Libraries" on page 270.

- You might want to review how checking for duplicate data is handled.

  For more information about checking for duplicate data, see "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation.

Cleanup

Delete the PDB and related structures that the QuickStart Wizard created.

- For the QuickStart Wizard PDB that is located in a UNIX or Windows environment: Delete the QuickStart Wizard's PDB, including the *qs* directory (and its subdirectories) that is related to the QuickStart Wizard's PDB.
- For the QuickStart Wizard PDB that is located in a z/OS environment: Delete the QuickStart Wizard's PDB, including any PDSs with *.QS* as a high-level qualifier and any directories with */qs* in the directory path that are related to the QuickStart Wizard's PDB.

CHAPTER

9

The MIB to Dictionary Compiler

The MIB to Dictionary Compiler

Using the MIB to Dictionary Compiler

Introduction

In the master data dictionary, SAS IT Resource Management supplies table and variable definitions for the data from many collectors. For that data, you can use the supplied table and variable definitions (after adding them from the master data dictionary to your PDB). For other data, you can create the table and variable definitions in your PDB by running the SAS IT Resource Management %CPDDUTL macro with %CPDDUTL control statements that define the tables and variables.

You can write the %CPDDUTL control statements yourself. But for SNMP data (all of which is described with a MIB), there is an easier way to generate the necessary control statements that add the MIB tables and variables to your PDB. The SAS IT Resource Management **mib2dict** compiler takes as input any SNMP V1 or V2 MIB and produces as output %CPDDUTL control statements that can be read directly by the %CPDDUTL macro to define SAS IT Resource Management tables and variables. The **mib2dict** compiler can optionally produce SAS statements for input to the SAS FORMAT procedure so that you can add formats for MIB enumeration types.

As an example, this chapter walks you through the steps that are needed to create table and variable definitions for MIB-II data. The same steps apply to any MIB. We use MIB-II as an example because of its simplicity and familiarity. If you want to use MIB-II tables in your PDB, you do not actually need to use **mib2dict** because all the MIB-II tables are included in the master data dictionary that is shipped with SAS IT Resource Management. (MIB-II table names begin with the character *M*.)

Note: For more information about the %CPDDUTL macro, see the macro reference documentation for SAS IT Resource Management. △

Note: If you need to know more about the MIB for one of your hardware devices, check the hardware documentation or call the manufacturer of the device. There are proprietary MIBs and public domain, Internet-standard MIBs.

If the MIB is a proprietary MIB, the hardware installation procedure probably copied it to a directory at your site and/or you can probably find a copy on the manufacturer's Web page. △

For information about the context of using the MIB to Dictionary Compiler, see "Overview of Setup Case 5" on page 119.

Prerequisites

- The SAS IT Resource Management server must be installed on your SAS host machine.
- The SAS IT Resource Management server software must be running, and the PDB to which you want to add the tables must be the active PDB.
- You must be familiar with the **mib2dict** man page.

  To view the man page, execute the following on UNIX (assuming that SAS has been installed in **/usr/local/sas**):

  ```
  cd /usr/local/sas/utilities/man/man1
  nroff -man mib2dict.1 | more
  ```

- The MIB that you want to compile must conform to SNMPv1 or SNMPv2 standards.

  In this example, we will use **rfc1213.mib**, which is a file that contains the definition of MIB-II. If you have installed SAS and the SAS IT Resource Management server in **/usr/local/sas**, you can find a copy of **rfc1213.mib** in **!sasroot/saspgm/cpe/mibs/** on UNIX, or **!sasroot\cpe\mibs** on Windows.

Actions

1 Compile the MIB.

In this action, we will compile **rfc1213.mib** and produce two output files: **rfc1213.ddu** and **rfc1213.sas**. The **.ddu** file will contain the %CPDDUTL control statements that are needed by the %CPDDUTL macro. The **.sas** file will contain the PROC FORMAT statements that are needed to create the formats for enumeration types.

To compile the MIB, execute the following command.

Note: This code assumes that SAS has been installed in **!sasroot** and that your PATH includes **!sasroot/sas/utilities/bin/**. △

On UNIX:

```
mib2dict -i rfc1213.mib -o rfc1213.ddu -f rfc1213.sas -c hp-ov
```

On Windows:

```
mib2dict.exe -i rfc1213.mib -o rfc1213.ddu -f rfc1213.sas -c hp-ov
```

where

- the **-i** option specifies **rfc1213.mib** as the input file to be compiled. If you do not specify **-i**, then **mib2dict** will expect input from stdin.
- the **-o** option specifies **rfc1213.ddu** as the output file to which %CPDDUTL control statements are printed. If you do not specify **-o**, %CPDDUTL control statements will print to *stdout*.

- the **-f** option specifies that **rfc1213.sas** is to receive the SAS statements for PROC FORMAT. If you do not specify **-f**, then no PROC FORMAT statements are written to **rfc1213.sas**.

  Note: When you specify the **-f** option, any %CPDDUTL CREATE VARIABLE control statements that are generated by **mib2dict** have a FORMAT= parameter that is set to the proper SAS format, as defined in **rfc1213.sas**. In a later action, you will execute **rfc1213.sas**. If you do not execute **rfc1213.sas**, the formats will not exist and you will receive format errors when a value that uses the nonexistent format is to be displayed. △

- the **-c** option specifies that HP-OV is the collector to use. If you do not specify **-c**, the collector will be set to an empty string.

  Note: When you specify the **-c** option, any %CPDDUTL CREATE TABLE control statements that are generated by **mib2dict** will have a COLLECTOR= parameter that is set to HP-OV. In this example, we use *hp-ov* because we plan to use HP OpenView Network Node Manager or IBM Tivoli NetView for AIX to collect the SNMP data. You can use any collector supported by SAS IT Resource Management that can collect SNMP data. Or, if you are logging SNMP data in an unsupported format, you can use **-c generic** and use the SAS IT Resource Management Generic Collector Facility to read the raw data. △

2 Review, and revise if necessary, the resulting %CPDDUTL control statements.

Although **mib2dict** is designed to create %CPDDUTL control statements that can be directly processed by the %CPDDUTL macro, there are circumstances in which you might want to edit the control statements before you use them:

- There might be tables or variables that were in the MIB but that you do not want to add to your PDB.

  If this is the case, prior to executing the %CPDDUTL macro you can simply delete the %CPDDUTL control statements that define the tables or variables that you do not want.

  To delete an entire table, look for the CREATE TABLE statement and delete everything up to the next CREATE TABLE statement.

  To delete a variable from a table, simply delete everything from CREATE VARIABLE up to and including the semicolon that ends the statement.

  If you delete a variable from a table but you do not delete the entire table, be sure that the variable that you deleted is not in the table's BYVARS or CLASSVARS lists. In particular, do not delete MACHINE, HOUR, or SHIFT because these variables are always on the BYVARS and CLASSVARS lists. For more information about the BY and CLASS variables lists, see "Specifying BY or CLASS Variables for a Level in a Table" on page 339.

- There might be some SAS IT Resource Management table or variable names that you want to change.

  Although **mib2dict** can squeeze long MIB table and variable names down to six-character names for the PDB tables and seven-character names for the PDB variables, the resulting names do not always look mnemonic.

  So that users of your PDB can read the names more easily, you can modify the PDB table and variable names in the file to which the %CPDDUTL control statements were written. For the naming conventions of user-defined tables and variables, see "Naming Conventions for UNIX and Windows Tables and Variables" on page 374.

3 Optionally, create formats for MIB enumeration types.

If you used the **-f rfc1213.sas** option when you ran **mib2dict**, then you need to execute **rfc1213.sas**. This SAS program executes the SAS FORMAT procedure, which creates the new formats and installs them in the active PDB's **DICTLIB.CPFMTS** catalog.

If your SAS Program Editor window has pull-down menus, select **File ▶ Open ▶ *locate rfc1213.sas* ▶ Open**

Otherwise, type **include rfc1213.sas** on the command line and press ENTER or RETURN. Then, select **Locals ▶ Submit**

Otherwise, type SUBMIT on the command line and press ENTER or RETURN.

4 Run the %CPDDUTL macro to create the table and variable definitions in your PDB.

In the body of the SAS Program Editor window, type

```
%cpddutl(filename='rfc1213.ddu');
```

Then if your SAS Program Editor window has pull-down menus, select **Locals ▶ Submit**

Otherwise, type SUBMIT on the command line and press ENTER or RETURN. %CPDDUTL creates the table and variable definitions in your PDB.

5 You are now ready to process raw data into your new PDB.

In our example, we used the mib2dict **-c hp-ov** option to specify HP-OV as the collector. Thus, to process data (for example, for **ifTable** in the newly built UIFTBL table), you could submit the following from the SAS Program Editor window:

```
%csproces (/usr/OV/databases/snmpCollect/,
           UIFTBL,
           collector=HP-OV);
```

CHAPTER
10

MXG-Based Tables (CMAPP2)

Using MXG-Based Tables (CMAPP2)

Using MXG-Based Tables (CMAPP2): Introduction

This document details any special SAS IT Resource Management or MXG actions that must be taken when you activate new tables in a PDB. It contains these topics:

- "Incorporating a Data Source into Your PDB" on page 133
- "Change Variable KEPT Status" on page 134
- "Customization Instructions for Each Data Source" on page 135
- "SMF Data Sources - All Operating Environments" on page 135
- "Non-SMF Data Sources - z/OS Operating Environment" on page 153.

Incorporating a Data Source into Your PDB

The following general instructions can be used to incorporate a data source into your PDB.

1. From the main menu of the SAS IT Resource Management GUI for z/OS, select **Explore ▶ Explore Supplied Tables** to see what tables and variables are available for the data source and to decide which tables you want to use in the PDB.
2. Customize any necessary IMAC and/or EX members in your MXG SOURCLIB PDS as documented in this chapter and in comments at the beginning of each member. The most common customization will be to put the SMF record number that you have assigned to the data source into the IMAC member.

   It is best to perform the customization of MXG SOURCLIB members by modifying a copy of the member in MXG.USERID.SOURCLIB and then to ensure that your %CPSTART macro specifies MXGSRC= to point to both MXG.USERID.SOURCLIB and MXG.MXG.SOURCLIB. For more information, see

the %CPSTART topic in the SAS IT Resource Management macro reference documentation.

3 Select the appropriate tables into your PDB.

4 If you are familiar with the data source, set retention period and class variables as appropriate for each table and begin process and reduction runs.

5 If you are not familiar with the data source, process some of the data, probably no more than one day's worth, into your PDB and then explore the resulting detail data sets using the browse, view, or graph options from the **Item Actions** pull-down menu on the Query PDB data window. You can also use the Design Reports subsystem for this purpose.

   Examining the data for potential class variables will help you decide whether the data is suitable for reduction, and if so, which variables to specify for class variables. In general, the default class variable specifications that are shipped with SAS IT Resource Management represent our best guess based on examination of sample data or the variable definitions for the table.

   The WHERE clause builder, available under "local where" from any Design Reports window, gives you a quick way to examine all the values of a potential class variable. Click on **VARNAME** to select the variable of interest. Then click on the VALUE(S) button to see the list of values that are found in the data for that variable.

   Note: Do not use the ampersand (&) to mean AND in WHERE expressions. △

   Examining the data might also show variables whose content is simply missing values. Variables like this waste space in the PDB, so you might want to define them as KEPT=NO. You might also notice other variables whose contents are not of any particular interest or use to you. These also can be set to KEPT=NO. Of course, if a variable becomes relevant later, you can always reset it to KEPT=YES.

Change Variable KEPT Status

Sometimes you might be asked to change the status of one or more variables in a table from KEPT=NO to KEPT=YES or vice versa. This can be done either by using the SAS IT Resource Management GUI for z/OS or through the batch data dictionary utility %CPDDUTL (see the SAS IT Resource Management macro reference documentation for details). Input statements to the dictionary utility can reside either in a SAS catalog source entry or in an OS data set. The following is an example of how to use the batch utility with OS data set input.

Here are the assumptions for the example:

- SAS IT Resource Management software root is SAS.ITRM.
- PDB to be updated is NEW.PDB.
- OS data set containing card-image input for the dictionary utility is DDUTL.CARDS.
- Variables VAR1 and VAR2 are to be set to KEPT=YES in table TABLEA.

These steps are necessary to change the status of a variable:

1 Edit DDUTL.CARDS to contain the following lines:

```
set table name=TABLEA;
update variable name=VAR1 kept=YES;
update variable name=VAR2 kept=YES;
build views name=TABLEA;
```

   Note: Do not include sequence numbers in the data set. △

2 Submit the following JCL fragment as part of a SAS batch job:

```
//S1 EXEC SAS,
    %cpstart(system=MVS,mode=BATCH,
     root=SAS.ITRM,pdb=NEW.PDB,disp=OLD);
    %cpddutl(filename='DDUTL.CARDS');
//
```

Customization Instructions for Each Data Source

The format of this section is as follows:

TABLEA - TABLEB

Data Source: Vendor product that produces these SMF record(s).

SAS IT Resource Management status:

- tested - we have tested SAS IT Resource Management with this data
- not tested - dictionary entries have been defined for this data, but we have not used them to process data.

MXG customization: MXG customization necessary to use this data source.

MXG Documentation member: none - there is no documentation for this data source in the MXG SOURCLIB.

ADOC*xxxx* - indicates that documentation can be found in this member in the SOURCLIB PDB. These ADOC members can be accessed from the main menu in the SAS IT Resource Management GUI for z/OS by selecting **Help ▸ MXG Online Doc**

SAS IT Resource Management customization: Any additional SAS IT Resource Management considerations for using this data source.

Note: SMF tables in the range XTY0 through XTY99 that are associated with standard IBM SMF records that require no MXG or SAS IT Resource Management customization or special considerations will not be listed. MXG documentation for these SMF records can be found in members ADOC*nn*, where "*nn*" is the SMF record number.

As to the testing status of these standard SMF tables, the following tables have been tested; all others have not:

Table 10.1 SMF Tables That Have Been Tested

XTY26J2	XTY50	XTY73PC
XTY30_V	XTY6	XTY74
XTY30_1	XTY70	XTY75
XTY30_4	XTY71	XTY76
XTY30_5	XTY72	XTY8
XTY30OM	XTY73	XTY78IO

△

SMF Data Sources - All Operating Environments

XACFAR - XACFVR

Data Source: CA-ACF2 security system from Computer Associates

SAS IT Resource Management status: not tested

MXG customization:

IMACACF2 - Specify the SMF record number

MXG Documentation member: none

SAS IT Resource Management customization: none

XBETA0 - XBETA42

Data Source: BETA93 report distribution and print management.

SAS IT Resource Management status: not tested

MXG customization:

IMACBETA - Specify the SMF record number

MXG Documentation member: none

SAS IT Resource Management customization: none

XB1ONCP - XB1ONPG

Data Source: BGS Best/1 I/O monitor

SAS IT Resource Management status: not tested

MXG customization:

IMACBGSI - Specify the SMF record number

MXG Documentation member: none

SAS IT Resource Management customization: none

XCAPAG - XCA90

Data Source: IBM RMF (TM) Cache Reporter

SAS IT Resource Management status: Table XCATY has been tested; the others have not

MXG customization:

IMACACHE - Specify the SMF record number(s)

MXG Documentation member: ADOCACHE (3990 records only)

SAS IT Resource Management customization: none

XCIACCT - XCIYSTM

Data Source: CICS (TM) Monitor Facility type 110 SMF record. XCITRAN (CICS transaction) also contains support for additional variables available through OMEGAMON (R) for CICS from Candle Corporation. Also, includes support for data collected by Boole and Babbage's CICS Manager.

SAS IT Resource Management status: Standard type 110 records have been tested. OMEGAMON CICS extensions and Boole and Babbage extensions have not been tested.

MXG customization:

- IMACKEEP - Add the following lines: `MACRO _KCICTRN _XKCICTR  %`
  `MACRO _LCICTRN WORK.CICSTRAN %`

  You can add them anywhere between the following statements:
  `%INCLUDE SOURCLIB(IMACOLDV);`

  and
  `&MACKEEP ;`

- IMACPTF - Read instructions in IMACPTF. This member accommodates changes to Monitor record format that were introduced by PTFs to older versions of CICS.
- IMACEXCL - You need to modify this member only if you have made any customization to the CICS Monitor record. See instructions in IMACEXCL.
- ANALDB2C and ASUMUOW - SAS IT Resource Management now supports both the ANALDB2C and its preferred form, ASUMUOW, in the forms of two tables named XCICDB2 and XUOW. If you have customized ANALDB2C or ASUMUOW to create extra variables compared to the MXG default members, please ensure that you have the associated SAS IT Resource Management variables marked KEPT=YES in the appropriate data dictionary tables. To process XUOW, you must also modify the IMACUOW MXG member as follows. Comment out or delete the first definition of the two macros, _NOOBS and _YESOBS, and remove the comment delimiters from the second definition, so that the macros are defined as follows:

```
MACRO _NOOBS    %
MACRO _YESOBS   %
```

  You need to look at the following members only if you (or another program product) make use of user segments in the monitor record.
- IMACICDA - Controls sequence of decoding user segments in the CICS Monitor record. Controls which of the following members are invoked and in what order.
- IMACICDL - DL/I counters.
- IMACICDU - User counters.
- IMACICDB - Optional DBCTL counters in CICS 3.1 and later.
- IMACICOB - Optional OMEGAMON DB2 data segment.
- IMACICOC - Optional OMEGAMON basic data segment.
- IMACICOL - Optional OMEGAMON DL/I data segment.
- IMACICUS - Additional User Data decoding and Hogan data decoding.

MXG Documentation member: ADOC110

SAS IT Resource Management customization:

XCITRAN

The variables with each of the optional segments mentioned above are set in table XCITRAN as KEPT=NO. Therefore, if you want to keep any of these variables, you will need to set the variables of interest to KEPT=YES. The variables that are associated with each segment are listed below.

Table 10.2 IMACICDB - Optional DBCTL Counters in CICS 3.1 and Later

STTBFWT	STATGN	STTTLOC
STTCTM1	STATGNP	STTTMIO
STTDATN	STATGU	STTTOTC
STTDATS	STTINTC	STTTSDQ
STTDBIO	STTISRT	STTUENQ
STTDECL	STTMSCL	STTUOWC
STTDERD	STTNPSB	STTUPDQ
STTDLET	STTOVFN	STTUSSN

STTEXDQ	STTPOOL	STTWEXQ
STATGHN	STTREPL	STTWTEQ
STTGHNP	STTSCHT	STTWUEQ
STATGHU	STTTENQ	

Table 10.3 IMACICDL - DL/I Counters

DBALLCN	DLIGN	IWAITTM
DBALLTM	DLIGNP	PSACTCN
DLDELET	DLIGU	PSACTTM
DLIGHN	DLNSERT	PSSCHCN
DLIGHNP	DLREPLC	PSCHTM
DLIGHU	IWAITCN	

Table 10.4 IMACICOB - Optional OMEGAMON DB2 Data Segment

OBCLOCN	OBFETTM	OBPRECN
OBCLOTM	OBINSCN	OBPRETM
OBDELCN	OBINSTM	OBSELCN
OBDELTM	OBMISCN	OBSELTM
OBEXICN	OBMISTM	OBUPDCN
OBEXITM	OBOPNCN	OBUPDTM
OBFETCN	OBOPNTM	

Table 10.5 IMACICOC - Optional OMEGAMON Basic Data Segment

OADABRQ	OMFLAG3	OSUPRTM
OADABTM	OMGLAG4	OUMBPTC
ODCOMRQ	OGMTOFF	OUMBUSR
ODCOMTM	OIDMSRQ	OUSRWRK
ODEXFIL	OIDMSTM	OMVRSN
ODEXTYP	OSUPRRQ	

Table 10.6 IMACICOL - Optional OMEGAMON DL/I Data Segment

OIDELCN	OIGHUTM	OISCHCN
OIDELTM	OIINSCN	OISCHTM
OIGHNCN	OIINSTM	OITERCN

OIGHNTM	OIREPCN	OITERTM
OIGHUCN	OIREPTM	

Table 10.7 IMACICUS - Optional Hogan Data

FPSACTN	FPSSCRN	TCTPLD
FPSFUNC	FPSTYPCD	TCBFUNC
FPSOPTN	PLDVERS	

XCICDB2 and XUOW: If you customize the ANALDB2C or ASUMUOW MXG SOURCLIB members to keep more variables from either CICSTRAN or DB2ACCT, you must ensure that these extra variables are marked KEPT=YES in the SAS IT Resource Management data dictionary. To process XUOW, you must also modify the IMACUOW MXG member as follows. Comment out or delete the first definition of the two macros, _NOOBS and _YESOBS, and remove the comment delimiter from the second definition, so that the macros are defined as follows:

```
MACRO _NOOBS    %
MACRO _YESOBS   %
```

XCMASMQ - XCM29CO

Data Source: Boole and Babbage, Inc. CMF

SAS IT Resource Management status: not tested

MXG customization:

IMACCMF - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XCOTROL

Data Source: 4th Dimension Software Inc., Control-D

SAS IT Resource Management status: not tested

MXG customization:

IMACCTLD - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XCOUCKP - XCOUTRM

Data Source: Software AG COM-PLETE

SAS IT Resource Management status: not tested

MXG customization:

IMACCOMP - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XDBACCT - XDBTATS

Data Source: DB2 SMF record types 100 and 101

SAS IT Resource Management status: tested

MXG customization:

IMACKEEP

1 Add the following lines:

```
MACRO _KDB2ACC _XKDB2AC    %
MACRO _KDB2ACP _XKDB2AP    %
MACRO _KDB2ACB _XKDB2AB    %
MACRO _KDB2ACG _XKDB2AG    %
MACRO _KDB2PAT _XKDB2GA    %
MACRO _KDB2PST _XKDB2GS    %
MACRO _KDB2ST0 _XKDB2S0    %
MACRO _KDB2ST1 _XKDB2S1    %
MACRO _KDB2ST2 _XKDB2S2    %
MACRO _KDB2STS _XKDB2SS    %
MACRO _KDB2STB _XKDB2SB    %
MACRO _KDB2STR _XKDB2SR    %
```

You can add them anywhere between the following statements:

```
%INCLUDE SOURCLIB(IMACOLDV);
```

and

```
&MACKEEP ;
```

2 Examine the instructions and code for the _DB2CORR macro that is contained in the VMAC110 member. This macro establishes correlation name and number. If you find it necessary to modify the definition, copy the macro definition to the IMACKEEP member and modify it there. The change you will most likely have to make is to specify the names for your CICS and/or IMS jobs or started tasks.

EXDB2ACB - If you want data to be written to table XDBCCTB (MXG data set DB2ACCTB), then modify this member and remove the comment block around the OUTPUT statement. You can also add IF-THEN style logic to the statement if required. Read the comments therein for further information.

MXG Documentation members: ADOCDB2, ADOCDB2R

SAS IT Resource Management customization: none

XDSNREC

See XTMS

XDMNDSN - XDMNVOL

Data Source: LEGENT Corporation ASTEX SMF record (formerly known as DASDMON)

SAS IT Resource Management status: not tested

MXG customization:

IMACDMON - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XEDSARE - XEDSVRE

Data Source: DFSMS*rmm* (IBM Corporation) SMF records

SAS IT Resource Management status: not tested

MXG customization: IMACEDGS

1 Specify the SMF record number for the DFSMS*rmm* audit record.

2 Specify the SMF record number for the DFSMS*rmm* security record.

MXG Documentation member: none

SAS IT Resource Management customization: none

XEPMVEP - XEPVEPI

Data Source: Candle Corporation EPILOG/MVS record

SAS IT Resource Management status: not tested

MXG customization:

IMACEPMV - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XFOUSMS

Data Source: Information Builders FOCUS multi-session option accounting record

SAS IT Resource Management status: not tested

MXG customization:

IMACFOCU - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XFTP01X - XFTP51X

Data Source: NETVIEW File Transfer Program User SMF Record

SAS IT Resource Management status: not tested

MXG customization:

IMACFTP - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XHIRSAM - XHIRVSB

Data Source: Empact Hipercache user SMF record.

SAS IT Resource Management status: not tested

MXG customization:

IMACHIPR - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XHSDSRF - XHSWVOL

Data Source: IBM DFHSM user SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACHSM - Specify the SMF record number by following the instructions within the member.

EXHSMFST - If you want data to be written to table XHSFSRT (MXG data set HSMFSRTP), then modify this member and remove the comment block around the OUTPUT statement. You can also add IF-THEN style logic to the statement if required. Read the comments therein for further information.

EXHSMWWV - If you want data to be written to table XHSWVOL (MXG data set HSMWWVOL), then modify this member and remove the comment block around the OUTPUT statement. You can also add IF-THEN style logic to the statement if required. Read the comments therein for further information.

MXG Documentation member: ADOCHSM

SAS IT Resource Management customization: none

XICBRGC - XICB2RGD

Data Source: STK Iceberg IXFP SMF records

SAS IT Resource Management status: not tested

MXG customization:

IMACICE - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCICE

SAS IT Resource Management customization: none

XIDSARA - XIDSYPE

Data Source: Computer Associates International, Inc. CA-IDMS Performance Monitor SMF record.

SAS IT Resource Management status: not tested

MXG customization:

IMACIDMS - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XILAT20 - XILVTRT

Data Source: Interlink user SMF records.

SAS IT Resource Management status: not tested

MXG customization:

IMACILKA - Specify the SMF record number for the Interlink SNS/TCPACCESS SMF record.

IMACILKG - Specify the SMF record number for the Interlink SNS/SNA Gateway SMF record.

IMACILKV - Specify the SMF record number for the Interlink SNS/TCPVT SMF record.

MXG Documentation member: none

SAS IT Resource Management customization: none

XIPAC01 - XIPAC05

Data Source: INFOPAC Report Distribution System

SAS IT Resource Management status: not tested

MXG customization:

IMACIPAC - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCIPAC

SAS IT Resource Management customization: none

XJM5FSE - XJM5WTD

Data Source: IBM JES3 Measurement Facility type 84 SMF record

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: none

XJOBS, XPRINT, XSTEPS, XSPNJOB, XNJPURG, XSMFINT

Data Source: MXG consolidation of several SMF job accounting and utilization tables.

SAS IT Resource Management status: tested

MXG customization:

IMACSPIN - Specify the number of processing cycles for which incomplete jobs will be retained in SPIN data sets. The default is zero. See the documentation in IMACSPIN and, more importantly, paragraph (4) of the INSTALL member in your MXG.MXG.SOURCLIB. The latter contains important information regarding the consequences of setting spin count.

Note: In SAS IT Resource Management, SPIN data sets are kept in the COLLECT library of the PDB. △

IMACSHFT - Copy the member into your MXG user library and delete all lines except the length statement for shift and the label statement for shift. If you want to use your own code for calculating SHIFT, this is where it should be placed. You can use the value of DATETIME in the calculation of SHIFT, but DO NOT change the value of DATETIME.

Note: For XPRINT, XSTEPS, and XSPNJOB: Changes to MXG 15.15 and SAS IT Resource Management 2.1 mean that no code changes are necessary to collect these tables. Moreover, any customizations performed as a result of having read these instructions in previous releases must be removed. This includes the DATA step to create a copy of the STEPS file as KEEPSTPS. △

Note: For XSMFINT: Support for the XSMFINT (SMFINTRV) table became available in TS45001 of SAS/CPE. To enable the collection of this data, you must first ensure that your system is capable of cutting the SMF interval accounting records. The z/OS system data set, SYS1.PARMLIB(SMFPRMxx), is where you tell SMF not only what records you want cutting, but also if you want interval accounting enabled. Contact your systems programming team for information regarding this change.

Even with the above change in place, you must also modify the IMACINTV MXG SOURCLIB member. MXG's default is that no TYPE30_V (SMF interval accounting) records are output to the SAS data sets. So, you must copy IMACINTV from MXG.MXG.SOURCLIB into your MXG.USERID.SOURCLIB and remove the comments from around the OUTPUT statement, as detailed in the comments within the member. △

Note: In addition, you can be selective about how many observations/ records are written, by coding IF-THEN type statements. See the comments for details. △

MXG Documentation members: ADOC26J2, ADOC26J3, ADOC30, ADOC6

SAS IT Resource Management customization: none

XLMADOW - XLMUNLD

Data Source: Memorex Tape Library Management System (LMS) Performance Monitor SMF record.

SAS IT Resource Management status: not tested

MXG customization: IMACLMS - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCLMS

SAS IT Resource Management customization: none

XMITAPE

Data Source: LEGENT Multi-Image Manager (MIM) tape statistics SMF record.

SAS IT Resource Management status: not tested

MXG customization:

IMACMIM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XMQMACCT - XMQMSGD

Data Source: IBM Corporation MQSeries (formerly MQM) SMF record.

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: none

XM2LOGO - XM2SINC

Data Source: Computer Corporation of America MODEL 204 LOGOFF and SINCE SMF records.

SAS IT Resource Management status: not tested

MXG customization:

IMACM204

1. Specify the SMF record numbers for the MODEL 204 LOGOFF and SINCE SMF records.
2. Specify the release number of your MODEL 204 system.

MXG Documentation member: none

SAS IT Resource Management customization: none

XNAENTA - XNAVOGN

Data Source: Candle Network Accounting Facility (NAF) SMF record. NAF records are produced by Candle products CL/SUPERSESSION, CL/GATEWAY, and VTAMPLUS.

SAS IT Resource Management status: not tested

MXG customization:

IMACACCT - Creates account code variables. Follow the instructions within the member header.

IMACNAF - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XNDMAE - XNDMWO

Data Source: Sterling Software's Network Data Mover (NDM) SMF record.

SAS IT Resource Management status: not tested

MXG customization:

IMACNDM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XNPANAT - XNPWDWD

Data Source: IBM Network Performance Management (NPM) type 28 SMF record.

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: ADOC28

SAS IT Resource Management customization: none

XNSACCT - XNSVTAM

Data Source: LEGENT NetSpy and LanSpy SMF record

SAS IT Resource Management status: NetSpy has been tested. LanSpy has not.

MXG customization:

IMACNSPY - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XNTPACT

Data Source: Software AG NATURAL Process Accounting SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACNATP - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XOCADA - XOCVVSA

Data Source: Candle OMEGAMON for CICS SMF record.

SAS IT Resource Management status: not tested.

MXG customization:

IMACOMCI - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XOMMSDE - XOMMSJO

Data Source: Candle OMEGAMON II for SMS SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACOMSM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XORACLE

Data Source: ORACLE dbms SMF record.

SAS IT Resource Management status: not tested.

MXG customization:

IMACORAC - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XOVEXCE - XOVTVR

Data Source: Candle OMEGAMON for VTAM (R) SMF record.

SAS IT Resource Management status: not tested.

MXG customization:

IMACOMVT - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XPRINT

See XJOBS.

XPRSHLP - XPRSXIO

Data Source: Softworks Performance Solutions SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACPRFS - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCPRFS

SAS IT Resource Management customization: none

XRMFINT

Data Source: Consolidation of several RMF tables into a single table

SAS IT Resource Management status: tested

MXG customization:

IMACWORK: defines which performance groups (or service classes for those in goal mode) make up which workloads at your site. Copy IMACWORK from MXG.MXG.SOURCLIB to MXG.USERID.SOURCLIB. Edit this copy of IMACWORK according to the instructions (comments) within it. It also contains important notes regarding the use of service class with reporting class (in goal mode) and control performance groups with report performance groups (not in goal mode). A good job of customizing this member ensures useful information in the XRMFINT table.

Instead of copying the EXRMFINT member from MXG.MXG.SOURCLIB to MXG.USERID.SOURCLIB and changing the labels in the copy, change the OTH0-OTH9 "Other" workloads labels for variables in the SAS IT Resource Management table XRMFINT.

MXG Documentation member: See the ADOC7*x* members that are associated with the underlying RMF data sets.

SAS IT Resource Management customization: none

XSASRQU

Data Source: Legent's Sysout Archival and Retrieval (SAR) SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACSAR - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCSAR

SAS IT Resource Management customization: none

XSPSCED - XSPSEXT

Data Source: Amdahl Storage Product Management Services (SPMS) SMF record for the AMDAHL 6100 DASD controller

SAS IT Resource Management status: not tested

MXG customization:

IMACSPMS - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCSPMS

SAS IT Resource Management customization: none

XSTBLOS - XSTVARY

Data Source: STC 4400 Tape Silo SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACSTC - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XSTEPS

See XJOBS.

XSXCNOF - XSXSART

Data Source: LEGENT STX (multi-session product) SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACSTX - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XSYCSOR

Data Source: SyncSort SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACSYNC - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTEEAPP - XTEETER

Data Source: Computer Associates Teleview SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACTELE - Specify the SMF record number for the Teleview SMF record.

EXTPXINT - Insert the following line before the OUTPUT statement:

```
IF DIFFTPX=1 THEN STARTIME=SMFTIME-DELTATM;
```

MXG Documentation member: none

SAS IT Resource Management customization: none

XTPECRD - XTPTART

Data Source: LEGENT TPX (multi-session product) SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACTPX - Specify the SMF record number by following the instructions within the member.

EXTPXINT - Insert the following line before the OUTPUT statement:

```
IF DIFFTPX=1 THEN STARTIME=SMFTIME-DELTATM;
```

MXG Documentation member: none

SAS IT Resource Management customization: none

XTSCALL - XTSSYST

Data Source: LEGENT TSO/MON SMF records

SAS IT Resource Management status: tested

MXG customization:

IMACTSOM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYDLMN

Data Source: Candle Corporation's Deltamon SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACDLMN - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYHMFB - XTYHMF9

Data Source: Network System's Host Monitoring Facility (HMF)

SAS IT Resource Management status: not tested

MXG customization:

IMACHMF - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYIAM

Data Source: Innovation Processing's IAM SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACIAM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYMDF

Data Source: AMDAHL MDFTRACK SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACMDF - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYRMDS

Data Source: IBM Report Management and Distribution System SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACRMDS

1. Specify the SMF record number by following the instructions within the member.
2. Specify the RMDS version number in macro _RMDSVER, if necessary, by coding

```
MACRO _RMDSVER 3 %
```

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYSASU

Data Source: SAS SMF record that can be optionally written at PROC and DATA step boundaries.

SAS IT Resource Management status: not tested

MXG customization:

IMACSASU - Specify the SAS 5 and SAS 6 SMF record numbers.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYSIM

Data Source: SIMWARE SIM/XFER VTAM SMF record.

SAS IT Resource Management status: not tested

MXG customization:

IMACSIM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCSIM

SAS IT Resource Management customization: none

XTPALOC, XTYMNTS, XTYTALO, XTYTMNT, XTYTSWP

Data Source: MXG Tape Mount Monitor SMF record

SAS IT Resource Management status: not tested

MXG customization:

ASMTMNT - See this member for installation and usage instructions.

IMACTMNT - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCTMNT

SAS IT Resource Management customization: none

XTYTAO

Data Source: Fischer Totally Automated Office (TAO) SMF record

SAS IT Resource Management status: not tested.

MXG customization:

IMACTAO - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYTCPA - XTYTCPT

Data Source: IBM TCP/IP SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACTCP

- For TCP/IP releases earlier than 2.1.1 and for 2.1.1 without APAR PN40511, specify the SMF record numbers for TELNET and FTP.
- If you are running TCP/IP 2.1.1 with APAR PN40511 or later, then do not modify IMACTCP. At the newer levels, TCP/IP will always use SMF record number 118.

MXG Documentation member: ADOCTCP

SAS IT Resource Management customization: none

XTYTPMF - XTYTPMV

Data Source: MVS Solutions Inc. Thruput Manager SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACTPM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTYX37

Data Source: EMPACT Software STOPX37 SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACX37 - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: none

XTY23

Data Source: SMF interval statistics

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: ADOC23

SAS IT Resource Management customization: none

XTY24

Data Source: JES2 Spool Offload SMF record

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: ADOC24

SAS IT Resource Management customization: none

XTY50

Data Source: VTAM Tuning Statistics SMF record

SAS IT Resource Management status: tested

MXG customization: none

MXG Documentation member: ADOC50

SAS IT Resource Management customization: Set the retention period to zero in all PDB levels except DETAIL. This is interval data, but the data cannot be reduced because there is no duration value for the interval.

XTY7204

Data Source: IBM RMF subtype 04 SMF record

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: none

XTY78PA - XTY78SP

Data Source: IBM RMF monitor job SMF records

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: none

XTY89

Data Source: IBM Measured Usage Interval SMF record

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: none

XTY94

Data Source: IBM Tape Library Data Server SMF record

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: none

XWSACCT - XWSFERD

Data Source: RSD's WSF SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACWSF - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCWSF

SAS IT Resource Management customization: none

XXCMDAT

Data Source: XCOM File Transfer Facility SMF record

SAS IT Resource Management status: not tested

MXG customization:

IMACXCOM - Specify the SMF record number by following the instructions within the member.

MXG Documentation member: ADOCXCOM

SAS IT Resource Management customization: none

Non-SMF Data Sources - z/OS Operating Environment

VAPLEDT - VUSETRE

Data Source: VM Monitor data

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: ADOCVMXA

SAS IT Resource Management customization: Tailor member CMVMLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your VMMON PDB.

XDCBKUP - XDCVOLS

Data Source: DCOLLECT data

SAS IT Resource Management status: tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: Tailor member CMDCOLLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce DCOLLECT log data into your PDB.

XERPCRW - XERPTIM

Data Source: EREP data

SAS IT Resource Management status: tested

MXG customization: none

MXG Documentation member: ADOCEREP

SAS IT Resource Management customization: Tailor member CMEREPLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the EREP data into your PDB.

XIMBMPS - XIMTRAN

Data Source: IMS log records.

SAS IT Resource Management status: tested

MXG customization:

ASMIMSLG - Tailor and run ASMIMSLG following the instructions within the member.

IMACKEEP - Add the following lines:

```
MACRO _LIMSBMP IMSDATA.BMPS     %
MACRO _LSAPIBA IMSDATA.SAPIMSBA %
MACRO _LSAPION IMSDATA.SAPIMSON %
MACRO _LSAPISP IMSDATA.SAPIMSSP %
```

You can add them anywhere between the following statements:

```
%INCLUDE SOURCLIB(IMACOLDV);
```

and

```
&MACKEEP  ;
```

MXG Documentation member: none

SAS IT Resource Management customization: Tailor member CMIMSLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce IMS log data into your PDB.

XIMDBDS - XIMPROG

Data Source: Boole and Babbage IMS Measurement Facility (IMF) (formerly Control/IMS)

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization: Tailor member CMIMFLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce IMF log data into your PDB.

XQAAPPN - XQAX25

Data Source: OS/400 performance/accounting data

SAS IT Resource Management status: not tested

MXG customization:

IMACQAPM - Add the variable DTETIM to the _KQAPCON macro.

EXQAPCON - Insert the following line before the OUTPUT statement:

```
RETAIN DTETIM "&SYSDATE:&SYSTIME"DT;
```

MXG Documentation member: ADOCQAPM

SAS IT Resource Management customization: Tailor member CMAS4LD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your AS400 PDB.

XTC2CMX - XTC2XMC

Data Source: TMON for CICS/ESA 2.0 from Landmark

SAS IT Resource Management status: not tested

MXG customization:

EXITMON6 - If you are using compressed TMON data, then tailor and run EXITMON6 following the instructions within EXITMON6.

IMACMONI - Tailor IMACMONI following the instructions within the member.

MXG Documentation member: none

SAS IT Resource Management customization: Tailor member CMTMC2LD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your PDB.

XTCCDBD - XTCUSER

Data Source: TMON/CICS from Landmark

SAS IT Resource Management status: not tested

MXG Documentation member: none

SAS IT Resource Management customization:

For TMON/CICS versions 8.0, 9.0, 1.0, and 1.1, tailor member CMTMCLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your PDB.

For TMON/CICS ESA version 1.3 and later, tailor member CMTMCCLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your PDB.

MXG customization:

(for TMON/CICS versions 8.0, 9.0, 1.0, and 1.1):

EXITMON6 - If you are using compressed TMON data, then tailor and run EXITMON6 following the instructions within EXITMON6.

IMACMONI - Tailor IMACMONI following the instructions within the member.

(for TMON/CICS ESA version 1.3 and later):

EXITMON6 - If you are using compressed TMON data, then tailor and run EXITMON6 following the instructions within EXITMON6.

IMACTMON - Tailor IMACTMON following the instructions within the member.

XTMDBA - XTMDDW

Data Source: Landmark's The Monitor for DB2

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: none

SAS IT Resource Management customization:

For TMON/DB2 version 1, tailor member CMTMDBLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your TMON/DB2 (V1) PDB.

For TMON/DB2 version 2 and later, tailor member CMTMD2LD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your TMON/DB2 (V2+) PDB.

XTMS and XDSNREC

Data Source: Computer Associates' Tape Management System

SAS IT Resource Management status: not tested

MXG customization:

IMACTMS5 - Add the variable DATETIME to the _KTMSTMS and _KTMSDSN macros.

EXTMSDSN - Insert the following line before the OUTPUT statement:

```
RETAIN DATETIME "&SYSDATE:&SYSTIME"DT;
```

MXG Documentation member: ADOCTMS5

SAS IT Resource Management customization: Tailor member CMTMSLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce the log data into your TMS PDB.

XTPFDA - XTPFTT

Data Source: TPF data

SAS IT Resource Management status: not tested

MXG customization: none

MXG Documentation member: ADOCTPF

SAS IT Resource Management customization: Tailor member CMTPFLD in your *prefix*.CPMISC data set. This member is the daily job that you will run to process and reduce TPF data into your job.

CHAPTER

11

Generic Collector Facility

Generic Collector Facility: Overview

Using the Generic Collector Facility

The Generic Collector Facility enables you to process data from data collectors or data sources for which SAS IT Resource Management does not supply table definitions.

- "Appendix 13: Character-Delimited Support in GENERATE SOURCE" on page 196.

Getting Started with the Generic Collector Facility

Types of Data Collector Support

Data collectors write logs. SAS IT Resource Management provides two techniques for processing the data from those logs into tables in your PDB. The technique that you choose depends on whether your data is in a user-defined table or in a table whose definition is supplied with SAS IT Resource Management.

The technique that is described in this document works with user-defined tables.

The other technique works with tables whose definition is supplied with SAS IT Resource Management. For a short description of that technique, see "Appendix 11: Support by Means of Supplied Table Definitions" on page 193.

Support by Means of the Generic Collector Facility

SAS IT Resource Management supports many data collectors or data sources by means of the Generic Collector Facility. One example of such a data source is a log file that is generated by a user-written application. For each of these data collectors or data sources, SAS IT Resource Management provides

- support for running code that provides staged data (as staging data sets or DATA step views)
- support for creating one or more user-generated table definitions and support for creating the table(s) in your PDB
- a macro that invokes SAS IT Resource Management
- a macro that processes the staged data into the tables in your PDB
- a macro that reduces the data in the PDB
- macros that report on the data in the PDB.

Some of these components are unique for each of the data collector or data sources and some are the same for all data collectors or data sources.

You can use the Generic Collector Facility with data from almost any time-based data source.

This document describes how to use the Generic Collector Facility. It also includes an example of using the Generic Collector Facility for fax data. The example includes

- fax data
- SAS code that stages the fax data
- code that generates fax table and variable definitions
- a fax table definition and its variable definitions that were created interactively or by running the SAS IT Resource Management data dictionary utility against the staging data set, and updates to these definitions
- a daily job that invokes SAS IT Resource Management, runs the SAS code to provide staged fax data, processes the staged fax data into the table in the PDB, and reduces the fax data. (For examples of macros that report on the data in the PDB, see the other Getting Started documents and/or the documentation for your release of SAS.)

Before proceeding, you should do the following:

- Locate the documentation for the data collector or data source that logs the data. Your collector software probably has the same name and might be the same

package as your logging software. The documentation should be supplied by the vendor of the data collector or data source.

- Read "Overview of Setup Case 3" on page 81 (setup case 3 is for character-delimited data) or "Overview of Setup Case 4" on page 99 (setup case 4 is for data that is not character-delimited).
- Obtain write access to an existing PDB that you want to use for this data or create a new PDB and give yourself write access. (You can use the default space estimate for now.)
- You need to be running SAS IT Resource Management 2.3 or later.

Example Used in This Document: Data from a Fax Log

The best way to understand how to create a table for use with the Generic Collector Facility is to walk through an actual example, as if you were doing it for your site. Although each collector is unique, the same basic steps are involved.

For the fax example, consider an environment in which there are several fax machines, each of which records its activity in an ASCII text log file. Your task is to take the combined daily log for all of the fax machines and load the log's data into the PDB. Assume that the fax log for each machine consists of a single-line message for each fax send attempt and a single-line message for each fax receive attempt. The fields in the log file are separated by blanks, and the fields also start in specific columns, as in the following table.

Table 11.1 Layout of the Fax Log

Columns	Description	Name
1-8	Name of fax machine	MACHINE
10	Send (S) or Receive (R) attempt	TYPE
12-14	Country code (blank unless international)	COUNTRY
16-18	Area code	ACODE
20-27	Phone number (xxx-yyyy)	PHNUM
29-30	Pages sent or received (99 page max)	PAGES
32-39	Date fax send/receive began (MM/DD/YY)	DATE
41-48	Time fax send/receive began (HH:MM:SS)	TIME
50-51	Pages re-sent (99 max)	RESENT
53-60	Total (elapsed) time of send/receive (HH:MM:SS)	CONNECT
62	Final Status Code:	STATUS
	Aborted Due to Errors (A)	
	Receiver Was Busy (B)	
	Completed Successfully (C)	

Note: The **Name** column in this table is simply a list of the variable names that will be used later in examples. △

For an example of the log file, see "Appendix 1: Fax: Sample Log" on page 176.

Generic Collector Facility: The Steps

The Steps

The creation of a table for use with the Generic Collector Facility requires the following four basic steps, which are described in more detail after this overview.

1 *Staging the Data*

In this step you prepare the raw data. Typically, preparation involves writing SAS DATA step code to stage the data (as a SAS data set or SAS DATA step view). By doing the staging first, you can use the staged data as a model for building the table and variable definitions.

2 *Creating the Table and Variable Definitions*

One component of a PDB is its data dictionary. The data dictionary contains table and variable definitions, also known as metadata. The metadata for the data source is one or more table definitions and the variable definitions that are associated with the table definition(s). There are several ways to define a table and variables to the PDB. Each way is described.

3 *Customizing (Updating) the Table and Variable Definitions*

Depending on how you create the table and variable definitions, you might find it necessary to apply additional specifications. This is usually the case when staged data is used as a model for building the table and variable definitions. Although tools are provided with SAS IT Resource Management to aid in constructing table and variable definitions, the algorithm that is used will not always yield exactly what you want. Therefore, you might want to customize such things as the BY and CLASS variables used, the default age limits for each level of the PDB, what interpretation (type of data) is used for each variable, and what statistics are kept.

These customizations can be performed as a series of update statements that follow the statements that you generated in the previous step.

4 *Setting Up for Production Use*

After you define the table and variables to the PDB, you set up the process-and-reduce job to populate the table with data.

You might find these topics useful:

Step 1 - Staging the Data

Staging the data means to transform the raw log file into a SAS data set or view, from which data can then be loaded into the PDB. In most cases, the creation of the data set or view is easily accomplished by a simple SAS DATA step. Usually, only rudimentary SAS DATA step programming skills are required. For more information about managing the staging code, see "What to Do with the Staging Code (Optional)" on page 171. For an example, see "Appendix 2: Fax: Sample Code for Staging the Data" on page 177.

Note: In some cases, you are not required to stage the data. In these cases, you must be using character-delimited data and specific formatting criteria. For more information about this kind of data, see "Appendix 9: Using Character-Delimited Data" on page 190. △

Before writing the code to stage the data, it is important to understand the raw data, as well as general table-specific and variable-specific concepts and requirements.

Analyzing the Raw Data

You should have a basic understanding of what each field in the raw data means. For example, you should be able to answer these questions:

- Does the information break down into distinct types, such as incoming and outgoing, or system and user?
- What information exists for providing a datetime stamp?
- Does each data item report about a discrete time interval or about an asynchronous (event) activity?
- Can you identify any metrics in the data and how they are to be interpreted?
- Are there any useful metrics that could be calculated from the raw data?
- Can you identify what fields would be useful for categorizing and grouping the data for reporting purposes?

In the case of the data in Table 11.1 on page 160, all of these questions can be answered. (To view the data, see "Appendix 1: Fax: Sample Log" on page 176.)

- The information breaks down neatly into faxes that are sent and faxes that are received.
- The fields DATE and TIME provide a means of creating a datetime stamp called *TIMESTMP*.
- Because each record corresponds to the sending or receiving of a fax, the records represent events rather than intervals.
- We see that we have counts of pages sent and received, as well as pages re-sent. Additionally, we have data for the elapsed time of each fax, as well as the fax's final status.
- We can use the count of pages that were re-sent and divide it by the total number of pages to arrive at a percentage of efficiency.

- Last, we can categorize the data by MACHINE, TYPE, COUNTRY CODE, AREA CODE, and TIMESTMP.

Table-Specific Considerations

SAS IT Resource Management organizes data into tables within the PDB. Typically, each table corresponds to a specific type of record from a data collector or data source.

- *DATETIME Variable*

  Each table must have a variable called *DATETIME*, whose value is the datetime of the record.

  - For type Interval tables, DATETIME is typically set to the start of the interval, for ease of reporting and for convenience when merging multiple data sources.
  - For type Even tables, DATETIME is usually the datetime stamp of the event. In our example, the TIMESTMP that we create from the DATE and TIME fields will be used as the DATETIME variable.

  Although it is easier in the staged data to name the datetime stamp DATETIME in order to avoid confusion, you might find a situation in which you are forced to use existing names (for example, if your data already exists in SAS data sets to which you have only read access). Therefore, the variable used for DATETIME is not required to be called *DATETIME* in the staged data.

- *DURATION Variable*

  Additionally, type Interval tables must have a variable called *DURATION* and the variable must contain the elapsed time of the intervals that are being recorded. Like DATETIME, this variable is not required to be called DURATION in the staged data.

  In the situation where the log does not already contain values for DURATION, the values can be calculated by using the DIF() function or similar techniques on the datetime stamps of adjacent records.

- *Attributes*

  Each table definition consists of attribute information about the table itself and about the type of record to which it corresponds. For instance, the table definition specifies whether the table is type Interval or type Event.

  Table attribute information includes the descriptive label for the table, how long to keep data (called the *age limits*) at each of the PDB levels, and how data is to be sorted (at the detail level) and classified (at the day, week, month, and year levels).

  Another important table attribute is the *external name*, which is the name of the staged data set or view that corresponds to the table. A table name in a PDB is limited to seven characters, but the table's external name can contain eight characters. To prevent name collisions with supplied table definitions, user-defined PDB (internal) table names should begin with the letter *U*.

Variable-Specific Considerations

The variable definitions that are associated with a table definition contain information that corresponds to the fields in the record.

- *Attributes*

  Variables in SAS IT Resource Management tables also have attributes. For example, the variable definition specifies whether the variable's data type is *Character* or *Numeric* and specifies a descriptive label.

Also, each variable is assigned a specific interpretation type. The interpretation type fine-tunes the data type and determines default characteristics for several other attributes, such as how a variable is to be displayed, what summary statistics to collect for it, and how these summary statistics are to be calculated.

- For more information about interpretation types, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation and "Appendix 8: Algorithm Used by GENERATE SOURCE" on page 188.
- It is important to understand the interpretation types as implemented by SAS IT Resource Management so that you can check the default interpretation types for the variables in your table and edit (or update) the interpretation types, if necessary. Also, it is important to understand the summary statistics that are available within SAS IT Resource Management so that you can check, for each variable, which summary statistics are to be calculated by default and modify (update) the summary statistics specification, if necessary.

Another important variable attribute is the *external name*, which is the name of the corresponding field in the staged data. Variable names in SAS IT Resource Management tables are limited to 7 characters, but a variable's *external name* can be up to 32 characters.

Step 2 - Creating the Table and Variable Definitions

Before data can be processed into the PDB, appropriate table and variable definitions must be generated that define the data source to the PDB and that create one or more tables in the PDB for storing the data.

Typically, you will use the %CPDDUTL utility macro to run a GENERATE SOURCE control statement against the model data set or view in order to automatically generate CREATE TABLE and CREATE VARIABLE control statements. In Step 3, you will review the control statements, write UPDATE statements to change the defaults, if necessary, and apply the CREATE and UPDATE statements to the PDB to create the table and its variables into which you can process the staged data.

Note: In some cases, you are not required to stage the data. In those cases, you have no model data set or view to run against. However, you can use the Character Delimited Collector support function in GENERATE SOURCE. For more information, see "Appendix 13: Character-Delimited Support in GENERATE SOURCE" on page 196. △

The automatic technique is strongly recommended for most work. In this technique, the DATASET= parameter on the GENERATE SOURCE statement points to the model data set (in this case, the model of the staged data). Other parameters of the GENERATE SOURCE control statement provide information such as

NAME=	the name of the table we want to create
TYPE=	whether the table is type Interval or type Event
ENTRYNAME=	entry in SAS catalog .SOURCE that points to where the generated statements are to be stored
DATETIME=	what variable to use for the DATETIME variable
DURATION=	what variable to use for the DURATION variable

For more information about the GENERATE SOURCE control statement, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation.

For the fax example, see the SAS code in "Appendix 3: Fax: Sample Code for Generating the Definitions" on page 179. This code runs after the code (described in step

1) that created the staging data set and creates the PDB table and variable definitions shown in "Appendix 4: Fax: Sample Table and Variable Definitions" on page 180.

Step 3 - Customizing the Table and Variable Definitions

After GENERATE SOURCE has built the table and variable definitions, you should review the generated definitions, because it is quite likely that further customization is required. The recommended procedure for updating the table and variable definitions is

1. Review the generated control statements, and build a set of update control statements that correct any undesired table and/or variable characteristics.
2. Use the %CPDDUTL macro to apply the generated table and variable definitions to the PDB (and thus create the table).
3. Use the %CPDDUTL macro to apply the update statements to the table and variable definitions in the PDB (and thus edit the table).

Review the Generated Control Statements and Build the Update Control Statements

It is important that you review the generated control statements in order to check that the following characteristics are set appropriately for your data source:

Variable's Interpretation Type

As previously discussed, the interpretation type that is chosen for a variable is critical in guiding the summarization and display of the data. You should review the generated control statements, paying attention to the value of the INTERPRET= parameter on each of the CREATE VARIABLE control statements. You might find that the algorithm used by GENERATE SOURCE did not specify the interpretation type that is most appropriate. In such a case, you should use the UPDATE VARIABLE control statement to specify the correct value for the INTERPRET= parameter. For information about how to use this parameter, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

For more information about the GENERATE SOURCE control statement, refer to the %CPDDUTL GENERATE SOURCE control statement in "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation.

Variable's Format

The variable format (specified by the FORMAT= parameter of the CREATE VARIABLE control statement) specifies the SAS display format to be used when displaying the data. By default, this is set according to the interpretation type selected by GENERATE SOURCE. If you change the interpretation type, you might also want to change the display format. In such a case, you should use the UPDATE VARIABLE control statement to specify the appropriate value for the FORMAT= parameter. For more information about SAS display formats, see the SAS Language Reference documentation for your release of SAS.

Statistics

Each variable definition includes a list of requested statistics. You should review this list to make sure that any statistics that you want will be calculated. Likewise, if statistics have been requested that you do not need, you can modify the list to specify that the statistics not be calculated.

If no summary statistics are selected for a given reduction level of the PDB (day, week, month, or year), then no data will be reduced for the table at that reduction level.

KEPT=YES/NO

By default, all variables are marked with a KEPT status of YES so that the PDB detail level keeps data for the variable. However, you might not have any need for the data that supports some of the variables in the staged data. In such cases, you can save disk space by marking the variable(s) as KEPT=NO instead of the default KEPT=YES. This allows the *definition* of the variable to be stored in the data dictionary, without storing the *data* for this particular variable.

Derived Variables and Formula Variables

In the fax example you create some new variables that do not exist in the raw data. Such data variables are called derived variables, in that they are derived from existing variables. Derived variables store real data in the PDB, just as normal variables do. Therefore, you can use a derived variable as a BY or CLASS variable. You can also request that statistics be calculated automatically for a derived variable.

In contrast, a formula variable can be defined in a table at one or more levels of the PDB. SAS IT Resource Management uses views to access the data in each table. Formula variables are included in the view definition and thus are calculated dynamically. This means that formula variables cannot be used as BY or CLASS variables, nor can statistics be calculated for them. Formula variables are, however, sometimes preferable to derived variables because formula variables do not require any disk space for data storage.

BY Variables

The BY variables list designates the sort order for detail-level data for this table. The BY variables list can also be used for detecting data with duplicate values (of the variables in the set of BY variables). For more information about duplicate-data checking, see "Appendix 4: Working with Duplicate Data" in the IT Service Version macro reference documentation.

The BY variables list is also important for reporting purposes. The generated source control statements use the sort order of the model data set (only if SAS knows about the sort order because SAS sorted it) to determine the BY variables list for the detail level. The list can be specified in the UPDATE TABLE control statement. If the model data set is not sorted (or, to be more precise, not sorted as known by SAS), the default BY variables list is *DATETIME HOUR SHIFT*.

To change the BY variables list, use the UPDATE TABLE control statement. Be sure that DATETIME is in the updated BY variables list.

Class Variables

The CLASS variables lists are used

- to designate the sort order for data in the summary levels of this table (week, month, day, and year) in the PDB
- to control the granularity of the summary levels
- to accumulate statistics for distinct classes of data, based upon the unique values of the CLASS variables.

The CLASS variables lists are also useful for reporting purposes. By default, the generated control statements specify, for each summary level, a CLASS variables list that is identical to the sort order (as known to SAS) of the model data set. If the model data set is not sorted (or, to be more precise, not sorted using a method that is known by SAS), the default CLASS variables lists is *DATETIME HOUR SHIFT*. The CLASS variables lists are not required to match each other, nor are they required to match the BY list for the detail level; however, the reduction step (%CPREDUCE) runs fastest when the BY variables list and all CLASS variables lists are the same.

To change the CLASS variables lists, use the UPDATE TABLE control statement. Be sure that DATETIME is in each updated CLASS variables list.

Age Limits

By default, when SAS IT Resource Management creates a table, it uses the age limits in the following table to determine how much data to store at each PDB level.

Table 11.2 Default PDB Age Limits

PDB Level	Default Age Limit of Data to Keep
detail	10 days
day	45 days
week	15 weeks
month	18 months
year	5 years

If you prefer to change the age limit for any or all of the levels, use the UPDATE TABLE control statement with the AGELIMIT= parameter to specify your preferred age limits. For an example of some updates for the generated definitions in the fax example, see “Appendix 5: Fax: Sample Code for Updating the Definitions” on page 185.

For information about setting age limits for your data, see “Specifying the Age Limit for a Level in a Table” on page 346.

For more information about using %CPDDUTL to store and applying control statements, see “Appendix 3: Fax: Sample Code for Generating the Definitions” on page 179. You can also refer to “Chapter 4: Data Dictionary Macro and Control Statements” in the SAS IT Resource Management macro reference documentation.

Step 4 - Setting Up for Production Use

Prior to this step, you have run, within the same SAS session, the code from step 1 and the control statements from steps 2 and 3. The PDB now has one or more "empty" tables (without data but with descriptors) for the data from this data source. You can now process data through the staging data sets or views to the table(s) in the PDB and, after that, you can reduce the data in the table(s).

Setting Up the Batch Job

You probably already have a batch job that processes and reduces data each day. This job can also run report definitions. This production job (which typically runs each night) probably has at least the following three steps:

1. The %CPSTART macro is invoked, which starts the SAS IT Resource Management server software. This macro specifies the active PDB and also requests write access to that PDB.
2. The %CPPROCES macro is invoked to process the new data into the detail level of the PDB. In time, this fully populates the detail level of the tables involved.
3. The %CPREDUCE macro is invoked to reduce the new data. In time, this fully populates the day, week, month, and year levels of the tables involved.

For more information about setting up the daily production job, see “Overview of Setup Case 3” on page 81 (if your data is character delimited) or “Overview of Setup Case 4” on page 99 (if your data is not character delimited).

The daily production job should be modified as follows:

- The staging code should run after the last %CMPROCES, %CPPROCES, %CSPROCES, or %CWPROCES macro that currently is in the daily production job. (For information about this, see “Appendix 2: Fax: Sample Code for Staging the Data” on page 177.)
- Following that code, the %CPPROCES macro should be invoked to process the staged data into the detail level of the PDB. (For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation
- You do not need to add another invocation of the %CPREDUCE macro. The existing invocation of %CPREDUCE can be used to reduce data for all tables in the PDB.

Alternatively, you can set up a separate daily production job for the new data source.

For an example of a separate daily production job for the fax data, see “Appendix 6: Fax: Sample Daily Process-and-Reduce Job” on page 185.

Adding Report Definitions

In addition to the report definitions that are already in the daily production job, you would typically add to the daily job some report definitions for reporting on the fax data. You would create new report definitions or modify existing report definitions to generate the reports that you want.

Because creation and modification of report definitions is the same whether or not the Generic Collector Facility is used, this document does not discuss report definitions. For more about report definitions, refer to the definitions that are already in your daily production job. For additional information, see “Using the Manage Report Definitions Tool” on page 468.

Generic Collector Facility: Additional Information

Other Considerations

Verifying That the Logging and Collector Software Is Installed Appropriately

Before you start your data collection software, ensure that the installation is correct for your operating system.

z/OS

- *Preferably, use one host for collecting data.*

  If more than one host logs data, then typically, tapes of the logs are brought to a single host for consolidation, whether in SMF-style records or not. Similarly, much of the data is preprocessed by MXG into a single database of staging data, whether the data was in SMF-style records or not. SAS IT Resource Management treats all data that MXG puts in this data base as if the data had been written on SMF-style records and collected by SMF.
- *If you have massive amounts of data, for space or time reasons, you might not be able to collect the data on one host.*

If you have data on CICS or DB2 transactions, on multiple hosts or even a single host, you might not want to collect it (for space or time reasons). In that case, you can process the data directly from the logs to the summary levels of one or more PDBs by using views.

You can also reduce the data directly from the logs to the summary levels of one or more PDBs. This is described in the %CMEXTDET section of the macro reference documentation.

UNIX

- *Preferably, do not use your workstation as the collector host.*

  Typically, the collector host is a different machine from your workstation for performance reasons.
- *If the collector host is a different machine from your workstation, verify that*
  - the collector host is network-accessible to your workstation
  - X Windows software is installed and running on your workstation
  - the display is redirected from the collector host to your workstation.

Windows

- *Preferably, do not use your workstation as the collector host.*

  Typically, the collector host is a different machine from your workstation for performance reasons.
- *If the collector host is a different machine from your workstation, verify that the collector host is network-accessible to your workstation.*

Installing Table Definitions in the Master Data Dictionary (Optional)

Note: The *Package Collector* enables the collector-support developer to package all the entities for a data collector or data source and install them into PGMLIB or other locations, so that the functionality can be shared or treated as if it were supplied with SAS IT Resource Management. For more information, see "Creating and Installing a Collector Support Package" on page 426. △

Other users at your site might want to use the same table definition(s) in their PDBs. You can use %CPDDUTL to apply an INSTALL TABLE control statement that copies the table and variable definitions from the data dictionary in this PDB into the supplied (master) data dictionary. (For more information about the INSTALL TABLE statement, see "Chapter 4: Data Dictionary Macros and Control Statements" in the SAS IT Resource Management macro reference documentation.) In order for the INSTALL TABLE control statement to work, you must have a SAS libref named *PGMLIBW*. The libref must be established with write access and must point to the same physical location that the PGMLIB libref points to.

For our example, the following SAS statements, which can be submitted from the SAS Program Editor window or in batch, use the %CPCAT macro to store the INSTALL TABLE control statement and then use the %CPDDUTL macro to run the INSTALL TABLE statement. This example assumes that the SAS IT Resource Management server software is in SAS.ITRM.PGMLIB. It also assumes that the active PDB was specified earlier in the job (by a %CPSTART invocation).

Figure 1 Invocation of INSTALL TABLE in Order to Copy Table and Variable Definitions into the Master Data Dictionary

```
libname pgmlibw 'SAS.ITRM.PGMLIB' ;
%cpcat;
```

```
cards4;
   INSTALL TABLE NAME=UFAXES REPLACE ;
;;;;
 /* ;;;; must begin in column 1 to terminate the
input stream */
%cpcat(cat=sasuser.cpddutl.install.source) ;
%cpddutl(entrynam=sasuser.cpddutl.install.source);
```

Note: When you install a new version of the SAS IT Resource Management server software, the PGMLIB library is replaced. Thus, after you have installed a new version of SAS IT Resource Management you will need to reinstall your table definitions, as shown in Figure 1 above. △

User-Defined Formats (Optional)

Note: The *Package Collector* enables the collector-support developer to package all the entities for a data collector or data source and install them into PGMLIB or other locations, so that the functionality can be shared or treated as if it were supplied with SAS IT Resource Management. For more information see "Creating and Installing a Collector Support Package" on page 426. △

As shown in the example, the staging code that is used by the Generic Collector Facility can be user-written. Therefore, you can also use your own SAS formats.

SAS display formats provide a means to transform the data into meaningful descriptions for reporting purposes. In our example, we could have defined a SAS display format to be used with the fax machine name and have the format identify the office location of the machine. Because this SAS display format would be user-specific and not shipped with SAS IT Resource Management, it would be the user's responsibility to maintain the format.

SAS IT Resource Management finds SAS display formats by searching first in the PDB's catalog DICTLIB.CPFMTS and then in the SITELIB's catalog SITELIB.CPFMTS. Therefore, any user-defined SAS display formats should be stored in either SITELIB.CPFMTS (for all users at the site to use) or DICTLIB.CPFMTS (if the format is PDB-specific).

For the fax example, you could implement a SAS display format for the STATUS variable, to display a meaningful description for each value, as in the following table.

Table 11.3 Display Values for the STATUS Variable

Value of STATUS variable	Description to display
A	Aborted Due to Errors
B	Receiver Busy
C	Completed Successfully

To implement this, you would first construct a SAS display format. For the fax example, the following SAS statements, which can be submitted from the SAS Program Editor window or in batch, define the format *$STATUS* and store it in the catalog DICTLIB.CPFMTS. This code must be submitted after %CPSTART has been invoked with update access to the PDB.

Figure 2 Defining the SAS Display Format for the STATUS Variable

```
proc format library=dictlib.cpfmts;
    value $STATUS
        'A' = 'Aborted Due to Errors'
        'B' = 'Receiver Busy'
```

```
        'C' = 'Completed Successfully' ;
 run;
```

Note: For any values other than A, B, or C, the actual value of the STATUS variable will be displayed instead of the description that is given by the format. △

You now update the definition of the variable STATUS to use the newly created format. The following SAS statements use the %CPCAT macro to store the control statements and the %CPDDUTL macro to apply the control statements. This code, which can be submitted from the SAS Program Editor window or in batch, modifies the data dictionary to use the format $STATUS for the variable STATUS in the table UFAXES.

Figure 3 Updating the Data Dictionary to Associate a SAS Display Format as an Attribute of a Variable

```
%cpcat; cards4;
   SET TABLE NAME=UFAXES ;
   UPDATE VARIABLE NAME=STATUS FORMAT=$STATUS ;
;;;;
 /* ;;;; must begin in column 1 to terminate the
input stream */
%cpcat(cat=sasuser.cpddutl.status.source);
%cpddutl(entrynam=sasuser.cpddutl.status.source);
```

For more information about SAS display formats, see the *SAS Language: Reference* documentation for your release of SAS.

What to Do with the Staging Code (Optional)

Note: The *Package Collector* software enables the collector-support developer to package all the entities for a data collector or data source and install them into PGMLIB or other locations, so that the functionality can be shared or treated as if it were supplied with SAS IT Resource Management. For more information, see "Creating and Installing a Collector Support Package" on page 426. △

The staging code is written by the user and must be managed by the user. It is the responsibility of the user to run the staging code before processing the data with the Generic Collector Facility. The user is also responsible for any modifications, changes, and/or fixes to the staging code.

You can manage the staging code by

- using the package collector procedure. For information about this topic, see "Creating and Installing a Collector Support Package" on page 426.
- doing it yourself.

To facilitate managing the code yourself, you might want to create a SAS macro to execute the staging code. This can be accomplished through the following steps:

- Determine what to call the macro. For the fax example, call the macro *FAXES*.
- Create a file called FAXES in an autocall library that you concatenate to your SAS autocall library. (An autocall library contains files that define SAS macros. For more about autocall libraries, see the SAS Companion for your operating environment.)
  - The first line in the FAXES file should be

    ```
    %MACRO FAXES;
    ```

  - Include the staging code as the body of the macro (see the staging sample code in "Appendix 2: Fax: Sample Code for Staging the Data" on page 177).

- The last line should be

```
%MEND FAXES;
```

You can now simply invoke the %FAXES macro to run the staging code.

To process the staged data into the PDB by using the %FAXES macro, submit the following code:

```
%INCLUDE FAXES.SAS;
%FAXES;
RUN;

%CPPROCES(COLLECTR=GENERIC,TOOLNM=SASDS,GENLIB=WORK,
          _RC=CPPRRC);
%PUT CPPROCES return code is &CPPRRC;
```

Note: For more information about submitting SAS programs and code, see "Working with the Interface for Batch Mode" on page 18. △

For more detailed samples of code that uses the %FAXES macro to process staged data into the PDB, see "Appendix 6: Fax: Sample Daily Process-and-Reduce Job" on page 185.

The following topics describe more about handling staging code yourself:

- Include MXG Staging Code in Your Batch Job

  The staging code must be executed in the same batch job as the %CPSTART macro and the %CMPROCES or %CPPROCES macro. The MXG code that stages the data must be executed after the %CPSTART macro and before the %CMPROCES or %CPPROCES macro. The name of the MXG member that you need to include usually begins with the letters TYPE, but this might vary, based on your data collector or data source. For example, if you use IBM DCOLLECT data, then your reference to the MXG staging code would look like this:

  ```
  %INCLUDE SOURCLIB(TYPEDCOL);
  RUN;
  ```

  The RUN statement in this example must be placed immediately after the INCLUDE statement. These statements execute the MXG staging code, including your changes to the IMACDCOL member and to either the IHDRDCOL or the EXDCO members, and write the staging data sets to the WORK library. When the %CMPROCES or %CPPROCES macro executes with the COLLECTR=GENERIC parameter, your data is processed into the detail level of the PDB.

  Note: Remember to specify the GENLIB= parameter on %CxPROCES macro in order to indicate the libref of the library in which the staged data is stored. In MXG staging code, typically the staged data is stored in the WORK library. △

- Include Non-MXG Staging Code in Your Batch Job

  The staging code must be executed in the same batch job as the %CPSTART macro and the %CMPROCES, %CPPROCES, %CSPROCES, or %CWPROCES macro. The staging code should be inserted after the call to the %CPSTART macro and before the call to the %CMPROCES, %CPPROCES, %CSPROCES, or %CWPROCES macro. This is because the %CPDUPCHK macro writes the temporary control data set to the WORK library and the data set is subsequently used by the %CPDUPUPD macro. The control data would be lost if %CPSTART executed in a different SAS session, because the WORK library is deleted by default at the end of a SAS session.

  If you run on z/OS and you store your staging code in MXG.USERID.SOURCLIB so that your staging code and MXG code are in one

location, then you can use the following example to include the SAS code (assuming that your code is in a member that is named *stage*):

```
%INCLUDE SOURCLIB(stage);
RUN;
```

If you run in a UNIX or Windows environment, then you can use one of the following pairs of statements to include the staging code. The first example applies to UNIX and the second applies to Windows environments.

```
%INCLUDE 'fully-qualified-directory/stage.sas';
RUN;

%INCLUDE 'fully-qualified-directory\stage.sas';
RUN;
```

If a RUN command appears at the end of your staging code, then the RUN command that is shown immediately after the %INCLUDE statement in this example is not required.

This %INCLUDE statement executes your staging code and executes your changes to implement the duplicate-data-checking macros. When you call the %CxPROCES macro with the COLLECTR=GENERIC parameter, your data is processed into the detail level of the PDB from the staging data set where it was written by your staging code.

Note: Remember to specify the GENLIB= parameter on the call to %CxPROCES in order to indicate the libref of the library to which the staging code writes the staged data. △

Handling Duplicate Data

Note: The *Package Collector* enables the collector-support developer to package all the entities for a data collector or data source and install them into PGMLIB or other locations, so that the functionality can be shared or treated as if it were supplied with SAS IT Resource Management. For more information, see "Creating and Installing a Collector Support Package" on page 426. △

Note: Depending on the data collector or data source from which you obtain the data, you might (or might not) need to implement support for handling duplicate data. △

For more information about duplicate-data checking, see "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation. Also see "Overview of Using Process Exits" on page 442.

Using the C2RATE and D2RATE Interpretation Types (Optional)

The C2RATE and D2RATE interpretation types are often the most difficult to understand and use. An example will clarify the distinction between these two. Consider a situation where a data collector or data source regularly collects data every 15 minutes. Each collected record describes the entire 15-minute interval that was just completed. If a metric exists as some sort of accumulated count, you might want to see it as a rate in the detail level. An example of this would be the number of pages that a fax machine sends during the recorded interval. Intuitively, you might consider this metric to be a COUNT. However, because you know that this count will be converted to a rate in the reduction levels, you might want to work with it as a RATE in the detail level as well. Therefore, you could choose to assign the C2RATE or D2RATE interpretation type to this metric. (For more information about weighting, see

"Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.)

The C2RATE interpretation type should be used in situations where the data collector or data source reports a metric in each interval as a counter that was not reset at the start of the interval. For example, suppose that in the first interval the metric reports a value of 50, and in the second interval the metric reports a value of 75. The true value for the count that was reported by the metric is 50 for the first interval (because the first interval is assumed to start from 0) and 25 for the second interval (because the second interval starts where the first interval ends).

The D2RATE interpretation type should be used in situations where the data collector or data source reports a metric (in each interval) as a counter that was reset at the start of the interval. Using the same example, the metric that was recorded in the first interval would report a value of 50, and in the second interval it would report a value of 25.

%C2RATE processing divides the event count by the value of the variable DURATION in order to determine the rate. As such, the C2RATE and D2RATE interpretation types are meaningful only for type Interval tables, because only type Interval tables have a DURATION variable.

For more information about the C2RATE and D2RATE interpretation types, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

Advanced Concepts Related to Staging Data

At this point the fax example is complete. The fax example was fairly straightforward with regard to staging the data. This section discusses other data-staging situations that you might encounter. Each of these cases requires more experience with writing SAS DATA step code for step 1. In most cases, you would still perform steps 2, 3, and 4 as previously discussed.

Case 1: Data Already in a SAS Data Set

You might find that the raw data that you want to use is already in a SAS data set. If so, you can immediately use GENERATE SOURCE to build your table and variable definitions. You will need to be sure that you have a variable that can serve as your DATETIME variable. If you are creating a type Interval table, you will also need a variable to serve as your DURATION variable. Again, these variables do not need to be named DATETIME and DURATION, respectively, in the staged data; by using the DATETIME= and DURATION= parameters on the GENERATE SOURCE statement, you can identify the names that are used in the staged data.

You might find that you need to construct the DATETIME and/or DURATION variables. In such a case, you might find it easier to create a SAS DATA step view in which you build the DATETIME and DURATION variables. In %CPPROCES, you would use the data view in place of the data set in the first parameter.

For more information about the use of SAS DATA step views, refer to the documentation for your release of SAS.

Regardless of whether you have the data already in a SAS data set or whether you use a SAS DATA step view, one consequence is that you do not need to run any staging code, because the existing data set or view is essentially already the staged data set or view.

Case 2: Data Already in an External DBMS

You might also encounter a situation in which the data that you want to stage is stored in an external database management system (DBMS). You can use SAS/ACCESS to create a SAS/ACCESS view descriptor, which can be used just like a SAS data set or view. SAS/ACCESS can read data from the following relational databases:

- DB2 under z/OS
- DB2 under VM
- DB2 under UNIX or PC
- CA-OpenIngres
- Informix
- ODBC
- OLE DB
- ORACLE
- Oracle Rdb
- SYBASE
- MS SQL Server
- Teradata

SAS/ACCESS can also read data from the following non-relational databases and other data sources:

- SAS/ACCESS Interface to ADABAS
- SAS/ACCESS Interface to CA-DATACOM/DB
- SAS/ACCESS Interface to CA-IDMS
- SAS/ACCESS Interface to IMS-DL/I
- SAS/ACCESS Interface to PC File Formats
- SAS/ACCESS Interface to SYSTEM 2000 software

And you can access Enterprise Resource Planning (ERP) systems by using:

- SAS/ACCESS Interface to Baan
- SAS/ACCESS Interface to PeopleSoft
- SAS/ACCESS Interface to R/3
- SAS/ACCESS Interface to SAP BW
- SAS Solution Adapters for SAP

For more information about SAS/ACCESS, contact your SAS Sales Representative.

Case 3: Using a SAS DATA Step View instead of a SAS Data Set

In situations where you are staging data for one table from one external data source (as in the fax example), you might find it advantageous to define a SAS DATA step view instead of a SAS data set. In essence, your staging code defines the view. This technique works only in situations where the entire staging process can be accomplished in one DATA step (that is, no extra steps, such as sorts, are required). The advantage of using a view is that only one pass through the data is required—when the data is processed into the PDB. In the flow that is shown in the fax example, two passes through the data are required: one pass to run the staging code, and one pass when processing the staged data into the PDB.

For more information about the use of SAS DATA step views, refer to the documentation for your release of SAS.

Note: Suppose you have data for multiple tables (such as CICS tables) and suppose the amount of data for one particular table (such as CICS transaction data) is extremely large. To avoid one pass of the large amount of data, you can stage the large amount of data as a view and stage the rest of the data as data sets. *To make this work, in the invocation of the process macro, you must explicitly list the tables (in the tablist parameter) and the first table must be the one whose data is staged with a view.* △

Case 4: Reading a Report ("Screen Scraping")

Sometimes the data that you want to read is not in a simple log or data file. Instead, the data is actually part of a report. Staging the data essentially becomes an exercise in parsing the report to extract the data values. There is no standard method for doing this. An advanced knowledge of SAS DATA step programming might be required in such cases. If these skills do not exist at your site, you might want to contact your SAS Sales Representative to arrange for consulting services.

Case 5: Multiple Tables from a Single Input Source

In our example, a simple log file was used to populate a single table. However, two tables could have been created: one for faxes sent, and another for faxes received. The decision whether to create multiple tables from a data source depends upon the nature and intended use of the data.

If you decide to stage data for multiple tables from one data source, you need to run the staging code only once per daily production job. And you need to invoke the %CPPROCES macro only once to process the staged data into the multiple tables.

Case 6: Character-Delimited Data

Some data collectors and data sources provide the ability to store the raw data in a character-delimited flat file. Quite often the delimiter is the comma character, space character, and/or tab "character." SAS IT Resource Management supports the use of character-delimited data as input for processing data with the Generic Collector Facility. Note that in order to use this feature, you can omit the staging code but you must still define the table and variables into which the data is to be processed. In the case of character-delimited data, you can use the INTYPE=CDC parameter with the GENERATE SOURCE control statement to build the necessary table and variable definitions from the raw character delimited data. For further information, see "Appendix 13: Character-Delimited Support in GENERATE SOURCE" on page 196.

Support for character-delimited data is available on all hosts, by using the %CPPROCES macro. For more information, see "Appendix 9: Using Character-Delimited Data" on page 190.

Generic Collector Facility: Appendices

Appendix 1: Fax: Sample Log

This is an example of what the fax log looks like. The column ruler indicates the location of the data, as shown in Table 11.1 on page 160.

```
Column          1         2         3         4         5         6
Ruler  12345678901234567890123456789012345678901234567890123456789012
```

```
larry    S     502 677-4444  5 10/12/96 13:14:23  0 00:01:45 C
curly    R     502 677-4444  5 10/13/96 01:01:12  2 00:05:14 C
moe      S 052 014 752-1234  5 10/14/96 01:01:12  2 00:05:14 C
```

Appendix 2: Fax: Sample Code for Staging the Data

This is the SAS code used to stage the data from the fax log. When you want to run this code, you can submit it from the SAS Program Editor window or in batch.

Note: Before you submit it, be sure that you have appended the code for generating the definitions. For information about this, see "Appendix 3: Fax: Sample Code for Generating the Definitions" on page 179. △

```
filename faxlog 'fax.log' ;

data work.faxes(drop=date time label='Fax Activity');

 attrib machine  label='Machine name'                  length=$8;
 attrib type     label='Fax Type: (S)end or (R)eceive' length=$1;
 attrib country  label='Country Code'                  length=$3;
 attrib acode    label='Area Code'                     length=$3;
 attrib phnum    label='Phone Number'                  length=$8;
 attrib pages    label='Pages Sent or Received';
 attrib timestmp label='Datetime stamp of fax' format=datetime21.2;
 attrib resent   label='Pages Resent';
 attrib connect  label='Total connect time'    format=time12.2;
 attrib pctrtry  label='Percent retries'       format=percent7.2;
 attrib status   label='Final Status Code'             length=$1;

 infile faxlog;

 input @1  machine          @10 type          @12 country
       @16 acode            @20 phnum         @29 pages
       @32 date : mmddyy8.  @41 time : time8. @50 resent
       @53 connect : time8. @62 status;

 status = upcase(status);
 if status not in('A','B','C') then do;
   put 'Invalid status code found in record ' _n_ ;
   put 'This record will not be included in the staged data';
   put machine= type= country= acode= phnum= pages= date= time=
       resent= connect= status=;
   delete;
 end;

 timestmp=dhms(date,hour(time),minute(time),second(time));

 if pages = 0 then pctrtry = .;
 else pctrtry = resent / pages ;
run;
```

Notes

- The FILENAME statement assigns a fileref, which is a temporary name within the current SAS session, to a file. In this case, the fileref FAXLOG is assigned to the file FAX.LOG.
- The DATA statement creates a SAS data set and specifies characteristics of the data set. The data set is to be created in the SAS library named WORK, have the name FAXES, and have the descriptive label FAX ACTIVITY. (WORK is a predefined SAS library that is automatically created at the beginning of a SAS session and automatically deleted at the end of a SAS session.)
- The DATA statement also begins a DATA step, which in this case ends with a RUN statement. Within the DATA step, there is an implied loop. In this case, one pass through the loop reads one record's worth of data from the input file to a buffer, modifies the data as described in the DATA step, and writes the data (except for the dropped variables DATE and TIME) from the buffer as a row in the data set. (The data set has rows that correspond to records in the input file and columns that correspond to fields on the records.)
- The ATTRIB statements specify some of the attributes of the data. These statements are associated with the INPUT statement later in the step. For more about attributes that help to simplify later steps, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation and see "Appendix 8: Algorithm Used by GENERATE SOURCE" on page 188.
- The INFILE statement specifies the fileref of the file that supplies the input records. In this case, the file to which FAXLOG points supplies the input records.
- The INPUT statement specifies the location of each field of interest on an input record and assigns a name to the data in that field. In this case, the locations are specified by the beginning column symbol, @. This statement also causes one record's worth of data to be read into the buffer.
- The value of the fax's final transmission status is checked. If the status is not valid, the DATA step puts (writes) an error message in the SAS log and deletes the data from the buffer.
- The DATE and TIME variables are used to create the TIMESTMP variable.
- A variable (PCTRTRY), for percent retries, is created to monitor the efficiency of the transfer. Both TIMESTMP and PCTRTRY are derived variables. A *derived variable* is one that does not exist in the raw data, but is derived from it in the staged data. You could instead create a formula variable in the SAS IT Resource Management table, but formula variables cannot be used as BY or CLASS variables, nor can statistics be calculated automatically for formula variables.
- Preceding the RUN statement, the contents of the buffer, if any, are written to the staged data (as a row in the staged data).
- The RUN statement specifies that SAS is to run the previous statements now. As a consequence, looping begins in the DATA step. The loop automatically terminates when the end-of-file is encountered in the input file. Thus, when the processing that was triggered by the RUN is completed, a FAXES data set exists in the (temporary) WORK library. In this example, the data set would have three observations, corresponding to the three records in the input file.
- In this document, you will use these statements in two ways. First, you will add them to a one-time job (or SAS session) that creates an empty staged data set, reads the staged data set to create PDB table and variable definitions, applies the definitions to the PDB, and deletes the empty staged data set. Later, you will place them in a macro and put that macro in the daily production job. In the daily production job, one SAS session will start SAS IT Resource Management, stage the

data (by means of this macro), process the staged data into the PDB, reduce the data, (optionally) report on the data, and delete the staged data set.

- For more about the implied loop and the statements in the DATA step, see the documentation for your release of SAS.

Appendix 3: Fax: Sample Code for Generating the Definitions

Because the GENERATE SOURCE control statement runs against the WORK.FAXES data set, and the WORK library is deleted at the end of the active SAS session (or, in batch, SAS procedure), you must submit this code after the staging code in the same SAS session or procedure. Thus, if you submit the staging code from the SAS Program Editor window, you would follow by submitting this code from the SAS Program Editor window during the same SAS session. Alternatively, you can place both sets of code in the same SAS procedure in a batch job.

Here is the code for generating the table and variable definitions for the fax data:

```
%cpcat; cards4;
GENERATE SOURCE NAME=ufaxes
                TYPE=event
                DATETIME=timestmp
                ENTRYNAME=
                 'sasuser.cpddutl.faxes.source'
                DATASET='work.faxes'
                REPLACE ;
;;;; /* ;;;; must begin in column 1 to terminate
   the input stream */
 %cpcat(cat=sasuser.cpddutl.generate.source);
 %cpddutl(entrynam=
          sasuser.cpddutl.generate.source);
```

Notes

- *For purposes of constructing the table and variable definitions*, the model data set can be created by running the staging code with a null input file. This results in a model data set that is empty of data but has the needed data descriptors.
- The first invocation of the SAS IT Resource Management %CPCAT utility macro copies instream text to a buffer. The SAS language CARDS4 statement indicates the beginning of the instream text. The four semicolons, starting in column 1, indicate the end of the instream text. In this case, the instream text consists of a GENERATE SOURCE control statement.
- The second invocation of the %CPCAT macro finds the stream of cards in the buffer and stores them in the indicated SAS catalog .SOURCE entry. In this case, it stores them in your SASUSER library, in the CPDDUTL catalog, in the GENERATE entry, which is of type SOURCE. (The SASUSER library is predefined and is not deleted when the SAS session ends.) For more information about the %CPCAT utility macro, see the topic %CPCAT in the SAS IT Resource Management macro reference documentation.
- The invocation of the SAS IT Resource Management %CPDDUTL macro finds the %CPDDUTL control statements in the specified SAS catalog entry and runs them. In this case, it reads the SASUSER.CPDDUTL.GENERATE.SOURCE entry and runs the GENERATE SOURCE control statement that it finds there. The GENERATE SOURCE statement specifies that the WORK.FAXES data set is to be read, and that the corresponding table and variable definitions are to be generated and written to the SAS catalog entry that is named

SASUSER.CPDDUTL.FAXES.SOURCE. The GENERATE SOURCE statement also specifies that the definitions are to replace whatever is in that entry (if anything is in that entry), that the definitions are to name the table UFAXES, that the table is to be specified as type Event, and that the staged data set's name of the variable that contains date and time is TIMESTMP.

- At any point, if you want to see the contents of a SAS catalog entry, you can define the libref for that library (if it is not already defined) and then use the COPY command to display the entry in your SAS Program Editor window. The libref SASUSER is already defined, so in this case, you only need to issue the command

```
copy sasuser.cpddutl.faxes.source
```

- For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macros and Control Statements" in the SAS IT Resource Management macro reference documentation.

To see the CREATE TABLE and CREATE VARIABLE control statements that are generated, see "Appendix 4: Fax: Sample Table and Variable Definitions" on page 180.

The generated control statements include attribute information such as the variable interpretation type, the format to use for displaying the data, the default statistics to collect, and the BY and CLASS variables. For more information about how the GENERATE SOURCE control statement determines what information to use, see "Appendix 8: Algorithm Used by GENERATE SOURCE" on page 188.

Appendix 4: Fax: Sample Table and Variable Definitions

The following source statements are generated by running the code in Appendix 2 and Appendix 3. For more information about these topics, see "Appendix 2: Fax: Sample Code for Staging the Data" on page 177 and "Appendix 3: Fax: Sample Code for Generating the Definitions" on page 179.

```
/*-------------------------------------------------------------*
| Table definition statements generated on
| 27NOV96:15:12:58.92
| For PDB table....: UFAXES
| Table type.......: EVENT
| Now sorted by....: MACHINE TYPE COUNTRY
|                    ACODE TIMESTMP
| From SAS data set: WORK.FAXES
| Data set type....: DATA
| Will be sorted by: MACHINE TYPE COUNTRY
|                    ACODE DATETIME HOUR
|                    SHIFT
*-------------------------------------------------------------*/

  DELETE TABLE NAME=UFAXES NOERROR ;

  CREATE TABLE NAME=UFAXES
   COLLECTOR=GENERIC TOOLNAME=SASDS TYPE=EVENT
   KEPT=YES
   EXTNAME=FAXES
   LABEL='Fax Activity'
   DESCRIPTION='Fax Activity'
   DETAIL=(agelimit=10 )
   DAY=(agelimit=45 )
   WEEK=(agelimit=15 )
```

```
   MONTH=(agelimit=18 )
   YEAR=(agelimit=5 ) ;

/*-----------------------------------------------------------*
| Variable definitions for table: UFAXES
*-----------------------------------------------------------*/

    CREATE VARIABLE NAME=MACHINE EXTNAME=MACHINE
     SUBJECT='N/A' LABEL='Machine name'
     DESCRIPTION='Machine name'
     FORMAT=.
     TYPE=CHARACTER LENGTH=8 INTERPRET=STRING
     KEPT=YES
     ;

    CREATE VARIABLE NAME=TYPE EXTNAME=TYPE
     SUBJECT='N/A' LABEL=
             'Fax Type: (S)end or (R)eceive'
     DESCRIPTION='Fax Type: (S)end or (R)eceive'
     FORMAT=.
     TYPE=CHARACTER LENGTH=1 INTERPRET=STRING
     KEPT=YES
     ;

    CREATE VARIABLE NAME=COUNTRY EXTNAME=COUNTRY
     SUBJECT='N/A' LABEL='Country Code'
     DESCRIPTION='Country Code'
     FORMAT=.
     TYPE=CHARACTER LENGTH=3 INTERPRET=STRING
     KEPT=YES
     ;

    CREATE VARIABLE NAME=ACODE EXTNAME=ACODE
     SUBJECT='N/A' LABEL='Area Code'
     DESCRIPTION='Area Code'
     FORMAT=.
     TYPE=CHARACTER LENGTH=3 INTERPRET=STRING
     KEPT=YES
     ;

    CREATE VARIABLE NAME=PHNUM EXTNAME=PHNUM
     SUBJECT='N/A' LABEL='Phone Number'
     DESCRIPTION='Phone Number'
     FORMAT=.
     TYPE=CHARACTER LENGTH=8 INTERPRET=STRING
     KEPT=YES
     ;
    CREATE VARIABLE NAME=PAGES EXTNAME=PAGES
     SUBJECT='N/A' LABEL='Pages Sent or Received'
     DESCRIPTION='Pages Sent or Received'
     FORMAT=BEST12.
     TYPE=NUMERIC LENGTH=8 INTERPRET=COUNT
     KEPT=YES
     ISTATS =(
```

```
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 DAY =(
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 WEEK =(
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 MONTH =(
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 YEAR =(
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 ;
 UPDATE VARIABLE NAME=DATETIME
                 EXTNAME=TIMESTMP
 SUBJECT='N/A' LABEL='Datetime stamp of fax'
 DESCRIPTION='Datetime stamp of fax'
 FORMAT=DATETIME21.2
 ;

CREATE VARIABLE NAME=RESENT EXTNAME=RESENT
 SUBJECT='N/A' LABEL='Pages Resent'
 DESCRIPTION='Pages Resent'
 FORMAT=BEST12.
 TYPE=NUMERIC LENGTH=8 INTERPRET=COUNT
 KEPT=YES
 ISTATS =(
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 DAY =(
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 WEEK =(
    NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
    )
 MONTH =(
```

```
     NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
     NOUSS NOCV NORANGE NOVARIANCE
     NOSTD NONMISS
     )
 YEAR =(
     NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
     NOUSS NOCV NORANGE NOVARIANCE
     NOSTD NONMISS
     )
 ;

CREATE VARIABLE NAME=CONNECT EXTNAME=CONNECT
 SUBJECT='N/A' LABEL='Total connect time'
 DESCRIPTION='Total connect time'
 FORMAT=TIME12.2
 TYPE=NUMERIC LENGTH=8 INTERPRET=TIME
 KEPT=YES
 ISTATS =(
     NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
     NOUSS NOCV NORANGE NOVARIANCE
     NOSTD NONMISS
     )
 DAY =(
     NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
     NOUSS NOCV NORANGE NOVARIANCE
     NOSTD NONMISS
     )
 WEEK =(
     NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
     NOUSS NOCV NORANGE NOVARIANCE
     NOSTD NONMISS
     )
 MONTH =(
     NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
     NOUSS NOCV NORANGE NOVARIANCE
     NOSTD NONMISS
     )
 YEAR =(
     NOCOUNT SUM AVERAGE NOMAXIMUM NOMINIMUM
     NOUSS NOCV NORANGE NOVARIANCE
     NOSTD NONMISS
     )
 ;

CREATE VARIABLE NAME=PCTRTRY EXTNAME=PCTRTRY
 SUBJECT='N/A' LABEL='Percent retries'
 DESCRIPTION='Percent retries'
 FORMAT=PERCENT7.2
 TYPE=NUMERIC LENGTH=8 INTERPRET=PERCENT
 KEPT=YES
 ISTATS =(
    NOCOUNT NOSUM AVERAGE NOMAXIMUM NOMINIMUM
    NOUSS NOCV NORANGE NOVARIANCE
    NOSTD NONMISS
```

```
        )
     DAY =(
        NOCOUNT NOSUM AVERAGE NOMAXIMUM NOMINIMUM
        NOUSS NOCV NORANGE NOVARIANCE
        NOSTD NONMISS
        )
     WEEK =(
        NOCOUNT NOSUM AVERAGE NOMAXIMUM NOMINIMUM
        NOUSS NOCV NORANGE NOVARIANCE
        NOSTD NONMISS
        )
     MONTH =(
        NOCOUNT NOSUM AVERAGE NOMAXIMUM NOMINIMUM
        NOUSS NOCV NORANGE NOVARIANCE
        NOSTD NONMISS
        )
     YEAR =(
        NOCOUNT NOSUM AVERAGE NOMAXIMUM NOMINIMUM
        NOUSS NOCV NORANGE NOVARIANCE
        NOSTD NONMISS
        )
     ;

    CREATE VARIABLE NAME=STATUS EXTNAME=STATUS
     SUBJECT='N/A' LABEL='Final Status Code'
     DESCRIPTION='Final Status Code'
     FORMAT=.
     TYPE=CHARACTER LENGTH=1 INTERPRET=STRING
     KEPT=YES
    ;

    CREATE VARIABLE NAME= HOUR
     EXTNAME='HOUR'
     LABEL='Hour of day'
     DESCRIPTION='Hour_of_day'
     KEPT=YES VALIDITY=. INTERPRET=INT
     TYPE=NUMERIC
     LENGTH=4 FORMAT=2. INFORMAT=. IDNUM=.
     OID= .     SUBJECT='N/A'
  ;

    CREATE VARIABLE NAME= SHIFT
     EXTNAME='SHIFT'
     LABEL='Operations shift'
     DESCRIPTION='Operations_shift'
     KEPT=YES VALIDITY=. INTERPRET=STRING
     TYPE=CHARACTER
     LENGTH=1 FORMAT=. INFORMAT=. IDNUM=.
     OID= .     SUBJECT='N/A'
  ;

  /*-----------------------------------------------------------*
   | Default table ordered using DATETIME HOUR and
   | SHIFT.
```

```
*-------------------------------------------------------------*/
 UPDATE TABLE NAME=UFAXES
  DETAIL=(byvars='DATETIME HOUR SHIFT ' )
  DAY=(classvars='DATETIME HOUR SHIFT ' )
  WEEK=(classvars='DATETIME HOUR SHIFT ' )
  MONTH=(classvars='DATETIME HOUR SHIFT ' )
  YEAR=(classvars='DATETIME HOUR SHIFT ' )
;
```

Appendix 5: Fax: Sample Code for Updating the Definitions

You could apply the following %CPDDUTL control statement to the PDB after applying the generated control statements. For information about this topic, see "Appendix 4: Fax: Sample Table and Variable Definitions" on page 180. This UPDATE TABLE control statement modifies the lists of BY and CLASS variables for this table, thus adding further categorization information.

```
UPDATE TABLE NAME=UFAXES
    DETAIL=(byvars='MACHINE TYPE COUNTRY ACODE
                   DATETIME HOUR SHIFT ' )
    DAY=(classvars='MACHINE TYPE COUNTRY ACODE
                   DATETIME HOUR SHIFT ' )
    WEEK=(classvars='MACHINE TYPE COUNTRY ACODE
                    DATETIME HOUR SHIFT ' )
    MONTH=(classvars='MACHINE TYPE COUNTRY ACODE
                     DATETIME HOUR SHIFT ' )
    YEAR=(classvars='MACHINE TYPE COUNTRY ACODE
                    DATETIME HOUR SHIFT ' )
;
```

Additionally, you could update variable attributes. A good example of such an update would include updating a variable's display format, as shown in Figure 3, "Updating the Data Dictionary to Associate a SAS Display Format as an Attribute of a Variable." (To see this figure, refer to "User-Defined Formats (Optional)" on page 170.)

For more information about applying (that is, running) a control statement, see "Appendix 3: Fax: Sample Code for Generating the Definitions" on page 179. For more information about the %CPDDUTL macro and its control statements, see "Chapter 4: Data Dictionary Macros and Control Statements" in the SAS IT Resource Management macro reference documentation.

Appendix 6: Fax: Sample Daily Process-and-Reduce Job

In a typical daily process-and-reduce job, four main actions occur:

- %CPSTART starts SAS IT Resource Management.
- %FAXES, the user-created macro, stages the raw data into the WORK library.
- %CPPROCES processes the staged data into the PDB.
- %CPREDUCE summarizes the data from the detail level into the summary levels.

The following examples illustrate what a daily production job (which typically runs at night) would look like, using the sample fax data. This code would typically be submitted each night by an automated job submission facility (such as *cron* on UNIX or *OPC/ESA* on z/OS).

z/OS Sample Code

```
%CPSTART(MODE=BATCH,
         ROOT='your.itrm.',
         SITELIB='your.itrm.sitelib',
         PDB='your.fax.pdb.',
         DISP=OLD,
         MXGSRC=('mxg.userid.sourclib' 'mxg.mxg.sourclib'),
         MXGLIB=mxg.mxg.formats,
         _RC=CPSTRC
         );
%PUT CPSTART return code is &CPSTRC;
%INCLUDE FAXES.SAS;
%FAXES;
RUN;

%CPPROCES(COLLECTR=GENERIC,TOOLNM=SASDS,GENLIB=WORK,
          _RC=CPPRRC);
%PUT CPPROCES return code is &CPPRRC;
```

Note: The call to the %CPSTART macro starts SAS IT Resource Management and specifies the PDB for fax data. The PDB= parameter is set to the name (high-level-qualifiers) that you used for the new PDB. The MXGSRC= parameter and the MXGLIB= parameter are optional for this job, because your data is not from SMF. For more information about %CPSTART, including parameter values, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. △

Note: The line

```
%PUT CPSTART return code is &CPSTRC;
```

prints the value of the %CPSTART macro's return code, stored in CPSTRC. △

Note: The code for staging the fax data is invoked next. In this example, the code is available in the %FAXES macro. △

Note: The %CPPROCES macro processes the generic collector's staged data into the detail level of the PDB. In this example, the macro processes data to the UFAXES table from the FAXES data set (named as the UFAXES table's external name) in the WORK library. (By default, the COLLECTR= parameter is set to GENERIC in the %CPPROCES macro.) The COLLECTOR=GENERIC and TOOLNM= SASDS parameters are not required because these are the default values for the %CPPROCES macro. For more about the %CPPROCES macro and its parameters, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation

This execution of the %CPPROCES macro processes data into *all* tables in your active PDB that are associated with the generic collector and have a Kept status of *Yes*. If you want to limit this execution to fax tables only, then code the table names specifically as a parameter of %CPPROCES. Here is an example:

```
%CPPROCESS (UFAXES, COLLECTOR=GENERIC,
TOOLNM=SASDS, GENLIB=WORK, _RC=CPPRRC);
```

△

Note: The final line of code prints the value of the %CPPROCES return code, stored in CPPRRC. △

Note: You might want to add report macro invocations that create a standard set of reports each day (typically at night), saving the output into SAS catalogs or external files. △

UNIX/Windows Code Sample

```
%CPSTART(MODE=BATCH,
         ROOT='your/itrm',
         SITELIB='your/itrm/sitelib',
         PDB='your/fax/pdb.',
         ACCESS=write,
         _RC=CPSTRC
         );
%PUT CPSTART return code is %CPSTRC;
%INCLUDE FAXES.SAS;
%FAXES;
RUN;

%CPPROCES(COLLECTR=GENERIC,TOOLNM=SASDS,GENLIB=WORK,
          _RC=CPPRRC);
%PUT CPPROCES return code is %CPPRRC;
```

Note: The %CPSTART macro starts SAS IT Resource Management and specifies the PDB for FAX data. The PDB= parameter is set to the name (high-level-qualifiers) that you used for the new PDB. △

Note: The line

```
%PUT CPSTART return code is &CPSTRC;
```

prints the value of the %CPSTART macro's return code, stored in CPSTRC. △

Note: The code for staging the fax data is invoked next. In this example, the code is available in the %FAXES macro. △

Note: The %CPPROCES macro processes the generic collector's staged data into the detail level of the PDB. In this example, the macro processes data to the UFAXES table from the FAXES data set (named as the UFAXES table's external name) in the WORK library. (By default, the COLLECTR= parameter is set to GENERIC in the %CPPROCES macro.) The COLLECTOR=GENERIC and TOOLNM= SASDS parameters are not required because these are the default values for the %CPPROCES macro. For more information about the %CPPROCES macro and its parameters, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. △

Note: The final line of code prints the value of the %CPPROCES macro's return code, stored in CPPRRC. △

Note: You might want to add report macro invocations that create a standard set of reports each day (typically at night), saving the output into SAS catalogs or external files. △

Appendix 7: Variable Interpretations and Default Statistics

For information about variable interpretation types, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management the macro reference documentation.

Appendix 8: Algorithm Used by GENERATE SOURCE

The control statement GENERATE SOURCE uses model data as the basis for the table and variable definitions that it generates. As the table and variable statements are constructed, certain assumptions are made by the rules that are embedded in the GENERATE SOURCE logic. By understanding the algorithm that GENERATE SOURCE uses and by building your model data set to take advantage of the algorithm, you can decrease the amount of work that you must do to customize the table and variable characteristics that are assigned by default.

The algorithm determines the specific *interpretation type* for each variable. This interpretation type determines how a variable is to be displayed, what summary statistics to collect, and how these summary statistics are to be calculated. For more information about interpretation types, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

The determination of the interpretation type is performed in a cascading series of IF-THEN/ELSE... tests, until an assignment is made. Therefore it is important to pay attention to the order in which the tests operate, as follows:

1 *Character-type data:* the algorithm handles the simple case of character-type data, as follows:

   if the variable's SAS display format contains the string **"HEX"**

   then set the interpretation type to **HEXFLAGS**

   else set the interpretation type to **STRING**

2 *Numeric-type data:* For numeric-type data, the algorithm is far more complicated. Remember that the IF-THEN/ELSE... tests start at the beginning of these lists and go to the end of these lists. For example, any match that is found in a test late in the lists has already been a non-match in all the tests above it in the lists.

   The variable's SAS display format is examined to try to set the interpretation type, as follows:

Table 11.4 Set Interpretation Type by Examining Variable's Format

IF the variable format contains the string:	THEN set the interpretation type to:
"DATE"	DATE
"DATETIME"	DATETIME
"TIME"	TIME
"PERCENT"	PERCENT

If this does not yield a match, the algorithm then begins to examine the variable's label. It scans the label for distinct words that match those in the following list. If no match is found, then it searches the label again for a character string that matches those in the following list. The complete algorithm for this stage is shown in the following table (where each common right-column value indicates a new ELSE IF).

Table 11.5 Set Interpretation Type by Examining Variable's Label

IF the variable label contains the string:	THEN set the interpretation type to:
" PERCENT"	PERCENT100
"%"	PERCENT100
"UTILIZ"	PERCENT100
"UTILIS"	PERCENT100
"PERCENT"	PERCENT100
"PCT"	PERCENT100
"PERCENT"	PERCENT100
"PCT"	PERCENT100
"RATE"	RATE
"PER"	RATE
"/HR"	RATE
"/HOUR"	RATE
"/MIN"	RATE
"/SEC"	RATE
"PEAK"	MAXIMUM
"MAX"	MAXIMUM
"HWM"	MAXIMUM
"MAXIMUM"	MAXIMUM
"HIGHEST"	MAXIMUM
"MIN"	MINIMUM
"LWM"	MINIMUM
"MINIMUM"	MINIMUM
"LOWEST"	MINIMUM
"RATIO"	PERCENT
"CURRENT"	GAUGE
"SIZE"	GAUGE
"GAUGE"	GAUGE
"SUM"	SUM
"DATETIME"	DATETIME
"DATE"	DATE
"TIME"	TIME
"AVERAGE"	AVERAGE
"MEAN"	AVERAGE
"AVG"	AVERAGE
"AVE"	AVERAGE
"TOTAL"	COUNT

IF the variable label contains the string:	THEN set the interpretation type to:
"COUNT"	COUNT
"PAGES"	COUNT
"VERSION"	INT
"NUMBER"	INT
"TYPE"	INT
"CLASS"	INT
"PRIORITY"	INT
"PTY"	INT

If the algorithm has still not assigned an interpretation type, then the interpretation type is set to GAUGE.

Appendix 9: Using Character-Delimited Data

Purpose

The CHARDELIM option of the Generic Collector Facility extends SAS IT Resource Management support to virtually any non-binary, flat-file type data logs as long as the data can be reformatted into a standard character-delimited format.

This enables programmers who are not familiar with SAS to use their preferred tools to format logged systems data and to process that data into a SAS IT Resource Management PDB. It also allows quick on-site additions of data sources that are not already supported by SAS IT Resource Management.

Data Format

- Flat (non-binary) source, default maximum record length 16384 bytes.
- The first line should contain a delimiter-separated list of variable names. If the variable names in the first line do not match the variable names in the SAS IT Resource Management PDB data dictionary, then they must match the variable's external name.
- Delimiters must follow the rules for the DELIMITER= option of the SAS INFILE statement. For more information, refer to the *SAS Language: Reference* documentation for your release of SAS.
- You must supply a timestamp for each data record. You can supply the timestamp either as a single variable called DATETIME or EXTDTM, or as two variables called DATE and TIME. Although the variable names must be DATETIME or DATE and TIME, you must set the external name of these variables in the SAS IT Resource Management data dictionary to match your raw data. If you use the EXTDTM variable to provide the timestamp, you must set the external name of the DATETIME variable to be EXTDTM. If your raw data uses "datetime" or "date" and "time," then you can leave the external names blank. You must also set the variable's INFORMAT specifications in the data dictionary to match the format of your raw data. See the *SAS Language: Reference* documentation for your current version of SAS for more information about acceptable informats. For more information about changing the informats on variables, see "Overview of Variables" on page 378

- If your raw data is formatted as something other than simple numeric or strings values, then you must set the variable's INFORMAT in the data dictionary. For example, if your raw data specifies a cpu busy value as 75.6%, then you will need to set INFORMAT to "PERCENT".
- If a string variable contains a delimiter character, then the string must be included in double quotes ("). Otherwise, enclosing strings in double quotes is optional.
- The presence of two consecutive delimiter characters indicates a missing value.
- If your table contains interval data (as opposed to event data), you might want to include a variable called DURATION in order to indicate the length of the interval. (If you do not supply a value for DURATION, it will be calculated for you by using successive DATETIME values only if you have at least one variable with an interpretation type of C2RATE or D2RATE.)

Controlling the Behavior of the Character-Delimited Collector

You can control the behavior of the character-delimited collector by using

- Macro Variables

  By default, the first line of the file should contain the field names, and the data should follow on the second line. The logical record length should not exceed 16384 bytes (16K). However, in some situations there might be extra non-data lines before or after the header, or the record length might exceed 16384 bytes. If you set specific macro variables prior to running %CPPROCES, the character-delimited collector can read this data. The macro variables are shown in the following table.

Table 11.6 Table of Macro Variables

Variable Name	Default Value	Description
CDCLRECL	16384	Record Length
CDCHSKIP	0	Number of lines to skip in order to read the header record.
CDCDSKIP	0	Number of lines to skip AFTER the header record, in order to read the first line of data.
CDCIFLDS	_blank_	Space-separated list of variables that are contained in the input file in the order in which they appear in the input data.
CSDELIM	_blank_	The character or characters that delimit the incoming data. This variable is used in conjunction with the user exits to override the value that is supplied on the %CPPROCES macro invocation.
CDCFOBS	0	Number of the first line of input data. This variable is used with the macro variable CDCIFLDS for the special case when the input data does not contain a header.

Variable Name	Default Value	Description
CPUSEVEW	_blank_	Set to NO to force %CPPROCES to stage the data into a SAS data set instead of using a SAS data set view.
CDCDTFMT	_blank_	The informat that is used when reading in the datetime value from the input data. The value of this macro variable overrides the informat that is assigned in the SAS IT Resource Management data dictionary.

- Exits

  You can use several exit points with character-delimited data.

  The first exit point is invoked prior to the DATA step that is used to stage the incoming data. This exit can be used to parse the headers of the input files and to set any macro variables necessary such as CDCIFLDS. This exit must contain a stand-alone SAS DATA step code.

  The code for the exit should be stored in a SAS catalog entry called *libname.collector*.CDCINIT.SOURCE, where *libname* is either ADMIN or SITELIB, and *collector* is the value that is specified for the COLLECTR= parameter on the %CPPROCES macro invocation. The exit should be stored in the ADMIN library if it is intended to be used only for this PDB; the exit should be stored in the SITELIB library if it is intended to be used for all PDBs that use the same SITELIB.

  The second exit point is invoked in the DATA statement of the generated DATA step program. This exit point can be used to add additional data sets to the data statement so that multiple data sets can be staged from a single pass of the input data. A corresponding output statement should be added to the CDCINPUT exit point in order to populate this data set. The contents of this exit is the name of a SAS data set, including any applicable data set options such as a keep list. The code for the exit should be stored in a SAS catalog entry called ***libname.collector*.CDC020.SOURCE**.

  The next exit point is invoked prior to the input statement of the DATA step before any variables have been read. This exit point can be used to parse the filename that is being processed in order to set the value of any variables that are encoded in the filenames and are not contained in the data file. The name of the file that is being processed is captured in the variable _FNAME. The contents of the exit are any valid SAS DATA step statements. The code for the exit should be stored in a SAS catalog entry called ***libname.collector*.CDC025.SOURCE**.

  You can also use an exit point that is invoked immediately after each line of data has been read from the input file. This exit can be used to perform data manipulation, such as adjusting the value of DATETIME for handling different time zones. The contents of the exit are any valid SAS DATA step statements. The code for the exit should be stored in a SAS catalog entry called ***libname.collector*.CDCINPUT.SOURCE**.

- Invoking the Process Macro
  - Use the %CPPROCES macro for all SAS IT Resource Management platforms.
  - Setting *COLLECTR=GENERIC* is required.
  - Setting *TOOLNM=CHARDELIM* is required.
  - Setting **DELIM='*delim_chars*'** is optional. If you specify **DELIM=**, then delim_chars must be enclosed in single quotation marks. If you do not specify delimiter characters, then a comma is used as the default delimiter.

A typical macro invocation might look like this:

```
%LET CDCLRECL=4096;  * record are only 4K;
%LET CDCHSKIP=1;     * skip one line to read
                       the header;
%LET CDCDSKIP=2;     * two blank lines between
                       header and data;
%CPPROCES(tablename, rawdata=/u/sasabc/rawdata,
          collectr=generic,
          toolnm=chardelim, delim='|');
```

Note: To help you get started with processing character-delimited data, you can refer to the VMSTAT examples. For details about these VMSTAT examples, see "CSVMSCOL: UNIX Shell Script" on page 233 and "CSVMSDDU: VMSTAT DDUTL" on page 204. △

Appendix 10: Defining Tables and Variables without Using GENERATE SOURCE

For creating tables, see "Overview of Tables" on page 353 and "Creating a Table" on page 357.

For creating variables, see "Overview of Variables" on page 378 and "Creating/Editing/Viewing a Derived Variable" on page 381, "Creating/Editing/Viewing a Formula Variable" on page 388, and "Creating/Editing/Viewing a Regular Variable" on page 394.

Appendix 11: Support by Means of Supplied Table Definitions

Overview

In the technique described in this appendix, you use a predefined, supplied table definition to add (to your PDB) a table to hold data from a specific data collector or data source. Then, you use the COLLECTR= parameter on your processing macro to run predefined, supplied SAS code that specifically stages (pre-processes) the data from that collector and points the processing macro to the staged data, which is then processed. You can use the table and software stream "as is," or you can modify them. For updating the table, there are both interactive and batch techniques. For modifying the software stream, there are exits in your processing macro that enable you to run user-written (in SAS code) exit routines.

Brief Description

For many data sources, SAS IT Resource Management supplies specific table definitions and staging software for using the data with SAS IT Resource Management. For each of the data collectors or data sources for which SAS IT Resource Management supplies collector support, SAS IT Resource Management provides

- SAS code that stages the data (as staging data sets or DATA step views)
- one or more table definitions that you can add to your performance data warehouse (PDB) and several ways to add the table definition(s) to your PDB
- a macro that invokes SAS IT Resource Management
- a macro that processes the staged data into the tables in your PDB
- a macro that reduces the data in the PDB
- macros that report on the data in the PDB.

Some of these components are unique for each of the collectors and some are the same for all of the collectors.

For a complete list of the supplied collector support and the platforms on which it runs, refer to COLLECTR= parameter on the topic %CMPROCES in the SAS IT Resource Management macro reference documentation, the topic %CPPROCES in the SAS IT Resource Management macro reference documentation, the topic %CSPROCES in the SAS IT Resource Management macro reference documentation, or the topic %CWPROCES in the SAS IT Resource Management macro reference documentation.

SAS IT Resource Management includes table and variable definitions for the metrics (data) collected by these collectors.

Appendix 12: The %CPPROCES Macro with the Generic Collector Facility

Note: This appendix discusses the use of the %CPPROCES macro only with the Generic Collector Facility. For a more complete description of these parameters and of the many different collectors that are available with the %CPPROCES macro, see the description of the COLLECTR= parameter on the the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. △

Purpose

The %CPPROCES macro is used to process data into a SAS IT Resource Management performance data warehouse (PDB). This macro should be used only when you are processing data with the Generic Collector Facility.

If TOOLNM=SASDS, before using this macro, you must convert your data into one or more SAS data sets or views; you can then use the %CPPROCES macro to process the data from the SAS data sets or views to the detail level of the PDB.

Syntax

%CPPROCES(
<*table-list*>
,**COLLECTR=**GENERIC
,**DUPMODE=**DISCARD | INACTIVE | FORCE | TERMINATE
,**EXITSRC=***exit-source*
,**GENLIB=***libref*
,<**RAWDATA=***filename*>
,<**TOOLNM=**SASDS | CHARDELIM>
,**_RC=***macro-var-name*) ;

%CPPROCES Parameters

table-list

lists the tables into which the logged data should be processed. This positional parameter is not required but, if specified, it must be specified in the first position. Use blanks to separate the tables in the list. If you omit this parameter, you must specify a comma as a placeholder. Additionally, if you do not specify a table or list of tables, the PDB is read and data is processed for the tables that meet the following three criteria:

- COLLECTOR=GENERIC
- TOOLNM=SASDS *or* TOOLNM=CHARDELIM
- KEPT=YES.

COLLECTR=GENERIC

specifies the name of the collector (or type of collector) that created the log file data. The value for the Generic Collector Facility is GENERIC. If you do not specify a value for this parameter, the default value is GENERIC.

DUPMODE=DISCARD | INACTIVE | FORCE | TERMINATE

specifies whether or not to allow duplicate data to be processed into the active PDB. Duplicate-data checking involves comparing the timestamp information for incoming data to the timestamp information for existing data, and recording the timestamp information for new, non-duplicate data. The default value is INACTIVE. Valid values for the %CPPROCES macro as follows:

DISCARD	reviews the incoming data and rejects (discards) duplicate data, but allows processing to continue. The filtering report that is printed in the SAS log provides information about the data that were processed and/or discarded.
INACTIVE	does not review the incoming data and, therefore, allows duplicate data to be processed into the active PDB.
FORCE	reviews the incoming data and allows duplicate data into the PDB.
TERMINATE	reviews the incoming data and stops processing if duplicate data is encountered.

Note: Before you can use duplicate-data checking, you must implement it as described in "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation. △

EXITSRC=*exit-source*

specifies the location where exit source programs for this application are stored. The value of this parameter can be the name of a SAS catalog that is specified in the form of *libref.catalog-name* or a specified location on your host system. For UNIX and Windows, this would be the complete pathname of a directory, and for z/OS this would be the fully qualified name of a partitioned data set. If you specify a SAS catalog name, use a SAS LIBNAME statement to associate the libref with the SAS library that contains the catalog. If you want to process your data without using any process exits, do not specify this parameter.

For more information about exit processing, see the "Overview of Using Process Exits" on page 442.

GENLIB=*libref*

(use with TOOLNM=SASDS only) specifies the libref for the SAS library that contains the staged data to be processed into the PDB. The staged data has been created by user-provided software and is to be processed into a PDB by using COLLECTR=GENERIC. The data can be a SAS data set, a SAS DATA step view, a SAS SQL view, or a SAS/ACCESS view. Use this parameter to specify the input data set for the SAS IT Resource Management process macro that you are using to process your data, only if you specify COLLECTR=GENERIC and TOOLNM=SASDS.

RAWDATA=*filename*

(use with TOOLNM=CHARDELIM only) specifies the full path and filename of the raw data file that you want to process (UNIX and Windows) or the fully qualified data set name for a sequential data set or the fully qualified PDS member name (z/OS).

TOOLNM=SASDS | CHARDELIM

is an optional parameter. If you do not specify this parameter, the default value, SASDS, is used.

_RC=*macro-var-name*

identifies a macro variable that will contain the return code from this macro. For example, you might specify the variable name *retcode* to show the return code value from this macro. Values that can be returned are 0 and nonzero. A value of 0 indicates that the macro was successful. A nonzero value indicates that the macro failed. Check the explanatory message in the SAS log. The macro variable value that is set by this parameter should be used only in batch.

%CPPROCES Example

This example processes data for a table named *TABABC* into the detail level of the PDB. The data has already been staged into the SAS library named *WORK*.

```
%cpproces(TABABC,genlib=WORK) ;
```

Appendix 13: Character-Delimited Support in GENERATE SOURCE

Introduction

The GENERATE SOURCE functionality of the %CPDDUTL macro has been enhanced to enable customers to generate a "first cut" of table and variable definitions for data sources directly from the raw data.

Note: GENERATE SOURCE is more strict than %C*x*PROCES (where *x*=M, P, S, or W) in terms of how the character-delimited data is presented to it; however, this should not be a problem because generating table and variable definitions is, typically, not a daily event, and the data can be manipulated to meet the requirements. △

Requirements

- The data must contain one header record. The header record must contain delimited values that describe the data that follows.
- It is recommended that you include about 10 lines of data after the header record, so that this data can be processed in order to assist in determining characteristics of the data.
- If the raw data does not follow the rules in the preceding two points, then the data should be modified so that it follows those rules.
- Although specifying the DELIM= parameter is not necessary, it is recommended. GENERATE SOURCE will try to decide on the delimiter automatically; however, this is not always possible, because of the format of the data.
- Only one table will be constructed per run of GENERATE SOURCE when the INTYPE=CDC option is specified.
- The NAME= option can be used to specify the name of the table that you want to create. If it is not specified, then *UCDCTAB* is the default name.

Example

The following is an example of how this facility can be used. For more information about the parameters to use with the GENERATE SOURCE control statement, see “Chapter 4: Data Dictionary Macros and Control Statements” in the SAS IT Resource Management macro reference documentation.

Here is the data:

```
DATA FIELDS:
machine,date,time,cpu utilization,value pct,
i/o per sec,device name,average val,max val,
min val,

DATA RECORDS:
tom,01JAN2000,01:00:00,0.5,20,6,devHarry,5,7,2,
tom,01JAN2000,01:05:00,0.6,21,7,devHarry,6,8,5,
tom,01JAN2000,01:10:00,0.7,22,8,devHarry,7,8,8,
tom,01JAN2000,01:15:00,0.8,23,9,devHarry,8,9,8,
tom,01JAN2000,01:20:00,0.9,24,10,devHarry,9,3,6,
tom,01JAN2000,01:25:00,0.0,25,11,devHarry,10,7,4,
```

Note: The data above is intended only to illustrate the CDC support of GENERATE SOURCE. △

Start SAS IT Resource Management and submit the following code from the SAS Program Editor window:

```
%cpcat;
 cards4;
 generate source
 infile='c:\rawdata\cdctest.data'
 intype=cdc
 type=interval
 delim=' '
 name=utest
 filename='c:\temp\ddutl.txt';
;;;;  /* ;;;; must begin in col 1 to terminate
              the input stream. */
 %cpcat(cat=work.temp.temp.source);
 %cpddutl(entrynam=work.temp.temp.source);
```

The output of the above code is a file in *c:\temp* called *ddutl.txt* that contains table and variable definitions that will enable you to process the CDC data. The table and variable definitions that are created should be reviewed, and modified if necessary.

For more information about interpretation type, refer to "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

```
/*-------------------------------------------------------*
 | Table definition statements generated on
 | 24MAR2000:16:32:13.25
 | For PDB table.....: UTEST
 | Table type........: INTERVAL
 | Now sorted by.....:
 | Will be sorted by.:
 *-------------------------------------------------------*/
DELETE TABLE NAME=UTEST NOERROR ;
 CREATE TABLE NAME=UTEST
 COLLECTOR='GENERIC'
 TOOLNAME=CHARDELIM
 TYPE=INTERVAL
 KEPT=YES
 EXTNAME='CDCTABLE' IDNUM=0
```

```
LABEL='CDC Table' ARCHIVE=NO
DESCRIPTION='CDC Table'
DETAIL=(agelimit=10 )
DAY=(agelimit=45)
WEEK=(agelimit=15 )
MONTH=(agelimit=18 )
YEAR=(agelimit=5 ) ;

 /*-------------------------------------------------*
| Variable definitions for table: UTEST
*-------------------------------------------------*/

CREATE VARIABLE NAME=MACHINE
EXTNAME='machine'
SUBJECT='N/A' LABEL='machine'
DESCRIPTION='machine'
FORMAT=. OID=. IDNUM=.
INTERPRET=STRING LENGTH=32
TYPE=CHARACTER KEPT=YES
;

CREATE VARIABLE NAME=DATE
EXTNAME='date'
SUBJECT='N/A' LABEL='date'
DESCRIPTION='date'
FORMAT=DATE9. OID=. IDNUM=.
INTERPRET=DATE LENGTH=8
TYPE=NUMERIC KEPT=YES
INFORMAT=DATE9.
;

CREATE VARIABLE NAME=TIME
EXTNAME='time'
SUBJECT='N/A' LABEL='time'
DESCRIPTION='time'
FORMAT=TIME12.2 OID=. IDNUM=.
INTERPRET=TIME LENGTH=8
TYPE=NUMERIC KEPT=YES
INFORMAT=TIME8.
ISTATS =(
NOCOUNT SUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
DAY =(
NOCOUNT SUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
WEEK =(
NOCOUNT SUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
```

```
MONTH =(
NOCOUNT SUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
YEAR =(
NOCOUNT SUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
;

CREATE VARIABLE NAME=CUTLZTN
EXTNAME='cpu utilization'
SUBJECT='N/A'
LABEL='cpu utilization'
DESCRIPTION='cpu utilization'
FORMAT=PERCENT9.2 OID=. IDNUM=.
INTERPRET=PERCENT LENGTH=6
TYPE=NUMERIC KEPT=YES
ISTATS =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
DAY =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
WEEK =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
MONTH =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
YEAR =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
;

CREATE VARIABLE NAME=VLPCT
EXTNAME='value pct'
SUBJECT='N/A' LABEL='value pct'
DESCRIPTION='value pct'
FORMAT=6.2 OID=. IDNUM=.
INTERPRET=PERCENT100 LENGTH=6
TYPE=NUMERIC KEPT=YES
```

```
ISTATS =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
DAY =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
WEEK =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
MONTH =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
YEAR =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
;

CREATE VARIABLE NAME=IOPRSEC
EXTNAME='i/o per sec'
SUBJECT='N/A' LABEL='i/o per sec'
DESCRIPTION='i/o per sec'
FORMAT=BEST12.2 OID=. IDNUM=.
INTERPRET=RATE LENGTH=8
TYPE=NUMERIC KEPT=YES
ISTATS =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
DAY =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
WEEK =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
MONTH =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
```

```
YEAR =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
;

CREATE VARIABLE NAME=DVCNAME
EXTNAME='device name'
SUBJECT='N/A' LABEL='device name'
DESCRIPTION='device name'
FORMAT=. OID=. IDNUM=.
INTERPRET=STRING LENGTH=200
TYPE=CHARACTER KEPT=YES
;

CREATE VARIABLE NAME=AVRGVAL
EXTNAME='average val'
SUBJECT='N/A' LABEL='average val'
DESCRIPTION='average val'
FORMAT=BEST12.2 OID=. IDNUM=.
INTERPRET=AVERAGE LENGTH=8
TYPE=NUMERIC KEPT=YES
ISTATS =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
DAY =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
WEEK =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
MONTH =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
YEAR =(
NOCOUNT NOSUM AVERAGE NOMAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
;

CREATE VARIABLE NAME=MAXVAL
EXTNAME='max val'
SUBJECT='N/A' LABEL='max val'
DESCRIPTION='max val'
```

```
FORMAT=BEST12.2 OID=. IDNUM=.
INTERPRET=MAXIMUM LENGTH=8
TYPE=NUMERIC KEPT=YES
ISTATS =(
NOCOUNT NOSUM NOAVERAGE MAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
DAY =(
NOCOUNT NOSUM NOAVERAGE MAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
WEEK =(
NOCOUNT NOSUM NOAVERAGE MAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
MONTH =(
NOCOUNT NOSUM NOAVERAGE MAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
YEAR =(
NOCOUNT NOSUM NOAVERAGE MAXIMUM
NOMINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
;

CREATE VARIABLE NAME=MINVAL
EXTNAME='min val'
SUBJECT='N/A' LABEL='min val'
DESCRIPTION='min val'
FORMAT=BEST12.2 OID=. IDNUM=.
INTERPRET=MINIMUM LENGTH=8
TYPE=NUMERIC KEPT=YES
ISTATS =(
NOCOUNT NOSUM NOAVERAGE NOMAXIMUM
MINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
) DAY =(
NOCOUNT NOSUM NOAVERAGE NOMAXIMUM
MINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
)
WEEK =(
NOCOUNT NOSUM NOAVERAGE NOMAXIMUM
MINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
) MONTH =(
NOCOUNT NOSUM NOAVERAGE NOMAXIMUM
MINIMUM NOUSS NOCV NORANGE
NOVARIANCE NOSTD NONMISS
```

```
 )
 YEAR =(
 NOCOUNT NOSUM NOAVERAGE NOMAXIMUM
 MINIMUM NOUSS NOCV NORANGE
 NOVARIANCE NOSTD NONMISS
 )
 ;

 CREATE VARIABLE NAME=HOUR
 EXTNAME='HOUR'
 LABEL='Hour of day'
 DESCRIPTION='Hour_of_day'
 KEPT=YES VALIDITY=. INTERPRET=INT
 TYPE=NUMERIC
 LENGTH=4 FORMAT=2. INFORMAT=.
 IDNUM=. OID= .
 SUBJECT='N/A' ;

 CREATE VARIABLE NAME=SHIFT
 EXTNAME='SHIFT'
 LABEL='Operations shift'
 DESCRIPTION='Operations_shift'
 KEPT=YES VALIDITY=.
 INTERPRET=STRING
 TYPE=CHARACTER
 LENGTH=1 FORMAT=.
 INFORMAT=. IDNUM=.
 OID= .
 SUBJECT='N/A' ;

/*-------------------------------------------------*
| By, Class, ID, and Index variable lists
| for table: UTEST
*-------------------------------------------------*/
 UPDATE TABLE NAME= UTEST
 DETAIL =(
 BYVARS= ' DATETIME HOUR SHIFT'
 )
 DAY =(
 CLASSVARS= ' DATETIME HOUR SHIFT'
 ) WEEK =(
 CLASSVARS= ' DATETIME HOUR SHIFT'
 ) MONTH =(
 CLASSVARS= ' DATETIME HOUR SHIFT'
 ) YEAR =(
 CLASSVARS= ' DATETIME HOUR SHIFT'
 )
 ;
```

Generic Collector Facility: Examples

CSVMSDDU: VMSTAT DDUTL

```
/*******************************************
* VMSTAT DDUTL
*******************************************/
/*=============================================
= Begin MIB Module: VMSTAT
=============================================*/
/*---------------------------------------------
- TABLE: UVMSTT (vmstat)
- MODULE: VMSTAT
---------------------------------------------*/
DELETE TABLE  NAME=UVMSTT NOERROR;
CREATE TABLE  NAME=UVMSTT
  EXTNAME='vmstat'
  LABEL='VMSTAT.vmstat'
  TOOLNAME='CHARDELIM'
  COLLECTOR='GENERIC'
  TYPE=INTERVAL
  KEPT=YES
  IDNUM=0
  DETAIL=(AGELIMIT=10)
  DAY=(AGELIMIT=60)
  WEEK=(AGELIMIT=12)
  MONTH=(AGELIMIT=24)
  YEAR=(AGELIMIT=0)
  DESCRIPTION=  'VMSTAT.vmstat: Group'
;
/*---------------------------------------------
- VARIABLE: PRCRUNQ (prcRunQ)
- TABLE: UVMSTT (vmstat)
---------------------------------------------*/
CREATE VARIABLE NAME=PRCRUNQ
   EXTNAME='prcRunQ'
   LABEL='prcRunQ'
   KEPT=YES
   LENGTH=8
   INTERPRET=RATE
   TYPE=NUMERIC
   FORMAT=BEST12.
   INFORMAT=.
   DAY=(AVERAGE)
   WEEK=(AVERAGE)
   MONTH=(AVERAGE)
   YEAR=(AVERAGE)
   ISTATS=(AVERAGE)
   DESCRIPTION=  'prcRunQ: The number of
                  processes per second
                  placed on the run queue
```

```
                during the sample interval.'
;
/*---------------------------------------------
- VARIABLE: UNAME (uname)
- TABLE: UVMSTT (vmstat)
---------------------------------------------*/
CREATE VARIABLE NAME=UNAME
  EXTNAME='uname'
  LABEL='uname'
  KEPT=YES
  LENGTH=10
  INTERPRET=STRING
  TYPE=CHARACTER
  FORMAT=$CHAR.
  INFORMAT=.
  DAY=(NOAVERAGE)
  WEEK=(NOAVERAGE)
  MONTH=(NOAVERAGE)
  YEAR=(NOAVERAGE)
  ISTATS=(NOAVERAGE)
  DESCRIPTION=  'Output from the unix 'uname'
                command'
;
/*---------------------------------------------
- VARIABLE: PRCBLKD (prcBlkd)
- TABLE: UVMSTT (vmstat)
---------------------------------------------*/
CREATE VARIABLE NAME=PRCBLKD
  EXTNAME='prcBlkd'
  LABEL='prcBlkd'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION=  'prcBlkd: The number of processes
                per second blocked for resources
                during the sample interval.'
                ;
/*---------------------------------------------
- VARIABLE: PRCSWPD (prcSwpd)
- TABLE: UVMSTT (vmstat)
---------------------------------------------*/
CREATE VARIABLE NAME=PRCSWPD
  EXTNAME='prcSwpd'
  LABEL='prcSwpd'
  KEPT=YES
  LENGTH=8
```

```
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION=  'prcSwpd: The number of processes
                 per second runnable but swapped
                 during the sample interval.'
;
/*-----------------------------------------------
- VARIABLE: MAVVLPS (memActvVrtlPgs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=MAVVLPS
  EXTNAME='memActvVrtlPgs'
  LABEL='memActvVrtlPgs'
  KEPT=YES
  LENGTH=8
  INTERPRET=GAUGE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'memActvVrtlPgs: The number of
                active virtual memory pages at
                the time of the interval sample.'
;
/*-----------------------------------------------
- VARIABLE: MALSPSC (memAvlSwpSpc)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=MALSPSC
  EXTNAME='memAvlSwpSpc'
  LABEL='memAvlSwpSpc'
  KEPT=YES
  LENGTH=8
  INTERPRET=GAUGE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'memAvlSwpSpc: The number of
```

```
                pages of swap space available at
                the time of the interval sample.'
;
/*---------------------------------------------
- VARIABLE: MFRLTPS (memFreLstPgs)
- TABLE: UVMSTT (vmstat)
---------------------------------------------*/
CREATE VARIABLE NAME=MFRLTPS
  EXTNAME='memFreLstPgs'
  LABEL='memFreLstPgs'
  KEPT=YES
  LENGTH=8
  INTERPRET=GAUGE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'memFreLstPgs: The number of
                real memory pages on the free
                list at the time of the interval
                sample.'
;
/*---------------------------------------------
- VARIABLE: PGRCLMS (pagRclms)
- TABLE: UVMSTT (vmstat)
---------------------------------------------*/
CREATE VARIABLE NAME=PGRCLMS
  EXTNAME='pagRclms'
  LABEL='pagRclms'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagRclms: page reclaims'
;
/*---------------------------------------------
- VARIABLE: PARTNFS (pagAdrTrnFlts)
- TABLE: UVMSTT (vmstat)  -
---------------------------------------------*/
CREATE VARIABLE NAME=PARTNFS
  EXTNAME='pagAdrTrnFlts'
  LABEL='pagAdrTrnFlts'
  KEPT=YES
```

```
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagAdrTrnFlts: address
               translation faults'
;
/*-----------------------------------------------
- VARIABLE: PGMRFLS (pagMinorFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
 CREATE VARIABLE NAME=PGMRFLS
  EXTNAME='pagMinorFlts'
  LABEL='pagMinorFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagMinorFlts: minor faults'
;
/*-----------------------------------------------
- VARIABLE: PGPGINS (pagPageIns)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=PGPGINS
  EXTNAME='pagPageIns'
  LABEL='pagPageIns'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagPageIns: pages paged in'
;
/*-----------------------------------------------
```

```
- VARIABLE: PGPGOTS (pagPageOuts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=PGPGOTS
  EXTNAME='pagPageOuts'
  LABEL='pagPageOuts'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagPageOuts: pages paged out'
;
/*-----------------------------------------------
- VARIABLE: PGPSFRD (pagPgsFrd)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=PGPSFRD
  EXTNAME='pagPgsFrd'
  LABEL='pagPgsFrd'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagPgsFrd: pages freed'
;
/*-----------------------------------------------
- VARIABLE: PGATMSL (pagAntMemShrtfl)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=PGATMSL
  EXTNAME='pagAntMemShrtfl'
  LABEL='pagAntMemShrtfl'
  KEPT=YES
  LENGTH=8
  INTERPRET=GAUGE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
```

```
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagAntMemShrtfl: anticipated
               short term memory shortfall'
;
/*-----------------------------------------------
- VARIABLE: PCKSDPS (pagClkScndPgs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=PCKSDPS
  EXTNAME='pagClkScndPgs'
  LABEL='pagClkScndPgs'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)      YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagClkScndPgs: pages scanned
               by clock algorithm'
;
/*-----------------------------------------------
- VARIABLE: PGCKRVS (pagClkRevs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=PGCKRVS
  EXTNAME='pagClkRevs'
  LABEL='pagClkRevs'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'pagClkRevs: clock revolutions
                            observed'
;
/*-----------------------------------------------
- VARIABLE: FTDVINS (fltDevInts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=FTDVINS
  EXTNAME='fltDevInts'
  LABEL='fltDevInts'
```

```
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'fltDevInts: (non-clock) device
               interrupts'
;
 /*-----------------------------------------------
- VARIABLE: FLTSCLS (fltSysCls)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=FLTSCLS
  EXTNAME='fltSysCls'
  LABEL='fltSysCls'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'fltSysCls: system calls'
;
/*-----------------------------------------------
- VARIABLE: FTCXSWH (fltCntxSwtch)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=FTCXSWH
  EXTNAME='fltCntxSwtch'
  LABEL='fltCntxSwtch'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'fltCntxSwtch: cpu context
                              switches'
```

```
;
/*-----------------------------------------------
- VARIABLE: CPUUSER (cpuUser)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=CPUUSER
  EXTNAME='cpuUser'
  LABEL='cpuUser'
  KEPT=YES
  LENGTH=8
  INTERPRET=GAUGE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'cpuUser: cpu user time'
;
/*-----------------------------------------------
- VARIABLE: CPUSYS (cpuSys)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=CPUSYS
  EXTNAME='cpuSys'
  LABEL='cpuSys'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'cpuSys: cpu system time'
;
/*-----------------------------------------------
- VARIABLE: CPUIDLE (cpuIdle)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=CPUIDLE
  EXTNAME='cpuIdle'
  LABEL='cpuIdle'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
```

```
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'cpuIdle: cpu idle time'
;
/*-------------------------------------------
- VARIABLE: CPUWAIT (cpuWait)
- TABLE: UVMSTT (vmstat)
-------------------------------------------------*/
CREATE VARIABLE NAME=CPUWAIT
  EXTNAME='cpuWait'
  LABEL='cpuWait'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'cpuWait: cpu idle but had
               outstanding i/o'
;
/*-------------------------------------------------
- VARIABLE: DSK1 (dsk1)
- TABLE: UVMSTT (vmstat)
-------------------------------------------------*/
/ CREATE VARIABLE NAME=DSK1
  EXTNAME='dsk1'
  LABEL='dsk1'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'dsk1: disk 1 transfers'
;
/*-------------------------------------------
- VARIABLE: DSK2 (dsk2)
- TABLE: UVMSTT (vmstat)
-------------------------------------------------*/
CREATE VARIABLE NAME=DSK2
  EXTNAME='dsk2'
```

```
  LABEL='dsk2'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'dsk2: disk 2 transfers'
;
/*-----------------------------------------------
- VARIABLE: DSK3 (dsk3)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=DSK3
  EXTNAME='dsk3'
  LABEL='dsk3'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'dsk3: disk 3 transfers'
;
/*-----------------------------------------------
- VARIABLE: DSK4 (dsk4)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=DSK4
  EXTNAME='dsk4'
  LABEL='dsk4'
  KEPT=YES
  LENGTH=8
  INTERPRET=RATE
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'dsk4: disk 4 transfers'
;
```

```
/*-----------------------------------------------
- VARIABLE: SADTNFS (sumAddTrnFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SADTNFS
  EXTNAME='sumAddTrnFlts'
  LABEL='sumAddTrnFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumAddTrnFlts: address trans.
               faults taken'
;
/*-----------------------------------------------
- VARIABLE: SMSPINS (sumSwapIns)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMSPINS
  EXTNAME='sumSwapIns'
  LABEL='sumSwapIns'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumSwapIns: swap ins'
;
/*-----------------------------------------------
- VARIABLE: SMSPOTS (sumSwapOuts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMSPOTS
  EXTNAME='sumSwapOuts'
  LABEL='sumSwapOuts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
```

```
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumSwapOuts: swap outs'
;
/*-----------------------------------------------
- VARIABLE: SPSSDIN (sumPgsSwpdIn)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SPSSDIN
  EXTNAME='sumPgsSwpdIn'
  LABEL='sumPgsSwpdIn'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPgsSwpdIn: pages swapped in'
;
/*-----------------------------------------------
- VARIABLE: SPSSDOT (sumPgsSwpdOut)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SPSSDOT
  EXTNAME='sumPgsSwpdOut'
  LABEL='sumPgsSwpdOut'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPgsSwpdOut: pages swapped out'
;
/*-----------------------------------------------
- VARIABLE: SMPGINS (sumPgIns)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMPGINS
  EXTNAME='sumPgIns'
  LABEL='sumPgIns'
```

```
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPgIns: page outs'
;
/*-----------------------------------------------
- VARIABLE: SPSPDIN (sumPgsPgdIn)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SPSPDIN
  EXTNAME='sumPgsPgdIn'
  LABEL='sumPgsPgdIn'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPgsPgdIn: pages paged in'
;
/*-----------------------------------------------
- VARIABLE: SPSPDIA (sumPgsPgdIn)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SPSPDIA
  EXTNAME='sumPgsPgdIn'
  LABEL='sumPgsPgdIn'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPgsPgdIn: pages paged in'
;
/*-----------------------------------------------
```

```
- VARIABLE: SPSPDOT (sumPgsPgdOut)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SPSPDOT
  EXTNAME='sumPgsPgdOut'
  LABEL='sumPgsPgdOut'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPgsPgdOut: pages paged out'
;
/*-----------------------------------------------
- VARIABLE: SFRLTRS (sumFreLstReclms)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SFRLTRS
  EXTNAME='sumFreLstReclms'
  LABEL='sumFreLstReclms'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumFreLstReclms: reclaims from free list'
;
/*-----------------------------------------------
- VARIABLE: SMPGRCS (sumPgReclms)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMPGRCS
  EXTNAME='sumPgReclms'
  LABEL='sumPgReclms'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
```

```
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPgReclms: page reclaims'
;
/*-----------------------------------------------
- VARIABLE: SIBPGFS (sumInstBlkPgFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SIBPGFS
  EXTNAME='sumInstBlkPgFlts'
  LABEL='sumInstBlkPgFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumInstBlkPgFlts: intransit blocking page
               faults'
;
/*-----------------------------------------------
- VARIABLE: SIBPGFA (sumInstBlkPgFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SIBPGFA
  EXTNAME='sumInstBlkPgFlts'
  LABEL='sumInstBlkPgFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumInstBlkPgFlts: intransit blocking page
               faults'
;
/*-----------------------------------------------
- VARIABLE: SZFPSCD (sumZroFilPgsCrtd)
- TABLE: UVMSTT (vmstat)
-------------------------------------------------------------*/
CREATE VARIABLE NAME=SZFPSCD
  EXTNAME='sumZroFilPgsCrtd'
  LABEL='sumZroFilPgsCrtd'
```

```
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumZroFilPgsCrtd: zero fill pages created'
;
/*-----------------------------------------------
- VARIABLE: SEFPSCD (sumExeFilPgsCrtd)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SEFPSCD
  EXTNAME='sumExeFilPgsCrtd'
  LABEL='sumExeFilPgsCrtd'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumExeFilPgsCrtd: executable
                fill pages created'
;
/*-----------------------------------------------
- VARIABLE: SEFPGFS (sumExeFilPgFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SEFPGFS
  EXTNAME='sumExeFilPgFlts'
  LABEL='sumExeFilPgFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumExeFilPgFlts: executable
                fill page faults'
```

```
;
/*-----------------------------------------------
- VARIABLE: SSTPFLS (sumSwpTxtPgsFrLs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SSTPFLS
  EXTNAME='sumSwpTxtPgsFrLs'
  LABEL='sumSwpTxtPgsFrLs'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumSwpTxtPgsFrLs: swap text
                pages found in free list'
;
/*-----------------------------------------------
- VARIABLE: SMRHTFS (sumMcrHatFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMRHTFS
  EXTNAME='sumMcrHatFlts'
  LABEL='sumMcrHatFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumMcrHatFlts: micro (hat) faults'
;
/*-----------------------------------------------
- VARIABLE: SMRASFS (sumMnrAsFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMRASFS
 EXTNAME='sumMnrAsFlts'
 LABEL='sumMnrAsFlts'
 KEPT=YES
 LENGTH=8
 INTERPRET=C2RATE
 TYPE=NUMERIC
 FORMAT=BEST12.2
```

```
 INFORMAT=10.
 DAY=(AVERAGE)
 WEEK=(AVERAGE)
 MONTH=(AVERAGE)
 YEAR=(AVERAGE)
 ISTATS=(AVERAGE)
 DESCRIPTION= 'sumMnrAsFlts: minor (as) faults'
;
/*-----------------------------------------------
- VARIABLE: SMMRFLS (sumMajorFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMMRFLS
  EXTNAME='sumMajorFlts'
  LABEL='sumMajorFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumMajorFlts: major faults'
;
/*-----------------------------------------------
- VARIABLE: SCOWTFS (sumCpyOnWrtFlts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SCOWTFS
  EXTNAME='sumCpyOnWrtFlts'
  LABEL='sumCpyOnWrtFlts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumCpyOnWrtFlts: copy-on-write
                faults'
;
/*-----------------------------------------------
- VARIABLE: SITPFLS (sumIndTxtPgsFrLs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SITPFLS
```

```
  EXTNAME='sumIndTxtPgsFrLs'
  LABEL='sumIndTxtPgsFrLs'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumIndTxtPgsFrLs: inode text
               pages found in free list'
;
/*-----------------------------------------------
- VARIABLE: SCKHDRS (sumClkHndRevs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SCKHDRS
  EXTNAME='sumClkHndRevs'
  LABEL='sumClkHndRevs'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumClkHndRevs: revolutions of
               the clock hand'
;
/*-----------------------------------------------
- VARIABLE: SCKEDPS (sumClkExmdPgs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SCKEDPS
  EXTNAME='sumClkExmdPgs'
  LABEL='sumClkExmdPgs'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
```

```
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumClkExmdPgs: pages examined
               by clock'
;
/*-----------------------------------------------
- VARIABLE: SCKFDPS (sumClkFrdPgs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SCKFDPS
  EXTNAME='sumClkFrdPgs'
  LABEL='sumClkFrdPgs'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumClkFrdPgs: pages freed by
              the clock daemon'
;
/*-----------------------------------------------
- VARIABLE: SMFORKS (sumForks)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMFORKS
  EXTNAME='sumForks'
  LABEL='sumForks'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumForks: forks'
;
/*-----------------------------------------------
- VARIABLE: SMVFRKS (sumVforks)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMVFRKS
  EXTNAME='sumVforks'
  LABEL='sumVforks'
  KEPT=YES
  LENGTH=8
```

```
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumVforks: vforks'
;
/*-----------------------------------------------
- VARIABLE: SCPCXSH (sumCpuCntxSwtch)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SCPCXSH
  EXTNAME='sumCpuCntxSwtch'
  LABEL='sumCpuCntxSwtch'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumCpuCntxSwtch: cpu context
               switches'
;
/*-----------------------------------------------
- VARIABLE: SMDVINS (sumDevIntrpts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMDVINS
  EXTNAME='sumDevIntrpts'
  LABEL='sumDevIntrpts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumDevIntrpts: device interrupts'
;
/*-----------------------------------------------
- VARIABLE: SMTRAPS (sumTraps)
```

```
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMTRAPS
  EXTNAME='sumTraps'
  LABEL='sumTraps'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumTraps: traps'
;
/*-----------------------------------------------
- VARIABLE: SMSCLLS (sumSysCalls)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMSCLLS
  EXTNAME='sumSysCalls'
  LABEL='sumSysCalls'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumSysCalls: system calls'
;
/*-----------------------------------------------
- VARIABLE: SMEXECS (sumExecs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMEXECS
  EXTNAME='sumExecs'
  LABEL='sumExecs'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
```

```
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumExecs: execs'
;
/*-----------------------------------------------
- VARIABLE: SMBCKKS (sumBcktrcks)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMBCKKS
  EXTNAME='sumBcktrcks'
  LABEL='sumBcktrcks'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumBcktrcks: backtracks'
;
/*-----------------------------------------------
- VARIABLE: SMLCKMS (sumLockMisses)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMLCKMS
  EXTNAME='sumLockMisses'
  LABEL='sumLockMisses'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumLockMisses: lock misses'
;
/*-----------------------------------------------
- VARIABLE: SFRFMWS (sumFreFrmWaits)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SFRFMWS
  EXTNAME='sumFreFrmWaits'
  LABEL='sumFreFrmWaits'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
```

```
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumFreFrmWaits: free frame waits'
;
/*-----------------------------------------------
- VARIABLE: SETXTWS (sumExtXPTWaits)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SETXTWS
  EXTNAME='sumExtXPTWaits'
  LABEL='sumExtXPTWaits'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumExtXPTWaits: extend XPT waits'
;
/*-----------------------------------------------
- VARIABLE: SMPGIWS (sumPndgIOWaits)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMPGIWS
  EXTNAME='sumPndgIOWaits'
  LABEL='sumPndgIOWaits'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumPndgIOWaits: pending I/O waits'
;
/*-----------------------------------------------
- VARIABLE: SMSTIOS (sumStrtIOs)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
```

```
CREATE VARIABLE NAME=SMSTIOS
  EXTNAME='sumStrtIOs'
  LABEL='sumStrtIOs'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumStrtIOs: start I/Os'
;
/*-----------------------------------------------
- VARIABLE: SMIDNES (sumIoDones)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMIDNES
  EXTNAME='sumIoDones'
  LABEL='sumIoDones'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumIoDones: iodones'
;
/*-----------------------------------------------
- VARIABLE: SMSRINS (sumSftwrInts)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMSRINS
  EXTNAME='sumSftwrInts'
  LABEL='sumSftwrInts'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
```

```
  DESCRIPTION= 'sumSftwrInts: software interrupts'
;
/*-----------------------------------------------
- VARIABLE: SMNMLKS (sumNameLookups)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMNMLKS
  EXTNAME='sumNameLookups'
  LABEL='sumNameLookups'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumNameLookups: total name
               lookups (cache hits 2%)'
;
/*-----------------------------------------------
- VARIABLE: SMTLONG (sumTooLong)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMTLONG
  EXTNAME='sumTooLong'
  LABEL='sumTooLong'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumTooLong: toolong'
;
/*-----------------------------------------------
- VARIABLE: SMUSRCP (sumUserCPU)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMUSRCP
  EXTNAME='sumUserCPU'
  LABEL='sumUserCPU'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
```

```
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumUserCPU: user cpu'
;
/*-----------------------------------------------
- VARIABLE: SMSSCPU (sumSysCPU)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMSSCPU
  EXTNAME='sumSysCPU'
  LABEL='sumSysCPU'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumSysCPU: system cpu'
;
/*-----------------------------------------------
- VARIABLE: SMIDLCP (sumIdleCPU)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMIDLCP
  EXTNAME='sumIdleCPU'
  LABEL='sumIdleCPU'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumIdleCPU: idle cpu'
;
/*-----------------------------------------------
- VARIABLE: SMWTCPU (sumWaitCPU)
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SMWTCPU
```

```
  EXTNAME='sumWaitCPU'
  LABEL='sumWaitCPU'
  KEPT=YES
  LENGTH=8
  INTERPRET=C2RATE
  TYPE=NUMERIC
  FORMAT=BEST12.2
  INFORMAT=10.
  DAY=(AVERAGE)
  WEEK=(AVERAGE)
  MONTH=(AVERAGE)
  YEAR=(AVERAGE)
  ISTATS=(AVERAGE)
  DESCRIPTION= 'sumWaitCPU: wait cpu'
;
/*-----------------------------------------------
- VARIABLE: HOUR ()
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=HOUR
  EXTNAME=''
  LABEL='HOUR'
  KEPT=YES
  LENGTH=3
  INTERPRET=INT
  TYPE=NUMERIC
  FORMAT=BEST12.
  INFORMAT=.
  DAY=(NOAVERAGE)
  WEEK=(NOAVERAGE)
  MONTH=(NOAVERAGE)
  YEAR=(NOAVERAGE)
  ISTATS=(NOAVERAGE)
  DESCRIPTION= 'HOUR is a default variable.'
; /
*-----------------------------------------------
- VARIABLE: MACHINE ()
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=MACHINE
  EXTNAME='machine'
  LABEL='MACHINE'
  KEPT=YES
  LENGTH=32
  INTERPRET=STRING
  TYPE=CHARACTER
  FORMAT=$CHAR.
  INFORMAT=.
  DAY=(NOAVERAGE)
  WEEK=(NOAVERAGE)
  MONTH=(NOAVERAGE)
  YEAR=(NOAVERAGE)
  ISTATS=(NOAVERAGE)
  DESCRIPTION= 'MACHINE is a default variable.'
```

```
;
/*-----------------------------------------------
- VARIABLE: SHIFT ()
- TABLE: UVMSTT (vmstat)
-----------------------------------------------*/
CREATE VARIABLE NAME=SHIFT
  EXTNAME=''
  LABEL='SHIFT'
  KEPT=YES
  LENGTH=1
  INTERPRET=STRING
  TYPE=CHARACTER
  FORMAT=$CHAR.
  INFORMAT=.
  DAY=(NOAVERAGE)
  WEEK=(NOAVERAGE)
  MONTH=(NOAVERAGE)
  YEAR=(NOAVERAGE)
  ISTATS=(NOAVERAGE)
  DESCRIPTION= 'SHIFT is a default variable.'
; /
*-----------------------------------------------
- UPDATE TABLE with BY and CLASS Variables
- TABLE: UVMSTT (vmstat)

-----------------------------------------------*/
UPDATE TABLE NAME=UVMSTT
DETAIL=(BYVARS='MACHINE  DATETIME HOUR SHIFT')
  DAY   =(CLASSVARS='MACHINE  DATETIME HOUR
          SHIFT')
  WEEK  =(CLASSVARS='MACHINE  DATETIME HOUR
          SHIFT')
  MONTH =(CLASSVARS='MACHINE  DATETIME HOUR
          SHIFT')
  YEAR  =(CLASSVARS='MACHINE  DATETIME HOUR
          SHIFT')
;
/*===============================================
= End VMSTAT
===============================================*/
```

CSVMSCOL: UNIX Shell Script

```
#!/usr/local/bin/perl
############################################################
# Program: csvmscol
# Purpose: Executes vmstat command, time-stamping
#          and logging output.
# Support: none
#
#  Usage: csvmscol [ -o file ] [ -i interval ]
#                  [ -d delim ] [ -h ]
```

```
#    Where:
#       file      is where collected data is
#                 written;
#                 file is appended with a dot '.'
#                 followed by the day of year
#                 (default output is to stdout)
#
#       interval  is seconds between polling;
#                 interval must be >= 5
#                 (default interval is 5 seconds)
#
#       delim     is the string used to separate
#                 data values in output
#                 (default delim is a comma ',')
#
#       -h        prints this usage
#
#          This script is Intended to be run
#          continuously in background (from
#          inittab
#          perhaps) to collect system performance
#          data. As each new day begins the
#          previous day's log file can
#          be processed without
#          interrupting the program.
#
# Output:  Formatted vmstat output is written to
#          STDOUT unless an output file is
#          specified with -o option.
#          If an output file is specified then
#          the day of year (julian date)
#          is appended to the file name. Data
#          values are separated by a delimiter
#          character(s). The first line of the
#          output is a delimiter-separated
#          list of variable names.
#
#
#          As each new day is entered (on the
#          sample interval boundry) the file
#          from the previous day is closed and a
#          new file with the current day of year
#          is opened for output.
#
#          If the output file already exists
#          then data is appended to it.
#          This is so that if the program is
#          restarted within a single day,
#          the day's previous data won't be lost.
#          The header is written only once.
#
#          When the script is first started,
#          its pid is written to stderr.
#          You can capture this pid to a file
```

```
#          to make it easier to kill.
#
#
# Portability:
#          This script has been test on AIX 3.2,
#          SunOS 5.4 and HP-UX 9.05. It expects
#          the data to be printed in a certain
#          order, which is different for each os.
#          Check the %vars array below for data
#          ordering expectations.
#
# Change History:
# Date      Who          Description
# ---------------------------------------------
# 20May96   Don Koch     Created
#
################################################
#
################################################
####  Start of User Configuration Constants  ####
################################################
# If perl complains that it can't find
# "getopts.pl" then put its
# path here $PERL_LIBS="/usr/lib/perl/";
# Define commands we use.
# These may depend on os.
$CMD_VMSTAT    ="/usr/bin/vmstat";
$CMD_HOSTNAME  ="/bin/hostname";
$CMD_UNAME     ="/bin/uname";
$CMD_BASENAME  ="/bin/basename";
# Define options for vmstat.
$OPT_VMSTAT{"AIX"}   = "";
$OPT_VMSTAT{"HP-UX"} = "";
$OPT_VMSTAT{"SunOS"} = "";
################################################
####  End of User Configuration Constants    ####
################################################
# USAGE USAGE USAGE USAGE
$thisProg = '$CMD_BASENAME $0';
chop($thisProg);
$USAGE = "
Usage: $thisProg [ -o file ] [ -i interval ]
                 [ -d delim ] [ -h ]
     Where:  file      is where collected data
                       is written;
                       file is appended with a
                       dot '.' followed by
                       the day of year (default
                       output is to stdout)
             interval  is seconds between
                       polling;
                       interval must be >= 5
                       (default interval is
                        5 seconds)
```

```
              delim     is the string used to
                        separate data
                        values in output
                        (default delim is a
                         comma ',')
              -h        prints this usage
";
# Grab command line arguments
push(@INC,$PERL_LIBS);
require 'getopts.pl';
if (&Getopts('o:i:d:h')) {
$outFile  = $opt_o;
$interval = $opt_i;
$delim    = $opt_d;
$help     = $opt_h;
}
else {
print STDERR $USAGE;
exit 1;
}
# Just needs a little help
if ($help) {
print STDERR $USAGE;
exit 0; }
# Print my pid to stderr (catch it & you'll know
#                         what to kill)
print STDERR "$$\n";
# If no delim specified then set default to comma
$delim = "," if ($delim eq "");
# Check for proper interval or set to default
# value
if ($interval eq "") {
$interval = 5;
}
else {
$interval !~ /^[0-9]+$/ &&  die "Interval must be
                                 an integer : ";
}  if ($interval < 5)
      { die "Interval must be >= 5 : " };
# Find out our hostname and os
$hostname = `$CMD_HOSTNAME`;
chop($hostname);
$uname = `$CMD_UNAME`;
chop($uname);

# Build variable name array.
# The "keys" are the variable names.
# The "values" are space-separated lists
# of positions. The positions are the order
# in which each os's vmstat command prints
# that variable.
# A 00 indicates the os doesn't support
# that variable.
# For %oss: key = uname output,
```

```
#   value = os's position in the values of %vars
$oss{"HP-UX"} = 0;
$oss{"AIX"}   = 1;
$oss{"SunOS"} = 2;
# vmstat output position for "HP,AIX,SUN"
$vars{"prcRunQ"}  = "01,01,01";
                # procs in run queue: rate
$vars{"prcBlkd"}  = "02,02,02";
            # procs blocked for resources: rate
$vars{"prcSwpd"}  = "03,00,03";
            # procs runnable but swapped: rate
$vars{"memActvVrtlPgs"} = "04,03,00";
            # active virtual pages: gauge
$vars{"memAvlSwpSpc"}   = "00,00,04";
         # amount of swapspace available: gauge
$vars{"memFreLstPgs"}   = "05,04,05";
         # size of free list: gauge
$vars{"pagRclms"}       = "06,05,06";
           # page reclaims: rate
$vars{"pagAdrTrnFlts"}  = "07,00,00";
           # address translation faults: rate
$vars{"pagMinorFlts"}   = "00,00,07";
           # minor faults: rate
$vars{"pagPageIns"}     = "08,06,08";
                  # pages paged in: rate
$vars{"pagPageOuts"}    = "09,07,09";
                  # pages paged out: rate
$vars{"pagPgsFrd"}      = "10,08,10";
                  # pages freed: rate
$vars{"pagAntMemShrtfl"} = "11,00,11";
                  # anticipated short term
                  # memory shortfall: gauge
$vars{"pagClkScndPgs"}  = "12,09,12";
        # pages scanned by clock algorithm: rate
$vars{"dsk1"}             = "00,00,13";
                  # disk 1 transfers: rate
$vars{"dsk2"}             = "00,00,14";
                  # disk 2 transfers: rate
$vars{"dsk3"}             = "00,00,15";
                  # disk 3 transfers: rate
$vars{"dsk4"}             = "00,00,16";
                  # disk 4 transfers: rate
$vars{"pagClkRevs"}       = "00,10,00";
           # clock revolutions observed: rate
$vars{"fltDevInts"}       = "13,11,17";
          # (non-clock) device interrupts: rate
$vars{"fltSysCls"}        = "14,12,18";
                  # system calls: rate
$vars{"fltCntxSwtch"}     = "15,13,19";
                  # cpu context switches: rate
$vars{"cpuUser"}          = "16,14,20";
                  # cpu user time: percent
$vars{"cpuSys"}           = "17,15,21";
                  # cpu system time: percent
```

```
$vars{"cpuIdle"}          = "18,16,22";
                    # cpu idle time: percent
$vars{"cpuWait"}          = "00,17,00";
                    # cpu idle but had
                    # outstanding i/o: percent
# The following variables are output from
# "vmstat -s".
# These numbers should pick up where
# the above numbers left off.
$vars{"sumAddTrnFlts"}    = "23,18,27";
                    # address trans. faults
                    # taken: counter
$vars{"sumSwapIns"}       = "19,00,23";
                    # swap ins: counter
$vars{"sumSwapOuts"}      = "20,00,24";
                    # swap outs: counter
$vars{"sumPgsSwpdIn"}     = "21,00,25";
                    # pages swapped in: counter
$vars{"sumPgsSwpdOut"}    = "22,00,26";
                    # pages swapped out: counter
$vars{"sumPgIns"}         = "24,19,28";
                    # page ins: counter
$vars{"sumPgOuts"}        = "25,20,29";
                    # page outs: counter
$vars{"sumPgsPgdIn"}      = "26,21,30";
                    # pages paged in : counter
$vars{"sumPgsPgdOut"}     = "27,22,31";
                    # pages paged out : counter
$vars{"sumFreLstReclms"}  = "28,00,33";
                    # reclaims from free list:
                    # counter
$vars{"sumPgReclms"}      = "29,23,32";
                    # page reclaims: counter
$vars{"sumInstBlkPgFlts"} = "30,00,00";
                    # intransit blocking page
                    # faults: counter
$vars{"sumZroFilPgsCrtd"} = "31,00,00";
                    # zero fill pages created:
                    # counter
$vars{"sumMcrHatFlts"}    = "00,00,34";
                    # micro (hat) faults: counter
$vars{"sumMnrAsFlts"}     = "00,00,35";
                    # minor (as) faults: counter
$vars{"sumMajorFlts"}     = "00,00,36";
                    # major faults: counter
$vars{"sumCpyOnWrtFlts"}  = "00,00,37";
                    # copy-on-write faults:
                    # counter
$vars{"sumZroFilPgFlts"}  = "32,24,38";
                    # zero fill page faults:
                    # counter
$vars{"sumExeFilPgsCrtd"} = "33,00,00";
                    # executable fill pages
                    # created: counter
```

```
$vars{"sumExeFilPgFlts"}  = "34,25,00";
                  # executable fill page
                  # faults: counter
$vars{"sumSwpTxtPgsFrLs"} = "35,00,00";
                  # swap text pages found in
                  # free list: counter
$vars{"sumIndTxtPgsFrLs"} = "36,00,00";
                  # inode text pages found in
                  #  free list: counter
$vars{"sumClkHndRevs"}    = "37,27,40";
                  # revolutions of the clock
                  # hand: counter
$vars{"sumClkExmdPgs"}    = "38,26,39";
                  # pages examined by clock:
                  # counter
$vars{"sumClkFrdPgs"}     = "39,28,41";
                 # pages freed by the clock
                 # daemon: counter
$vars{"sumForks"}         = "00,00,42";
                 # forks: counter
$vars{"sumVforks"}        = "00,00,43";
                 # vforks: counter
$vars{"sumCpuCntxSwtch"}  = "40,36,45";
                # cpu context switches: counter

 $vars{"sumDevIntrpts"}    = "41,37,46";
                 # device interrupts: counter
$vars{"sumTraps"}         = "42,39,47";
                 # traps: counter
$vars{"sumSysCalls"}      = "43,40,48";
                 # system calls: counter
$vars{"sumExecs"}         = "00,00,44";
                 # execs: counter
$vars{"sumBcktrcks"}      = "00,29,00";
                 # backtracks: counter
$vars{"sumLockMisses"}    = "00,30,00";
                 # lock misses: counter
$vars{"sumFreFrmWaits"}   = "00,31,00";
                 # free frame waits: counter
$vars{"sumExtXPTWaits"}   = "00,32,00";
                 # extend XPT waits: counter
$vars{"sumPndgIOWaits"}   = "00,33,00";
                 # pending I/O waits: counter
$vars{"sumStrtIOs"}       = "00,34,00";
                 # start I/Os: counter
$vars{"sumIoDones"}       = "00,35,00";
                 # iodones: counter
$vars{"sumSftwrInts"}     = "00,38,00";
                 # software interrupts: counter


$vars{"sumNameLookups"}   = "00,00,49";
 # total name lookups
```

```
 # (cache hits 2%): counter
$vars{"sumTooLong"}       = "00,00,50";
 # toolong: counter
$vars{"sumUserCPU"}       = "00,00,51";
 # user   cpu: counter
$vars{"sumSysCPU"}        = "00,00,52";
  # system cpu: counter
$vars{"sumIdleCPU"}       = "00,00,53";
                           # idle   cpu: counter
$vars{"sumWaitCPU"}       = "00,00,54";
                           # wait   cpu: counter
# Which number in the os array are we
$osNum = -1;
foreach $os (keys %oss) {
   if ($os eq $uname) {
       $osNum = $oss{$os};
   }
}
die "I am not programmed for your operating
     system :   " if ($osNum < 0);
# Build an ordered array of the variable names
# for this os
foreach $var (keys %vars) {
  # turn comma-separated list of numbers
  # into an array
  @headerNum = split(/,/,$vars{$var});
  # if this os supports the variable then
  # store var name in header array
    if ($headerNum[$osNum] != 0) {
  $headerVars[$headerNum[$osNum]] = $var;
  }
}
# Collect data until interrupted.
while (1) {
  # Get day of year (add 1 since Jan 1 is day 0)
  $doy = (localtime)[7] + 1;
  # Set header printing flag to print
  $printHeader = 1;
  # Open output file or redirect stdout if
  # none specified
  if ($outFile ne "") {
    # Build new file name
    $newFile = $outFile . "." . $doy;
    # If new file exists then program was
    # restarted during same day Check first line
    # to see if header was written so we don't
    # write a another one in the middle of the
    # data.
    if (open(NEW,"<$newFile")) {
      $_ = <NEW>;
      $printHeader = 0 if
                  (/^datetime.+machine.+uname/)
}
# Open for append in case of restart during
```

```
# same day.
open (OUT,">>$newFile") ||
      die "Can't open $newFile for output : ";
# Force flush after every write
select OUT;
$| = 1;
}
else {
   # Dup OUT to stdout
   open (OUT,">&STDOUT") ||
             die "Can't dup STDOUT : ";
}
# Print the header if it needs one
if ($printHeader == 1) {
    print OUT "datetime" . $delim . "machine" .
               $delim . "uname";
    for ($i = 1; $i <= $#headerVars; $i++) {
         print OUT $delim . $headerVars[$i];

   }
   print OUT "\n";
 }
# Open vmstat command pipe
open (VMSTAT,"$CMD_VMSTAT $OPT_VMSTAT{$uname}
      $interval |") ||
      die "Can't run $CMD_VMSTAT $interval : ";
 # Since we are starting a new vmstat process
 # at this point we must skip the non-data
 # header output AND the first data line which
 # contains stats
 # summary statistics since system boot.
$_ = <VMSTAT>;
while (/[a-zA-Z\-]+/) {
    $_ = ;
 }
# Collect real data until interrupted
while ($intervalRecord = ) {
# Record time stamp for active data
  $ts = &timestamp
# We have vmstat interval data from  in
# $intervalRecord.
# Additionally we get vmstat summary stats from
# which is stored in $summaryRecord for later
# processing.
open (VMSTAT_S,"$CMD_VMSTAT -s |");
$k = 0;
while () {
    $summaryRecord[$k++] = $_;
}
close (VMSTAT_S);
# If we have entered a new day, exit "while
# ($intervalRecord = )". The current log file
# (previous day) will be closed and a new log
# file (current day) will be opened.
```

```
# The data records we just read will be lost.
last if ($doy != (localtime)[7] + 1);
# Skip over non-data header output from
# and store the next data record.
while ($intervalRecord =~ /[a-zA-Z]+/) {
    $intervalRecord = ;
}
chop($intervalRecord);

# Convert space-separated interval data to a
# $delim-separated string and print to log.
# Note: forced space on split string so
#       $intervalRecord
# will lead with a delim after join.
$intervalRecord = " ".$intervalRecord;
split(/[\ \t\n]+/,$intervalRecord);
$intervalRecord = join("$delim",@_);
print OUT $ts . $delim . $hostname . $delim .
          $uname . $intervalRecord;

# Parse and convert array of  output to
# $delim-separated record

for ($k=0; $k<=$#summaryRecord; $k++) {
   $summaryRecord[$k] =~ /^\s*([0-9]+).*/;
   print OUT $delim . $1;
 }
 print OUT "\n";
}
# Close I/O to start a new day
close(OUT);
close(VMSTAT); }
# Subroutine to give us a nicely formatted
# timestamp
sub timestamp {
     @months = ("","Jan","Feb","Mar","Apr","May",
                "Jun","Jul","Aug","Sep","Oct",
                "Nov","Dec");
     ($hour,$min,$sec,$mon,$day,$year) =
                      (localtime)[2,1,0,4,3,5];

     sprintf ("%2.2d%s%2.2d:%2.2d:%2.2d:%2.2d",
              $day,$months[++$mon],$year+1900,
              $hour,$min,$sec);
 }
# Subroutine to print a timestamp to stderr
#  for debugging
# Usage: xtime("comment") will print
#                         "12:31:02 - comment"
sub xtime {
   ($hour,$min,$sec) = (localtime)[2,1,0];
    print STDERR "$hour:$min:$sec - @_\n";
}
```

CHAPTER
12

Migration

Migrating Your SAS IT Resource Management Environment from SAS Version 6 to SAS Version 8

Migrating Your SAS IT Resource Management Environment from SAS Version 6 to SAS Version 8

Introduction

This document is for sites at which SAS IT Resource Management currently runs under SAS Version 6 (V6) but is migrating to run under SAS Version 8 (V8). You do not need this document if you are installing SAS IT Resource Management for the first time or if you already migrated your SAS IT Resource Management environment to V8.

Introductory Notes

- The AIX Platform

  If you are migrating from V6 to V8 on the AIX platform, see "Special Considerations for the AIX Platform" on page 256 (at the end of this document) before continuing.

- Difference between Production Work and Non-Production Work

  SAS V6 and V8 have different structures for objects. The objects need to be converted. In this document, "convert" means to create a V8 object that corresponds to a V6 object.

  Before you use SAS IT Resource Management software for production work under V8, you need to convert SAS IT Resource Management objects to V8. You can use V6 objects for V8 non-production work as long as you use ACCESS=READONLY (for UNIX/Windows) or DISP=SHR (for z/OS) for V6 objects; however, we do not recommend reading V6 PDBs from V8.

- Supplied Software and Objects

  Before you follow the rest of the instructions in this document, install SAS IT Resource Management on V8. This installs the supplied software and supplied objects (already in their converted form). For more details, see the installation instructions.

- Site-Specific (Customized) Software and Objects

  Now, the objects that are specific (customized) to you or your site (your SASUSER library, your site library/libraries, your PDBs, and so on) need to be converted.

  Note: Convert the objects in the order in which they are presented below. △

Your SASUSER Library

The first time that you use V8, it creates a V8 SASUSER library. The new library is empty.

Your V6 SASUSER library probably contains information that you want to continue to use. The V6 library might contain information that is related to SAS settings. It might also contain information that is related to SAS IT Resource Management settings, your list of known PDBs, your remote profiles, and perhaps your custom report definitions, custom rule definitions, custom palette definitions, and so on. Additionally, it might contain similar information about other SAS products.

Following are the instructions to copy the contents of the V6 library to the V8 library so that the V8 SASUSER library contains the V6 contents that have been structured correctly for V8. In the instructions, you have two decisions to make:

1. Decision 1: You can copy the contents by using the SAS Program Editor window in the V8 GUI or by running the equivalent SAS batch job under V8. One of these methods might be more convenient than the other at your site.
2. Decision 2: You can copy the contents by using the EXCLUDE or SELECT statement.
   - Using the EXCLUDE statement to copy the contents is recommended if you are using the V8 SASUSER library for the first time. You might copy more items than you need and thus be required to delete some items later, but you will be assured of copying all the items that you need.
   - Using the SELECT statement to copy the contents is recommended if you have already used your V8 SASUSER library. You might not copy all the items that you need and thus be required to copy more items later, but you will not overwrite any items already in the V8 SASUSER library.

Read through all the instructions before acting on them, because the choices might make an important difference at your site.

1. Determine the location of your V6 SASUSER library and the names of the items in it. The easiest way to do this is to bring up SAS IT Resource Management under

V6 in your usual way, and submit the following statements from the SAS Program Editor window:

Note: You can also submit this code in a V6 SAS batch job. △

```
OPTIONS SOURCE NOTES ;
LIBNAME SASUSER LIST ;
PROC DATASETS DD=SASUSER ;
RUN ;
QUIT ;
```

In the SAS log window, the location of your V6 SASUSER library is identified by the label Physical Name, and the items in your V6 SASUSER library are listed in the Name column.

2 Using a method that does not also invoke SAS IT Resource Management (or that invokes SAS IT Resource Management and then exits to SAS), invoke the V8 GUI. SAS defines the SASUSER libref to point to your V8 SASUSER library (and creates the library if this is your first use of V8).

3 In the SAS Program Editor window of the V8 GUI, type the following code, edit the location of your V6 SASUSER library, and submit the edited code. Or else you can do the same thing in a SAS batch job under V8.

At this point, choose whether you will copy with either the EXCLUDE statement instructions or the SELECT statement instructions.

- To copy your library by using the EXCLUDE statement (which copies everything except the item(s) listed in the EXCLUDE statement):
  - For z/OS:

    ```
    LIBNAME olduser 'old_library_name'
    DISP=SHR ;
    PROC COPY IN=olduser OUT=sasuser ;
    EXCLUDE profile;
    RUN ;
    ```

    where 'old_library_name' is the location of your V6 SASUSER library.
  - For UNIX or Windows:

    ```
    LIBNAME olduser 'old_library_name' ACCESS=READONLY ;
    PROC COPY IN=olduser OUT=sasuser ;
    EXCLUDE profile;
    RUN ;
    ```

    where *old_library_name* is the location of your V6 SASUSER library.

  Note: The profile that is being excluded is your Base SAS profile. △

  Note: If you use your V8 SASUSER library with other products or applications, some of the items that you copied might be associated with the other products and applications. Verify the correct operation of those products or applications after you copy the contents of the library. △
- To copy your library by using the SELECT statement (which copies only the item(s) listed in the SELECT statement):
  - For z/OS:

    ```
    LIBNAME olduser 'old_library_name' DISP=SHR ;
    PROC COPY IN=olduser OUT=sasuser ;
    SELECT cpusropt cpver itsv workarea tskinfo
    ```

```
       cpupgms cpuout cpulog cpugseg cpusrc;
RUN ;
```

where 'old_library_name' is the location of your V6 SASUSER library.

- For UNIX or Windows:

```
LIBNAME olduser 'old_library_name' ACCESS=READONLY ;
PROC COPY IN=olduser OUT=sasuser ;
SELECT cpusropt cpver itsv workarea tskinfo
       cpupgms cpuout cpulog cpugseg cpusrc;
RUN ;
```

where *old_library_name* is the location of your V6 SASUSER library.

Note: The select list above contains the names of the items that SAS IT Resource Management might have created in the V6 SASUSER library. Add to the select list the names of any objects that you created (report definition folders, rule definition folders, palette definition folders, data sets, views, catalogs, and so on) in the V6 SASUSER library and that you want to continue to use under V8. For example, if you have report definitions in SASUSER.MYDEFS, then add MYDEFS to the list. However, do not add your SAS base profile (PROFILE) to the list. △

Note: Some of the items in the select list above might not be present in your V6 SASUSER library. Missing items are not a problem. You can omit them from the list or leave them in the list. △

Note: If you use your V8 SASUSER library with other products or applications, then verify that those products or applications work properly after you finish copying. You might need to copy some of the items that are associated with those products and applications. △

Your Site Library/Libraries for SAS IT Resource Management

Note: The instructions in this section offer some choices. Read through the entire section before acting on the instructions, so that your choices are consistent and are appropriate for your site. △

When you install a new version or release of SAS IT Resource Management, a new V8 site library is created. We strongly recommend that, at that point, you run CPSITEUP to merge the contents of your V6 production site library with the new V8 site library. CPSITEUP ensures that your existing production site library information will be retained; it also makes any updates in the new V8 site library available. If you have already run CPSITEUP, then you can skip the rest of this section. If you have not yet run CPSITEUP, then please review the following two sets of instructions that describe the different methods by which you might copy or merge your site libraries. We strongly recommend the use of CPSITEUP, but whichever method you choose, follow that method's instructions only.

If you have not run CPSITEUP but want to do so, then follow these instructions:

1 Locate and read the CPSITEUP member. You can find it

- on z/OS, in the CPSITEUP member in the SAS IT Resource Management PDS named CPMISC
- on Windows, in the file named cpsiteup in the directory named !SASROOT\cpe\sasmisc
- on UNIX, in the file named cpsiteup in the directory named !SASROOT/misc/cpe.

2

Note: For customers who are running SAS IT Resource Management 2.3 or 2.4: The CPSITEUP that ships with SAS IT Resource Management 2.5 has a code change that you must make yourself if you are using the 2.3 or 2.4 version of CPSITEUP. So, if you are running 2.3 or 2.4, then locate CPSITEUP as described above and follow these instructions:

- In CPSITEUP, locate lines 273 and 274. In 2.3 and 2.4, those lines are

```
%else
%do;
```

- Expand those lines to be

```
%else %if %quote(&PRODSITE) ne %quote(&NEWSITE)
            and
          %quote(&PRODSITE) ne %quote(&OLDSITE)
      %then
%do;
```

△

3 Execute the CPSITEUP code according to the instructions within it.

4 If you require multiple V8 libraries:

- on z/OS, allocate space for the additional site library (using the same values for the DSORG=, RECFM=, LRECL=, BLKSIZE=, and SPACE= parameters as the new V8 site library users). Then use the SAS COPY procedure (PROC COPY) to copy the contents of the new V8 site library to the additional site library.
- on Windows, create an additional folder, and copy the contents of the new V8 site library's folder to the additional folder.
- on UNIX, create an additional directory, and copy the contents of the new V8 site library's directory to the additional directory.

If you have not run CPSITEUP and do not want to, then follow these instructions:
These instructions show how to copy the contents of a production V6 site library into the new V8 site library (or into a copy of the new V8 site library) so that the contents are structured correctly for V8. You can copy the contents by using the SAS V8 GUI or by running the equivalent SAS batch job under SAS V8. Because you are not using CPSITEUP, you will not be merging your production site library with any updates that might have been available in the new V8 site library.

1 Determine the location of your V6 site library. You can do this by invoking the SAS IT Resource Management GUI under SAS V6, and using the **Site Options** button on the **Administration** tab in the UNIX/Windows GUI. You can also find the location by following this path in the z/OS GUI:

From the main menu, select **Options** and **Site SAS Options**. Then select **Site Location**.

If in doubt, check with the SAS IT Resource Management administrator at your site.

Then determine the location of your SAS V8 site library (or the location of a copy of the new SAS V8 site library). You can do this by invoking the SAS IT Resource Management GUI under SAS V8, and using the method that is described in the previous paragraph. If in doubt, consult the SAS IT Resource Management administrator at your site or the person who installed SAS IT Resource Management.

2 *Only for z/OS:* Run IEFBR14 or use ISPF 3.2 to allocate space for a SAS V8 site library if the installation of SAS IT Resource Management did not already do this.

Except for using the DSN= and perhaps adding more space if the SAS V6 site library was running out of space, you can use the same amount of space and the same DCB= as for the SAS V6 site library.

3 Using a method that does not also invoke SAS IT Resource Management, invoke the SAS V8 GUI. Or invoke SAS IT Resource Management and exit to SAS. Or using a method that does not also invoke SAS IT Resource Management, construct a batch job that invokes SAS V8.

4 Type the following code in the SAS Program Editor window, edit the location of your SAS V6 and SAS V8 site library for SAS IT Resource Management, and submit the edited code. Or add the following code to the SAS job, edit the locations of your SAS V6 and SAS V8 site library for SAS IT Resource Management, and submit the batch job with the edited code.

   □ For z/OS:

```
LIBNAME OLDSITE 'old_library_name' DISP=SHR ;
LIBNAME SITELIB 'new_library_name'
DISP=OLD ;
PROC COPY IN=oldsite OUT=sitelib ;
RUN ;
```

   where *old_library_name* is the location of your SAS V6 site library for the earlier release of SAS IT Resource Management, and *new_library_name* is the location of your SAS V8 site library for SAS IT Resource Management.

   □ For UNIX or Windows:

```
LIBNAME oldsite 'old_library_name' ACCESS=READONLY ;
LIBNAME sitelib 'new_library_name' ;
PROC COPY IN=oldsite OUT=sitelib ;
RUN ;
```

   where *old_library_name* is the location of your SAS V6 site library for the earlier release of SAS IT Resource Management, and *new_library_name* is the location of your SAS V8 site library for SAS IT Resource Management.

5 If you require multiple SAS V8 site libraries:

   □ on z/OS, allocate space for the additional site library (using the same values for the DSORG=, RECFM=, LRECL=, BLKSIZE=, and SPACE= parameters as the new SAS V8 site library uses). Then use the SAS COPY procedure (PROC COPY) to copy the contents of the new SAS V8 site library to the additional site library.

   □ on Windows, create an additional folder, and copy the contents of the new SAS V8 site library's folder to the additional folder.

   □ on UNIX, create an additional directory, and copy the contents of the new SAS V8 site library's directory to the additional directory.

Note: You might need to edit some of the values in the SAS V8 site library, but do not edit until you finish converting the other objects. Instructions for editing appear later in this document. △

Your PDBs

You will need to make a new SAS IT Resource Management PDB for use with V8 and then copy to it the contents of the V6 PDB. (If you have more than one SAS IT Resource Management PDB, then you will need to do this one time for each PDB. However, you do not need to convert the demonstration PDBs. When SAS IT Resource Management is installed, already-converted demonstration PDBs are available.)

Below are the instructions to create the V8 PDB and copy the contents of the V6 PDB to the V8 PDB so that they are structured correctly for V8. You can copy the contents by using the SAS V8 GUI with SAS IT Resource Management. Or you can copy the contents by running a SAS batch job under SAS V8 and invoking SAS IT Resource Management. Read through the instructions before acting on them, because the choice of method can make a difference. (For example, on z/OS, performing this operation in a batch job is strongly recommended.)

1 Determine the location of your V6 PDB. To remind yourself of the PDB locations, you can invoke the SAS IT Resource Management GUI under SAS V6. Then, in the UNIX/Windows GUI select the **Manage PDBs** button on the **Administration** tab, or in the z/OS GUI follow this path:

   From the main menu, select **PDB Admin** and then **Manage PDBs/Switch PDB**.

   If in doubt, consult the SAS IT Resource Management administrator at your site.

   Then, plan the location of your V8 PDB. If in doubt, consult the SAS IT Resource Management administrator at your site or the person who installed SAS IT Resource Management.

2 On z/OS, allocate space for each of the PDB libraries. On UNIX and Windows, create a directory for the PDB and create subdirectories for each of the PDB libraries. There is no requirement at this point for any other structure or any content in the new PDB.

   - On a z/OS server host:

     Use the z/OS GUI (from the main menu, select **PDB Admin** and **Define New PDB**, and do not add tables). You can also accomplish this by running IEFBR14, or else by using ISPF 3.2 to allocate space for the V8 PDB libraries. Except for using the DSN= and perhaps adding more space if the V6 site library was running out of space, you can use the same amount of space and the same DCB= as for the V6 PDB libraries. You will need to allocate nine libraries for each PDB, all of whose names start with the PDB name as a prefix. If your PDB name is MYPDB, for example, then you will need to allocate the following libraries:

     - *MYPDB.DICTLIB*
     - *MYPDB.DETAIL*
     - *MYPDB.DAY*
     - *MYPDB.WEEK*
     - *MYPDB.MONTH*
     - *MYPDB.YEAR*
     - *MYPDB.ADMIN*
     - *MYPDB.PDBWORK*
     - *MYPDB.COLLECT*

     A sample job to allocate a PDB is in the CMPDBALC member of the SAS IT Resource Management CPMISC PDS. You might need to adjust the SPACE= parameters and possibly the UNIT= and BLKSIZE= parameters.

     The requirements for allocating multivolume SAS data sets have changed in SAS V8. Refer to SAS V8 OnlineDoc for further information.

   - On a UNIX or Windows server host:

     In the *next step*, you will copy the contents of the V6 PDB to the V8 PDB. If you plan to do this in batch/background mode, in *this* step you must create directories for the V8 PDB and its libraries. If you plan to copy interactively, skip *this* step; you do not need to pre-allocate the directories.

     You can create directories in any of these ways:

- use the UNIX/Windows GUI (from the main window, select the **QuickStart Wizard** or the **Create PDB Wizard**)
- use the SAS IT Resource Management %CPSTART macro in a SAS batch job
- use the mkdir command at the operating system prompt.

If you want your PDB to have a name like /my/pdb (on UNIX) or c:\My\Pdb (on Windows), for example, then you will need to create the following directories:

- */my/pdb* or *c:\My\Pdb*
- */my/pdb/dictlib* or *c:\My\Pdb\Dictlib*
- */my/pdb/detail* or *c:\My\Pdb\Detail*
- */my/pdb/day* or *c:\My\Pdb\Day*
- */my/pdb/week* or *c:\My\Pdb\Week*
- */my/pdb/month* or *c:\My\Pdb\Month*
- */my/pdb/year* or *c:\My\Pdb\Year*
- */my/pdb/admin* or *c:\My\Pdb\Admin*
- */my/pdb/pdbwork* or *c:\My\Pdb\Pdbwork*
- */my/pdb/collect* or *c:\My\Pdb\Collect*

3 Copy the contents of the V6 PDB to the V8 PDB.

- Using the GUI:

  If the V8 GUI is not already invoked, invoke it. Then, invoke SAS IT Resource Management software with one of the SAS V8 demonstration PDBs as the active PDB.

  - If the PDB to be converted is on a z/OS server:

    Follow this path from the main menu: Select PDB Admin and then select**Manage PDBs/Switch PDBs**. Select the name of the V6 PDB from the **ItemActions** menu and then select **Copy PDB**. Type or select the location of the V8 PDB and select **Run**.

    When the confirmation dialog box appears, select **OK**. When the copying is finished, select **Goback** to return to the PDB Admin window. From the list of known PDBs, select the name of the V6 PDB. From the **ItemActions** menu, select **Remove PDB Entry**.

    Note: The batch method of copying the PDB is preferred, because copying the PDB in a TSO session can take a very long time. △

  - If the PDB to be converted is on a UNIX or Windows server:

    From the **Administration** tab, select **Manage PDBs**. From the list of known PDBs, select the name of the V6 PDB. Then, from the **File** menu, select **Copy**. In the **To** field, type or select the name of the V8 PDB and select **OK**. When the copying is finished, from the list of known PDBs, select the name of the V6 PDB. From the **File** menu, select **Remove PDB Entry**.

    Note: The above z/OS and UNIX/Windows paragraphs assume that you retained the original name and location for the V6 PDB and used a new name and location for the V8 PDB. If instead you renamed and moved the V6 PDB and if you used the original name and location for the V8 PDB, then you do not need to remove the original name from your V8 list of known PDBs. However, you might want to change the name of the V6 PDB in your V6 list of known PDBs, V6 remote profiles, and so on. △

- Using a batch job:

  If a batch job that invokes SAS V8 has not already been constructed, construct one now. (On UNIX and Windows, be sure that the invocation uses your V8 SASUSER library.) Then, add a call to the %CPSTART macro to invoke SAS IT Resource Management software with one of the V8 demonstration PDBs as the active PDB. Add a call to the %CPDBCOPY macro to copy the contents of the V6 PDB to the V8 PDB, and add a call to the %CPSTART macro to activate the V8 PDB.

  Note that in the first call to the %CPSTART macro, you need to specify the ROOT= parameter so that it can point to the location of the SAS IT Resource Management software. Here are two examples:

  - If the PDB to be converted is on a z/OS server:

```
%cpstart(mode=batch,
         root=location-of-IT-Service-Vision,
         pdb=location-of-SASV8-demonstration-PDB,
         disp=shr) ;
%cpdbcopy(location-of-SASV6-PDB,location-of-SASV8-PDB);
%cpstart(mode=batch,
         root=location-of-IT-Service-Vision,
         pdb=location-of-SASV8-PDB,disp=old);
```

  - If the PDB to be converted is on a UNIX or Windows server:

    Note: For Windows: Before running the following code, check that you do not have the PDB directory or its subdirectories locked. That is, make sure that the **cd** command (or Windows Explorer or My Computer) is not set to the PDB directory or to any of its subdirectories. △

```
%cpstart(mode=batch,
         root=location-of-IT-Service-Vision,
         pdb=location-of-SASV8-demonstration-PDB,
         access=readonly);
%cpdbcopy(location-of-SASV6-PDB,location-of-SASV8-PDB);
%cpstart(mode=batch,
         root=location-of-IT-Service-Vision,
         pdb=location-of-SASV8-PDB,access=write);
```

  If you use the SAS IT Resource Management GUI, check that your V8 list of known PDBs has the appropriate names and locations, and update any that need to be changed. You might want to do the same for your V6 list of known PDBs. Access and, if necessary, edit your list of known PDBs as described above. (In the section "Your PDBs," see step 3.) For more information about calling these macros, see the SAS IT Resource Management macro reference documentation.

4 Your site might have a .QS PDS or /qs directory "under" your PDB. That PDS or directory is not a SAS object and its related PDSs and subdirectories are not SAS objects. If you want to make a copy of them "under" the V8 PDB, then you can use an operating system command, such as the following:

- On z/OS, you can use ISPF 3.3 or IEBCOPY in batch or you can use a similar utility. There are multiple PDSs with a QS qualifier. Make sure that you copy each of them.
- On Windows, you can use Windows Explorer or you can use XCOPY or a similar command.

- On UNIX, you can use

```
cp -R location-of-SASV6-PDB/qs location-of-SASV8-PDB/qs
```

Also, if you want to use the QuickStart jobs with the V8 PDB, then you will need to edit the *x*REPORT, *x*PROCESS, and possibly *x*FTPHTML files in the *location-of-SASV8-PDB*/*qs*/*cntl* or *location-of-SASV8-PDB**qs**cntl* directory or in the *location-of-SASV8-PDB.QS.CNTL* PDS, and change the values of the PDB= and ROOT= parameters that are contained in those files.

5 Your site might have an archive "under" or associated with your PDB. An archive is used only in read mode and thus its archive libraries do not require conversion. (If you need to restore data from these archive libraries later, then you will need to remember their original names and locations.)

MXG

If you install a new version of MXG, then part of the installation procedure is to run a job that creates and populates a SAS library of formats for use with MXG. When you run the job under SAS V8, a V8 library will be the result.

If you want to continue to use your existing (SAS V6) format library until you install the next version of MXG, then you can convert the existing format library from V6 to V8. First, allocate space for the new library, as described in the section "Your Site Library/Libraries for SAS IT Resource Management" on page 246. Then, to copy the catalog that contains the formats, submit SAS code like this through the SAS Program Editor window or by using a SAS batch job:

```
LIBNAME SASV6LIB 'location-of-SASV6-format-library-for-MXG'
        DISP=SHR ;
LIBNAME SASV8LIB 'location-of-SASV8-format-library-for-MXG'
        DISP=OLD ;
PROC CATALOG CAT=SASV6LIB.FORMATS ;
COPY OUT=SASV8LIB.FORMATS ;
RUN ;
QUIT ;
```

Other Libraries Related to SAS IT Resource Management

You might have other V6 libraries that are related to SAS IT Resource Management, such as one or more libraries for custom report definitions, custom rule definitions, custom palette definitions, and source code for exits.

Convert each of these libraries in the same way that you converted the site library. However, for clarity you might want to use different librefs. For example, instead of *oldsite* you might want to use *oldlib*, and instead of *sitelib* you might want to use *newlib*. And the locations that you edit are the locations of the old and new custom library, not the old and new site library.

Your DeskTop Reporter

The DeskTop Reporter, as shipped with SAS IT Resource Management, is already converted.

If the only customization or modification that you have made to your reports is to use your own data, then you need to convert the job that contains the call to the %CPEISSUM macro. To convert the job, follow the instructions below, in "Your Batch Jobs (Scripts and CLISTs)" on page 253.

If you have added new reports or modified existing reports by using SAS/EIS software, then follow the instructions in the section "Additional Tasks to Consider for DeskTop Reporter" on page 255 to convert.

Remote Profiles

If you use any remote profiles, then edit the locations in them to refer to the V8 objects. To do this, you can follow these steps:

1. From the **Administration** tab, select **Manage PDBs ▶ Locals ▶ Remote Profile**.
2. Then, for each profile that you use, select the profile name and then select **File ▶ Open**.
3. Review the contents of all fields on all of the tabs, and edit the values for any software or objects whose location has changed.
4. Then select **OK ▶ OK ▶ Close**.

Note: These instructions assume that the original names and locations were retained for the V6 objects and that new names and locations were used for the V8 objects. If instead the V6 objects were renamed and moved and the V8 objects used the original names and locations, then you do not need to edit your V8 remote profiles. However, you might want to edit the V6 remote profiles. △

PDB Properties

If you have a PDB that has an archive, then decide whether you want the V8 PDB to use the same archive as the V6 PDB or to use a new archive. (Either location is acceptable.) Then, edit those properties of the V8 PDB that refer to the location of the archive. (If nothing was specified, then the location of the archive is probably changing, because the default location is "under the PDB" and the location of the PDB is probably changing.)

- To edit in batch:

  Call the %CPSTART macro to activate that PDB. Then, call the %CPPDBOPT macro to specify the values of its archive parameters.

- To edit by using the z/OS GUI:

  Activate that PDB and then, on the main menu, select **PDB Admin ▶ Set Active PDB Options**. Then, edit the values of the archive parameters and select **OK**.

- To edit by using the UNIX/Windows GUI:

  From the **Applications** tab, select **Manage PDBs**. Activate the PDB if it is not already activated, and then select **Properties** and the **Archive** tab. Then, edit the values of the archive parameters, and select **OK** and then **Close**.

Your List of Known PDBs

If you use the SAS IT Resource Management GUI, then edit your list of known PDBs if you are on a client host. For instructions, see step 3 in the section “Your PDBs” on page 248.

Your Batch Jobs (Scripts and CLISTs)

If you have one or more batch jobs, then save a V6 version of the job and edit one of the copies to refer to the locations of the SAS IT Resource Management software and the V8 versions of the objects. Also make the following changes where you invoke SAS:

- On z/OS:
  - In SAS V6, where you used the CONFIG file in

    ```
    CPMISC(CMCONFIG)
    ```

for SAS V8, instead use the one in

```
CPMISC(CMCONFI8)
```

- In SAS V6, where you used, in your own custom SAS code,

  ```
  ERRORABEND
  ```

  for SAS V8, because SAS stops on more conditions, you might instead want to use

  ```
  NOERRORABEND
  ```

- On UNIX:
  - In SAS V6, where you used the options

    ```
    -dmsbatch, -batch,-terminal, and -fsdevice ascii.vt100
    ```

    for SAS V8, remove those options and instead add the option

    ```
    -noterminal
    ```

- On Windows:
  - In SAS V6, where you used the option

    ```
    -dmsbatch
    ```

    for SAS V8, remove that option and instead add the option

    ```
    -noterminal
    ```

Similarly, save and then edit one of the copies of scripts, CLISTs, and any other such files that refer to locations of the software and objects.

Your PGMLIB

Note: The following paragraphs describe the same operation that you perform with every new PGMLIB if you have an installed user-written table in the earlier PGMLIB. The operation is not related to the conversion from V6 to V8. △

SAS IT Resource Management includes a V8 PGMLIB. If you installed a user-written table definition in your V6 PGMLIB, then you might want to install that table definition in your V8 PGMLIB.

To do so, write a batch job that does all of the following:

- invokes SAS V8
- calls the %CPSTART macro to start SAS IT Resource Management and to activate a PDB that contains the table
- then calls the %CPDDUTL macro to apply the INSTALL TABLE control statement. (For more information about using the %CPDDUTL macro and about the %CPDDUTL control statement INSTALL TABLE, see “Chapter 4: Using the %CPDDUTL Macro” in the SAS IT Resource Management macro reference documentation.)

Verifying

Ideally, run both the V6 and V8 systems in parallel for two weeks and compare the results each day. If you have a very large site, running in parallel might not be possible, but you should at least run parallel test systems before converting the production system to V8.

At minimum, compare the process summary table and reduce summary table in the V6 and V8 logs.

Additional Tasks to Consider for DeskTop Reporter

The following instructions describe how to update your DeskTop Reporter if you have added or modified reports (other than modifying reports to use your own data).

- Data Conversion

  Data can be converted from V6 to V8 by using the SAS COPY procedure (PROC COPY) with the appropriate library engines. You can submit the following SAS code:

  - For z/OS:

```
LIBNAME SASV6Data SASV6 'SASV6 path' DISP=SHR ;
LIBNAME SASV8Data SASV8 'SASV8 path' DISP=OLD ;
PROC COPY IN=SASV6Data OUT=SASV8Data
           MEMTYPE=(DATA MDDB VIEW) ;
RUN ;
```

  - For UNIX and Windows:

```
LIBNAME SASV6Data SASV6 'SASV6 path' ACCESS=READONLY ;
LIBNAME SASV8Data SASV8 'SASV8 path' ;
PROC COPY IN=SASV6Data OUT=SASV8Data
          MEMTYPE=(DATA MDDB VIEW) ;
RUN ;
```

  Note: Views do not need to be re-created from the source code. The VIEW member can be copied by using PROC COPY. It is recommended that you keep the source code in any case. △

- SAS V8 Repositories

  In V8, metadata is stored in repositories, not in libraries. When you are migrating the metabase, the repository name in V8 should be the same as the metabase libname in V6. If they are not the same, then the applications that use the metabase definitions will be unable to find them, and most of the attributes will need to be set up again.

  Note: Repository names are case sensitive. Always use uppercase for repository names that are the target of V6 conversions. △

  To Create a Repository

  1. Type REPOSMGR on the command line and select **Enter** or **Return**.
  2. Select the Repository Registration.
  3. Select the **New** button.
  4. Type the repository name (in uppercase for V6 conversions), path, and description.

  Note: Repositories will be created for you for each V6 metabase during Metabase Conversion. △

- Metabase Conversion

  This section covers only non-HOLAP entries in the metabase. The metabase libraries can be converted from the V6 libraries to the V8 repository paths by using the SAS COPY procedure (PROC COPY). You can submit the following SAS code:

  - For z/OS:

```
LIBNAME SASV6MetLib SASV6 'SASV6 path' DISP=SHR ;
LIBNAME SASV8RepLib SASV8 'SASV8 path' DISP=OLD ;
PROC COPY IN=SASV6metlib OUT=SASV8RepLib ;
SELECT metabase1 metabase2 . . . sasmbc sasmbi ;
```

```
RUN ;
```

- For UNIX and Windows:

```
LIBNAME SASV6MetLib SASV6 'SASV6 path' ACCESS=READONLY ;
LIBNAME SASV8RepLib SASV8 'SASV8 path' ;
PROC COPY IN=SASV6metlib OUT=SASV8RepLib ;
SELECT metabase1 metabase2 . . . sasmbc sasmbi ;
RUN ;
```

The metabase is now in V8 file format, but the internal structure of the metabase needs to be converted to a repository structure.

To convert the metadata:

1. Open the Metabase window in SAS/EIS.
2. Select **Edit ▶ Convert**.
3. From the list, select the repository path to which you copied your V6 metabase.
4. Add your metabases to the selected list.
5. Select **OK** to perform the conversion.

- Application Conversion

Applications can be converted by using the SAS COPY procedure (PROC COPY). You can submit the following SAS code:

- For z/OS:

```
LIBNAME SASV6AppLib SASV6 'SASV6 Path' DISP=SHR ;
LIBNAME SASV8AppLib SASV8 'SASV8 Path' DISP=OLD ;
PROC COPY IN=SASV6AppLib OUT=SASV8AppLib ;
SELECT SASV6ApplDB ;
RUN ;
```

- For UNIX and Windows:

```
LIBNAME SASV6AppLib SASV6 'SASV6 Path' ACCESS=READONLY ;
LIBNAME SASV8AppLib SASV8 'SASV8 Path' ;
PROC COPY IN=SASV6AppLib OUT=SASV8AppLib ;
SELECT SASV6ApplDB ;
RUN ;
```

Individual applications will work if the libname that is used in V8 is not the same as the libname that is used in V6. However, application windows will be unable to find the applications unless they are referenced by the same libref as they were in V6. Therefore, always keep the libref the same.

- Mixing V6 and V8 Objects

Both SAS V6 applications that are ported to V8 and native V8 objects can be mixed on the same application window.

Special Considerations for the AIX Platform

You cannot use the SAS COPY procedure (PROC COPY) to copy SAS V6 catalogs on AIX to SAS V8 catalogs. Thus, as you follow the procedures that are outlined in this document, you might see error messages similar to the following:

```
ERROR: File format version 6 catalogs are not supported by this release.
```

```
ERROR: File TDICTLIB.CPFMTS.CATALOG has not been saved because copy could
not be completed.
```

You might also see these error messages when running %CPDBCOPY, because this macro also uses the SAS COPY procedure.

If, based on the information in this migration document, you decide that it is necessary to migrate the catalog(s), or if the instructions state that the catalog(s) must be migrated, then you must do the following:

- Use the SAS CPORT procedure (PROC CPORT) on SAS V6 to convert the catalog(s) into transport file format.
- Then, use the SAS CIMPORT procedure (PROC CIMPORT) on SAS V8 to add the catalog(s) to a SAS V8 library.
- For example, first edit the following code and submit it on your SAS V6 system to create a transport file that contains the necessary catalogs. (The example below adds two SAS catalogs, CPFMTS and CPVER, to the transport file. You can include additional catalogs in the SELECT statement as necessary.)

```
LIBNAME SASV6LIB 'SASV6_library' ;
PROC CPORT LIB=SASV6LIB FILE='location-of-external-file' ;
SELECT CPFMTS CPVER ;
QUIT ;
```

  In this example, *SASV6_library* is the location of the SAS V6 library that contains the catalogs to be ported.
- Next, if you need to FTP the transport file, then you should ensure that it is transferred in binary mode.
- Finally, you need to import the catalogs in the transport file to your SAS V8 system. Edit the following code and submit it on your SAS V8 system:

```
LIBNAME SASV8LIB 'SASV8_library_name' ;
PROC CPORT LIB=SASV8LIB INFILE='location-of-external-file' ;
QUIT ;
```

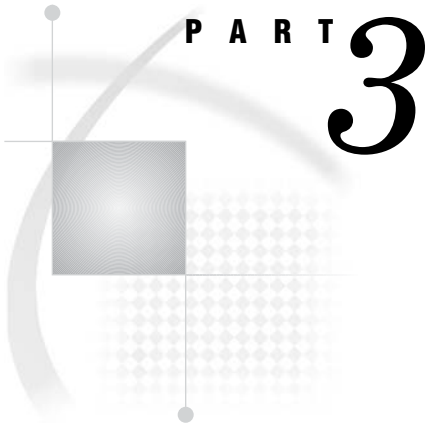
where *SASV8_library_name* is the location of the SAS V8 library to contain catalogs.

Note: For further information about the SAS CPORT and SAS CIMPORT procedures, refer to your SAS System documentation. △

Migrating Your SAS IT Resource Management Environment from SAS 8 to SAS 9

This document is for sites at which SAS IT Resource Management currently runs under SAS 8 but is migrating to run under SAS 9. You do not need this document if you are installing SAS IT Resource Management for the first time or if you already migrated your SAS IT Resource Management environment to SAS 9.

You can view and/or print the document from this Web site: http://support.sas.com/documentation/onlinedoc/itsv.



PART 3

Administration

CHAPTER 13

Administration: Concepts and Tools

Overview of Administration

Overview of Administration Concepts and Tools

After SAS IT Resource Management is installed:

1. Set up one or more SAS IT Resource Management servers with one or more data collectors. As a result of setting up the server(s) and collectors,
   - the data from your collectors can be extracted, transformed, and loaded into one or more SAS IT Resource Management PDBs
   - batch jobs are scheduled (typically, one time each night) to process the data, summarize the data, and analyze and report on the data.

   Note: The server interfaces have some interactive functionality for reporting, but the full interactive functionality for reporting is in the client interface. △
2. Set up one or more SAS IT Resource Management clients. As a result of setting up the client(s),
   - you and other users can use interactive tools for ad hoc data investigation, analysis, and reporting
   - you and other users can use interactive tools to create and save report definitions and rule definitions. Saved report and rule definitions can be run interactively and/or run (typically, each night) from the batch job that analyzes and reports on the data.
3. Set up your Web browser (and others set up their Web browsers) to view the reports that are generated interactively or in batch mode.

After setup, you might want to use SAS IT Resource Management

- without administrative changes between releases.
- with simple administrative changes that you can make through the GUIs or in batch mode. For example, you might want to change which data is kept and how long it is kept, and you might want to change which summary statistics are calculated on the data.
- with more substantial changes that you can make by using the SAS IT Resource Management extensions. For example, you might want to change whether there is checking for duplicate data and how the checking is done, and you might want to use process exits to remove data about test systems.

The main objects that you work with when you do administrative work are

- one or more site libraries. The main uses of a site library are
  - to contain site-wide defaults that are copied to a PDB when the PDB is created
  - to contain site-wide information (such as the list of site holidays) that is used when data is processed into a PDB.
- one or more PDBs. The main uses of a PDB are
  - to contain data that is extracted, transformed, and loaded from one or more collectors
  - to contain summarized data
  - to provide data for analysis and reporting.

  Logically, a PDB contains detailed and summarized data, as well as information (metadata) that is needed for managing the data. Physically, each PDB consists of a coordinated set of nine SAS libraries that contain data and metadata.

  Within a PDB, data is grouped by one or more levels and one or more tables. A group is named *level-name.table-name*.
  - *Level-name* groups data with a common amount of summarization. Unsummarized data is in the detail level. Summarized data is in the day level (which is summarized at least by day), week level (which is summarized at least by week), month level (which is summarized at least by month), and year level (which is summarized at least by year).

    Level is a logical term, and is represented physically by a SAS library within the PDB. Thus, detail-level data is in the PDB's DETAIL library, day-level data is in the PDB's DAY library, and so on.
  - *Table-name* groups data about a common topic.

  Here is a brief summary about PDBs:
  - A PDB has properties and contains tables. A PDB corresponds to one (or more) collectors.
  - A table has properties and contains variables. A table corresponds to one (or more) topics.
  - A variable has properties and contains values. A variable corresponds to one (or more) metrics.

    For each measurement in the detail level, you can choose to have zero, one, or more statistics calculated at a summary level. For example, you can choose mean, minimum, maximum, range, and so on.
- For information about less commonly encountered administration objects, see " SAS IT Resource Management Software Components" on page 263.

For more information about the main administration tasks and objects, see the following topics:

- "Overview of Setup" on page 37
- "Overview of Site Libraries" on page 270
- "Overview of Whole PDBs" on page 296
- "Overview of Levels" on page 337
- "Overview of Tables" on page 353
- "Overview of Variables" on page 378
- "Overview of Data" on page 409
- "Overview of Extensions" on page 425.

SAS IT Resource Management Software Components

Before you use SAS IT Resource Management to collect, analyze, or report on data, you might want to become familiar with the structure of the software. You will be using the following libraries and software components. (If an item is specific to an operating environment, the operating environment is indicated.)

- "Program Library" on page 263
- "Site Library" on page 264
- "Performance Data Base and Its Libraries" on page 264
- "SASUSER Library" on page 264
- "MISC Directory (UNIX) and SASMISC Directory (Windows)" on page 265
- "CPMISC PDS (z/OS)" on page 265
- " SAS IT Resource Management Macros" on page 265
- "SCHEMA Directory (UNIX) and MIBS Directory (UNIX and Windows)" on page 266
- "MXG Software" on page 266.

These libraries and components are described in the following sections.

Program Library

The SAS IT Resource Management program library (whose SAS libref is PGMLIB) is the SAS data library that contains the SAS IT Resource Management software for your current host. It contains the master data dictionary (supplied table and variable definitions), the supplied report and rule definitions, and the SAS IT Resource Management client or server software programs for your current host.

When SAS IT Resource Management is updated, the libraries and information in PGMLIB are updated. You must have write access to add or modify any of the items in PGMLIB. However, you do not need to have write access to PGMLIB in order to view, search, or copy from the list of supplied table definitions, supplied variable definitions (regular, derived, and formula), supplied report definitions, and supplied rule definitions.

If you have more than one release of SAS IT Resource Management installed, then you will have a separate version of the program library for each release.

Note: On some hosts, additional software for the SAS IT Resource Management GUI is installed in the CPSYSLIB library. △

The PGMLIB library is created when you install SAS IT Resource Management.

- On z/OS, the location is

```
the.location.where.you.installed.sas.at.your.site.itrm.pgmlib
```

.

- On UNIX, the location is

  ```
  .../saspgm/cpe/pgmlib
  ```

- On Windows, the location is

  ```
  ...\cpe\pgmlib
  ```

Site Library

The site library contains your site-related information, such as the list of site holidays, the default work-shift schedules, the default start-of-week setting, and other defaults. When you or others at your site create a new PDB, the default properties for that PDB are copied from the site library. Then, you can customize the settings for that PDB.

If you are processing data for multiple sites or groups within your organization, then you can have a separate site library for each site or group so that you can customize the default settings for each group.

The locations of the default site library are as follows:

- On z/OS, the location is

  ```
  the.location.where.you.installed.sas.at.your.site.itrm.sitelib
  ```

.

- On UNIX, the location is

  ```
  .../saspgm/cpe/sitelib
  ```

- On Windows, the location is

  ```
  ...\cpe\sitelib
  ```

To modify settings or add items to your site library, you must have write access to this library. However, you can read and copy items from the active site library if you have read access to the library.

For more information about accessing your site library or setting site-specific values such as holidays, the shift code for holidays, and PDB defaults, see “Overview of Site Libraries” on page 270.

Note: If your site processes data for multiple sites, then you might have a separate site library for each site. △

Performance Data Base and Its Libraries

For more information, see “What Is a PDB?” on page 7 and “Overview of Whole PDBs” on page 296.

SASUSER Library

The SASUSER library is a SAS library that is used by your interactive SAS session for storing user-specific information, such as the locations of PDBs that you have used and any customizations that you specify for SAS options.

When you start the SAS System, a SASUSER library is automatically created for you, if one does not already exist. When you customize keys, menus, or other items within SAS IT Resource Management, these modifications are automatically saved in your SASUSER library. By default, any source, log, or report files that you save are saved in this library.

Note: By default, this library *is not* used in batch mode on z/OS and *is* used in batch mode on UNIX and Windows. For information about the SASUSER library that is specific to your operating environment, refer to the SAS Companion documentation for your operating environment and your current version of SAS. △

MISC Directory (UNIX) and SASMISC Directory (Windows)

The MISC directory in UNIX and the SASMISC directory in Windows contain the following information: examples for running SAS IT Resource Management, the autoexec file, and the *itsv* command for starting SAS and SAS IT Resource Management. By default, this library is located in the directory where SAS is installed.

- The default location for UNIX is

```
.../misc/cpe
```

- The default location for Windows is

```
...\cpe\sasmisc
```

If you need to modify a file in the MISC or SASMISC directories, copy the file from this directory to another location and modify the copy.

On z/OS, CPMISC PDS is used for a similar purpose. For more information about CPMISC, see the next topic.

CPMISC PDS (z/OS)

The CPMISC data set is a standard z/OS partitioned data set (PDS).

- The default location for z/OS is

```
the.location.where.you.installed.sas.at.your.site.itrm.cpmisc
```

.

CPMISC contains information such as

- JCL examples for processing, reducing, and reporting on supported data collectors in batch.
- CMAUTOEX, which is the SAS autoexec that contains start-up parameters for SAS IT Resource Management.
- CMCPE CLIST, which is the default name for the CLIST with which you start SAS and automatically invoke the SAS IT Resource Management server on z/OS.

A list of all of the members in this PDS and a short description of each member's content are in the $README member.

If you or the SAS IT Resource Management administrator at your site needs to modify a member from CPMISC, copy the member to another location and modify the copy.

On UNIX the MISC directory is used for a similar purpose, and on Windows the SASMISC directory is used for a similar purpose. For more information about MISC and SASMISC, see the previous topic.

SAS IT Resource Management Macros

SAS IT Resource Management provides macros that can run in batch mode. Macros that require write access to the PDB, such as %CPPROCES and %CPREDUCE, or any macros that update the data dictionary must run on one of the SAS IT Resource Management server hosts. Reporting macros can run on a SAS IT Resource Management server host or client host.

The SAS IT Resource Management GUI is based on these macros. When you select tasks within the GUI, the macros are executed to perform the tasks. If you want to run jobs in batch mode, you can also write batch jobs that include calls to these macros.

- On z/OS, the macros are located in

  `the.location.where.you.installed.sas.at.your.site`.autolib

  .
- On UNIX, the macros are located in the SAS autocall library in

  `.../sasautos`
- On Windows, the macros are located in the directory

  `...\cpe\sasmacro`

For more information about SAS IT Resource Management macros, see the SAS IT Resource Management macro reference documentation .

SCHEMA Directory (UNIX) and MIBS Directory (UNIX and Windows)

On UNIX, the schema directory contains schema files and the MIBS directory contains MIB files. These are copies of the files that are used by various network and systems management platforms.

- On UNIX, MIB files are in the directory

  `!sasroot/saspgm/cpe/mibs`

  and schema files are in the directory

  `!sasroot/saspgm/cpe/schema`
- On Windows, MIB files are in the directory

  `...\cpe\mibs`

  There are no schema files on Windows.
- There are no MIB or schema files on z/OS.

MXG Software

When you are running the SAS IT Resource Management server on z/OS, MXG software is available with SAS IT Resource Management.

Your site can customize the MXG code and also utilize the SAS formats that are provided by MXG so that the data values are formatted as they are read in and written to the data sets.

MXG software consists of three libraries, which have the following names in MXG documentation:

- *MXG.MXG.SOURCLIB*: This library contains the SAS macro code that is used to build the staging SAS data set for each table. This is a standard z/OS PDS (partitioned data set), and it is created when MXG is installed.
- *MXG.USERID.SOURCLIB*: This library contains any modified versions (for example, to make use of exits) of the members in MXG.MXG.SOURCLIB. This is a standard z/OS PDS, just like MXG.MXG.SOURCLIB, but it is created by your site's SAS IT Resource Management and/or MXG administrator.
- *MXG.MXG.FORMATS*: This library contains the SAS formats and informats that are used by MXG. When MXG is installed, this SAS library is created by running the JCLINSTL job in MXG.MXG.SOURCLIB.

Note: MXG uses the DDname SOURCLIB for the concatenation of SOURCLIB libraries. The order of the concatenation is important. For any given member name, MXG uses the first member that it encounters in the concatenation. Thus, the intended

order is MXG.USERID.SOURCLIB and then MXG.MXG.SOURCLIB. MXG and SAS IT Resource Management use the DDname LIBRARY for the SAS library named MXG.MXG.FORMATS. △

For more information about setting up MXG and SAS IT Resource Management to process data into MXG-based tables, see "Using MXG-Based Tables (CMAPP2): Introduction" on page 133.

CHAPTER

14

Administration: Working with Site Libraries

Working with Site Libraries: Main Topics

Overview of Site Libraries

An SAS IT Resource Management site library is a SAS library that contains site-specific information for use with SAS IT Resource Management. A site library contains some site-wide information and also contains some site-wide defaults for PDB properties.

- The side-wide information consists of the list of holidays and the shift code for holidays.
- The defaults for PDB properties are the default archive device, the default archive path, the default archive parameters, the default archive engine, the default active collector, the default Daylight Saving Time (DST) definition, the default Greenwich Mean Time (GMT) deviation, the default description of the shift code for holidays, the default for the start of a week, the defaults for the descriptions of work-shift codes, and the default work-shift schedule.

  On z/OS, the default MXG views property is an additional default.

  Note: The site-wide information and the defaults for PDBs are called site library properties (also known as site library options). △

If you are a SAS IT Resource Management administrator, you might want to have more than one site library on a server host. For example, you might want to

- have more than one set of defaults for PDBs
- process data from subsidiaries or divisions that have different lists of holidays
- have, for analysts, both a "read-only" site library (with information that you provide) and a "write" site library (that enables users to upload palettes, report definitions, report definition folders, and exception rules folders to make them available for site-wide use).

If you are using the SAS IT Resource Management client, you have a site library on the client host.

When you specify to SAS IT Resource Management which site library you want to use, SAS IT Resource Management assigns a SAS libref of SITELIB to the site library. If you later change to a different site library as your active site library, SAS IT Resource Management assigns SITELIB to that site library. Thus, at any one time, only one site library can be designated as your active site library in your SAS IT Resource Management session, and it is known as SITELIB.

The values of the defaults for PDB properties are obtained, changed, and used as follows:

- When a site library is first used (and, to be used, it must at that time be the active site library) SAS IT Resource Management automatically copies the values of the defaults from PGMLIB to the site library. You can then modify the values in the site library if you prefer other values.
- When a PDB is created, SAS IT Resource Management automatically copies the values of the defaults from the active site library to the PDB. You can then modify the values in the PDB if you prefer other values.

A site library might contain formats, report definition folders, palette folders, exception rule folders, and any other information of site-wide interest.

- An administrator can use a site library to store formats, to create folders for report definitions, palettes, and exception rules, and to store any other information that needs to be available for site-wide use.
- Similarly, an analyst can use a site library to store formats and palettes, report definitions, report definition folders, exception rule folders, and any other information that the analyst wants to make available for site-wide use.

  Note: An administrator might not want to give analysts "write" access to the main site library. In that case, the administrator can provide an alternate, writable site library (or any other SAS library) for this purpose. △

The details for working with a site library are described in the topics below, in alphabetical order. For additional information, select **`Help`** in the windows that are described in the topics.

Note: You can specify which site library to use as the active site library while you start or restart SAS IT Resource Management. You can also specify which site library to use as the active site library while you are running SAS IT Resource Management. (For more information about specifying the site library, see "Activating a Site Library" on page 272.) If you do not specify which site library to use, for a given task SAS IT Resource Management uses the default site library for the host on which that task will run. (For more information about the default site library, see "Viewing/Editing the Value of Default Site Library" on page 280.) △

When you access a site library, you access it in "read-only" mode or "write" mode. The mode places a lock of one kind or another on the site library. The lock can prevent another person from accessing the site library, so it is good to release the lock as soon as you finish using the site library. You can release the lock on a particular site library by activating a different site library or by exiting from SAS IT Resource Management and SAS.

The following topics about site libraries are in alphabetical order. In the topics, assume that SAS IT Resource Management is already started in batch mode or in interactive mode:

Activating a Site Library

You can activate a site library

□ by using the %CPSTART macro in batch mode. For more information about the %CPSTART macro, see the SAS IT Resource Management macro reference documentation .
□ by using the SAS IT Resource Management GUI for z/OS.
□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Activating a Site Library (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can activate a site library by following these steps:

1 From the main menu, select **Exit**. A menu opens. From the menu, select **Exit SAS IT Resource Management only**.
2 If the SAS Program Editor window is not visible, issue the NEXT command repeatedly to cycle through the windows until the Program Editor window is visible.

   Note that, by default, the NEXT command is mapped to the F8 key or SHIFT+F8 key.
3 Do one of the following actions:
   □ If the SAS Program Editor window has pull-down menus, in the body of the Program Editor window, type

```
%CPSTART(SITELIB=name.of.site.library.that.you.want.to.use,
         SITEACC=access.mode.that.you.want.to.use) ;
```

   where the name is fully qualified and the access mode is SHR or OLD.

   Select **Run**. A menu opens. From the menu, select **Submit**.
   □ If the SAS **Program Editor** window has a command prompt, type the command

```
ITRM SITELIB=name.of.site.library.that.you.want.to.use
     SITEACC=access.mode.that.you.want.to.use
```

   where the name is fully qualified and the access mode is SHR or OLD.

   Press RETURN or ENTER.
4 Issue the NEXT command repeatedly to cycle through the windows until the SAS IT Resource Management window is visible.

Activating a Site Library (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can activate a site library by following these steps:

1 On the main window’s **Administration** tab, select **Site Options**. The Set Site-wide Options window opens.

2 In the upper-left corner of the window, select the **Change Sitelib** icon.
3 In the **Path** field, edit the value so that it has the name of the site library that you want to activate.
4 In the **Access** field, select the access mode that you want to use.
5 In the **Location** field, select the file system on which the site library is located.
6 To return to the Set Site-wide Options window, select **OK**.
7 To return to the main window, select **Close**.

Backing Up a Site Library

To back up a site library:

- On z/OS, a site library is a single z/OS data set. Back up the site library as you would back up any operating system data set.

  Note: Do not reblock the library. △
- On Windows, a site library is a folder. Back up the site library as you would back up any folder.
- On UNIX, a site library is a directory. Back up the site library as you would back up any directory.

Copying a PDB's Options to a Site Library

You can copy one or more of a PDB's properties to a site library's same-named properties in order to make them the default values of the PDB properties for the site

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Copying a PDB's Options to a Site Library (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can copy one or more of the options from the active PDB to the active site library by following these steps:

1 If your active PDB is not the one that you want to use, change your active PDB. For more information about specifying the active PDB, see "Activating a PDB" on page 299.
2 If your active site library is not the one that you want to use, change your active site library. Change your access mode to "write" if you do not have already have write access to your active site library. For more information about changing your site library and/or changing your access mode to the active site library, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
3 On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens. The **Copy** button is in the lower-left corner of the window.
4 Select **Copy**. The Copy Options window opens.
5 In the **Copy From** field, select **PDB Options**.
6 In the **Copy To** field, select **Site Options**.

7 In the **Copy Selected Options** field, select the option(s) that you want to copy and deselect the options that you do not want to copy.

Note: Not all PDB options are available in this field. For others, use the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation. △

8 To return to the Set Site-wide Options window, select **OK**.

9 To return to the main window, select **OK**.

Copying a Site Library's Options from PGMLIB

You can restore the original values of one or more of a site library's site properties to the original (as shipped) value(s) from PGMLIB

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Copying a Site Library's Options from PGMLIB (UNIX, Windows)

From the SAS IT Resource Management GUI for UNIX and Windows, you can copy one or more options from PGMLIB to the active site library by following these steps:

1 If your active site library is not the one that you want to use, change your active site library. Change your access mode to "write" if you do not already have write access to your active site library. For more information about specifying the active site library and/or changing your access mode to the active site library, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.

2 On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens. The **Copy** button is in the lower-left corner of the window.

3 Select **Copy**. The Copy Options window opens.

4 In the **Copy From** field, select **Supplied Default**.

5 In the **Copy To** field, select **Site Options**.

6 In the **Copy Selected Options** field, select the option(s) that you want to restore and deselect the ones that you do not want to restore.

Note: Not all site library options are available in this field. For others, use the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation. △

7 To return to the Set Site-wide Options window, select **OK**.

8 To return to the main window, select **OK**.

Copying a Site Library's Options to a PDB

When a PDB is created, SAS IT Resource Management automatically copies the values of the site options in the active site library to the same-named PDB options in the new PDB. Thus, you do not need to copy options to a new PDB.

If you want to copy one or more of the site library's site options (default values for PDB options) to an existing PDB, you can do so

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

You do not need to exit from SAS IT Resource Management after this operation for the changes to take effect in the PDB.

Note: The PDB's PDB options are in the PDB's DICTLIB. △

Copying a Site Library's Options to a PDB (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can copy one or more options from the active site library to the active PDB by following these steps:

1. If your active site library is not the one that you want to use, change your active site library. Change your access mode to "write" if you do not have already have write access to your active site library. For more information about changing which site library is active and/or changing your access mode to the active site library, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. If your active PDB is not the PDB that you want to use, change your active PDB. Change your access mode to "write" if you do not have already have write access to your active PDB. For more information about changing which PDB is active and/or changing your access mode to the active PDB, see "Activating a PDB" on page 299.
3. On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens. The **Copy** button is in the lower-left corner of the window.
4. Select **Copy**. The Copy Options window opens.
5. In the **Copy From** field, select **Site Options**.
6. In the **Copy To** field, select **PDB Options**.
7. In the **Copy Selected Options** field, select the option(s) that you want to copy and deselect the ones that you do not want to copy.

   Note: Not all PDB options are available in this field. For others, use the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation. △
8. To return to the Set Site-wide Options window, select **OK**.
9. To return to the main window, select **OK**.

Creating a Site Library

You can create a site library

- by allocating a new z/OS data set (on z/OS).
- by creating a new folder (on Windows).
- by creating a new directory (on UNIX).

Creating a Site Library (z/OS)

1. Allocate an empty data set to be used as a new site library:

- DSN=*high_level_qualifiers.xxxx*, where *high_level_qualifiers* specifies a location and *xxxx* is the name of the site library. A site library can have any name and be on any accessible disk. Typically, one of the site libraries has the same value for *high_level_qualifiers* as the SAS IT Resource Management software and has the site library name *SITELIB*.
- DSORG=PS
- RECFM=FS
- LRECL=6144
- BLKSIZE=6144
- SPACE=(CYL,(1,1))

Note: You do not need directory blocks. This is not a PDS. △

2 Submit a SAS program that copies the default values for PDB options from PGMLIB to the site library:

```
%CPSTART(...
        , SITELIB=location.of.site.library
        , SITEACC=OLD
        , ... ) ;

%CPPDBOPT(TYPE=SITELIB, COPY=PGMLIB) ;
```

For more information about submitting a SAS program, see "Using SAS Windows" on page 24 and "Working with the Interface for Batch Mode" on page 18.

Note: When a new PDB is created, if the active site library does not have the default values for PDB options, the values for the PDB's options are automatically copied from PGMLIB. △

Creating a Site Library (UNIX)

Create a directory to be used as a new site library. Name the folder *full_path_name*/*xxxx*, where *xxxx* is the name of the site library (directory). A site library can have any name and be on any accessible disk, including those that are accessed by means of NFS, AFS, or DFS. Typically, one of the site libraries has the same value for *full_path_name* as the SAS IT Resource Management software and has the site library name *sitelib*.

Note: The first time that the site library is used, SAS IT Resource Management automatically populates the site library with default (as shipped) settings from PGMLIB. △

Creating a Site Library (Windows)

Create a folder to be used as a new site library. Name the folder *full_path_name**xxxx*, where *xxxx* is the name of the site library (folder). A site library can have any name and be on any accessible disk, including those that are accessed by means of Network Neighborhood. Typically, one of the site libraries has the same value for *full_path_name* as the SAS IT Resource Management software and has the site library name *sitelib*.

Note: The first time that the site library is used, SAS IT Resource Management automatically populates the site library with default (as shipped) settings from PGMLIB. △

Deleting a Site Library

To delete a site library, follow these steps:

1 Notify the users of the site library that you intend to delete the site library. Also, recommend another site library for them to use.
2 (Optional) Make a backup of the site library. For more information about backing up a site library, see "Backing Up a Site Library" on page 273.
3 Display the value of Default Site Library. For more information about displaying which site library is the default, see "Viewing/Editing the Value of Default Site Library" on page 280.
4 If your active default site library is the site library that you want to delete, change the value of default site library to a different site library. For more information about changing which site library is the default, see "Viewing/Editing the Value of Default Site Library" on page 280.
5 Delete the site library. Use an operating-system command to delete
   - the z/OS data set (on z/OS)
   - the folder (on Windows)
   - the directory (on UNIX).
6 Check batch jobs, commands, CLISTs, and so on, for references to the deleted site library and change them to a site library that exists.

Releasing the Lock on the Active Site Library

When a site library PDB is activated, it is activated in the mode ("write" or "read-only") that you specified. The mode causes a lock of one kind or another to be placed on the site library. As soon as you finish using the site library, remember to release the lock.

You can release the lock on the active site library by doing one of the following:

- activating a different site library. For more information, see "Activating a Site Library" on page 272.
- exiting from SAS IT Resource Management and SAS.

Storing a Custom Format in Your Active Site Library

You can store a format for site-wide use in SITELIB.CPFMTS by using the SAS FORMAT procedure in batch mode.

Note: The formats that you store in SITELIB.CPFMTS might not be the only formats in that catalog. For example, if you run the %CSCSIFMT macro in batch mode to create formats for Aprisma SPECTRUM data, those formats are also stored in SITELIB.CPFMTS. For more information about the %CSCSIFMT macro, see the topic %CSCSIFMT in the SAS IT Resource Management macro reference documentation. △

Note: The format search path is as follows

- If you have not specified a format search path in SAS, the format search path that is used in SAS IT Resource Management is

```
WORK.FORMATS (first)
LIBRARY.FORMATS
DICTLIB.CPFMTS
SITELIB.CPFMTS
PGMLIB.CPFMTS (last)
```

- If you have specified a format search path in SAS, catalogs are appended to it within SAS IT Resource Management, and the format search path that is used in SAS IT Resource Management is

```
existing format search path (first)
DICTLIB.CPFMTS
SITELIB.CPFMTS
PGMLIB.CPFMTS (last)
```

△

Note: If there are multiple formats with the same name, the first one that is found in the search order is used. △

Note: For more about formats, see "The FORMAT Procedure" in the *Base SAS Procedures Guide* for your release of SAS software. △

Note: For more information about specifying search paths and the SAS FMTSEARCH option, see the documentation for your release of SAS or see SAS Technical Report P-222, "Changes and Enhancements to Base SAS Software, Release 6.07." △

Note: Typically, PDB-specific formats are stored in the appropriate PDB's DICTLIB.CPFMTS library, not in a site library. △

Storing a Custom Format in Your Active Site Library (z/OS, UNIX, Windows)

You can store a custom format in a site library by following these steps:

1. If your active site library is not the one that you want to use, change your active site library. If you do not already have write access to your active site library, change to "write" access mode. For more information about changing your active site library and/or changing your access mode to the active site library, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. To check that the name that you want to use for the format is available:
   - □ In the SAS IT Resource Management GUI for UNIX and Windows, select **Globals ▶ Access ▶ Display Libraries**

     The Libname window opens. In the **Name** column, double-click on **Sitelib**. The list of the contents in the library whose libref is SITELIB is displayed. Double-click on **Cpfmts**. The list of the contents in SITELIB.CPFMTS is displayed.

     Select **X** to close the window.
   - □ In the SAS IT Resource Management GUI for z/OS, issue the NEXT command to cycle to a SAS window (for example, the Output window). (By default, the NEXT command is mapped to the F8 key or SHIFT+F8 key combination.) If the SAS window has a command line, type *LIBNAME* and press RETURN or ENTER; if the window has pull-down menus, select **Tools ▶ Options ▶ Command** and then type *LIBNAME* and select **OK**. Find *Sitelib* in the **Name** column. Preceding *Sitelib*, type *S* and then press RETURN or ENTER. The list of the contents of *SITELIB* is displayed. Find *CPFMTS* in the **Name** column. Preceding *CPFMTS*, type *S* and then press RETURN or ENTER. The list of contents of *SITELIB.CPFMTS* is displayed.

     Issue the NEXT command repeatedly until you return to the SAS IT Resource Management window.
3. Submit a SAS program that is similar to the following one. (For more information about submitting a SAS program, see the topic "Use SAS Windows" in the online help that is delivered with this product and "Working with the Interface for Batch Mode" on page 18.)

```
proc format library=sitelib.cpfmts;
   .
   .
   .
   the name and the definition of the format
   .
   .
   .
run;
```

The format name and definition are saved to the CPFMTS catalog in the active site library. If the CPFMTS catalog does not exist, it is created. If a format with the same name exists, the new version replaces the previous version.

Viewing the Name of Your Active Site Library

Your active site library is the site library with which you are currently working. You can view the name of your active site library

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing the Name of Your Active Site Library (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display the name of your active site library by following these steps:

1. From the main menu, issue the NEXT command to cycle to a SAS window (for example, the Output window). (By default, the NEXT command is mapped to the F8 key or SHIFT+F8 key combination.)
2. If the SAS window has a command line, type *LIBNAME* and press RETURN or ENTER. If the SAS window has pull-down menus, select **Tools ▶ Options ▶ Command** and then type *LIBNAME* and select **OK**.
3. On the line that has *Sitelib* in the **Name** column, the full name of the active site library is displayed in the **Host Path Name** column.
4. To return to the main menu, issue the NEXT command repeatedly until you return to the SAS IT Resource Management window.

Viewing the Name of Your Active Site Library (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display the name of your active site library by following these steps:

1. On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens. The name of the active site library is displayed at the top of the window.
2. To return to the main window, select **Close**.

Viewing Your Active Site Library's File System and Profile

On z/OS, your active site library can be on only the local file system.

On UNIX and Windows with a SAS IT Resource Management client license, your active site library can be on the local file system or on a remote file system. You can view whether your active site library is on the local file system or a remote file system

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing Your Active Site Library's File System and Profile (UNIX, Windows)

On the SAS IT Resource Management GUI for UNIX and Windows, you can view your active site library's file system and remote profile, if any, by following these steps:

1. On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens. There is a **Change Sitelib** icon in the upper-left corner of the window.
2. Select the **Change Sitelib** icon. The Change Site Library window opens.
3. If the value in the **Location** field is *Local*, the site library is on the local file system.

   If the value in the **Location** field is *Remote Server xxxx*, the site library is on a remote file system and the connection to that file system uses the remote server profile named *xxxx*.

   For more information about remote server profiles, see "Overview of Remote Server Profiles" on page 571.
4. To return to the Set Site-wide Options window, select **OK**.
5. To return to the main window, select **Close**.

Viewing/Editing the Value of Default Site Library

The default site library is the site library that is used when you or anyone else starts SAS IT Resource Management without specifying an active site library. The value for the default site library is stored in PGMLIB.

You can

- display the value of the default site library by submitting a SAS program
- edit the value of the default site library by submitting a SAS program.

Viewing the Value of Default Site Library (z/OS, UNIX, Windows)

You can view the value of the default site library by submitting this SAS program:

```
PROC PRINT DATA=PGMLIB.CPSITE ;
   RUN;
```

(For more information about submitting a SAS program, see "Using SAS Windows" on page 24 and "Working with the Interface for Batch Mode" on page 18.)

Editing the Value of Default Site Library (z/OS, UNIX, Windows)

The original default site library is

- *high_level_qualifiers.SITELIB* on z/OS, where *high_level_qualifiers* is the location of the SAS IT Resource Management software
- *full_path_name**sitelib* on Windows, where *full_path_name* is the location of the SAS IT Resource Management software
- *full_path_name*/*sitelib* on UNIX, where *full_path_name* is the location of the SAS IT Resource Management software.

Before another site library is designated as the default site library, that other site library should already exist.

To edit the value of default site library:

1 You must know the location of the site library that you want to designate as the default site library. That is, obtain the complete name, including the high-level qualifiers of its data set (z/OS), the path name of its folder (Windows), or the path name of its directory (UNIX).

2 You must know the location of the program library that SAS IT Resource Management uses (and refers to as PGMLIB). That is, obtain the complete name, including the high-level qualifiers of its data set (z/OS), the path name of its folder (Windows), or the path name of its directory (UNIX). (If you do not already know this information, you can find it in your site's CLIST that invokes SAS IT Resource Management or in your site's command for starting SAS IT Resource Management, or, while SAS IT Resource Management is running, by submitting this SAS program through the SAS Program Editor window:

```
LIBNAME PGMLIB LIST ;
```

A RUN statement is not necessary after the LIBNAME statement.) For more information about submitting a SAS program, see "Using SAS Windows" on page 24 and "Working with the Interface for Batch Mode" on page 18.)

3 Because you will need exclusive access in the next step,

   a check that no one is running SAS IT Resource Management.
   b invoke the appropriate SAS IT Resource Management server GUI (the one for z/OS or the one for UNIX and Windows). For more information about starting SAS IT Resource Management, see "Starting the z/OS GUI" on page 19 or "Starting the UNIX and Windows GUI" on page 21.
   c on the main menu or main window, follow this path: **Exit ▶ Exit from SAS IT Resource Management Only**

4 For z/OS, submit the following SAS program:

```
LIBNAME PGMLIB CLEAR ;
LIBNAME PGMLIB "location-for-PGMLIB" DISP=OLD ;
DATA PGMLIB.CPSITE ;
   CPSITE = "location-for-default-site-library" ;
   RUN ;
LIBNAME PGMLIB CLEAR ;
```

A RUN statement is not necessary after the final LIBNAME statement.

For UNIX and Windows, submit the following SAS program:

```
LIBNAME PGMLIB CLEAR ;
LIBNAME PGMLIB "location-for-PGMLIB";
DATA PGMLIB.CPSITE ;
   CPSITE = "location-for-default-site-library" ;
   RUN ;
LIBNAME PGMLIB CLEAR ;
```

A RUN statement is not necessary after the final LIBNAME statement.

Note: To run this program successfully, you must have your file system's write permission for PGMLIB. It is recommended that you request "write" permission only during the time that you need it, so that you do not accidentally write to PGMLIB at any other time. △

Viewing/Editing Your Active Site Library's Access Mode

By starting or restarting SAS IT Resource Management, you can specify your active site library and your active site library's access mode for your SAS IT Resource

Management session. For more information about starting or restarting SAS IT Resource Management, see "Starting the z/OS GUI" on page 19 or "Starting the UNIX and Windows GUI" on page 21.

Within a SAS IT Resource Management session, you can view and/or edit the value of your active site library and the value of your active site library access mode by using

- the SAS IT Resource Management GUI for z/OS
- the SAS IT Resource Management GUI for UNIX and Windows.

Note: Prior to specifying your active site library's access mode in SAS IT Resource Management, you must arrange for permission from your site's security administrator or security system to have that level of access to the library. △

Viewing/Editing Your Active Site Library's Access Mode (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can view and/or edit the name of your active site library and the value of your active site library's access mode by following these steps:

1. From the main menu, select **EXIT**. A menu opens. From the menu, select **Exit SAS IT Resource Management only**. You return to SAS (but the librefs that SAS IT Resource Management defined are still defined).
2. If the SAS Program Editor window has a command-line prompt (Command ==>), type **COMMAND** and press ENTER or RETURN. If you later want to restore the command-line prompt, follow this selection path at a Program Editor window that has menus: **Tools ▶ Options ▶ Turn Menus Off**
3. Select **Tools**. A menu opens. From the menu, select **Options**. A menu opens. Select **Command**. A dialog box opens.
4. At the Command ==> prompt, type **FNAME** and then select **OK**. The SAS FNAME window opens.
5. Page down or scroll down in the window to the line that has SITELIB in the DDname column. On that line, the Data Set Name column lists the name of the active site library and the Status column lists the access mode of the active site library.
6. Select **File**. A menu opens. From the menu, select **Close**.
7. In the body of the Program Editor window, type

```
%CPSTART();
```

   if you want to invoke SAS IT Resource Management again with the same site library and the same access mode to the site library

```
%CPSTART(siteacc=OLD);
```

   or

```
%CPSTART(siteacc=SHR);
```

   if you want to invoke SAS IT Resource Management again with the same site library but a different access mode (OLD or SHR) to the site library

```
%CPSTART(sitelib=name-of-another-site-library, siteacc=OLD);
```

   or

```
%CPSTART(sitelib=name-of-another-site-library, siteacc=SHR);
```

   if you want to invoke SAS IT Resource Management again with a different site library (whose name replaces *name-of-another-site-library*).

8 Select **Run**. A menu opens. From the menu, select **Submit**. SAS IT Resource Management opens the main menu.

Viewing/Editing Your Active Site Library's Access Mode (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view and/or edit the name of your active site library and the value of your active site library's access mode by following these steps:

1 On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens. The name of your active site library and your access mode to it are displayed at the top of the window.

2 If your active site library is not the one that you want to use, and if the site library that you want to use is on a remote host, begin by activating a PDB that uses a remote profile that specifies that remote host and that site library on the remote host. (For more information about activating a PDB that uses a remote profile, see "Activating a PDB" on page 299 and "Overview of Remote Server Profiles" on page 571.)

3 Next, regardless of the location of the site library that you want to use, on the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens. The name of your active site library and your active access mode to it are displayed at the top of the window.

4 If your active site library is not the one that you want to use or if you want to edit your access mode to it, select the icon in the upper-left corner of the window.

5 To change to a different site library, type over the name of the site library in the **Path** field.

6 In the **Access** field, select the access mode that you want to use.

7 Select the location of the site library in the **Location** field. (If you select **Remote**, the remote profile that is used is the one that you selected above when activating a PDB on that remote host.)

8 Select **OK**.

9 To return to the main menu, select **OK**.

Working with Site Libraries: Properties

Viewing/Editing Your Active Site Library's Active Collector

The active collector designates the collector that is used if you do not specify a collector when you perform an operation (such as the process task) that involves a collector.

The default value for the active collector is stored as a site option. SAS IT Resource Management automatically copies the value in the site option to a new PDB's same-named option when you create the PDB. You can change the value in the PDB if you prefer another.

The original value in the site option comes from PGMLIB's same-named option when you create the site library. You can change the value in the site option if you prefer another. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit a site library's option for active collector

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.

Viewing/Editing Your Active Site Library's Archive Information

If one or more tables have archiving turned on (in their table properties), then there is an archive.

An archive is a set of archive libraries. Each archive library is the result of one execution of the processing task. Each archive library contains a copy of the incoming data to the detail level of the tables for which archiving is turned on.

The default values for archive information are stored as site options. SAS IT Resource Management automatically copies the values in the site options to a new PDB's same-named options when you create the PDB. You can change the values in the PDB if you prefer others.

The original values in the site options come from PGMLIB's same-named options when you create the site library. You can change the values in the site options if you prefer others. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit a site library's archiving information

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.

Viewing/Editing Your Active Site Library's DST Definition

The Daylight Saving Time (DST) definition describes the Daylight Saving Time period. Some collectors log data by using DST timestamps during the DST period. SAS IT Resource Management can change those timestamps to standard time by using the Daylight Saving Time definition. (The only supplied collector that uses the Daylight Saving Time definition is the one for SunNet Manager data, and the Generic Collector Facility does not access the Daylight Saving Time definition.)

The default value for the Daylight Saving Time definition is stored as a site option. SAS IT Resource Management automatically copies the value in the site option to a new PDB's same-named option when you create the PDB. You can change the value in the PDB if you prefer another.

The original value in the site option comes from PGMLIB's same-named option when you create the site library. You can change the value in the site option if you prefer another. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit a site library's default value for the Daylight Saving Time definition

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: There are two ways to express the definition: as a concatenation of ranges (one each for next year, this year, last year, and the year before) and as a formula that takes the system's year into account. Both forms of the definition work correctly. The advantage of the formula is that it does not require periodic updates. △

Viewing/Editing Your Active Site Library's DST Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view the active site library's option for the Daylight Saving Time definition by following these steps:

1. If your active site library is not the site library that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will change the option's value, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. On the main window's **`Administration`** tab, select **`Site Options`**. The Set Site-wide Options window opens.
3. Select the **`Time Zone`** tab. The second field on the tab is the Daylight Saving Time definition.
4. If you want to change the Daylight Saving Time definition, you can use any one of these methods:
   - Type over the value and then select **`OK`**.
   - Copy the value from a PDB. For more information about copying from a PDB, see "Copying a PDB's Options to a Site Library" on page 273.
   - You can copy the original (as shipped) value from PGMLIB. For more information about copying from PGMLIB, see "Copying a Site Library's Options from PGMLIB" on page 274.

Viewing/Editing Your Active Site Library's GMT Deviation

The Greenwich Mean Time (GMT) deviation describes the number of hours' difference between the time where the data was logged and the time in Greenwich, UK (Coordinated Universal Time). Some collectors log data by using GMT timestamps (the time in Greenwich, not the local time). SAS IT Resource Management can change these timestamps to local time by using the GMT deviation.

Note: The only supplied collector that uses the Greenwich Mean Time deviation is the one for SunNet Manager data, and the Generic Collector Facility does not access the Greenwich Mean Time deviation. △

The default value for the Greenwich Mean Time deviation is stored as a site option. SAS IT Resource Management automatically copies the value in the site option to a new PDB's same-named option when you create the PDB. You can change the value in the PDB if you prefer another.

The original value in the site option comes from PGMLIB's same-named option when you create the site library. You can change the value in the site option if you prefer another. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit a site library's default value for the Greenwich Mean Time (GMT) deviation

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing/Editing Your Active Site Library's GMT Deviation (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view and/or edit the active site library's option for the Greenwich Mean Time (GMT) deviation by following these steps:

1. If your active site library is not the site library that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change the option's value, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. From the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens.
3. Select the **Time Zone** tab. The first field on the tab is the Greenwich Mean Time deviation.
4. If you want to change the Greenwich Mean Time deviation, you can use any one of these methods:
   - Type over the value and then select **OK**.

     Note: Remember to use a minus sign, if needed. △
   - Copy the value from a PDB. For more information about copying from a PDB, see "Copying a PDB's Options to a Site Library" on page 273.
   - Copy the original (as shipped) value from PGMLIB. For more information about copying from PGMLIB, see "Copying a Site Library's Options from PGMLIB" on page 274.

Viewing/Editing Your Active Site Library's List of Holidays

The list of holidays is a list of the inactive and active holidays at the site. SAS IT Resource Management ignores the holidays that are marked "inactive" and uses the holidays that are marked "active." The "active" holidays are used in the following ways:

- If an observation has a datetime stamp whose date is an "active" holiday and if, by the time that the data is ready to be processed into the PDB, the value of the shift code in the observation is *blank*, then the value of the shift code in the observation is set equal to the value of the shift code for holidays.
- If an observation has a datetime stamp whose date is an "active" holiday and if, by the time that the data is ready to be processed into the PDB, the value of the shift code in the observation is *non-blank*, then the value of the shift code in the observation is not changed.

At run time, the active PDB uses the list of holidays in the active site library.

In a site library, the original list of holidays is from PGMLIB. You can change the list in the active site library to match the holidays at your site.

You can view and/or edit the list of holidays in the active site library

- by using the %CPHDAY macro in batch mode. For more information about the %CPHDAY macro, see the topic %CPHDAY in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing/Editing Your Active Site Library's List of Holidays (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can view and/or edit the active site library's list of holidays by following these steps:

1. If your active site library is not the site library that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change one or more holidays, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Define Work Shifts`**. The Define Work Shifts window opens.
3. Select **`Site Defaults`**. (This selection is not necessary for displaying or editing the list of holidays or the shift code for holidays. However, if you edit the description of the shift code that is assigned to holidays, you will need this setting to adjust the description.)
4. Select **`Define Holidays`**. The Holidays Definition window opens and displays the list of holidays.
5. Holidays with a status of "active" are used by SAS IT Resource Management.. Holidays with a status of "inactive" are ignored by SAS IT Resource Management.
   - To change the status of a holiday, follow this path: **select the holiday ▶ ItemActions ▶ Edit Holiday ▶ select the status that you want to use ▶ OK**
   - To add a holiday, follow this path: **Actions ▶ Add holiday ▶ click in the Date field ▶ specify the date ▶ check that the status is "active" ▶ click in the Description field ▶ type the Description ▶ optionally, click in the Comment field and type a comment ▶ OK**
   - To delete a holiday, follow this path: **select the holiday ▶ ItemActions ▶ Delete Holiday ▶ OK**

     Note: You can make multiple selections in the list before activating one of the item actions: **`Edit Holiday`** or **`Delete Holiday`**. △
6. To return to the Define Work Shifts window, select **`File`**. A menu opens. From the menu, select **`End`**.
7. To return to the main menu, select **`OK`** if you edited the list of holidays and want to save your changes; otherwise, select **`Cancel`**.

Viewing/Editing Your Active Site Library's List of Holidays (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view and/or edit the active site library's list of holidays by following these steps:

1. If your active site library is not the one that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change one or more values, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. On the main window's **`Administration`** tab, select **`Define Work Shifts`**. The Define Work Shifts window opens.
3. Select **`Site Defaults`**. (This selection is not necessary for displaying or editing the list of holidays. However, if you go on to edit the shift code for holidays and the description of the shift code for holidays, you will need this setting to adjust the description.)
4. Select **`Define Holidays`**. The Define Holidays window opens and displays the list of holidays.

5 Holidays with a status of "active" are used by SAS IT Resource Management. Holidays with a status of "inactive" are ignored by SAS IT Resource Management.
   - To change the status of a holiday, follow this path: **select the holiday ▶ File ▶ Edit Holiday ▶ select the status ▶ OK**
   - To add a holiday, follow this path: **File ▶ Add holiday ▶ specify the date ▶ select the status ▶ type the description ▶ OK**
   - To delete a holiday, follow this path: **select the holiday ▶ File ▶ Delete Holiday ▶ Yes ▶ OK**

6 To return to the Define Work Shifts window, select **Close**.

7 To return to the main window, select **OK**.

Viewing/Editing Your Active Site Library's MXG Views Property

In SAS IT Resource Management software for z/OS, MXG views of a PDB enable you to run report programs that are written in SAS that use the MXG names for tables and variables. This can be useful if you have old MXG report programs that you want to run against one or more SAS IT Resource Management PDBs. Note that the MXG views are provided in addition to the SAS IT Resource Management views. If you request no MXG views, you still have the SAS IT Resource Management views to use with the SAS IT Resource Management reporting facilities.

The default value for the MXG Views property is stored as a site option. SAS IT Resource Management automatically copies the value in the site option to a new PDB's same-named option when you create the PDB. You can change the value in the PDB if you prefer another.

The original value in the site option comes from PGMLIB's same-named option when you create the site library. You can change the value in the site option if you prefer another. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit the status of the MXG views site option

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.

Note: By default, MXG views are requested. △

Viewing/Editing Your Active Site Library's Shift Code for Holidays and Its Description

The list of holidays is a list of inactive and active holidays at the site. SAS IT Resource Management ignores the holidays that are marked "inactive" and uses the holidays that are marked "active." The "active" holidays are used in the following ways:

- If an observation has a datetime stamp whose date is an "active" holiday and if, by the time that the data is ready to be processed into the PDB, the value of the shift code in the data is *blank*, then the value of the shift code in the observation is set equal to the value of the shift code for holidays.
- If an observation has a datetime stamp whose date is an "active" holiday and if, by the time that the data is ready to be processed into the PDB, the value of the shift code in the data is *non-blank*, then the value of the shift code in the observation is not changed.

The description that is used for the shift code for holidays is the shift description that is associated with the value that is used as the shift code for holidays. For example, if the shift code for holidays is "0", then the description of the shift code for holidays is the description that is associated with the shift code of "0". (For more information about shift descriptions, see "Viewing/Editing Your Active Site Library's Shift Descriptions" on page 291.)

At run time, the active PDB accesses the shift code for holidays in the active site library.

When a site library is created, it obtains its value of the shift code for holidays and its description of the shift code for holidays from PGMLIB. By default, the shift code for holidays is "0". By default, the description for the shift code of "0" is "HOLIDAY." You can change the values in the active site library if you prefer others.

You can view and/or edit the active site library's shift code for holidays and default description of the shift code for holidays

- by using the %CPHDAY and %CPPDBOPT macros in batch mode. For more information about the %CPHDAY macro, see the topic %CPHDAY in the SAS IT Resource Management macro reference documentation. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: The shift code for holidays and the default description of the shift code for holidays are stored as site options (properties).

Viewing/Editing Your Active Site Library's Shift Code for Holidays and Its Description (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view and/or edit the active site library's shift code for holidays and the active site library's default description of the shift code for holidays by following these steps:

1. If your active site library is not the one that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change one or more of the options' values, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. On the main window's **`Administration`** tab, select **`Define Work Shifts`**. The Define Work Shifts window opens.
3. At the question **`Define at which level?`**, select **`Site Defaults`**. (This selection is not necessary if you want to change the shift code for holidays or the list of holidays. However, this selection is necessary if you later want to adjust the default description of the shift code for holidays.)
4. Select **`Define Holidays`**. The Define Holidays window opens and displays, at the top of the window, the value of the shift code for holidays.
5. If you want to change the value, select **`File`**. A menu opens. From the menu, select **`Modify Holiday Shift`**. The Define Holiday Shift window opens.
6. Type over the old value with the new value that you want to assign to data on days that are active holidays.
7. To return to the Holiday Definition window, select **`OK`**.
8. To return to the Define Work Shifts window, select **`Close`**. If you changed the value, also select **`OK`**.
9. Select **`Shift Descriptions`**. The Modify Shift Descriptions window opens.

10 For each value of shift code there is a description. If you did not change the value of the shift code for holidays, then the description next to its value is still the description of the shift code for holidays. If you did change the value of the shift code for holidays, make the appropriate adjustments so that the values of the descriptions are now appropriate. For example, if the shift code "4" is for holidays and is not used in the work-shift schedule, you might want to change the description for "0" to blank and the description for "4" to "HOLIDAY." Or if "2" is now the shift code for holidays and is also used for weekends in the work schedule, you might want to change the description for "0" to blank and the description for "2" to "WEEKEND OR HOLIDAY."

11 To return to the Define Work Shifts window, select **OK** if you made changes and want to save them; otherwise, select **Cancel**.

12 To return to the main window, select **OK** if you made changes and want to save them; otherwise, select **Cancel**.

Viewing/Editing Your Active Site Library's Start of Week

Start of week defines the beginning of a week for the week level. Some sites consider a week to be Sunday through Saturday; some sites consider a week to be Monday through Sunday; and so on.

The default value for Start of Week is stored as a site option. SAS IT Resource Management automatically copies the value in the site option to a new PDB's same-named option when you create the PDB. You can change the value in the PDB if you prefer another.

The original value in the site option comes from PGMLIB's same-named option when you create the site library. You can change the value in the site option if you prefer another. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit a site library's option for the start of week

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: The default value for Start of Week is *Sunday*. △

Viewing/Editing Your Active Site Library's Start of Week (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view and/or edit the active site library's value Start of Week by following these steps:

1 If your active site library is not the one that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change the option's value, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.

2 On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens.

3 Select the **Start-of-Week** tab. The tab displays the active value of Start of Week.

4 To change the site library's option for start of week, you can use any one of these methods:

   - You can follow this path: **select the down arrow ▶ select the day that you want to use ▶ OK**

- You can copy the value from a PDB. For more information about copying from a PDB, see "Copying a PDB's Options to a Site Library" on page 273.
- You can copy the original (as shipped) value from PGMLIB. For more information about copying from PGMLIB, see "Copying a Site Library's Options from PGMLIB" on page 274.

5 To return to the main window, select **OK**.

Viewing/Editing Your Active Site Library's Shift Descriptions

The work-shift schedule is defined by using work-shift codes. (For more information about the work-shift schedule, see "Viewing/Editing Your Active Site Library's Work-Shift Schedule" on page 292.) The descriptions of the codes assign meaning to the codes. By default,

- for shift code 0, the description is "HOLIDAY"
- for shift code 1, the description is "WEEKDAY"
- for shift code 2, the description is "WEEKNIGHT"
- for shift code 3, the description is "WEEKEND"
- for shift code 4, the description is "Undefined"
- for shift code 5, the description is "Undefined"
- for shift code 6, the description is "Undefined"
- for all other shift codes, the descriptions are blank.

The default values for the descriptions are stored as site options. SAS IT Resource Management automatically copies the values in the site options to a new PDB's same-named options when you create the PDB. You can change the values in the PDB if you prefer other values.

The original values in the site options come from PGMLIB's same-named options when you create the site library. You can change the values of the shift descriptions if you prefer others. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit a site library's options for descriptions

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing/Editing Your Active Site Library's Descriptions of Work-Shift Codes (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can view and/or edit the active site library's descriptions for the work-shift codes by following these steps:

1 If your active site library is not the one that you want to use, change your active site library. If your access mode to your active site library is not "write" (that is, "OLD") and if there is a possibility that you will want to change one or more of the options' values, change your access mode to "write" (OLD). For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.

2 On the main menu, select **PDB Admin**. A menu opens. From the menu, select **Define Work Shifts**. The Define Work Shifts window opens.
3 In the **Define at Which Level?** field, select **Site Defaults**.
4 The descriptions of the work-shift codes appear in the frame in the lower-left corner of the window.
5 If the codes and descriptions do not provide the distinctions that you want to make among shifts, in the **Shift Descriptions** frame add or remove codes and type over the existing descriptions with the descriptions that you prefer.
6 To exit to the main menu, select **OK**.

Viewing/Editing Your Active Site Library's Descriptions of Work-Shift Codes (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view and/or edit the active site library's descriptions for the work-shift codes by following these steps:

1 If your active site library is not the one that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change one or more of the options' values, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2 On the main window's **Administration** tab, select **Define Work Shifts**. The Define Work Shifts window opens.
3 In the **Define at Which Level?** field, select **Site Defaults**.
4 To display and/or edit the descriptions of the work-shift codes, select **Shift Descriptions**.
5 If the codes and descriptions do not provide the distinctions that you want to make among shifts, add or remove codes and type over the existing descriptions with the descriptions that you prefer, and then select **OK**.
6 To return to the main window, select **OK**. A dialog box opens. Select **OK**.

Viewing/Editing Your Active Site Library's Work-Shift Schedule

The work-shift schedule has 24 rows (one for each hour of the day) and 7 columns (on for each day of the week). SAS IT Resource Management uses the work-shift schedule in the logic for assigning a shift code to each observation:

- If the observation already has a *non-blank* value for the shift code, no change is made to the value of the shift code in the observation;
- otherwise, if the date in the observation's date time stamp is an *active* holiday, the value of the shift code for holidays is used in the observation;
- otherwise, the date in the observation's datetime stamp is converted into its day-of-week and the time in the observation's datetime stamp is converted into its hour, and the shift code for that day-of-week and hour is obtained from the work-shift schedule and used in the observation.

For more information about the active holidays, see "Viewing/Editing Your Active Site Library's List of Holidays" on page 286. For more information about the shift code for holidays, see "Viewing/Editing Your Active Site Library's Shift Code for Holidays and Its Description" on page 288.

The default work-shift schedule is as follows:

- shift code "1" is assigned to data that has a datetime stamp whose date is a Monday, Tuesday, Wednesday, Thursday, or Friday, and whose time is in the range from 8 a.m. through 4:59 p.m. (08:00 through 16:59)
- shift code "2" is assigned to data that has a datetime stamp whose date is a Monday, Tuesday, Wednesday, Thursday, or Friday and whose time is in the range from midnight through 7:59 a.m. (00:00 through 07:59) or in the range from 5 p.m. to 11:59 p.m. (17:00 through 23:59)
- shift code "3" is assigned to data that has a datetime stamp whose date is a Saturday or Sunday.

The work-shift schedule is composed of seven site options (the schedule for Sunday, the schedule for Monday, and so on, through the schedule for Saturday). SAS IT Resource Management automatically copies the values in the site options to a new PDB's same-named options when you create the PDB. You can change the values in the PDB if you prefer others.

The original values in the site options come from PGMLIB's same-named options when you create the site library. You can change one or more of the values if you prefer others. (For more information about creating a site library, see "Creating a Site Library" on page 275.)

You can view and/or edit a site library's options for the work-shift schedule

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing/Editing Your Active Site Library's Work-Shift Schedule (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can view and/or edit the active site library's work-shift schedule by following these steps:

1. If your active site library is not the one that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change the schedule, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. On the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Define Work Shifts`**. The Define Work Shifts window opens.
3. Select **`Site Defaults`**.
4. The work-shift schedule is displayed in the frame in the upper-left corner of the window. To see other hours, use the scroll bar on the right of the frame. To see other days, select an arrow at the bottom of the frame.

   Note that the hours are displayed in terms of the 24-hour clock (0:00 to 23:00). Each hour, *n*:00, represents the range *n*:00:00 to *n*:59:59. For example, 08:00 represents 08:00:00 to 08:59:59.

   Remember to check (and change if necessary) the shift codes for every day of the week.
5. To edit the work-shift schedule, you can use any one of these methods:
   - Type over the active shift code(s) with the shift code(s) that you want to use, and then select **`OK`**.
   - Copy the value(s) from a PDB. For more information about copying from a PDB, see "Copying a PDB's Options to a Site Library" on page 273.

- Copy the original (as shipped) values from PGMLIB. For more information about copying from PGMLIB, see "Copying a Site Library's Options from PGMLIB" on page 274.

Viewing/Editing Your Active Site Library's Work-Shift Schedule (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view and/or edit the active site library's work-shift schedule by following these steps:

1. If your active site library is not the one that you want to use, change your active site library. If your access mode to your active site library is not "write" and if there is a possibility that you will want to change the schedule, change your access mode to "write." For more information about changing your active site library and/or your access mode to it, see "Activating a Site Library" on page 272 or "Viewing/Editing Your Active Site Library's Access Mode" on page 281.
2. On the main window's **Administration** tab, select **Define Work Shifts**. The Define Work Shifts window opens.
3. Select **Site Defaults**.
4. The work-shift schedule displays in the frame in the left of the window. To see other hours, use the scroll bar on the right of the frame. To see other days, select an arrow at the bottom of the frame.

   Note that the hours are displayed in terms of the 24-hour clock (0:00 to 23:00). Each hour, *n*:00, represents the range *n*:00:00 to *n*:59:59. For example, 08:00 represents 08:00:00 to 08:59:59.

   Remember to check (and change if necessary) the shift codes for every day of the week.
5. To edit the work-shift schedule, you can use any one of these methods:
   - Type over the schedule with the value(s) that you want to use and then select **OK**.
   - Copy the value(s) from a PDB. For more information about copying from a PDB, see "Copying a PDB's Options to a Site Library" on page 273.
   - Copy the original (as shipped) value(s) from PGMLIB. For more information about copying from PGMLIB, see "Copying a Site Library's Options from PGMLIB" on page 274.
6. To return to the main window, select **OK**. A dialog box opens. Select **OK**.

CHAPTER 15

Administration: Working with Whole PDBs

Working with Whole PDBs: Main Topics

Overview of Whole PDBs

Logically, a PDB is a performance data warehouse (formerly a performance database) that contains performance data that is related to one or more data sources. Physically, a PDB is a coordinated set of nine SAS libraries. This overview and the topics that are associated with it correspond to a PDB that is being treated as a whole structure.

The main operations with respect to a whole PDB are as follows:

- creating a PDB, backing up a PDB, and deleting a PDB.

  Typically, a site has one PDB for each data source. But your site can have data from multiple data sources in the same PDB and/or have data from one data source in multiple PDBs.

- creating a remote-server profile, displaying and editing a remote-server profile, and deleting a remote-server profile.

  If a PDB that you want to access is not local to your host, you need a profile for the remote server host on which the PDB is located.

- adding the name of a PDB to your list of known PDBs, displaying and editing your list of known PDBs, and removing the name of a PDB from your list of known PDBs.

  Remote PDBs on your list of known PDBs need an appropriate remote-server profile to be associated with them.

- making a PDB your active PDB, and displaying the name of your active PDB and your access mode for it ("write" or "readonly").

  Your active PDB is whichever PDB you are creating or working with at the moment. Making a PDB your active PDB is also known as activating the PDB.

- acquiring PDB properties (also known as PDB options) by default, displaying and editing PDB properties, resetting PDB properties, uploading PDB properties to a site library; displaying and editing defaults for all table definitions and all regular, derived, and formula variable definitions, and rebuilding views for all tables.

  Some properties, settings, and operations apply to all of the data and/or all of the components within the PDB.

  - The Holiday List and Holiday Shift Code that the PDB uses are obtained from the site library that is current at the time of *use*.
  - The active site library at the time that the PDB is *created* automatically provides these defaults for PDB properties: the default archive device, the default archive path, the default archive parameters, the default archive engine, the default active collector, the default Daylight Saving Time (DST) definition, the default Greenwich Mean Time (GMT) deviation, the default description of the shift code for holidays, the default for the start of a week, the descriptions of work-shift codes, and the work-shift schedule. The values in the PDB can be displayed and edited, and if necessary they can be reset to the values that are provided by the site library. If you prefer, some or all of the values in the PDB can also be uploaded to the site library.
  - The PDB stores data in tables and the tables hold data in variables. For some tables and their variables, the table definition (and the definitions of its variables) is added from the master data dictionary and a few attributes acquire values by default. For other tables and their variables, the table definition (and the definitions of its variables) is defined by the user and attributes whose values are not specified acquire values by default.

    In either case, you can specify the default values that take effect during the addition or creation of table definitions (and the definitions of their variables).
  - For performance reasons, the tables use multiple data sets to hold the data. So that the existence of these data sets is transparent to the user, views are provided that access the data sets and make the data appear to be together. If you edit the table definitions or variable definitions, the views are removed and are rebuilt automatically before any additional data is processed into the tables or summarized for the tables. If you want to report on the data in the tables before the views are automatically rebuilt, you can "manually" request that the views be rebuilt.

Note: These views (SAS IT Resource Management views) are not the same as MXG views. SAS IT Resource Management views enable SAS IT Resource Management report definitions (and other SAS IT Resource Management software) to access the data in the SAS IT Resource Management PDB. MXG views are optional, supplementary views. MXG views enable old MXG report definitions to access the data in the SAS IT Resource Management PDB as if the PDB were an MXG PDB. △

Other operations that affect the whole PDB are as follows:

- copying a PDB within a local file system, and exporting, transferring, and importing a PDB.

  You might want to make a copy of a PDB and its contents. If the copy does not take place within the local file system, you need to package the PDB for transfer (export it), do the transfer, and then un-package the PDB (import it).

- combining two PDBs into one, and splitting one PDB into two.

  If you have two PDBs, each with tables of data from its own data source, you can combine the PDB into one with tables from both data sources, and vice versa. You can also combine and split PDBs for other reasons and for more than two data sources.

  The combining is done by copying tables (including the data in the tables) from one PDB to another. The split is done in the same way.

- converting a PDB from the SAS IT Resource Management 1 structure to the SAS IT Resource Management 2 structure; migrating a PDB from SAS Version 6 to SAS Version 8.

  If you have an existing PDB that has an old structure, you can update the PDB to a new structure. That is, if you acquire a new version of SAS IT Resource Management and/or SAS software, you can update your PDB(s) to match.

  Note: To convert a PDB from ITRM2 to ITRM1, see the %CPUNCVT topic in "Chapter 2: Administrative Macros" in the macro reference documentation. △

- verifying and/or correcting the PDB's data dictionary.

  Because a PDB is a coordinated set of SAS libraries, it is important to do any editing through SAS IT Resource Management rather than by using SAS software or operating system software to edit the libraries independently. If, however, SAS IT Resource Management fails because of a power outage, out-of-space condition, or similar circumstance, the coordination might be lost and thus might damage the PDB.

  If something has potentially damaged the PDB, you can check for damages and, in some cases, repair the damages automatically. If the damages cannot be repaired automatically, there are also instructions for "manual" recovery; the instructions depend on what was running when the failure occurred. Or you can retrieve the backup copy of the PDB and rerun the job that was interrupted by the failure.

- releasing the lock on a PDB.

  When you access a PDB, you access it in "readonly" mode or "write" mode. The mode places a lock of one kind or another on the PDB. The lock can prevent another person from accessing the PDB, so it is good to release the lock as soon as you finish using the PDB. You can release the lock on a particular PDB by activating a different PDB or exiting from SAS IT Resource Management and SAS.

For additional information about whole PDBs, see the following topics:

- "Activating a PDB" on page 299

Activating a PDB

Your active PDB is the PDB that you are creating or working with at the moment. Each time that you start SAS IT Resource Management (after your initial use), by default SAS IT Resource Management activates the PDB that you were using when you most recently exited from SAS IT Resource Management.

You can activate another PDB (that is, make another PDB your active PDB, or switch from using one PDB to using another)

- by using the %CPSTART macro (in batch mode). For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Activating a PDB causes a lock of one kind or another to be placed on the PDB, and that lock can prevent others from working with the PDB. Thus, as soon as you finish using the PDB, remember to release the lock. For more information, see “Releasing the Lock on the Active PDB” on page 319. △

Note: To activate a PDB on a remote host, you must use the SAS IT Resource Management client software. When you activate a remote PDB, the client software makes a connection to the remote server that is specified by the associated profile. When you deactivate that remote PDB (by activating a local PDB), typically you agree to disconnect from the remote server on which the remote PDB is located. △

Activating a PDB (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can activate a PDB by following these steps:

1. From the main menu, select the **PDB Admin** button. A menu opens. From the menu, select **Manage PDB/Switch PDB**. The Performance Database Operations window opens.
2. If the name of the PDB that you want to use is not on the PDB list, add the name to the list. For information about adding a name, see "Adding a PDB to Your List of Known PDBs" on page 301
3. From the PDB list, select the name of the PDB that you want to make your active PDB. The PDB must be on your local file system.

   Note: You do not specify a remote-server profile. If you are using the z/OS GUI, you are using the SAS IT Resource Management server software, which cannot access a remote PDB. To access a remote PDB, use the SAS IT Resource Management client software. △
4. Select the **ItemActions** menu. From the menu, select **Activate**. The PDB Access Mode window opens.
5. Select the level of access that you want to have for the PDB and then select **OK** to return to the Performance Database Operations window. The PDB that you selected is activated, and the active PDB's name is listed at the top of the window. (Farther down the window, its name is also in the list of all PDBs.)
6. To exit from the window, select the **File** menu. From the menu, select **End**.

Activating a PDB (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can activate a PDB by following these steps:

1. From the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2. If the name of the PDB that you want to use is not on the Other PDBs list, add it to the list. For information about adding a name, see "Adding a PDB to Your List of Known PDBs" on page 301.
3. In the Other PDBs list, select the name of the PDB that you want to make your active PDB.
4. Select **Activate**. The Activating PDB Confirmation window opens.
5. In the **Access Mode** box, select the level of access that you want to have to that PDB. You can access a local PDB in "Readonly" or "Write" mode. You can access a remote PDB in "Readonly" mode.

   Note: A *local PDB* is a PDB that you access from one or more disk drives that are attached to the local host or from one or more disk drives whose contents are accessible through a service such as Network Neighborhood, NFS, AFS, or DFS. Thus, if a PDB satisfies the definition of a local PDB, you can access the PDB without connecting to a remote server. △

   Note: A *remote PDB* is a PDB that you access by connecting to a remote host that has the PDB available as a local PDB. Thus, if a PDB does not satisfy the definition of a local PDB, you must connect to a remote server if you want to access the PDB. △
6. In the **Filesystem** box, select the kind of access that you want to use for that PDB.

7 If the PDB is remote, the Remote Profile box displays the name of the remote server profile that is associated with the PDB. If no profile is associated with the PDB or you want to change which profile is associated with the PDB,

   a select the right arrow. The Remote Server Connection Profiles window opens and lists the names of the sample profiles and any profiles that you have created.
   b If the appropriate remote-server profile is on the list, select it.

      If there is no appropriate remote-server profile, you can create one from this window and then select that profile. For more information about creating a remote-server profile, see "Creating a Remote Server Profile" on page 574.
   c Select **OK** in the Remote Server Connection Profiles window.

8 Select **OK** in the Activating PDB Confirmation window. If you selected **Remote** in the **Filesystem** box, SAS IT Resource Management prompts you for your user ID and password, and uses them to connect to the remote server (and gain access to the PDB and to any other objects that are specified in the remote profile).

   When the activation is completed, the name of the active PDB is in the **Active PDB** field at the top of the Manage PDBswindow, and the name of the active PDB is no longer in the **Other PDBs** list.

9 To exit from the window, select **Close**. The name of the active PDB is in the bar at the top of the window.

Note: You now have a lock on the libraries in the PDB. Remember to release the lock when you finish using the PDB. For more information, see "Releasing the Lock on the Active PDB" on page 319. △

Note: If the PDB is on a remote server, you are now connected to that server. When you finish using the PDB, remember to deactivate the PDB (by activating a local PDB and agreeing to disconnect from that server). For more information about connection and disconnection, see "Overview of Remote Server Profiles" on page 571. △

Adding a PDB to Your List of Known PDBs

In order to access a PDB, you must have that PDB's name on your list of known PDBs.

When you are a new SAS IT Resource Management user, the names of the demonstration PDBs that are local to your host are automatically added to your list of known PDBs. When you make a PDB interactively, its name is automatically added to your list of known PDBs. If you make or activate a PDB in batch mode and your SASUSER library is allocated by the job, the PDB's name is automatically added to your list of known PDBs. When another person makes a PDB, you might want to add the name of the PDB to your list of known PDBs.

You can add a PDB's name to your list of known PDBs

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: To add the name of a PDB that is on a remote server, use SAS IT Resource Management on a host that has a client license.

Note: For information about the PDBs that are automatically on your list, see "Viewing Your List of Known PDBs" on page 324.

Adding a PDB to Your List of Known PDBs (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can add a PDB's name to your list of known PDBs by following these steps:

1 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDB**. The Performance Database Operations window opens and displays your list of known PDBs in the center of this window.
2 To add a PDB to your list, select **Actions**. A menu opens. From the menu, select **Create/Add**. The Specify PDB Location window opens.
3 In the **Enter PDB Prefix** field, type the fully qualified name of the PDB. (The PDB must already exist.) The trailing period (.) is optional.

   Note that you do not specify a remote profile. If you are using the z/OS GUI, you are using the SAS IT Resource Management server software, which cannot access a remote PDB. To access a remote PDB, use the SAS IT Resource Management client software.
4 To return to the Performance Database Operations window, select **OK**. The PDB name that you typed is now on your list.
5 To return to the main menu, select **File**. A menu opens. From the menu, select **End**.

Adding a PDB to Your List of Known PDBs (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can add a PDB's Name to your list of known PDBs by following these steps:

1 From the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens. Your list of known PDBs consists of the active PDB (which is listed at the top of the window) and the PDBs that are in the Other PDBs list (in the center of the window).
2 Select **File**. A menu opens. From the menu, select **Add PDB Entry**. The Add PDB Entry window opens.
3 In the **Location of Existing PDB** field, type the full path and name of the PDB. (The PDB must already exist.) The trailing slash (/) if the PDB is on a UNIX server, trailing backslash (\) if the PDB is on a Windows server, or trailing period (.) if the PDB is on a z/OS server is optional.
4 A *local PDB* is a PDB that you access from one or more disk drives that are attached to the local host or from one or more disk drives whose contents are accessible through a service such as Network Neighborhood, NFS, AFS, or DFS. Thus, if a PDB satisfies the definition of a local PDB, you can access the PDB without connecting to a remote server.

   If the PDB whose name you want to add is not local,

   a select the right arrow on the **Remote Server Profile** field. The Remote Server Connection Profiles window opens, and lists the names of the sample profiles and any profiles that you have created.
   b If the appropriate remote-server profile is on the list, select it.

      If there is no appropriate remote-server profile, you can create one from this window and then select that profile. For more information about creating a remote-server profile, see "Creating a Remote Server Profile" on page 574.
   c Select **OK** in the Remote Server Connection Profiles window.
5 To return to the Manage PDBs window, select **OK** in the Add PDB Entry window.

   The name that you added is on the **Other PDBs** list.
6 To return to the main window, select **Close**.

Associating a Remote-Server Profile with a PDB

If the PDB is remote, you can associate the PDB with a remote-server profile at the same time that you add the name of the PDB to your list of known PDBs. For more

information about adding the name of the PDB, see "Adding a PDB to Your List of Known PDBs" on page 301.

Note: A *local PDB* is a PDB that you access from one or more disk drives that are attached to the local host or from one or more disk drives whose contents are accessible through a service such as Network Neighborhood, NFS, AFS, or DFS. Thus, if a PDB satisfies the definition of a local PDB, you can access the PDB without connecting to a remote server. △

Note: A *remote PDB* is a PDB that you access by connecting to a remote host that has the PDB available as a local PDB. Thus, if a PDB does not satisfy the definition of a local PDB, you must connect to a remote server if you want to access the PDB. △

Backing Up a PDB

The fastest way to back up a PDB is to use an operating system utility. You can back up a PDB

- by using a z/OS utility (on z/OS)
- by using a UNIX utility (on UNIX)
- by using a Windows utility (on Windows).

Backing Up a PDB (z/OS)

There is no preferred z/OS utility. Here is an example for z/OS:

```
//BACKUP  EXEC PGM=ADRDSSU                              #1
//TAPE     DD DSN=your.itrm.pdb,DISP=(OLD,KEEP),       #2
//            UNIT=cart,RETPD=30
//SYSPRINT DD SYSOUT=A
//SYSIN   DD *
  DUMP DATASET(INCLUDE(your.itrm.pdb.*))               #3
            OUTDD(TAPE) - SHARE COMPRESS
//
```

Notes:

1. This example uses the IBM DFDSS program. You might prefer to use the backup procedure that is commonly used at your site.
2. You might want to change characteristics on the TAPE allocation according to your needs or according to site-specific rules.
3. You can use the DFDSS DUMP command to back up volumes and data sets, and you can then use the RESTORE command to recover them. You can also make incremental backups of your data sets by doing a data set DUMP with RESET specified and filtering on the data set–changed flag. For more information about this facility, refer to the IBM manual Data Facility Data Set Services: Reference V2R5 (SC26-4389-08). You might want to change the DUMP command according to your needs or according to site-specific rules.

Backing Up a PDB (UNIX)

There is no preferred UNIX utility. Here is an example for UNIX:

- To make the backup for a PDB named "mypdb" to a file "/u/sasabc/mypdb.tar.Z", you can use these commands

```
cd <parent-dir-of-mypdb>
tar cf - mypdb | compress > /u/sasabc/mypdb.tar.Z
```

where

- <*parent-dir-of-mypdb*> is the full pathname of the PDB (that is, it contains nine subdirectories: DETAIL, DAY, etc.)
- *mypdb* is the PDB name (without the path)
- */u/sasabc/mypdb.tar.Z* is an arbitrary file name, in which the .Z indicates that it is compressed and the *.tar* indicates that the tar utility was used to convert a directory structure to a flat file.

- To restore the above PDB from the above tar file, you can use these commands:

```
cd <parent-dir-of-mypdb>
uncompress -c /u/sasabc/mypdb.tar.Z | tar xf -
```

Backing Up a PDB (Windows)

There is no preferred Windows utility. On Windows, you can use your site's standard backup utility.

Building Views

The PDB's data is made available through views. If the PDB's views do not exist or if one or more table or variable definitions changed after the views were built, SAS IT Resource Management automatically builds views at the process step or reduce step.

If you make changes to the table definitions and/or variable definitions and you want to use the views (for example, for reporting) before the next time that you run process or reduce, you can force a build of the views

- by using the %CPSTART macro and the %CPDDUTL macro (with the BUILD VIEWS control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see the topic %CPDDUTL in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL control statement BUILD VIEWS, see the BUILD VIEWS topic in the SAS IT Resource Management macro reference documentation.

Note: If you use the GUI to make one or more changes to a table definition or variable definition, the views are automatically rebuilt for you at that time. △

Combining PDBs

A PDB has properties and stores its data in tables, with one or more tables for each data source.

There are advantages to having data from different data sources in the same PDB:

- If you want to generate a report on data in more than one table, it is convenient to have the tables in the same PDB.
- If you want to generate reports interactively, you do not need to change the active PDB for each report.

You can combine PDBs (that is, copy their tables to the same PDB)

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: If you would prefer to have the copy delete the properties and tables in the "To PDB" before copying PDB properties and tables from the "From PDB," use the **Copy**

PDBs operation. For more information about copying PDBs, see "Copying a PDB within the Local File System" on page 306. △

Note: 1. The PDB's properties apply to all of the data in the PDB. Thus, combine data only if the PDB properties are appropriate for *all* of the data.

2. You can copy tables (including the data in the tables) into a combined PDB at any time that using the same PDB would be convenient. You can also copy tables into separate PDBs (that is, split a PDB) at any time that using different PDBs would be convenient. For information about splitting a PDB, see "Splitting One PDB into Two PDBs" on page 320.

3. To combine PDBs, you must be using SAS IT Resource Management on a host with a server license, and the PDBs must be local. △

Combining PDBs (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can combine PDBs by following these steps:

1. Display your list of known PDBs. For information about displaying your list of known PDBs, see "Viewing Your List of Known PDBs" on page 324.
2. Decide which PDB to use as the combined PDB.

   If you do not want to use a PDB that is already on your list of known PDBs, create a new PDB. For information about creating a PDB, see "Creating a PDB" on page 308. You can skip the step of adding tables when you create a PDB for this purpose.
3. If the PDB that you decided to use for the combined PDB is the active PDB, activate a different PDB (for example, a demonstration PDB). For information about activating a PDB, see "Activating a PDB" on page 299.
4. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDB**. The Performance Database Operations window opens.
5. Select **Actions**. A menu opens. From the menu, select **Combine PDBs**. The Combine Selected Tables to Other PDB window opens.
6. Select the down arrow that is associated with the **From** field. The Combine From PDB window opens. Select the name of the PDB from which you want to copy tables.
7. Select the down arrow that is associated with the **To** field. The Combine To PDB window opens. Select the name of the PDB to which you want to copy tables (that is, the PDB that you want to use as the combined PDB).

   Note: The tables already in the "To PDB" will not be affected by your copying additional tables to the "To PDB" as long as the names of the tables are different. Similarly, the PDB properties in the "To PDB" will not be affected by your copying additional tables to the "To PDB." △
8. In the lower-left corner of the window, a list is displayed of all the tables in the "From PDB" that do not already exist in the "To PDB." Select each of the tables that you want to copy.

   In the lower-right corner of the window, a list is displayed of each of the tables that you select.
9. When you finish selecting tables, select **Run**. A busy symbol is displayed in the SAS message area at the bottom of the window.
10. When the busy symbol disappears from the message area, select **Goback** to return to the Performance Database Operations window.
11. To return to the main menu, select **File**. A menu opens. From the menu, select **End**.

Reminder: The tables were copied, not moved. You might want to delete the tables from the "From PDB."

Reminder: You might want to change your batch job(s) to use the combined PDB.

Combining PDBs (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can combine tables into one PDB:

1. Display your list of known PDBs. For information about displaying your list of known PDBs, see "Viewing Your List of Known PDBs" on page 324.
2. Decide which PDB to use as the combined PDB.

   If you do not want to use a PDB that is already on your list of known PDBs, create a new PDB. For information about creating a PDB, see "Creating a PDB" on page 308. You can skip the step of adding tables when you are creating a PDB for this purpose.
3. If the PDB that you decided to use for the combined PDB is the active PDB, activate a different PDB (for example, a demonstration PDB). For information about activating a PDB, see "Activating a PDB" on page 299.
4. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
5. From your list of known PDBs, select the "To PDB" (the PDB that you want to use as the combined PDB).

   Note: The tables already in the "To PDB" will not be affected by your coping additional tables to the "To PDB" as long as the table names are different. Similarly, the PDB properties in the "To PDB" will not be affected by your copying additional tables to the "To PDB". △
6. Select **File**. A menu opens. From the menu, select **Combine PDBs**. The Combine PDB window opens.
7. Select the down arrow that is associated with the **From** field. The Combine From PDB window opens. Select the name of the PDB from which you want to copy tables.
8. In the lower-left corner of the window, a list is displayed of all the tables in the "From PDB" that do not already exist in the "To PDB." Select each of the tables that you want to copy.

   In the lower-right corner of the window, a list is displayed of each of the tables that you select.
9. When you finish selecting tables, select **OK**.

   When SAS IT Resource Management finishes copying the selected table(s), the following message is displayed in the SAS message area (at the bottom of the window): "PDB <name-of-"To-PDB"> combined."
10. To return to the main window, select **Close**.

Reminder: The tables were copied, not moved. You might want to delete the tables from the "From PDB."

Reminder: You might want to change your batch job(s) to use the combined PDB.

Copying a PDB within the Local File System

A PDB has properties and stores its data in tables, with one or more table for each data source.

Within the local file system, you can copy a PDB to another PDB (that is, copy one PDB's properties, tables, and data to another PDB)

- by using the %CPSTART macro and the %CPDBCOPY macro (in batch mode). For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDBCOPY macro, see the %CPDBCOPY topic in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

CAUTION:
The PDB properties and tables in the "To PDB" are deleted before the copying begins. If you do not want to lose existing properties and tables in the "To PDB," use a new PDB as the "To PDB" or use the combine PDBs operation. For more information about combining PDBs, see "Combining PDBs" on page 304. △

To copy a PDB to a remote location, see "Exporting a PDB" on page 316, "Transporting a PDB outside the Local File System" on page 322, and "Importing a PDB" on page 318.

Copying a PDB within the Local File System (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can copy a PDB within the local file system by following these steps:

1. Display your list of known PDBs. For more information about displaying your list, see "Viewing Your List of Known PDBs" on page 324.
2. If the "To PDB" does not already exist, create a PDB to use as the "To PDB." For more information about creating a PDB, see "Creating a PDB" on page 308.

   CAUTION:
   In the "To PDB," the existing PDB properties and all existing tables and data will be deleted before the copying begins. Thus, if you do not want to lose the existing properties or existing tables from an existing PDB, make a new PDB to use as the "To PDB." △
3. If the "To PDB" is the active PDB, make a different PDB active (for example, activate a demonstration PDB). For more information about activating a PDB, see "Activating a PDB" on page 299.
4. From the Performance Database Operations window, select **Actions**. A menu opens. From the menu, select **Copy PDB**. The Copy an Entire PDB to Another PDB window opens.
5. Select the down arrow that is associated with the **From** field. The Copy From PDB window opens. Select the "From PDB."
6. Select the down arrow that is associated with the **To** field. The Copy to PDB window opens. Select the "To PDB."
7. Select **Run**. The Confirmation window opens as a reminder that all tables in the "To PDB" will be deleted before the copying begins. Select **OK**. A busy symbol is displayed in the SAS message area (at the bottom of the window).
8. When the busy symbol disappears, the copying is complete. Select **Goback** to return to the Performance Database Operations window.
9. To return to the main menu, select **File**. A menu opens. From the menu, select **End**.

Copying a PDB within the Local File System (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can copy a PDB within the local file system by following these steps:

1 Display your list of known PDBs. For more information about displaying your list, see "Viewing Your List of Known PDBs" on page 324.

2 If the "To PDB" does not already exist, create a PDB to use as the "To PDB." For more information about creating a PDB, see "Creating a PDB" on page 308.

CAUTION:
In the "To PDB," before the copying begins, the existing PDB properties are removed and all existing tables are deleted. Thus, if you do not want to lose the existing properties or existing tables from an existing PDB, make a new PDB to use as the "To PDB." △

3 If the "To PDB" is the active PDB, make a different PDB active (for example, activate a demonstration PDB). For more information about activating a PDB, see "Activating a PDB" on page 299.

4 From the Manage PDBs window, in the Other PDBs list select the "From PDB."

5 From the Manage PDBs window, select **File**. A menu opens. From the menu, select **Copy PDB**. The Copy PDB window opens.

6 Select the down arrow that is associated with the **To** field. The Select window opens. Select the "From PDB." When the name of the "From PDB" is in the **Folder** field, select **OK**.

7 Select **OK**. A confirmation window reminds you that all properties and tables in the "To PDB" will be deleted before the copying begins. Select **Yes**.

8 When the copying is finished, another confirmation window states that the "From PDB" was copied to the "To PDB." Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.

9 To return to the main window, select **Close**.

Copying a PDB's Properties to a Site Library

A PDB's default properties are acquired automatically from the active site library at the time that the PDB is created. Typically, the PDB's properties are used "as is" or edited slightly. If, however, the PDB's properties are greatly changed and would be useful for other PDBs created in the future, you can copy the PDB's properties back to the site library. For more information about such a copy, see "Copying a PDB's Options to a Site Library" on page 273.

Reminder: PDB options is another name for PDB properties.

Copying a Site Library's Properties to a PDB

A PDB's default properties are acquired automatically from the active site library at the time that the PDB is created. Typically, the PDB's properties are used "as is" or edited slightly. If, however, the PDB's properties are greatly changed and you want to restore them to the default values, you can re-copy the PDB's default properties from the site library. For more information about such a copy, see "Copying a Site Library's Options to a PDB" on page 274.

Reminder: PDB options is another name for PDB properties.

Creating a PDB

You can create a PDB

- by using the %CPSTART macro in batch mode. For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

For information about the size of the PDB, see "Guidelines for Allocating Space in the PDB" in "Appendix 1: Recovery Procedures" in the macro reference documentation and "Guidelines for Revising Space Allocations in the PDB" in "Appendix 1: Recovery Procedures" in the macro reference documentation . For information about the size of the archive, which might or might not be in the same location as the PDB, see "Guidelines for Allocating Space in the Archive" in "Appendix 1: Recovery Procedures" in the macro reference documentation and "Guidelines for Revising Space Allocations in the Archive" in "Appendix 1: Recovery Procedures" in the macro reference documentation .

Note: In order to create a PDB, you must be using SAS IT Resource Management on a host that has a server license. △

Creating a PDB (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can create a PDB in two ways: by using the QuickStart Wizard, or by using the Define New PDB Wizard. The QuickStart Wizard does more for you than the Define New PDB Wizard does, so check the QuickStart Wizard first.

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **QuickStart Wizard**. The Welcome to the QuickStart Wizard window opens.
2. Select the down arrow that is associated with the **Collector** field. A menu opens. The items in the menu describe types of logs that collectors generate and that SAS IT Resource Management staging code reads.

   If the data that you intend to process into the new PDB matches any of these log formats, select the format and proceed through the wizard (for example, select **Next** at the bottom of this window). The wizard will generate a PDB for you, and (so that you can direct reports to the Web) the wizard will create a report structure for you. The wizard will also generate one or more jobs to process data into the PDB, reduce the data, and report on the data.
3. If the data that you intend to process into the new PDB does not match any of these log formats, select **Cancel** at the bottom of the window to return to the main menu (in order to select a different path).

If the QuickStart Wizard does not handle your type of data, use the Define New PDB Wizard instead.

1. On the main menu, select **PDB Admin**. A menu opens. From the menu, select **Define New PDB**. The Define New PDB window opens.
2. Select **Location**. The Specify PDB Location window opens. In the Enter PDB prefix field, type the fully qualified name of the PDB that you want to create. The trailing period (.) is optional.
3. Select **OK** to continue. The Confirmation window opens. Select **OK** to continue. The PDB Allocation window opens.
4. Select the down arrow that is associated with the **Unit** field. A menu opens. From the menu, select the appropriate value. SAS IT Resource Management returns you to an updated version of the PDB Allocation window.
5. If you want to specify a particular volume, fill in the **Volser** field.

If you want to specify a particular expiration date, fill in the **Expiration Day** field.

Check the other fields and make any changes that are appropriate for the PDB that you want to allocate.

6 Select **Run**. SAS IT Resource Management allocates the PDB and returns you to an updated version of the PDB Allocation window.

7 Select **Goback** to continue. The Display/Modify PDB Options window opens.

8 For each PDB option (property) that you want to specify, there is a field with that name. Select or provide the appropriate value for that field.

- **PDB Access Mode** field:

  You can change the default values in this window. You will need write access to do that. Also, you are likely to want to add table definitions to the PDB in the next step. You will need write access to do that.

  For more information about a PDB's access mode, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.

- **Provide MXG PDB Views** field:

  Views of the SAS IT Resource Management PDB are automatically built for you. If you have existing MXG reporting programs that refer to views by their names in the MXG PDB, and if you want to continue to use those programs without changing the view names to the names to the names of the views of the SAS IT Resource Management PDB, select **Yes**.

  For more information about MXG views, see "Viewing/Editing Your Active PDB's MXG Views Property" on page 330.

- **SAS/SHARE Server Name** field:

  If you will manage multi-user access to the PDB by using a SAS/SHARE server, type the name of the server in this field. For more information about the SAS/SHARE server, see the SAS/SHARE documentation for your version of SAS.

- **Start of Week** field:

  The week level of the PDB does a grouping by week. If your enterprise has a convention about the day that is considered to be the start of the week, select **Change Start of Week** and then select that day. If your enterprise has no convention about the start of week, select **Change Start of Week** and select **Sunday**, if Sunday is not already selected. (Sunday is the default value.) For more information about Start of Week, see "Viewing/Editing Your Active PDB's Start of Week" on page 333.

- **Archive** fields:

  Archiving the data that is coming into the PDB is optional. If you want an archive (that is, a set of archive libraries), do the following:

  - **Archive Type** field:

    Select whether you want the archive libraries to be on **DASD** or **Tape**.

  - **Archive Root** field:

    The name of an archive library is a concatenation of the high-level qualifiers that you specify for archive root and of the qualifier *.Znnnnnnn* (where *nnnnnnn* is *00000001*, *0000002*, etc.).

    By default, the high-level qualifiers that are used for archive root are identical to the PDB's name.

  - **Archive Unit** field:

Type in the value that you want to use for the UNIT= parameter on the archive libraries. For example, you might want to specify *3380* or *SYSDA*.

- **`Archive Library Allocation Parameter String`** field:

  Provide any other parameters that you want to use when an archive library is generated. For example, if the archive is on tape, you might want to specify *LABEL=(1,SL,,,RETPD=365)*.

Note that you are providing the archive information in case data is actually archived. Whether data is archived depends on the setting of one property of a PDB table. For more information about archiving, see "Viewing/ Editing Your Active PDB's Archive Information" on page 326 and "Appendix 2: Archiving Data" in the SAS IT Resource Management macro reference documentation.

9 Select **`OK`** to continue. SAS IT Resource Management returns you to the Define New PDB window.

10 Unless you want an empty PDB, select **`Add Table(s)`**. The Collector list opens. Select the collector that generates the logs that contain the data that you want in the PDB. SAS IT Resource Management displays an asterisk next to your selection.

11 Select **`OK`** to continue. The Select Table(s) from List window opens.

12 Select all of the tables whose definitions you want in the PDB. SAS IT Resource Management highlights the tables that you select.

13 After you have selected all of the tables, select **`OK`** to continue. SAS IT Resource Management displays a busy symbol in the SAS message area (at the bottom of the window).

14 After all the table definitions have been added for all the selected tables, SAS IT Resource Management returns you to the Define New PDBs window. The tables that are listed have definitions in the PDB.

Note: If you select **`Add Table(s)`** a second time, the previous selection is not lost. You can add additional tables from the same collector or a different collector. △

15 Select **`Goback`** to return to the main menu.

Copying a Test PDB (z/OS)

If you create a test PDB and then want to copy the structure, settings, and data to your production PDB, follow these steps:

1 Create an empty production PDB. You can do this interactively or in batch.

- Interactively:

  a From the main window, select **PDB Admin ▶ Manage PDBs/Switch PDB ▶ Actions ▶ Create/Add**
  b Type the name of the (high-level qualifiers of the) production PDB and then select **`OK`** on the next two windows. The PDB Allocation window opens.
  c Enter your space estimates and make any other required changes.
  d Select **`Run`** to allocate your production PDB.
  e Select **`Goback`**. The new PDB is now included in the list of known PDBs.
  f Select the following to activate the new PDB: **ItemActions ▶ Activate ▶ WRITE access ▶ OK**

- Batch:

  Review the ALLOC job step in the following code:

```
//CMQSTART JOB (accounting info),'ITRM Quick Start',              #1
//       MSGLEVEL=(1,1),TIME=20,REGION=32M,NOTIFY=
```

```
                .
                .                                                        #2
                .
//*****************************************************************
//*                                                                 **
//* Delete the old PDB data libraries, to make this job able to be **
//* rerun.                                                          **
//*                                                                 **
//*****************************************************************
//CLEANUP EXEC PGM=IDCAMS                                           #3
//SYSPRINT DD SYSOUT=*
//SYSIN DD DATA,DLM='$$'
  DELETE 'YOUR.NEWPDB.*'
  SET MAXCC=0
$$
//*****************************************************************
//*                                                                 **
//* Allocate  SAS IT Resource Management for z/OS Performance       **
//* Database data libraries.                                        **
//*                                                                 **
//*****************************************************************
//ALLOC   EXEC PGM=IEFBR14                                          #4
//DETAIL   DD DSN=YOUR.NEWPDB.DETAIL,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(120,20)),UNIT=SYSDA                       #5
//DAY      DD DSN=YOUR.NEWPDB.DAY,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(99,25)),UNIT=SYSDA
//WEEK     DD DSN=YOUR.NEWPDB.WEEK,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(30,25)),UNIT=SYSDA
//MONTH    DD DSN=YOUR.NEWPDB.MONTH,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(99,10)),UNIT=SYSDA
//YEAR     DD DSN=YOUR.NEWPDB.YEAR,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(20,5)),UNIT=SYSDA
//ADMIN    DD DSN=YOUR.NEWPDB.ADMIN,DISP=(NEW,CATLG,DELETE),
//            SPACE=(TRK,(15,10)),UNIT=SYSDA
//COLLECT  DD DSN=YOUR.NEWPDB.COLLECT,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(9,9)),UNIT=SYSDA
//PDBWORK  DD DSN=YOUR.NEWPDB.PDBWORK,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(180,80)),UNIT=SYSDA
//DICTLIB  DD DSN=YOUR.NEWPDB.DICTLIB,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(4,1)),UNIT=SYSDA,
//            DCB=(LRECL=6144,BLKSIZE=6144,RECFM=PS)
```

Note: 1. The JCL job card needs to be changed to make it be appropriate for your site.

2. Comments that explain the job and listing the customization instructions are omitted here for the sake of brevity.

3. The CLEANUP job step invokes IDCAMS to delete the data sets for the PDB that you will allocate. If you need to rerun the job, this job step makes certain that the previous PDB is out of the way before you allocate a new one with the same name.

Be certain that the PDB names in the CLEANUP job step and ALLOC job step match.

Warning: This will delete all data sets that start with this prefix, so be sure that you have no other data sets that begin with the same qualifiers!

4. The ALLOC job step invokes IEFBR14 to allocate the data sets for the PDB.

5. You might need to adjust the SPACE= parameter and/or the UNIT= parameter for the PDB data sets. △

2 Copy your test PDB's structure, setting, and data to your production PDB.

Set up and submit a batch job with a SAS job step (using JCL like that used in the LOAD job step in the CMQSTART job, except that you can remove the definition of the SMF ddname). In the job step, include the following macro calls:

```
%CPSTART(MODE=BATCH SYSTEM=MVS,
         ROOT='itrm.prefix.',
         PDB='your.test.pdb',
         DISP=OLD,SHARE=N/A,
         MXGSRC=('MXG.USERID.SOURCLIB' 'MXG.MXG.SOURCLIB'),
         MXGLIB=MXG.MXG.FORMATS
         );

%CPDBCOPY(your.test.pdb, your.production.pdb);
```

Creating a PDB (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create a PDB in two ways: by using the QuickStart Wizard or by using the Create PDB Wizard. The QuickStart Wizard does more for you than the Create PDB Wizard does, so check the QuickStart Wizard first.

1 On the main window's **Administration** tab, select **QuickStart Wizard**. The Welcome to the QuickStart Wizard window opens.

2 Select the down arrow that is associated with the **Collector** field. A menu opens. The items in the menu describe types of logs that collectors generate and that SAS IT Resource Management staging code reads.

3 If the data that you intend to process into the new PDB matches any of these log formats, select the format and proceed through the wizard (for example, select **Next** at the bottom of this window). The wizard will generate a PDB for you. The wizard will also generate one or more jobs to process data into the PDB, reduce the data, and report on the data. The wizard directs you to files that tell you how to set the PDB's properties (options).

4 If the data that you intend to process into the new PDB does not match any of these log formats, select **Cancel** to return to the main menu.

If the QuickStart Wizard does not handle your type of data, use the Create PDB Wizard instead.

1 On the main window's **Administration** tab, select **Create PDB Wizard**. The PDB Wizard window opens. Follow the instructions on this window and proceed through the wizard (for example, select **Next** at the bottom of this window). The wizard will create a PDB for you.

2 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens. Select **Properties**. The PDB Options window opens. Review, and adjust if necessary, the values of the PDB properties (options).

- Access Mode:

  On the **Access** tab, you can select whether you want "Write" or "Readonly" access to the PDB. You can change your access mode to the PDB at any time. This setting is in effect until the next time that you change it. To change any of the other properties of the PDB, the **Access Mode** setting must be "Write."

For more information about a PDB's access mode, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.

- Greenwich Mean Time (GMT) Deviation:

  On the **Time Zone** tab, you can specify the number of hours' difference between GMT and local time at the site where the data is logged. For some data sources, this information is needed to convert data logged with GMT datetime stamps to data logged with local datetime stamps.

  For more information about the GMT deviation, see "Viewing/Editing Your Active PDB's GMT Deviation" on page 328.

- Daylight Saving Time (DST) Expression:

  On the **Time Zone** tab, you can specify the period during which Daylight Saving Time is in effect. For some data sources, this information is needed to convert data logged with DST datetime stamps to data logged with standard datetime stamps.

  For more information about the DST expression, see "Viewing/Editing Your Active PDB's DST Definition" on page 327.

- Start of Week:

  The week level of the PDB does a grouping by week. If your enterprise has a convention about the day that is considered to be the start of the week, select the down arrow and that day. If your enterprise has no convention about the start of week, select the down arrow and select **Sunday**, if Sunday is not already selected. (Sunday is the default start of week.)

  For more information about the Start of Week, see "Viewing/Editing Your Active PDB's Start of Week" on page 333.

- **Archive** fields:

  Archiving the incoming data to SAS IT Resource Management is optional. If you do want an archive (that is, a set of archive libraries), do the following:

  - **Archive Type** field:

    Select whether you want the archive libraries to be on **Disk** or **Tape**.

  - **Archive Device** field:

    Type the name of the device where the archive libraries are to be written. For example, this could be *c:*. By default, the value that is used for archive device is the name of the device where the PDB is located.

  - **Archive Root** field:

    The name of an archive library is a concatenation (with slash or backslash, as appropriate) of the values that you supply for Archive Device, Archive Root, and *Znnnnnnn* (where *nnnnnnn* is *00000001*, *0000002*, etc.). By default, the value that is used for archive root is the name of the PDB.

    By default, the high-level qualifiers that are used for the archive's root are identical to the PDB's name.

  - **Archive Library Allocation Parameter String** field:

    Provide any parameters that you want to use when a SAS LIBNAME statement is used to reference an archive library.

  Note that you are providing the archive information in case data is actually archived. Whether data is archived depends on the setting of one property of a PDB table. For more information about archiving, see "Viewing/Editing Your Active PDB's Archive Information" on page 326 and "Appendix

2: Archiving Data" in the SAS IT Resource Management macro reference documentation.

3 Select **OK** to continue.

4 To return to the main window, select **Close**.

Copying a Test PDB (UNIX/Windows)

If you create a test PDB and then want to copy the structure, settings, and data to your production PDB, use these steps.

1 Create an empty production PDB.

   a On the main window's **Administration** tab, select **Create PDB Wizard**.
   b Name the new PDB. and select **Next.**
   c On each subsequent window, select **Next** without making any additional changes until the wizard's final window is displayed.
   d Select **Finish** in the wizard's final window to activate the new, empty PDB.

2 Copy your test PDB's structure, settings, and data to your production PDB.

   a On the main window's **Administration** tab, select **Manage PDBs**.
   b Select the PDB that you want to copy. Then select **File ▶ Copy PDB** The Copy PDB window is displayed.
   c In the Copy To window, scroll to the name of your test PDB and select it.
   d Select **OK**. The following message is displayed: "PDB *your_new_pdb* already exists. Do you want to overwrite it?"
   e Verify that you have selected the correct PDB, and select **Yes**. This message is displayed: "Copied *your_test_pdb* to *your_new_pdb*."
   f Select **OK**. The list of known PDBs is updated to reflect the new PDB.
   g Select **Close** to return to the main window.

Deleting a PDB

You can delete a PDB

□ by using the SAS IT Resource Management GUI for z/OS

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: 1. In order to delete a PDB, you must be using SAS IT Resource Management on a host that has a server license, and the PDB must be local.

2. After this operation, the PDB is not only removed from your list of known PDBs. It is also removed from the local file system. If you think that you might need the PDB later, you must back it up before you delete it. △

Deleting a PDB (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can delete a PDB by following these steps:

1 If the PDB that you want to delete is the active PDB, activate a different PDB. For more about activating a PDB, see "Activating a PDB" on page 299.

2 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDB**. The Performance Database Operations window opens.

3 Select the PDB that you want to delete. Then select **ItemActions**. A menu opens. From the menu, select **Delete PDB**.

4 The Confirmation window opens. Select **OK**.

5 To return to the main menu, select **File**. A menu opens. From the menu, select **End**.

Deleting a PDB (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete a PDB by following these steps:

1 If the PDB that you want to delete is the active PDB, activate a different PDB. For information about activating a PDB, see "Activating a PDB" on page 299.
2 From the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
3 From the **Other PDBs** list, select the name of the PDB that you want to delete.
4 Select **File ▶ Delete PDB ▶ Yes**
5 After you get a message that the PDB has been deleted, select **OK**.
6 To return to the main window, select **Close**.

Disassociating a Remote-Server Profile from a PDB

If you activated a remote PDB and specified a remote-server profile, you might need to make one of these corrections later:

- specify a different remote-server profile for that (remote) PDB
- specify that the PDB is not remote but local.

To make either of these changes, activate the PDB again and, during the activation, do one of these actions:

- specify a different remote-server profile for that (remote) PDB.
- specify that the PDB is local.

For more information about activating a PDB, see "Activating a PDB" on page 299.

Note: The SAS IT Resource Management GUI does not permit you to reactivate the currently active PDB. If the PDB for which you want to make these changes is the active PDB, activate another PDB and then reactivate the PDB for which you want to make these changes. △

Exporting a PDB

Exporting a PDB reads a normally formatted PDB as input and writes a transport-formatted PDB as output. (The transport-formatted PDB does not overwrite the normally formatted PDB.) Thus, exporting a PDB is equivalent to packaging it in preparation for transporting it (copying it) outside of the local file system.

You can export a PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

For more information about transporting a transport-formatted PDB to another file system, see "Transporting a PDB outside the Local File System" on page 322. For more information about importing (that is, un-packaging) the transported PDB on that other file system, see "Importing a PDB" on page 318.

Exporting a PDB (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can export a PDB by following these steps:

1 Allocate two data sets whose names are identical except for the lowest-level qualifier:
   a *YOURID.YOURPDB*.TRANFILE, where
      - TRANFILE is any name except TRANBUFF and the higher qualifiers could be any legal values at all (that is, they do not need to be YOURID or YOURPDB or even two levels)
      - LRECL=80
      - BLKSIZE=6160
      - DSORG=PS
      - RECFM=FB
      - The amount of space that is needed is somewhat difficult to predict. If you allocate a file that is four times the size of the PDB, that is probably more than enough space.
   b *YOURID.YOURPDB*.TRANBUFF, where
      - the TRANBUFF qualifier must literally be TRANBUFF. The other levels must be identical to the ones for TRANFILE.
      - LRECL, BLKSIZE, ORG, and RECFM are the same as for TRANFILE.
      - Again, the amount of space needed is difficult to predict. If you allocate a file that is four times the size of the largest library in the PDB, that is probably more than enough space.
2 If the PDB that you want to export is the active PDB, activate a different PDB. For more information about activating a PDB, see "Activating a PDB" on page 299.
3 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDB**. The Performance Database Operations window opens.
4 Select the PDB that you want to export.
5 Select **ItemActions**. A menu opens. Select **Export PDB**. The Export All Tables in a PDB to a Transportable File window opens.
6 In the Enter Output File Name window, type the name that you gave to *YOURID.YOURPDB*.TRANFILE.
7 Select **OK**.
8 After the PDB is exported, you can delete *YOURID.YOURPDB*.TRANBUFF, which was only a work space.

Exporting a PDB (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can export a PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2 If the PDB that you want to export is the active PDB, activate a different PDB. For more information about activating a PDB, see "Activating a PDB" on page 299.
3 From the **Other PDBs** list, select the PDB that you want to export.
4 Select **File**. A menu opens. Select **Export PDB**. The Export window opens.
5 In the **Enter Output File Name** field, type the full path and name of the file that will contain the transport-format PDB.
6 Select **OK**.
7 Select **Close** to return to the main window.

Note: This operation generates a work file that is in the same directory as the file that contains the transport-format PDB. The work file has the name TRANBUFF. After the file that contains the transport-format PDB is generated, the TRANBUFF file is automatically deleted. △

Importing a PDB

Importing a PDB reads a transport-formatted PDB as input and writes a normally formatted PDB as output. (The normally formatted PDB does not overwrite the transport-formatted PDB.) Thus, importing a PDB is equivalent to un-packaging it after transporting it (copying it) from somewhere outside of the local file system.

You can import a PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: 1.For the import function to work, the transport-formatted PDB must have been generated by the SAS IT Resource Management export function. For more information about exporting a PDB, see "Exporting a PDB" on page 316. For more information about transporting a PDB outside of the local file system, see "Transporting a PDB outside the Local File System" on page 322.

2. Before you import a PDB whose data involves the use of MXG, you will need to install MXG on the host where the PDB will be located. △

Importing a PDB (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can import a PDB by following these steps:

1. Create a new PDB, without adding tables. (For more information about creating a PDB, see "Creating a PDB" on page 308.) You should be able to estimate the amount of space needed by looking at the PDB on the host from which it was exported.
2. If the new PDB is the active PDB, activate a different PDB. For more information about activating a PDB, see "Activating a PDB" on page 299.
3. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDB**. The Performance Database Operations window opens.
4. Select the PDB that you created without tables.
5. Select **ItemActions**. A menu opens. From the menu, select **Import PDB**. The Transportable File window opens.
6. Type the name that you gave the transport-formatted PDB when you transported it to this host.
7. Select **OK**.
8. After the PDB is imported, you can delete the transport-format version of the PDB from this host and from the host where it was generated.
9. If you want to generate reports on data in this PDB before the next process task or reduce tasks runs against this PDB, rebuild the views of the data. (If this PDB is intended to be used only as a copy of the PDB that is in everyday use, do not process data into it or reduce data in it.) For more information about rebuilding views, see "Building Views" on page 304.

Importing a PDB (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can import a PDB by following these steps:

1 Create a new PDB, without adding tables. For more information about creating a PDB, see "Creating a PDB" on page 308.
2 If the new PDB is the active PDB, activate a different PDB. For more information about activating a PDB, see "Activating a PDB" on page 299.
3 On the main window's **Administration** tab, **Manage PDBs**. The Manage PDBs window opens.
4 Select the PDB that you created without tables.
5 Select **File**. A menu opens. From the menu, select **Import PDB**. The Transportable File window opens.
6 Type in the name that you gave the transport-formatted PDB when you transported it to this host.
7 Select **OK**.
8 After the PDB is imported, you can delete the transport-format version of the PDB from this host and from the host where it was generated.
9 If you want to generate reports on data in this PDB before the next process task or reduce task runs against this PDB, rebuild the views of the data. (If this PDB is intended to be used only as a copy of the PDB that is in everyday use, do not process data into it or reduce data in it.) For more information about rebuilding views, see "Building Views" on page 304.

Releasing the Lock on the Active PDB

When a PDB is activated, it is activated in the mode that you specified ("write" or "readonly" (for UNIX and Windows), *OLD* or *SHR* (for z/OS)). The mode causes a lock of one kind or another to be placed on the PDB. As soon as you finish using the PDB, remember to release the lock.

You can release the lock on the active PDB by doing one of the following:

- activating a different PDB. (If the PDB that you are deactivating is remote, agree to disconnect when SAS IT Resource Management prompts you to disconnect from the remote server.)
- exiting from SAS IT Resource Management and SAS. All remote connections close automatically.

Removing a PDB from Your List of Known PDBs

You can delete a PDB's name from your list of known PDBs

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: The PDB's name is removed from your list of known PDBs, but the PDB continues to exist on its file system. △

Removing a PDB from Your List of Known PDBs (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can remove a PDB's name from your list of known PDBs by following these steps:

1 If the PDB whose name you want to remove is the active PDB, activate a different PDB. For more about activating a PDB, see "Activating a PDB" on page 299.
2 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDB**. The Performance Database Operations window opens.

3 Select the name of the PDB that you want to remove from your list of known PDBs.
4 Select **ItemActions**. A menu opens. From the menu, select **Remove PDB Entry**. The Confirmation window opens.
5 Select **OK**.
6 To return to the main menu, select **File**. A menu opens. From that menu, select **End**.

Removing a PDB from Your List of Known PDBs (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can remove a PDB's name from your list of known PDBs by following these steps:

1 If the PDB whose name you want to remove is the active PDB, activate a different PDB. For more about activating a PDB, see "Activating a PDB" on page 299.
2 On the main window's **Administration** tab, select **Manage PDB**. The Manage PDBs window opens.
3 Select the PDB whose name you want to remove from your list of known PDBs.
4 Select **File**. A menu opens. From the menu, select **Remove PDB Entry**.
5 To return to the main window, select **Close**.

Splitting One PDB into Two PDBs

A PDB stores its data in tables, with one or more tables for each data source. There are advantages to having data from different sources in different PDBs:

- You can back up, process, reduce, and report on the data in parallel.
- You are less likely to run into file size problems (UNIX has a file size limitation of 2 gigabytes).
- It might be easier to handle sites with different time zones.
- It might be easier to handle multiple users that need update access at the same time.

You can split a PDB into two PDBs

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: You can copy some tables into another PDBs (that is, split a PDB) at any time that using different PDBs would be convenient. You can also copy tables into a combined PDB at any time that using the same PDB would be convenient. For information about combining PDBs, see "Combining PDBs" on page 304. △

Note: To split a PDB, you must be using SAS IT Resource Management on a host with a server license, and the PDBs must be local. △

Splitting One PDB into Two PDBs (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can split a PDB into two by following these steps:

1 Create a new PDB. For information about creating a PDB, see "Creating a PDB" on page 308. (You do not need to add any tables during creation when creating a PDB for this purpose.)
2 If the new PDB is the active PDB, activate some other PDB (for example, a demonstration PDB). For information about activating a PDB, see "Activating a PDB" on page 299.

3 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDB**. The Performance Database Operations window opens.

4 Select **Actions**. A menu opens. From the menu, select **Combine PDBs**. The Combine Selected Tables to Other PDB window opens.

5 Select the down arrow that is associated with the **From** field. The Combine From PDB window opens. Select the name of the PDB that you want to copy tables from (that is, select the name of the PDB that you are splitting).

6 Select the down arrow that is associated with the **To** field. The Combine To PDB window opens. Select the name of the PDB that you want to copy tables to (that is, select the name of the new PDB).

Note: The tables already in the "To PDB" will not be affected by the copy of additional tables to the "To PDB" as long as the names of the tables are different. Similarly, the PDB properties in the "To PDB" will not be affected by the copy of additional tables to the "To PDB." △

7 In the lower-left corner of the window, a list is displayed of all the tables in the "From PDB" that do not already exist in the "To PDB." Select each of the tables that you want to copy.

In the lower-right corner of the window, a list is displayed of each of the tables that you select.

8 When you finish selecting tables, select **Run**. A busy symbol is displayed in the SAS message area at the bottom of the window.

9 When the busy symbol disappears from the message area, select **Goback** to return to the Performance Database Operations window.

10 To return to the main menu, select **File**. A menu opens. From the menu, select **End**.

Reminder: The tables were copied, not moved. You might want to delete the tables from the "From PDB."

Reminder: You might want to duplicate and edit your set of batch jobs, so that one set of jobs uses one of the PDBs and the other set of jobs uses the other PDB.

Splitting One PDB into Two PDBs (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can split a PDB into two by following these steps:

1 Create a new PDB. For information about creating a PDB, see "Creating a PDB" on page 308. (You do not need to add any tables during creation when creating a PDB for this purpose.)

2 If the new PDB is the active PDB, activate some other PDB (for example, a demonstration PDB). For information about activating a PDB, see "Activating a PDB" on page 299.

3 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

4 On your list of known PDBs, select the "To PDB" (that is, select the new PDB).

Note: If there are already tables in the "To PDB," they will not be affected by the copy of additional tables to the "To PDB" as long as the table names are different. Similarly, the PDB properties in the "To PDB" will not be affected by the copy of additional tables to the "To PDB." △

5 Select **File**. A menu opens. From the menu, select **Combine PDBs**. The Combine PDB window opens.

6 Select the down arrow that is associated with the **From** field. The Combine From PDB window opens. Select the name of the PDB that you want to copy tables from (that is, select the name of the PDB that you want to split).

7 In the lower-left corner of the window, a list is displayed of all the tables in the "From PDB" that do not already exist in the "To PDB." Select each of the tables that you want to copy.

   In the lower-right corner of the window, a list is displayed of each of the tables that you select.

8 When you finish selecting tables, select **OK**.

   When SAS IT Resource Management finishes copying the selected table(s), the following message is displayed in the SAS message area (at the bottom of the window): "PDB <*name-of-"To-PDB"*> combined."

9 To return to the main window, select **Close**.

Reminder: The tables were copied, not moved. You might want to delete the tables from the "From PDB."

Reminder: You might want to duplicate and edit your set of batch jobs, so that one set of jobs uses one of the PDBs and the other set of jobs uses the other PDB.

Transporting a PDB outside the Local File System

An exported (also known as transport-formatted) PDB must be copied by using a bit-protecting cross-operating-environment mechanism such as binary FTP. (See your system administrator for more information about a binary FTP that is available on your system.) Also, the transport mechanism must not change the logical record length of the file that is being transported.

Note: Before the cross-operating-environment copying takes place, the PDB must be exported on the "from operating environment" (that is, the PDB in the format of the "from operating environment" must be copied to transport format). For more information about exporting, see "Exporting a PDB" on page 316. △

Note: After the cross-operating-environment copying takes place, the PDB must be imported on the "to operating environment" (that is, the PDB in transport format must be copied to a PDB in the format of the "to operating environment"). For more information about importing, see "Importing a PDB" on page 318. △

Verifying/Correcting the Active PDB's Data Dictionary

You can check the structural integrity of the active PDB's data dictionary. If there are errors, you might be able to correct them automatically. To do both of these operations, use the %CPSTART macro and the %CPDDUTL macro in batch mode. For more information about the %CPSTART macro, see the %CPSTART topic in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see the %CPDDUTL topic in the SAS IT Resource Management macro reference documentation , and

- to check the structural integrity, use the VERIFY DICTIONARY control statement (in batch mode).
- to correct the structural integrity, use the SYNCHRONIZE DICTIONARY control statement (in batch mode).

CAUTION:

If the SYNCHRONIZE DICTIONARY control statement does not succeed in correcting the problem, restore the PDB from the backup copy and rerun whatever is necessary to bring it up to date. △

Checking the Structure

To check the structure, run the %CPDDUTL macro with a file or with a SAS catalog entry. The file/entry should contain the VERIFY DICTIONARY control statement. For more information about this control statement, see the VERIFY DICTIONARY topic in the SAS IT Resource Management macro reference documentation.

Correcting the Structure

To correct the structure, run the %CPDDUTL macro with a file or with a SAS catalog entry. The file/entry should contain the SYNCHRONIZE DICTIONARY control statement. For more information about this control statement, see the SYNCHRONIZE DICTIONARY topic in the SAS IT Resource Management macro reference documentation.

Viewing the Name of Your Active PDB

Your active PDB is the PDB with which you are currently working.
You can view the name of your active PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and the PDB environments.

Viewing the Name of Your Active PDB (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display the name of your active PDB by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Manage PDBs/Switch PDB`**. The Performance Database Operations window opens. The name of your active PDB is displayed at the top of this window.
2. To return to the main menu, select **`File`**. A menu opens. From the menu, select **`End`**.

Viewing the Name of Your Active PDB (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display the name of your active PDB by following these steps:

1. From the main window's **`Administration`** tab, select **`Manage PDBs`**. The Manage PDBs window opens. The name of your active PDB is displayed at the top of this window.
2. To return to the main window, select **`Close`**.

Viewing Your Active PDB's File System and Profile

On z/OS, your active PDB is on the local file system, and no remote-server profile is associated with it.

On UNIX and Windows hosts without a SAS IT Resource Management client license, your active PDB is on the local file system, and no remote-server profile is associated with it.

On UNIX and Windows hosts with a SAS IT Resource Management client license, your active PDB can be on the local file system or on a remote file system. If it is on a

remote file system, a remote-server profile is associated with it. You can view the type of system (and name of the remote-server profile, if any)

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing Your Active PDB's File System and Profile (UNIX, Windows)

On the SAS IT Resource Management GUI for UNIX and Windows, you can view your active PDB's file system (and the name of the remote-server profile that is associated with it, if any) by following these steps:

1. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2. Select **Properties**. The Set Active PDB Options window opens. The current setting of File System is displayed on the Access tab. If the setting is **Remote**, then the **Remote Profile** field displays the name of the remote server profile that is associated with the PDB.
3. Select **OK**.
4. To return to the main window, select **Close**.

Viewing Your List of Known PDBs

Your list of known PDBs automatically contains

- the names of the demonstration PDBs on your local file system
- any PDB that you create interactively
- any PDB that you create in batch mode, if you allocate your SASUSER library in the batch job that does the creation
- any PDB that you activate in batch mode, if you allocate your SASUSER library in the batch job that does the activation.

You can also add or remove the names of PDBs that other people create. For more information about adding a PDB's name to your list of known PDBs, see "Adding a PDB to Your List of Known PDBs" on page 301. For more information about removing a PDB's name from your list of known PDBs, see "Removing a PDB from Your List of Known PDBs" on page 319.

You can display your list of known PDBs

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: The PDBs in your list are the ones that are known to you. They are in the list for selection convenience. Other PDBs might exist, but you cannot select them interactively until you add them to this list. △

Viewing Your List of Known PDBs (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display your list of known PDBs by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Manage PDBs/Switch PDBs**. The Performance Database Operations window opens.
2. Your list of known PDBs is displayed at the center of this window.
3. To return to the main menu, select **File**. A menu opens. Select **End**.

Viewing Your List of Known PDBs (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display your list of known PDBs by following these steps:

1. On the main window's **`Administration`** tab, select **`Manage PDBs`**. The Manage PDBs window opens.
2. Your list of known PDBs consists of the active PDB (which is displayed at the top of the window) and the PDBs that are in the **`Other PDBs`** list (which is displayed at the center of the window).
3. To return to the main window, select **`Close`**.

Viewing/Editing Your Active PDB's Access Mode

There are several factors that jointly affect the access mode that you can have to a PDB.

1. Operating system permissions: If you want to have read access to a PDB, you must make arrangements with the system or security administrator where the PDB is located for read permission to that PDB and its objects. Similarly, if you want to have update (read/write) access to a PDB, you must make arrangements with the system or security administrator where the PDB is located for update (read/write) permission to that PDB and its objects.

   Note: If you use SAS/CONNECT (or SAS/SHARE), your own user ID is required to access the SAS/CONNECT (or SAS/SHARE) server. You get permission to access the PDB by invoking your user ID for the SAS/CONNECT (or SAS/SHARE) server. △

2. SAS IT Resource Management software: The SAS IT Resource Management server software permits you to have read access or update (read/write) access to PDBs on its local file system. The SAS IT Resource Management client software permits you to have read access but not update (read/write) access to PDBs on its local file system and to remote PDBs.

   Note: If the client software is installed on the same host where the server software is installed, the access rights are additive. You can have read/write access to a local or remote PDB. △

   If you need update (read/write) access to a PDB and it is not available because another user has read/write access to that PDB, wait several minutes and try again. It is recommended that you request update (read/write) permission only during the time that you need it.

You can display and edit your access to a PDB that is currently your active PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

You can display and set your access to a PDB that is not currently your active PDB when you activate that PDB. For more information about activating a PDB, see "Activating a PDB" on page 299.

Viewing/Editing Your Active PDB's Access Mode (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display and/or edit the access mode of your active PDB by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens. The current setting of access mode is displayed under **`PDB Access Mode`**.

2 Under **PDB Access Mode**, select **Readonly** or **Write**.

3 To return to the main menu, select **OK**.

Viewing/Editing Your Active PDB's Access Mode (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit the access mode of your active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

2 Select **Properties**. The Set Active PDB Options window opens. The current setting of Access Mode is displayed on the **Access** tab.

3 On the **Access** tab, under Access Mode select **Readonly** or **Write**. Then select **OK**.

Note: You cannot select WRITE access for a remote PDB.) △

4 To return to the main window, select **Close**.

Working with Whole PDBs: Properties

Viewing/Editing Your Active PDB's Archive Information

If one or more tables have archiving turned on (in their table properties), then there is an archive.

An archive is a set of archive libraries. Each archive library is the result of one run of the processing step. Each archive library contains the incoming data to the detail level of the tables for which archiving is turned on.

The archiving information describes what the archive is to be like. You can display and edit your active PDB's archiving information

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: 1. Except for the archive location, all of the archiving information is inherited from the active site library when the PDB is created. The default location for the archive is the PDB. That is, by default the archive libraries are "under" the PDB (UNIX, Windows) or have the same high-level-qualifiers as the PDB (z/OS).

2. For some supplied collector support (for example, the collector support for Web data), the detail level is bypassed. In that case, there is no incoming data to the detail level. Thus, archiving is not relevant. △

For more information about archiving, see the topic "Archiving Overview" in "Appendix 2: Archiving Data" in the macro reference documentation.

If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs. (For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.)

Viewing/Editing Your Active PDB's Archive Information (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display and/or edit your active PDB's archiving information by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2. In the **Archive Type** field, select **DASD** or **Tape**.
3. In the **Archive Unit** field, type the value that you want to use for each archive library's UNIT= parameter.
4. In the **Archive Root** field, overwrite the location (that is, name) of the archive if you do not want to use the default value.
5. In the **Archive Library Allocation Parameter String** field, type any additional specifications. For example, if the archive is on tape, you might want to specify *LABEL=(1,SL,,,RETPD=365)*.
6. To return to the main menu, select **OK**.

Viewing/Editing Your Active PDB's Archive Information (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's archiving information by following these steps:

1. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens. The PDB that is active is listed at the top of the window.
2. Select **Properties**. The Set Active PDB Options window opens.
3. Select the **Archive** tab. The fields on the tab display the archiving information.
4. To change the value in the **Archive Type** field, select **Disk** or **Tape**.
5. To change the archive device, in the **Archive Device** field edit the name of the device. For example, type *c:* for Windows or */usr* for UNIX.
6. To change the archive root, in the **Archive Root** field edit the name of the archive. For example, type *\temp\newpdb* for Windows or */temp/newpdb* for UNIX. The value of archive path will be the concatenation of the value of archive device and the value of archive root. So in this case, the archive path will be c:\temp\newpdb on Windows and /usr/temp/newpdb on UNIX.
7. SAS IT Resource Management uses the SAS LIBNAME statement when it creates an archive library. For example, if *&ARCPARM* represents the value of the **Archive Library Allocation Parameter String** field, then an archive library is allocated by using the SAS LIBNAME statement as follows:

   ```
   LIBNAME libref 'library-location' &ARCPARM ;
   ```

   (The value of *library-location* is the archive path, concatenated with */Znnnnnnn*, where *nnnnnnn* is 0000001, 0000002, 0000003, etc.)

   To change the string, edit the **Archive Library Allocation Parameter String** field.
8. Select **OK** to return to the Manage PDBs window.
9. Select **Close** to return to the main window.

Viewing/Editing Your Active PDB's DST Definition

The Daylight Saving Time (DST) definition describes the Daylight Saving Time period. Some collectors log data by using DST timestamps during the DST period. SAS IT Resource Management can change those timestamps to standard time by using the DST definition. (The only supplied collector support that uses the DST definition is for

SunNet Manager data, and the Generic Collector Facility does not access the DST definition.)

There are two ways to express the definition: as a concatenation of ranges (one each for next year, this year, last year, and the year before) and as a formula that takes the system's year into account. Both forms of the definition work correctly. The advantage of the formula is that it does not require periodic updates.

You can display and edit your active PDB's value for the DST definition

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: 1. The default value of the DST definition is inherited from the active site library when the PDB is created.

2. If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs. (For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.) △

Viewing/Editing Your Active PDB's DST Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's DST definition by following these steps:

1. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens. The PDB that is active is listed at the top of the window.
2. Select **Properties**. The Set Active PDB Options window opens.
3. Select the **Time Zone** tab. One of the fields on the tab displays the DST definition.
4. To make changes in the DST definition, edit the field.
5. Select **OK** to return to the Manage PDBs window.
6. Select **Close** to return to the main window.

Viewing/Editing Your Active PDB's GMT Deviation

The Greenwich Mean Time (GMT) deviation describes the number of hours' difference between the time where the data was logged and the time in Greenwich, UK (Coordinated Universal Time). Some collectors log data by using GMT timestamps (the time in Greenwich, not the local time). SAS IT Resource Management can change these timestamps to local time by using the GMT deviation. (Note: The only supplied collector support that uses the GMT deviation is for SunNet Manager data, and the Generic Collector Facility does not access the GMT deviation.)

You can display and edit your active PDB's value for the GMT deviation

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note:

- The default value of the DST definition is inherited from the active site library when the PDB is created.

- HP OpenView Performance Agent records data in Greenwich Mean Time. For this data, however, SAS IT Resource Management does not use the constant above. Instead, it uses the constant that is provided in the data log (in a header).
- If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs. (For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.)

△

Viewing/Editing Your Active PDB's GMT Deviation (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's GMT deviation by following these steps:

1. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens. The PDB that is active is listed at the top of the window.
2. Select **Properties**. The Set Active PDB Options window opens.
3. Select the **Time Zone** tab. One of the fields on the tab displays the GMT deviation.
4. To make changes in the GMT deviation, edit the field.
5. Select **OK** to return to the Manage PDBs window.
6. Select **Close** to return to the main window.

Viewing/Editing Your Active PDB's Holiday Shift Code

The holiday shift code that your active PDB uses is not in the PDB. It is in the active site library. You can see and change the shift code that is used for holidays by

- changing the holiday shift code in the active site library. For more information about the holiday shift code, see "Viewing/Editing Your Active Site Library's Shift Code for Holidays and Its Description" on page 288.

The holiday shift code *description* that your active PDB uses is in the PDB.

You can display and edit your active PDB's value for the *description* of the holiday shift code

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs. (For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.) △

Viewing/Editing Your Active PDB's Holiday Shift Code Description (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's holiday shift code and description by following these steps:

1 On the main window's **Administration** tab, select **Define Work Shifts**. The Define Work Shifts window opens.

2 At the question **Define at which level?**, select **Site Defaults**. (This selection is not necessary if you want to change the holiday shift code or holiday list. However, this selection is necessary if you want to adjust the holiday shift code's description.)

3 Select **Shift Descriptions**. The Modify Shift Descriptions window opens.

4 For each shift code there is a description. If you did not change the value of the holiday shift code, then the description next to the value is still current. If you did change the value of the holiday shift code, make the appropriate adjustments so that the values of the descriptions are now appropriate. For example, if shift code "4" is for holidays and is not used in the work schedule, you might want to change the description for the old value to blank and the description for "4" to "HOLIDAYS". Or if shift code "2" is now for holidays and is also used for weekends in the work schedule, you might want to change the description for the old value to blank and the description for "2" to "WEEKENDS AND HOLIDAYS".

5 To return to the Define Work Shifts window, select **OK** if you made changes and want to use them; otherwise, select **Cancel**.

6 To return to the main window, select **OK** if you made changes and want to use them; otherwise, select **Cancel**.

Viewing/Editing Your Active PDB's MXG Views Property

In SAS IT Resource Management software for z/OS, MXG views of a PDB enable you to run report programs that are written in SAS that use the MXG names for tables and variables. This can be useful if you have old MXG report programs that you want to run against one or more SAS IT Resource Management PDBs. Note that the MXG views are in addition to the SAS IT Resource Management views. If you request no MXG views, you still have the SAS IT Resource Management views for use with the SAS IT Resource Management reporting facilities.

You can display and edit your active PDB's MXG Views property

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.

In addition, you can

- check for the existence, now, of MXG views
- force the existence, now, of MXG views.

Note: The setting of the MXG Views property is inherited from the active site library when the PDB is created. You can change the setting. △

If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs.

For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.

Viewing/Editing Your Active PDB's MXG Views Property (z/OS)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's MXG Views property by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.
2. The **`Provide MXG PDB Views`** field indicates the current setting of the option (Yes or No).
3. To change the setting, select the other value.
4. To return to the main menu, select **`OK`**.

The setting takes effect when views are checked. (Views are checked, and rebuilt if necessary, by the process step, by the reduce step, and by the %CPDDUTL control statement BUILD VIEWS.)

Checking for the Existence of MXG Views (Batch Mode)

To find out whether the MXG views are not only requested but also exist now, submit this SAS program:

```
LIBNAME PDB LIST ;
   RUN ;
```

For more information about submitting a SAS program, see "Using SAS Windows" on page 24 and "Working with the Interface for Batch Mode" on page 18. If the PDB libref is assigned, then submit this SAS program:

```
PROC DATASETS DD=PDB ;
   RUN ;
```

In the list that this program generates, if you see any views whose names start with the characters *TYPE*, then MXG views exist.

Note: You can run the programs in a batch job or submit the programs from the SAS Program Editor window.

- If you run the program in a batch job: in the step's JCL change the location to which the ddname PDB points
  - from: the MXG PDB
  - to: the DETAIL library of the appropriate SAS IT Resource Management PDB.
- If you submit the program from the Program Editor window: first start the SAS IT Resource Management GUI for z/OS and activate the appropriate SAS IT Resource Management PDB. The activation will point the libref named PDB to the DETAIL library of the active PDB. Then submit the program through the SAS Program Editor window.

△

Forcing the Existence of MXG Views (Batch Mode)

If MXG Views do not exist now and if you want to run old report programs before the next process or reduce step, you can force the MXG views to be built now by following these steps:

1. Start SAS IT Resource Management with the access mode that you want to use for the PDB. For more information about starting SAS IT Resource Management, see "Starting the z/OS GUI" on page 19.

2 To set the value of the global macro variable CPXPDBNM, submit one of the following two SAS programs.

(For more information about submitting a SAS program, see "Using SAS Windows" on page 24 and "Working with the Interface for Batch Mode" on page 18.

□ To change the status of MXG views to not requested:

```
%let cpxpdbnm = ;
   run;
```

□ To change the status of MXG views to requested:

```
%let cpxpdbnm = PDB ;
   run;
```

3 Create a file (of any name). In the file, put this %CPDDUTL control statement:

```
BUILD VIEWS NAME=_ALL_ ;
```

4 To apply the control statement to the active PDB, submit this program:

```
%CPDDUTL(filename='name-of-the-file-that-you-just-created');
   RUN;
```

For more information about the %CPDDUTL macro, see the topic %CPDDUTL in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL control statement BUILD VIEWS, see the topic BUILD VIEWS in the SAS IT Resource Management macro reference documentation.

Viewing/Editing Your Active PDB's Shift Descriptions

The work shift schedule is defined by using work shift codes. (For more information about the work shift schedule, see "Viewing/Editing Your Active PDB's Work Shift Schedule" on page 334.) The descriptions of the codes assign meaning to the codes.

You can display and edit the descriptions of your active PDB's work shift codes

□ by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.

□ by using the SAS IT Resource Management GUI for z/OS.

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: The descriptions are inherited from the active site library when the PDB is created. You can change the descriptions if you prefer others. △

If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs.

For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.

Viewing/Editing Your Active PDB's Shift Descriptions (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display and/or edit your active PDB's descriptions of the work shift codes by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Define Work Shifts**. The Define Work Shifts window opens.
2. In the **Define at Which Level?** field, select **Active PDB** if it is not already selected.
3. The descriptions of the work shift codes display in the frame in the lower-left corner of the window.
4. If the codes and descriptions do not provide the distinctions that you want to make among shifts, in the **Shift Descriptions** frame add or remove codes and type over the existing descriptions with the descriptions that you prefer. The changes will take effect at the next process step. (The changes affect only incoming data; they do not affect the data that is already in the PDB.)
5. To return to the main menu, select **OK**.

Viewing/Editing Your Active PDB's Shift Descriptions (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's descriptions of the work shift codes by following these steps:

1. On the main window's **Administration** tab, select **Define Work Shifts**. The Define Work Shifts window opens.
2. In the **Define at Which Level?** field, select **Active PDB** if it is not already selected.
3. To display and/or edit the descriptions of the work shift codes, select **Shift Descriptions**.
4. If the codes and descriptions do not provide the distinctions that you want to make among shifts, add or remove codes and type over the existing descriptions with the descriptions that you prefer, and then select **OK**. The changes will take effect at the next process step. (The changes affect only incoming data; they do not affect the data that is already in the PDB.
5. To return to the main window, select **OK**. A dialog box opens. Select **OK**.

Viewing/Editing Your Active PDB's Start of Week

Start of week defines the beginning of a week for the week level. (Some sites consider a week to be Sunday through Saturday; other sites consider a week to be Monday through Sunday.) The default start of week is Sunday.

You can display and edit your active PDB's Start of Week setting

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: The default value of Start of Week is inherited from the active site library when the PDB is created. △

If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs.

For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.

Viewing/Editing Your Active PDB's Start of Week (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display and/or edit your active PDB's Start of Week by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens. The current setting of Start of Week is displayed at the center of the window. (A value of blank is interpreted as Sunday.)
2. To use a different day, select **Change-Start-of-Week**. A menu opens. Select the day that you want to use.
3. To return to the main menu, select **OK**.

Viewing/Editing Your Active PDB's Start of Week (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's Start of Week by following these steps:

1. On the main window's **Administration** tab, select **Site Options**. The Set Site-wide Options window opens.
2. Select the **Start-of-Week** tab. The current setting of Start of Week is displayed on that tab.
3. To change the setting, select the down arrow to the right of the **Start of Week** field. A menu opens. From the menu, select the day that you want to use.
4. Select **OK** to return to the main window.

Viewing/Editing Your Active PDB's Work Shift Schedule

The work shift schedule is a matrix of 24 hours by 7 days. Unless a non-blank value for the shift code is already included in the data that is processed into the PDB, the shift code is set in the data to the appropriate value from the work shift schedule, based on the datetime stamp of the data. (For data that is logged on active holidays, the work shift schedule is not used; instead the shift code for holidays is assigned. For more information about the shift code for holidays, see "Viewing/Editing Your Active Site Library's Shift Code for Holidays and Its Description" on page 288.) Work shift schedule is composed of seven site options (the schedule for Sunday, the schedule for Monday, ..., and the schedule for Saturday).

You can display and edit your active PDB's work shift schedule

- by using the %CPPDBOPT macro in batch mode. For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: The work shift schedule in the PDB is inherited from the active site library when the PDB is created. You can change the PDB's schedule if you prefer another. △

If you decide to edit your active PDB's properties, you might later want to copy the new values to your active site library. From there, the values will be copied automatically to new PDBs and can be copied to existing PDBs. This takes less time than editing the values in each of your PDBs.

(For more information about copying the values from the PDB to the site library, see "Copying a PDB's Options to a Site Library" on page 273. For more information about copying of values from the site library to new or existing PDBs, see "Copying a Site Library's Options to a PDB" on page 274.)

Viewing/Editing Your Active PDB's Work Shift Schedule (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can display and/or edit your active PDB's work shift schedule by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Define Work Shifts`**. The Define Work Shifts window opens.
2. In the **`Define at Which Level?`** field, select **`Active PDB`** if it is not already selected.
3. The work shift schedule displays in the frame in the upper-left corner of the window. To see other hours, use the scroll bar that is on the right of the frame. To see other days, select an arrow at the bottom of the frame.

   Note that the hours are displayed in terms of the 24-hour clock (0:00 to 23:00). Each hour, *n*:00, represents the range *n*:00:00 to *n*:59:59. For example, 08:00 represents 08:00:00 to 08:59:59.

   Remember to check (and change if necessary) the shift codes for every day of the week.
4. You can type over the current shift code(s) with the shift code(s) that you want to use.
5. To return to the main menu, select **`OK`**. The changes will take effect at the next process step. (The changes affect only incoming data; they do not affect the existing data in the PDB.)

Viewing/Editing Your Active PDB's Work Shift Schedule (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit your active PDB's work shift schedule by following these steps:

1. On the main window's **`Administration`** tab, select **`Define Work Shifts`**. The Define Work Shifts window opens.
2. In the **`Define at Which Level?`** field, select **`Active PDB`**.
3. The work shift schedule displays in the frame on the left side of the window. To see other hours, use the scroll bar on the right of the frame. To see other days, select an arrow at the bottom of the frame.

   Note that the hours are displayed in terms of the 24-hour clock (0:00 to 23:00). Each hour, *n*:00, represents the range *n*:00:00 to *n*59:59. For example, 08:00 represents 08:00:00 to 08:59:59.

   Remember to check (and change if necessary) the shift codes for every day of the week.
4. To edit the work shift schedule, you can type over the schedule with the value(s) that you want to use.
5. To return to the main window, select **`OK`**. A dialog box opens. Select **`OK`**. The changes will take effect at the next process step. (The changes affect only incoming data; they do not affect the existing data in the PDB.)

Viewing/Editing Your Active PDB's Holiday List

The list of holidays that your active PDB uses is not in the PDB. It is in the active site library. You can see and change the holiday list by

- changing the holiday list in the active site library. For more information about the holiday list, see "Viewing/Editing Your Active Site Library's List of Holidays" on page 286.

CHAPTER

16

Administration: Working with Levels

Working with Levels: Main Topics

Overview of Levels

Each table in a PDB has data in one or more levels. Each level corresponds to the extent of summarization of the data. Typically,

- in the detail level, observations are not summarized
- in the day level, observations have summary statistics at least by day
- in the week level, observations have summary statistics at least by week
- in the month level, observations have summary statistics at least by month
- in the year level, observations have summary statistics at least by year.

For example, suppose the detail level of a table contains observations about a certain type of hardware and the observations are logged at 10-minute intervals. And suppose that the week level of a table contains observations that are grouped by machine, week, and hour.

- At detail level, a day is represented by 144 observations (6 per hour multiplied by 24 hours) for each piece of hardware for which data is collected. On each observation are the measurements collected over 10 minutes about that piece of hardware.
- At week level, a week is represented by 24 observations (1 per hour) for each piece of hardware for which data is collected. On each observation are statistics (for example, count, sum, maximum, minimum, etc.) that are based on the 42 detail-level observations (6 per hour multiplied by 7 days) for that machine and hour during the week.

  The detail-level observations on which the week-level observations are based might or might not be in the detail level. (They might have aged out of the detail level.)

The following topics about levels are listed in alphabetical order:

Specifying a Variable's Presence at a Level in a Table

Variable properties control the values of regular variables, derived variables, and formula variables at each level.

- Regular variables are variables whose detail-level values are in the logged data. For more information about how to specify a regular variable at detail level (and, at day, week, month, or year level, how to specify which statistics, if any, are to be calculated for the variable), see "Creating/Editing/Viewing a Regular Variable" on page 394.

  The values of regular variables (and their statistics) can be kept in the PDB.
- Derived variables are variables whose detail-level values are built from the values of regular variables, other derived variables, and/or constants. For more information about how to specify a derived variable at detail level (and, at day, week, month, or year level, how to specify which statistics, if any, are to be calculated for the variable), see "Creating/Editing/Viewing a Derived Variable" on page 381.

  The values of derived variables (and their statistics) can be kept in the PDB.
- Formula variables are variables whose detail-, day-, week-, month-, and year-level values are built from the values of regular variables, derived variables, other formula variables, constants, and/or the _LEVEL_ variable. For more information about how to specify a formula variable at detail, day, week, month, or year level, see "Creating/Editing/Viewing a Formula Variable" on page 388.

  The values of formula variables are calculated when requested, and are not kept in the PDB.

Table properties control which regular or derived variables are used as BY or CLASS variables, ID variables, or INDEX variables at each level. (Formula variables are not available for these purposes.)

- The BY and CLASS variables provide information about how to group the data in the tables. For more information about how to specify one or more variables as BY

variables at the detail level or CLASS variables at the day, week, month, or year levels, see "Specifying BY or CLASS Variables for a Level in a Table" on page 339.

- The ID variables provide alternate ways of identifying the data in the CLASS variables. For more information about how to specify one or more variables as ID variables at the day, week, month, or year level, see "Specifying ID Variables for a Level in a Table" on page 342.
- The INDEX variables often provide faster access to data during reporting. For more information about how to specify one or more variables as INDEX variables at the detail, day, week, month, or year levels, see "Specifying Index Variables for a Level in a Table" on page 344.

Table properties also control whether and how long the variable's data stays in each level of the PDB.

- The table's Kept status controls *whether* the table is to be used or ignored. For more information about how to specify the table's Kept status, see "Viewing/ Editing a Table" on page 364.
- The age limits at detail, day, week, month, and year provide information about *how long* the data for that table's variables is in the PDB (if the Kept status is *Yes*). For more information about how to specify the age limits, see "Specifying the Age Limit for a Level in a Table" on page 346.

Specifying BY or CLASS Variables for a Level in a Table

A table has BY and CLASS variables that are used to group observations within each level.

The BY variables list groups observations for the detail level of the table. The kept status of the variables in the BY variables list must be *Yes*, and the DATETIME variable must be on the BY variables list. To the right of DATETIME on the list, use only HOUR and/or SHIFT. To the left of DATETIME on the list, you can use any variables as long as they are of type Character, are not formula variables, and satisfy the restrictions in "Restrictions on Modifications of BY Variables Lists" on page 348. The purpose of the BY variables list is to sort the observations in the detail level to match the sort order of the observations that are incoming during the processing step and to detect observations that have duplicate values in the BY variables. Under normal circumstances, you do not need to modify the BY variables list, but you can modify the BY variables list if you need to. For example, if you add a variable in a process exit, you can add that variable to the BY variables list.

For some data sources, there are restrictions on the variables and the order of the variables in BY variables lists. For more information about the restrictions, see "Restrictions on Modifications of BY Variables Lists" on page 348.

The CLASS variables lists group observations for the summary levels of the table so that the same values of all CLASS variables are together for the summary step. For each of the summary levels of a table (i.e., day, week, month, and year), there is a list of CLASS variables that groups observations for that level. The lists also ensure that data is in the proper sort order for reporting on the summary levels. By default, the CLASS variable lists contain the same variables in the same order as the BY variables list. However, the CLASS variables list for a summary level can be different from the BY variables list and can be different from all other CLASS variables lists. Like the BY variables list, the CLASS variables list must contain the DATETIME variable.

If you want character string metrics like MACHINE and JOB to be carried into summary levels for reporting, add them to the CLASS variables lists immediately to the left of the DATETIME variable (or add them to the ID variables lists if equivalent variables are already in the CLASS variables list). Use of a string variable as a CLASS or ID variable requires that the Kept status of the string variable be *Yes*. A string

variable that is not on a level's CLASS variables list or ID variables list is dropped from the summary level, regardless of its Kept status.

Note: These lists apply to table definitions and are used to process and summarize data. This BY variables list is independent of the BY variables that you can select for a report definition. △

You can specify one or more grouping variables that are to be used for the observations in a particular level of a table

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE control statement and the UPDATE TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic "SET TABLE" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL "UPDATE TABLE" statement, see the topic "UPDATE TABLE" in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Specifying BY or CLASS Variables for a Level in a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can specify the BY or CLASS variables list for a particular level of a table by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2. In the **PDB Access Mode** field, select **Write** if it is not already selected.
3. Select **OK** to return to the main menu.
4. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens. The window displays a list of the tables in the active PDB.
5. Select the name of the table for which you want to change a level's BY or CLASS variables list.
6. Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.
7. In the **Constructional Variables to Display** field, select **By/class** if it is not already selected.
8. In the **Level** column, select the button with the name of the level that you want to change. If you select **Detail**, the Edit By Variable Order window opens. If you select the **Day**, **Week**, **Month**, or **Year** button, the Edit Class Variable Order window opens.

   Both windows have the same format. In the upper part of the window is an ordered list of the active variables on the table's BY or CLASS variables list for the level. In the lower part of the window is a list of the table's other variables that are eligible to be on the list of BY or CLASS variables for the level.
9. To move a variable from the upper list to the lower list, select the variable. To move a variable from the lower list to the upper list, select the variable.

   Note: You can undo all the moves by selecting **Cancel**. △

10 When you have modified the upper list so that it has the variables that you want in the order that you want, select **OK**. SAS IT Resource Management returns you to the Edit Table window.

11 Select **OK**. SAS IT Resource Management returns you to the Dictionary Tables window.

12 Select **File**. A menu opens. Select **End**. The Confirm Table View Rebuild window opens.

13 Select **OK**. SAS IT Resource Management returns you to the main menu.

Note: The above steps update a BY or CLASS variables list in the table definition. If you want additional information about the table definition, see "Overview of Tables" on page 353. △

Specifying BY or CLASS Variables for a Level in a Table (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can specify the BY or CLASS variables list for a particular level of a table by following these steps:

1 From the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

2 Select **Properties**. The Set Active PDB Options window opens.

3 From the **Access** tab, select **Write** if it is not already selected.

4 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.

5 Select **Close**. SAS IT Resource Management returns you to the main window.

6 From the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window displays a list of the tables in the active PDB.

7 Select the name of the table for which you want to change a level's BY or CLASS variables list.

8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

9 Select the **Structure Variables** tab.

10 Select the **By/Class** option if it is not already selected.

11 Select the button with the name of the level that you want to change.

   If you select **Detail**, the Edit BY Variable Order window opens. If you select the **Day**, **Week**, **Month**, or **Year** button, the Edit CLASS Variable Order window opens.

   Both windows have the same format. In the upper part of the window is an ordered list of the active variables on the table's BY or CLASS variables list for the level. In the lower part of the window is a list of the table's other variables that are eligible to be on the list of BY or CLASS variables for the level.

12 To move a variable from the upper list to the lower list, select the variable and then select the down arrow. To move a variable from the lower list to the upper list, select the variable and then select the up arrow.

   Note: You can undo all the moves by selecting **Cancel**. △

13 When you have modified the upper list so that it has the variables that you want in the order that you want, select **OK** to return to the Table Properties window.

14 Select **OK** to return to the Manage Tables window.

15 Select **Close**. A confirmation dialog box opens.

16 Select **OK** to return to the main window.

Note: The preceding steps update a BY and CLASS variables list in the table definition. If you want additional information about the table definition, see "Overview of Tables" on page 353. △

Note: For additional information about frequently used variables, from the **Administration** tab of the SAS IT Resource Management GUI for UNIX and Windows, select **OnlineHelp ▶ Other ITRM Documentation ▶ Server Setup Guide**

Click on **Section 2, Task 2: Customize and Verify Your Test PDB**. Then scroll down to item 2 of the Prerequisites topic and click **Supplied**. On the next page, click on **UNIX** or **Windows**. The next page contains a topic entitled "More about frequently used variables." △

Specifying ID Variables for a Level in a Table

ID variables provide alternate ways of identifying the data in the CLASS variables. ID variables are non-changing or slowly changing variables whose values you want to have available for reporting at one or more summary levels.

You can specify one or more ID variables that are to be used for the observations in a particular level of a table

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE control statement and the UPDATE TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic "SET TABLE" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL UPDATE TABLE statement, see the topic "UPDATE TABLE" in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Specifying ID Variables for a Level in a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can specify the ID variables list for a particular level of a table by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2. In the **PDB Access Mode** field, select **Write** if it is not already selected.
3. Select **OK** to return to the main menu.
4. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens. The window displays a list of the tables in the active PDB.
5. Select the name of the table for which you want to change a level's ID variables list.
6. Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.
7. In the **Constructional Variables to Display** field, select **ID**.
8. In the **Level** column, select the button with the name of the level that you want to change. The Edit ID Variable List window opens.

In the upper part of the window is a list of the active variables on the table's ID variables list for the level. In the lower part of the window is a list of the table's other variables that are eligible to be on the list of ID variables for the level.

9 To move a variable from the upper list to the lower list, select the variable. To move a variable from the lower list to the upper list, select the variable.

Note: You can undo all the moves by selecting **Cancel**. △

10 When you have modified the upper list so that it has the variables that you want, select **OK** to return to the Edit Table window.

11 Select **OK** to return to the Dictionary Tables window.

12 Select **File**. A menu opens. Select **End**. The Confirm Table View Rebuild window opens.

13 Select **OK** to return to the main menu.

Note: The preceding steps update an ID variables list in the table definition. If you want additional information about the table definition, see "Overview of Tables" on page 353. △

Specifying ID Variables for a Level in a Table (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can specify the ID variables list for a particular level of a table by following these steps:

1 From the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

2 Select **Properties**. The Set Active PDB Options window opens.

3 From the **Access** tab, select **Write** if it is not already selected.

4 Select **OK** to return to the Manage PDBs window.

5 Select **Close** to return to the main window.

6 From the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window displays a list of the tables in the active PDB.

7 Select the name of the table for which you want to change a level's ID variables list.

8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

9 Select the **Structure Variables** tab.

10 Select the **ID** option.

11 Select the button with the name of the level that you want to change. The Edit ID Variable window opens.

In the upper part of the window is a list of the active variables, if any, on the table's ID variables list for the level. In the lower part of the window is a list of the table's other variables, if any, that are eligible to be on the list of ID variables for the level.

12 To move a variable from the upper list to the lower list, select the variable and then select the down arrow. To move a variable from the lower list to the upper list, select the variable and then select the up arrow.

Note: You can undo all the moves by selecting **Cancel**. △

13 When you have modified the upper list so that it has the variables that you want in the order that you want, select **OK** to return to the Table Properties window.

14 Select **OK**. SAS IT Resource Management returns you to the Manage Tables window.

15 Select **Close**. A confirmation dialog box opens.

16 Select **OK** to return to the main window.

Note: The preceding steps update an ID variables list in the table definition. If you want additional information about the table definition, see "Overview of Tables" on page 353. △

Specifying Index Variables for a Level in a Table

Index variables are variables whose values you want to index at one or more levels. The Index variables provide faster access to data during reporting. Each index variable causes a simple index (single variable index) to be created.

You can specify one or more index variables that are to be used for the observations in a particular level of a table

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE control statement and the UPDATE TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic "SET TABLE" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL UPDATE TABLE statement, see the topic "UPDATE TABLE" in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Specifying Index Variables for a Level in a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can specify the index variables list for a particular level of a table by following these steps:

1 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.

2 In the **PDB Access Mode** field, select **Write** if it is not already selected.

3 Select **OK** to return to the main menu.

4 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens. The window displays a list of the tables in the active PDB.

5 Select the name of the table for which you want to change a level's Index variables list.

6 Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.

7 In the **Constructional Variables to Display** field, select **Index**.

8 In the Level column, select the level that you want to change. The Edit Index Variable List window opens.

  In the upper part of the window is a list of the active variables on the table's Index variables list for the level. In the lower part of the window is a list of the table's other variables that are eligible to be on the list of Index variables for the level.

9 To move a variable from the upper list to the lower list, select the variable. To move a variable from the lower list to the upper list, select the variable.

Note: You can undo all the moves by selecting **Cancel**. △

10 When you have modified the upper list so that it has the variables that you want, select **OK** to return to the Edit Table window.

11 Select **OK** to return to the Dictionary Tables window.

12 Select **File**. A menu opens. Select **End**. The Confirm Table View Rebuild window opens.

13 Select **OK** to return to the main menu.

Note: The preceding steps update an index variables list in the table definition. If you want additional information about the table definition, see"Overview of Tables" on page 353. △

Specifying Index Variables for a Level in a Table (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can specify the index variables list for a particular level of a table by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

2 Select **Properties**. The Set Active PDB Options window opens.

3 On the **Access**s tab, select **WRITE** if it is not already selected.

4 Select **OK** to return to the Manage PDBs window.

5 Select **Close** to return to the main window.

6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window displays a list of the tables in the active PDB.

7 Select the name of the table for which you want to change a level's Index variables list.

8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

9 Select the **Structure Variables** tab.

10 Select the **Index** option.

11 Select the button with the name of the level that you want to change. The Edit Index Variable window opens.

In the upper part of the window is a list of the active variables, if any, on the table's Index variables list for the level. In the lower part of the window is a list of the table's other variables, if any, that are eligible to be on the list of Index variables for the level.

12 To move a variable from the upper list to the lower list, select the variable and then select the down arrow. To move a variable from the lower list to the upper list, select the variable and then select the up arrow.

Note: You can undo all the moves by selecting **Cancel**. △

13 When you have modified the upper list so that it has the variables that you want in the order that you want, select **OK** to return to the Table Properties window.

14 Select **OK** to return to the Manage Tables window.

15 Select **Close**. A confirmation dialog box opens.

16 Select **OK** to return to the main window.

Note: The preceding steps update an index variables list in the table definition. If you want additional information about the table definition, see"Overview of Tables" on page 353. △

Specifying Statistics at a Level in a Table

Statistics are properties of a variable. See "Specifying a Variable's Presence at a Level in a Table" on page 338.

Specifying the Age Limit for a Level in a Table

By default, SAS IT Resource Management keeps 10 days of data at the detail level, but the age limit in your tables can be different. If you want your reports on the detail level to be able to cover a different number of days (for example, 14 days), overwrite the current value with the new value (in this case, 14).

Note: If you do not want to generate reports on detail-level data in this table, but you do want to generate reports on other levels, you can set the age limit for detail level 0. △

The process step deletes existing data in the detail level that is beyond the age limit for the detail level, but the process step keeps incoming data (and adds it to existing data in the detail level) regardless of the dates in the incoming data. When the age limit at the detail level is 0, the process step behaves just as it always does: it deletes existing detail-level data whose age is greater than the age limit (thus, in this case, it deletes all of the existing detail-level data), and it keeps the incoming data in the detail level.

The reduce step ages out data in summary levels that are beyond the age limits for those levels and uses the existing data in the detail level to update the summary levels. If the age limit at the detail level is 0, the reduce step also deletes the existing data in the detail level. If the age limit is 0, the existing data in the detail level would be cleared anyway (at the time of the next processing step) because the data by now is existing data and exceeds the age limit for the detail level. The purpose of this exception (the reduce step touching the detail level) is to remove the existing data in the detail level many hours sooner. This is useful when there is a tremendous volume of detail-level data, and the data can be safely discarded after the day, week, month, and year levels have been updated. The discard frees extra disk space between production jobs when the data will not be used again.

There is one caution: if you have a failure (for instance, a power outage) during the reduce step, and any age limits at detail level are zero, and any age limits at non-detail levels are nonzero, make sure that you resubmit the reduce step before you proceed; otherwise, the next day's production job will remove the detail data before it can be summarized in the other levels.

By default, the PDB keeps summary-level data as follows:

- Day: 45 days
- Week: 15 weeks
- Month: 18 months
- Year: 5 years.

If you want your reports on these levels to be able to cover a different number of days, weeks, months, or years, or if you want to keep less data in order to save disk space and shorten processing time, then overwrite the current value with a new value.

You can specify the age limit that is to be used for the observations in a particular level of a table in the active PDB

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE control statement and the CREATE TABLE or UPDATE TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic "SET TABLE" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL CREATE TABLE control statement, see the topic "CREATE TABLE" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL UPDATE TABLE statement, see the topic "UPDATE TABLE" in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Reference dates for aging out of data:

- At the detail level, if age limit is not zero, the %CMPROCES, %CPPROCES, %CSPROCES, %CWPROCES macros use, as reference date, the maximum of these two dates: the maximum date in the incoming data to the detail level, and the maximum date of the existing data in the detail level. At the detail level, if age limit is equal to zero, the %CPREDUCE macro uses, as reference date, the maximum of these two dates: the maximum date in the incoming data to the detail level, and the maximum date of the existing data in the detail level.
- At a summary level, the %CPREDUCE macro uses, as reference date, the maximum of these two dates: the maximum date of the not-yet-summarized data in the detail level, and the maximum date of the existing data in that summary level.

△

Specifying the Age Limit for a Level in a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can specify a table's age limit for a particular level by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.
2. In the **`PDB Access Mode`** field, select **`Write`** if it is not already selected.
3. Select **`OK`**. SAS IT Resource Management returns you to the main menu.
4. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Config Active PDB Dictionary`**. The Dictionary Tables window opens. The window displays a list of the tables in the active PDB.
5. Select the table for which you want to change one or more age limits.
6. Select **`ItemActions`**. A menu opens. From the menu, select **`Edit Definition`**. The Edit Table window opens. In the lower half of the window, the left column is the level identifier. The right column is the age limit value and units.
7. Find the row for the level that you want to change, and select the age limit for that level. Overwrite the existing age limit with the value that you want to use.
8. Select **`OK`**. SAS IT Resource Management returns you to the Dictionary Tables window.
9. Select **`File`**. A menu opens. From that menu, select **`End`**. SAS IT Resource Management returns you to the main menu.

Note: The preceding steps update the age limits in the table definition. If you want additional information about the table definition, see “Overview of Tables” on page 353. △

Specifying the Age Limit for a Level in a Table (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can specify a table's age limit for a particular level by following these steps:

1. From the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2. Select **Properties**. The Set Active PDB Options window opens.
3. From the **Access** tab, select **Write** if it is not already selected.
4. Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.
5. Select **Close** to return to the main window.
6. On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window displays a list of the tables in the active PDB.
7. Select the table for which you want to change one or more age limits.
8. Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.
9. Select the **Table Status** tab.
10. In the **Levels** column, find the row for the level that you want to change. On that row, find the level's age limit in the **Age Limit** column. Overwrite the existing age limit with the value that you want to use.
11. Select **OK**. SAS IT Resource Management returns you to the Manage Tables window.
12. Select **Close**. SAS IT Resource Management returns you to the main window.

Note: The preceding steps update the age limits in the table definition. If you want additional information about the table definition, see “Overview of Tables” on page 353. △

Working with Levels: Related Topics

The _LEVEL_ Variable

In the source statements that define how a formula variable is to be constructed, you can refer to an automatic variable named _LEVEL_.

For more information about the _LEVEL_ variable, see “Creating/Editing/Viewing a Formula Variable” on page 388 and the topic “CREATE FORMULA” in the SAS IT Resource Management macro reference documentation.

Restrictions on Modifications of BY Variables Lists

For more information about restrictions on modifications of BY variables lists, see the appendix ”Shared Appendix 14: Restrictions on Modifications of BY-Variables Lists“ in the *SAS IT Resource Management Server Setup Guide*.

Space Usage at a Level in a Table

If a table is "fully populated" (that is, every level is as full as its age limit allows), then the amount of data (in the data set, not the view) in a given level of the table is controlled by the length of the observations at that level of the table and by the number of observations at that level of the table. (Also, if any variables are indexed at a given level, the index for that level consumes some space.)

The length of observations at each level depends on the variables and their lengths at each level.

- At detail level, the observations have a DATETIME variable, have a DURATION variable if the table is of type Interval, have one variable for each BY variable that was specified for the table, and have one variable for each regular and derived variable that has a Kept status of *Yes*.
- At day level, the observations have a DATETIME variable, have a DURATION variable if the table is if type Interval, have one variable for each CLASS variable that was specified at the day level for the table, have one variable for each ID variable that was specified at the day level for the table, have one variable for each statistic that was requested on a regular or derived variable (that has a Kept status of *Yes*) for the day level of this table, and have any variables (not already present) that underlie formula variables that are requested for the day level of this table.
- At week level, the observations have a DATETIME variable, have a DURATION variable if the table is of type Interval, have one variable for each CLASS variable that was specified at the week level for the table, have one variable for each ID variable that was specified at the week level for this table, have one variable for each statistic that was requested on a regular or derived variable (that has a Kept status of *Yes*) for the week level of this table, and have any variables (not already present) that underlie formula variables that are requested for the week level of this table.
- At month level, the observations have a DATETIME variable, have a DURATION variable if the table is of type Interval, have one variable for each CLASS variable that was specified at the month level for the table, have one variable for each ID variable that was specified at the month level for this table, have one variable for each statistic that was requested on a regular or derived variable (that has a Kept status of *Yes*) for the month level of this table, and have any variables (not already present) that underlie formula variables that are requested for the month level of this table.
- At year level, the observations have a DATETIME variable, have a DURATION variable if the table is of type Interval, have one variable for each CLASS variable that was specified at the year level for the table, have one variable for each ID variable that was specified at the year level for this table, have one variable for each statistic that was requested on a regular or derived variable (that has a Kept status of *Yes*) for the year level of this table, and have any variables (not already present) that underlie formula variables that are requested for the year level of this table.

The number of observations at a given level of the table depends on the level.

- At detail level, typically the number of observations matches the number of records that were logged during the days specified by the table's age limit for detail level.

  However, if the data source generates a huge volume of data, the data can be pre-summarized. If the data is pre-summarized, it can be merged into the detail level. In that case, typically there is one observation for each unique combination of values of the table's detail-level BY variables. For example, if the BY variables

are DATETIME, HOUR, and SHIFT, and there is one value of shift for each hour (the shift value might differ from hour to hour) and 24 values of hour, there are 24 observations (24 hours multiplied by 1 shift) per day. The number of days is controlled by the table's age limit for detail level.

- At day level, there is one observation for each unique combination of values of the day-level CLASS variables for that table. For example, if the table's day-level CLASS variables are DATETIME, HOUR, and SHIFT, and there is one value of shift for each hour (the shift value might differ from hour to hour) and 24 values of hour, there are 24 observations (24 hours multiplied by 1 shift) per day. The number of days is controlled by the table's age limit for day level.
- At week level, there is one observation for each unique combination of values of the week-level CLASS variables for that table. For example, if the table's week-level CLASS variables are DATETIME, HOUR, and SHIFT, and there is one value of shift for each hour (the shift value might differ from hour to hour) and 24 values of hour, there are 24 observations (24 hours multiplied by 1 shift) per week. The number of weeks is controlled by the table's age limit for week level.
- At month level, there is one observation for each unique combination of values of the month-level CLASS variables for that table. For example, if the table's month-level CLASS variables are DATETIME, HOUR, and SHIFT, and there is one value of shift for each hour (the shift value might differ from hour to hour) and 24 values of hour, there are 24 observations (24 hours multiplied by 1 shift) per month. The number of months is controlled by the table's age limit for month level.
- At year level, there is one observation for each unique combination of values of the year-level CLASS variables for that table. For example, if the table's year-level CLASS variables are DATETIME, HOUR, and SHIFT, and there is one value of shift for each hour (the shift value might differ from hour to hour) and 24 values of hour, there are 24 observations (24 hours multiplied by 1 shift) per year. The number of years is controlled by the table's age limit for year level.

For additional information about space usage, see "Guidelines for Allocating Space in the PDB," "Guidelines for Revising Space Allocation in the PDB," "Guidelines for Allocating Space in the Archive," and "Guidelines for Revising Space Allocation in the Archive" in "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.

Variable Indicating Datetime, Day, Week, Month, Year

The same variable (DATETIME) is used on the observations at each level, but the contents of the variable are different at each level.

The *date* in DATETIME operates as follows:

- In the detail level: For event data, the date in the DATETIME variable represents the date of the event. For interval data, the date in the DATETIME variable typically represents the date of the beginning of the interval.

  If the data source generates large numbers of observations about transactions, the data can be pre-summarized. If the data is pre-summarized, it might or might not be merged into the detail level. For more information about the form of the pre-summarized data if it is merged into the detail level, see "Overview of Data" on page 409.
- In the day level: The date in the DATETIME variable is the date of the day that is represented by the summary statistics on the observation.
- In the week level: The date in the DATETIME variable is the date of the first day of the week that is represented by the summary statistics on the observation. (For

more information about Start of Week, see "Viewing/Editing Your Active PDB's Start of Week" on page 333.)

- In the month level: The date in the DATETIME variable is the date of the first day of the month that is represented by the summary statistics on the observation.
- In the year level: The date in the DATETIME variable is the date of the first day of the year that is represented by the summary statistics on the observation.

The *time* in DATETIME is as follows:

- In the detail level: The time in the variable DATETIME is the time at which the data was logged. (And the value in the variable HOUR is the same as the hour part of DATETIME. For example, if the time in DATETIME is 17:35:15, then HOUR has the value 17.)

  If the data source generates large numbers of observations about transactions, the data can be pre-summarized. If the data is pre-summarized, it might or might not be merged into the detail level. For more information about the form of the pre-summarized data if it is merged into the detail level, see "Overview of Data" on page 409.
- In the day, week, month, and year levels: The time in the variable DATETIME is equivalent to the time in the variable HOUR. For example, if HOUR has the value 17, then the time in DATETIME has the value 17:00:00. (The "official" value of hour is the one in HOUR, but when the observations need to be sorted by DATE and HOUR, then the sort is faster if there is a "copy" of the HOUR value next to the DATE value.)

Time Range in a Level of a Table

If a table is "fully populated" (that is, every level is as full as its age limit allows), then the time range of the data in each level of the table is controlled by the age limit of each level in the table. For example, if the table's age limits are the ones that are provided by default, then the time range of the data in each level is

- 10 days at detail level (default age limit at detail level is 10 days)
- 45 days at day level (default age limit at day level is 45 days)
- 15 weeks at week level (default age limit at week level is 15 weeks)
- 18 months at month level (default age limit at month level is 18 months)
- 5 years at year level (default age limit at year level is 5 years).

For more exact information about the effect of the age limits, see "Specifying the Age Limit for a Level in a Table" on page 346.

For more information about viewing the time range of the data in a table, whether or not the table is "fully populated", see "Viewing Amount of and Datetime Range of Data" on page 419

CHAPTER

17

Administration: Working with Tables

Working with Tables: Main Topics

Overview of Tables

Within a PDB, performance data is grouped into one or more tables.

- Logically, a table is a grouping of data about a particular topic. For example, a PDB with phone data could have one table for console data, one table for voice mail data, one table for trunk group data, and so on.

  A table has data at one or more levels:

- detail level
- day level
- week level
- month level
- year level.

You access the data in each level by using the name of the corresponding view: *level_name.table-name*. For example, if a PDB with phone data has the console information in a table named CONSOLE, then you would specify DETAIL.CONSOLE to access the detail-level data in that table.

A table has metadata in the PDB's data dictionary.

- One part of the metadata consists of the table's definition (also known as the table's properties) and the definitions/properties of its variables.

  For example, the settings of the flag that indicate whether the table is to be used, the type of observations that are to be in the table, the age limits that are to be applied to the data at each level, and so on, are table properties. The settings of the flag that indicate whether the variable is to be used, the data type, the display format, and so on, are variable properties.
- The other part of the metadata consists of the status information about the table's data.

  For example, the current number of observations at each level, the current oldest observation date at each level, and the current newest observation date at each level are status information.

- Physically, a table is a set of data sets that have data about a particular topic.

  A table has data sets in one or more libraries:

  - detail-level data is in the DETAIL library (except for availability tables; for more about availability tables, see the topic %CPAVAIL in the SAS IT Resource Management macro reference documentation).
  - day-level data is in the DAY library
  - week-level data is in the WEEK library
  - month-level data is in the MONTH library
  - year-level data is in the YEAR library.

  Also, while data is being staged, processed, and reduced, it might temporarily be located in other PDB libraries such as COLLECT and PDBWORK.

  Within the WEEK, MONTH, and YEAR libraries, the data is split data into multiple data sets. The split is for a performance reason: to decrease the size of the data set that is most likely to be active. The data set whose name ends with a P (which stands for *partial*) contains the data from the current week, month, or year. The data set whose name ends with a D contains the data from the previous weeks, months, or years. The split in no way affects the logical functioning of the table. Use the *view* to ensure that you have access to all of the data. *Do not access the data sets directly.*

  A table has metadata in the PDB's DICTLIB library. Access the metadata only through a SAS IT Resource Management GUI or through the SAS IT Resource Management macros, such as the %CPDDUTL macro. *Do not access the DICTLIB library directly.*

For the more popular data sources, table definitions (including the definitions of the variables in these tables) are supplied with SAS IT Resource Management. If you want to use one of these table definitions, you can *add* it to your active PDB's data dictionary from the master data dictionary. For more details, see the topic "Adding a Table" below.

For the other data sources, you can *create* table definitions (and the definitions of the variables in those tables) in your active PDB's data dictionary. For more information about creating table and variable definitions "from scratch," see the "Creating a Table" topic below. For more information about creating table and variable definitions by taking advantage of programs, control statements, and so on that shorten the task, see Setup Case 3, Setup Case 4, and Setup Case 5 in the "Setup the Server" topic in the "Administration: Setup" section of this user's guide.

For additional information about tables, see the following list of topics, which is in alphabetical order.

Note: If you are working with tables that are related to MXG, also see "Using MXG-Based Tables (CMAPP2): Introduction" on page 133 △

Adding a Table

When you add a table to the active PDB, the table definition is "copied" from the master data dictionary to the PDB's data dictionary, and then the data sets that the table consists of are created in the PDB. Until data is processed into the PDB and/or reduced in the PDB, the data sets have no data.

You can add a table

- by using the %CPSTART macro and the %CPDDUTL macro (with the ADD TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL ADD TABLE control statement, see the topic ADD TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Adding a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can add a table to the active PDB by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.
2. In the **`PDB Access Mode`** field, select **`Write`** if it is not already selected.
3. Select **`OK`**. SAS IT Resource Management returns you to the main menu.
4. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Config Active PDB Dictionary`**. The Dictionary Tables window opens.
5. Select **`Actions`**. A menu opens. From the menu, select **`Add Table`**. The Collector List window opens.

6 Page down to the description that represents your data. If that line is not already selected (marked with an asterisk), select the line.

7 Select **OK**. The Select Tables window opens.

8 Page down to the description that represents the table that you want to add. Select the line.

9 Select **OK**. SAS IT Resource Management adds the table and returns you to the Dictionary Tables window. The table that you added is now in the list of tables in the active PDB.

10 Select **File ▶ End ▶ OK**

11 You might want to edit the table definition. For example, you might want to override the default value of the **Archive** field or to override the default age limit at one or more levels.

For information about editing the table definition, see "Viewing/Editing a Table" on page 364.

Adding a Table (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can add a table to the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The **Manage PDBs** window opens.

2 Select **Properties**. The Set Active PDB Options window opens.

3 On the **Access** tab, select **WRITE** if it is not already selected.

4 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.

5 Select **Close**. SAS IT Resource Management returns you to the main window.

6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.

7 Select **File**. A menu opens. From the menu, select **Add Table Definition**. The Select Tables window opens.

8 In the **Collector** field, select **Selected**. The Select Collector window opens.

9 Select the line with the description of the data collector or data source for which you want to add the table. Select **OK**.

10 In the **Scope** field, select **All Supplied** if it is not already selected.

11 Select **OK**. The Explore Tables/Variables window opens. The window lists all of the table definitions for the data collector or data source that are in the master data dictionary.

12 Scroll to the definition that you want to add and select it.

13 Select **Add**. The Dictionary Utility in Progress window opens and SAS IT Resource Management adds the table definition to the active PDB.

14 To close the Dictionary Utility in Progress window, select **Close**. SAS IT Resource Management returns you to the Manage Tables window. The table that you added is now on the list of tables in the active PDB.

15 Select **Close**. SAS IT Resource Management returns you to the main window.

16 You might want to edit the table definition. For example, you might want to override the default value of the **Archive** field or to override the default age limit at one or more levels.

For information about editing the table definition, see "Viewing/Editing a Table" on page 364.

Creating a Table

When you create a table in the active PDB, the table definition is "copied" from the location where you saved it, and then the data sets that the table consists of are created in the PDB. Until data is processed into the PDB and/or reduced in the PDB, the data sets have no data.

You can create a table in the active PDB

- by using the %CPSTART macro and the %CPDDUTL macro (with the CREATE TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL CREATE TABLE control statement, see the topic CREATE TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Creating a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can create a table in the active PDB by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2. In the **PDB Access Mode** field, select **Write** if it is not already selected.
3. If you want to be able to archive incoming data for the table that you create below (or for any other table in the PDB), check that the archive information is specified in the PDB and, if it is not, specify it. For more information, see "Viewing/Editing Your Active PDB's Archive Information" on page 326.
4. Select **OK**. SAS IT Resource Management returns you to the main menu.
5. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens.
6. Select **Actions**. A menu opens. From the menu, select **Create Table**. The Create New Table window opens.
7. In the **Table** field, type the name of the table that you want to create.

   For the table-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for z/OS Tables and Variables" on page 371.
8. In the **Type** field, select **Interval** if each observation in the table will contain data that is related to an interval of time (synchronous data), or select **Event** if each observation in the table will contain data that is related to an event (asynchronous data).
9. Select **OK**. The Create Table window opens.
10. In the **Table Name** field, the name represents the choice that you made in the previous window. To change the name, select **Cancel**, and then reopen the previous window.
11. In the **Table Type** field, the specified type represents the choice that you made in the previous window. To change the type, select **Cancel** to exit from this window, and then reopen the previous window.

12 The **Table Kept** field specifies whether the table's definition is to be used or ignored by SAS IT Resource Management.

If the value is *Yes*, the table's definition will be saved (when you select **OK**) and used. If the value is *No*, the table's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

Note: If you are already processing data for one or more other tables, you might want to specify *No* for Table Kept when you are first working on this table definition. That way, if you are interrupted before you finish the table definition and its variable's definitions, you can save the definition(s) and SAS IT Resource Management will not attempt to use the table and its variables for processing.

Later, you can edit the definition(s) to continue working on them.

When the table definition and its variable's definitions are finished, you can specify *Yes* for Table Kept, and save the definition. Then, SAS IT Resource Management will use the definitions for processing. △

To specify the other value, select its radio button.

13 The **Collector** field has a default value of *GENERIC*.

For now, leave the value set to *GENERIC*.

14 The **Process Tool** field has a default value of *SASDS*.

For now, leave the value set to *SASDS*.

15 The **External Name** field has a default value of the table name.

For now, leave the value set to the table name.

16 In the **Archive** field, if you want incoming data to the table to be archived, select **On** if it is not already selected; otherwise, select **Off**. (For more information about archiving, see "Archiving Incoming Data" on page 411.)

17 Select **Description**. The Edit Description window opens.

a The default label is the table name. In the **Label** field, type a brief description of the data that is to be in the table. The label is used by the GUI and data dictionary to provide a short description of the table.

b The default description is the table name. In the **Description** field, type a long description of the data that is to be in the table. The description is used by the GUI and data dictionary to provide a long description of the table.

c Select **OK**. SAS IT Resource Management returns you to the Dictionary Tables window.

18 The age limit that is to be used for each level is specified under the Duration header. The values that you see are the default values. If you do not want to use the default values, change them by overwriting them with the values that you prefer.

19 Select **OK**. SAS IT Resource Management saves the table definition and returns you to the Dictionary Tables window.

The new table is on the list of tables in the active PDB.

20 Select **File ▶ End ▶ OK**

SAS IT Resource Management builds views based on the table definitions and the variable definitions for DATETIME, LSTPDATE (and, for tables of type Interval, DURATION), and returns you to the main menu.

21 Create variable definitions for any regular, derived, and formula variables that you want in the table.

For information about creating variables, see "Overview of Variables" on page 378.

22 Edit the table definition. Make at least these changes:

- Specify the BY, CLASS, ID, and/or INDEX variables for the table.
- If your data is character-delimited, specify a tool name of *CHARDELIM* for the table.

  Make a corresponding change to the table's external name, if necessary.
- If you intend to package the collector-support entities for installation later,
  - change the value of the **Collector** field from *GENERIC* to the collector name that you intend to use when you package and install the collector-support entities
  - change the value of the **Process Tool** field, if necessary, to the tool name that you intend to use when you package and install the collector-support entities. Make a corresponding change to the table's external name, if necessary.

For more information about editing a table definition, see "Viewing/Editing a Table" on page 364.

Creating a Table (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create a table in the active PDB by following these steps:

1. On the main window's **Administration** tab, select **Manage PDBs**. The **Manage PDBs** window opens.
2. Select **Properties**. The Set Active PDB Options window opens.
3. On the **Access** tab, select **WRITE** if it is not already selected.
4. If you want to be able to archive incoming data for the table that you create below (or for any other table in the PDB), check that the archive information is specified in the PDB and, if it is not, specify it. For more information, see "Viewing/Editing Your Active PDB's Archive Information" on page 326.
5. Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.
6. Select **Close**. SAS IT Resource Management returns you to the main window.
7. From the main window, select **Manage Tables**. The Manage Tables window opens.
8. Select **File**. A menu opens. From the menu, select **New Table Definition** . The Create New Table window opens.
9. In the **Enter New Table Name** field, type the name of the table that you want to create.

   For the table-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for UNIX and Windows Tables and Variables" on page 374.
10. Select **OK**. The Create Table Properties window opens.
11. In the **Table Name** field, the name represents the choice that you made in the previous window. If you want to change the name, select **Cancel** to exit from this window, and then reopen the previous window.
12. In the **Table Type** field, select **Interval** if each observation contains data that is related to an interval of time, or select **Event** if each observation contains data that is related to an event.
13. The **Table Kept** field specifies whether the table's definition is to be used or ignored by SAS IT Resource Management.

    If the value is *Yes*, the table's definition will be saved (when you select **OK**) and used. If the value is *No*, the table's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

Note: If you are already processing data for one or more other tables, you might want to specify *No* for Table Kept when you are first working on this table definition. That way, if you are interrupted before you finish the table definition and its variable's definitions, you can save the definition(s) and SAS IT Resource Management will not attempt to use the table and its variables for processing.

Later, you can edit the definition(s) to continue working on them.

When the table definition and its variable's definitions are finished, you can specify *Yes* for Table Kept, and save the definition. Then, SAS IT Resource Management will use the definitions for processing. △

To specify the other value, select its radio button.

14 The **Collector** field has a default value of *GENERIC*.

For now, leave the value set to *GENERIC*.

15 The **Process Tool** field has a default value of *SASDS*.

For now, leave the value set to *SASDS*.

16 The **External Name** field has a default value of the table name.

For now, leave the value set to the table name.

17 Select the **Advanced** tab. In the **Archive** field, if you want incoming data to the table to be archived, select **Yes** if it is not already selected; otherwise, select **No**. (For more information about archiving, see "Archiving Incoming Data" on page 411.)

18 Select the **General** tab. In the **Description** field, type a more complete description of the data that is to be in the table

19 Select the **Table Status** tab. The age limit that is to be used for each level is specified under the **Age Limit** header. The values that you see are the default values. If you do not want to use the default values, change them by overwriting them with the values that you prefer.

20 Select **OK ▶ OK** SAS IT Resource Management saves your changes to the table definition and returns you to the Manage Tables window.

21 Select **Close ▶ OK** SAS IT Resource Management builds views based on the table definitions and on the variable definitions for DATETIME, LSTPDATE (and, for tables of type Interval, DURATION), and returns you to the main menu.

22 Create variable definitions for any regular, derived, and formula variables that you want in the table.

For information about creating variables, see "Overview of Variables" on page 378.

23 Edit the table definition. Make at least these changes:

- Specify the BY, CLASS, ID, and/or INDEX variables for the table.
- If your data is character-delimited, specify a tool name of *CHARDELIM* for the table.

  Make a corresponding change to the table's external name, if necessary.
- If you intend to package the collector-support entities for installation later,
  - change the value of the **Collector** field from *GENERIC* to the collector name that you intend to use when you package and install the collector-support entities
  - change the value of the **Process Tool** field, if necessary, to the tool name that you intend to use when you package and install the collector-support entities. Make a corresponding change to the table's external name, if necessary.

For information about editing a table definition, see "Viewing/Editing a Table" on page 364.

Deleting a Table

When you delete a table, the table's data, data sets, definition, and other metadata are deleted.

You can delete a table from your active PDB

- by using the %CPSTART macro and the %CPDDUTL macro (with the DELETE TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL DELETE TABLE control statement, see the topic DELETE TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

To delete the data in the table without deleting the table definition, see "Purging Data" on page 415.

Deleting a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can delete a table from the active PDB by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.
2. In the **`PDB Access Mode`** field, select **`Write`** if it is not already selected.
3. Select **`OK`**. SAS IT Resource Management returns you to the main menu.
4. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Config Active PDB Dictionary`**. The Dictionary Tables window opens.
5. Select the table that you want to delete.
6. Select **`ItemActions`**. A menu opens. From the menu, select **`Delete Table`**. The Delete Table window opens.
7. Select **`Delete Table:  Delete data and dictionary entries`**.

   Note: For information about the other choice, see "Purging Data" on page 415. △
8. Select **`OK`**. The table definition (and the table's other metadata, data, and data sets) are deleted from the active PDB. SAS IT Resource Management returns you to the Dictionary Tables window. The table that you deleted is no longer in the list of tables in the active PDB.
9. Select **File ▶ End**

Deleting a Table (UNIX, Windows)

1. On the main window's **`Administration`** tab, select Manage PDBs. The **`Manage PDBs`** window opens.
2. Select **`Properties`**. The Set Active PDB Options window opens.
3. On the **`Access`** tab, select **`WRITE`** if it is not already selected.
4. Select **`OK`**. SAS IT Resource Management returns you to the Manage PDBs window.
5. Select **`Close`**. SAS IT Resource Management returns you to the main window.

6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.

7 Select the table whose definition you want to delete.

8 Select **File**. A menu opens. From the menu, select **Delete Table Definition**. The Delete Table window opens.

9 Select **Delete all data + definition**.

Note: For information about the other choice, see "Purging Data" on page 415. △

10 Select **OK**. SAS IT Resource Management deletes the table definition (and the table's other metadata, data, and data sets) and displays a "task done" window.

11 Select **OK**. SAS IT Resource Management returns you to the Manage Tables window. The table that you deleted is no longer in the list of tables in the active PDB.

12 Select **Close**. SAS IT Resource Management returns you to the main window.

Exploring Supplied Table Definitions

You can explore supplied table definitions in the active PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Write access or update access to the PDB is not required for you to view or search the list of supplied table definitions.

For more information about table definitions, see "Overview of Tables" on page 353.

Exploring Supplied Table Definitions (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can explore supplied table definitions in the active PDB by following these steps:

1 Select **Explore** from the SAS IT Resource Management main menu. A menu opens.

2 Select **All Supplied Tables**. The Supplied Tables window opens and displays a list of the supplied tables.

3 Select **Edit ▶ Find**

Next, in the **Search** field type the name of the table definition that you want to explore, and select the other search characteristics. Then, select **OK**.

SAS IT Resource Management searches the table description fields, highlights the table or tables whose descriptions contain the string that you entered in the **Search** field, and scrolls to the first highlighted table. (If there is more than one table, you might need to scroll to find the others.)

4 Select the table that you want to know more about, and then select **ItemActions ▶ Browse Definition** or **ItemActions ▶ List Variables**

Exploring Supplied Table Definitions (UNIX and Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can explore supplied table definitions in the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Explore Tables/Variables**. The Select Tables and Variables window opens.

2 In the **Choose** field, select **Tables**.

3 Select and/or type the remaining criteria that describe which tables you want to view or explore, and then select **OK**. A list displays the variables that meet your criteria.

Note: If you search by MIB name, you might need to try several versions of the MIB name. A common name of the MIB (like "MIB-II" or "RMON MIB") might not be in the description. SAS IT Resource Management names MIB tables after the name that is used in the MIB itself. For example, the opening line of the MIB that is defined by RFC1271 (also known as the RMON MIB) is

```
RFC1271-MIB DEFINITIONS
```

Thus, SAS IT Resource Management identifies tables in this MIB with the string RFC1271-MIB. The name of each individual table/group (like "ICMP" or "alarmTable") is also in the description field. For example, the description field for table HN2ICM begins nb-mibII.icmp and for table HRMATM begins

```
RFC1271-MIB.alarmTable
```

△

Viewing a Table's Status Information

You can view the table status information

- by using the %CPSTART macro and the %CPDDUTL macro (with the STATUS TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation . For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL STATUS TABLE control statement, see the topic STATUS TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing a Table's Status Information (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can view the status information about a table in the active PDB by following these steps:

- From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens.
- Select the table whose status information you want to view.
- Select **ItemActions**. A menu opens. From the menu, select **Table Status**. The Edit Table window opens.
- For each level, you can see
  - the datetime of the oldest data
  - the datetime of the newest data
  - the datetime of the last update
  - the number of observations.
- To return to the main menu, select **Goback ► File ► End**

Viewing a Table's Status Information (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view the status information about a table in the active PDB by following these steps:

- On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.
- Select the table whose status information you want to view.
- Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.
- Select the **Table Status** tab.
- For each level, you can see
  - the datetime of the oldest data
  - the datetime of the newest data
  - the datetime of the last update
  - the number of observations.
- To return to the main window, select **OK ► Close**

Viewing/Editing a Table

You can view and/or update a table definition (also known as the table's properties) in the active PDB

- by using the %CPSTART macro and the %CPDDUTL macro (with the UPDATE TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL UPDATE TABLE control statement, see the topic UPDATE TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing/Editing a Table (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can update the definition/properties of a table in the active PDB by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2. In the **PDB Access Mode** field, select **Write** if it is not already selected.
3. If you want to be able to archive incoming data for the table that you edit below (or for any other table in the PDB), check that the archive information is specified in the PDB and, if it is not, specify it. For more information, see "Viewing/Editing Your Active PDB's Archive Information" on page 326.
4. Select **OK**. SAS IT Resource Management returns you to the main menu.
5. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens.
6. Select the table whose definition you want to update.

7 Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.

8 In the **Table Name** field, you can see the table's name. You cannot change the table's name.

9 In the **Table Type** field, you can see the table's type. You cannot change the table's type.

The value *Interval* indicates that each observation in the table contains data that is related to an interval of time (synchronous data). The value *Event* indicates that each observation in the table contains data that is related to an event (asynchronous data).

10 The value of the **Table Kept** field specifies whether the table's definition is to be used or ignored by SAS IT Resource Management.

If the value is *Yes*, the table's definition will be saved (when you select **OK**) and used. If the value is *No*, the table's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

To specify the other value, select its radio button.

11 The **Collector** field, **Process Tool** field, and **External Name** field operate as follows:

- For tables that SAS IT Resource Management supplies, do not change the collector name, tool name, or external name.

  The collector name and tool name are documented in the table in the description of the COLLECTR= parameter in the topic %CPPROCES in the SAS IT Resource Management macro reference documentation.

  The only place that the external name is documented is in the table definition.

  When you process data into this table with a process macro, specify

  ```
  %CxPROCES(...COLLECTR=collector-name,TOOLNM=tool-name,...);
  ```

  where *x*=M, P, S, or W.

  For more information about the %CMPROCES macro, see the topic %CMPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CSPROCES macro, see the topic %CSPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CWPROCES macro, see the topic %CWPROCES in the SAS IT Resource Management macro reference documentation.

- For user-created tables, if you *have* created the table definition as part of a set of collector-support entities for installation and use with %CPPROCES,
  - the collector name and tool name should be the same as the ones that you have used or will use to "tag" the package. For more information about collector-support entity packages and installation, see "Create and Install a Collector Package" in the *SAS IT Resource Management Server Setup Guide*.

    When you process data into this table with a process macro, specify

    ```
    %CPPROCES(...COLLECTR=collector-name,TOOLNM=tool-name,...);
    ```

  - the external name should be the name of the SAS data set or view in the COLLECT library that represents the staged data.

    For example, if the staging code stages the data to *COLLECT.XYZ*, then *XYZ* is the external name.

- For user-created tables, if you *have not* created the table definition as part of a set of collector-support entities for installation and use with %CPPROCES,
  - use *GENERIC* as the collector name, and use *SASDS* or *CHARDELIM* as the tool name. (If you want to use the supplied staging code for character-delimited data, specify *CHARDELIM*. Otherwise, specify *SASDS*.)

    When you process data into this table with a process macro, specify

    ```
    %CxPROCES(...COLLECTR=GENERIC,TOOLNM=tool-name,...)
    ```

    where *x*=M, P, S, or W.
  - the external name should be the name of the SAS data set or view (in the COLLECT library) that represents the staged data.

    For example, if the staging code stages the data to *COLLECT.XYZ*, then *XYZ* is the external name.

To change the value in the **Collector** field, overwrite the existing value with the new value.

To change the value in the **Process Tool** field, overwrite the existing value with the new value.

To change the value in the **External Table Name** field, select the Advanced table and overwrite the existing value with the new value.

12 The **Archive** field specifies whether you want incoming data to the table to be archived.

If you want to change the value, select the other radio button.

13 Select **Description**. The Edit Description window opens. The value in the **Label** field is a brief description of the data that is in the table. The value in the **Description** field is a long description of the data that is in the table.

If you want to update the value in the **Label** field, type over it. If you want to update the value in the **Description** field, type over it.

Select **OK**. SAS IT Resource Management returns you to the Dictionary Tables window.

14 The age limit that is to be used for each level is specified under the **Duration** header. The values that you see are the default values. If you do not want to use the default values, change them by overwriting them with the values that you prefer.

15 Review the variables in your table to verify that their Kept status and the statistics to be calculated are appropriate for your site. Within your tables, the variables that are generally most useful for analysis are the ones whose default value for the Kept status is *Yes*. If you at first decide on a Kept status of *Yes*, you can change it to *No* later. If you change the Kept status to *No*, the existing values are deleted and new values are not processed into the detail level from that time forward. Similarly, if you decide on a Kept status of *No*, you can decide later to change it to *Yes*, in which case values are processed into the detail level from that time forward.

16 You can create, delete, edit, and view the table's variables by selecting **Variable List**. The Variables window opens and displays a list of the variables in the table.

For information about creating, deleting, editing and viewing variables, see "Overview of Variables" on page 378.

To return to the main table definition window, select **File ▶ End**

17 If you want to view the list of BY or CLASS variables for each level in the table, in the **Constructional Variables to Display** field, select **By/Class**. The list of

BY/CLASS variables for each level is displayed under the **By/Class Variables** heading.

- If you want to edit the list for the detail level, select **Detail**. The Edit By-Variable Order window opens and displays the list of the table's BY variables for the detail level (with numbers indicating order), followed by a list of all other variables in the detail level that are eligible to be BY variables.

  To add a variable to the BY variables list, select that variable in the lower list. To remove a variable from the BY variables list, select that variable in the upper list.

  When you finish specifying the list of BY variables (in which the last three variables are typically DATETIME HOUR SHIFT), select **OK**. SAS IT Resource Management returns you to the Edit Table window and displays the new list.
- If you want to edit the list for the day level, select **Day**. The Edit Class-Variable Order window opens and displays the list of the table's CLASS variables for the day level (with numbers indicating order), followed by a list of all other variables in the day level that are eligible to be CLASS variables.

  To add a variable to the CLASS variables list, select that variable. To remove a variable from the CLASS variables list, select that variable.

  When you finish specifying the list of the CLASS variables for the day level (in which the last three variables are typically DATETIME HOUR SHIFT), select **OK**. SAS IT Resource Management returns you to the Edit Table window and displays the new list.
- If you want to edit the value for the week level, select **Week** and specify the table's CLASS variables list for the week level. Use the technique that is described above for day level.
- If you want to edit the value for the month level, select **Month** and specify the table's CLASS variables list for the month level. Use the technique that is described above for day level.
- If you want to edit the value for the year level, select **Year** and specify the table's CLASS variables list for the year level. Use the technique that is described above for day level.

18 If you want to view the list of ID variables for each level in the table (except detail level, which does not have a list of ID variables), in the **Constructional Variables to Display** field, select **ID**. The list of ID variables for each level displays under the ID Variables heading.

If you want to edit the list for any level, use the technique that is described above for editing the BY/CLASS variables lists.

19 If you want to view the list of Index variables for each level in the table, in the **Constructional Variables to Display** field, select **Index**. The list of Index variables for each level displays under the Index Variables heading.

If you want to edit the list for any level, use the techniques that is described above for editing the BY/CLASS variables lists.

Note that there is no order to the index variables because each index is simple (that is, each index variable defines a different index).

20 Select **OK ▶ File ▶ End ▶ OK**

Viewing/Updating a Table (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can update the definition/properties of a table in the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2 Select **Properties**. The Set Active PDB Options window opens.
3 On the **Access** tab, select **WRITE** if it is not already selected.
4 If you want to be able to archive incoming data for the table that you edit below (or for any other table in the PDB), check that the archive information is specified in the PDB and, if it is not, specify it. For more information, see "Viewing/Editing Your Active PDB's Archive Information" on page 326.
5 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.
6 Select **Close**. SAS IT Resource Management returns you to the main window.
7 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.
8 Select the table whose definition you want to edit.
9 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.
10 In the **Table Name** field, you can see the table's name. You cannot change the table's name.
11 In the **Table Type** field, you can see the table's type. You cannot change the table's type.

   The value *Interval* indicates that each observation in the table contains data related to an interval of time (synchronous data). The value *Event* indicates that each observation in the table contains data related to an event (asynchronous data)
12 The value of the **Table Kept** field specifies whether the table's definition is to be used or ignored by SAS IT Resource Management.

   If the value is *Yes*, the table's definition will be saved (when you select **OK**) and used. If the value is *No*, the table's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

   To specify the other value, select its radio button.
13 The **Collector** field and **Process Tool** field are on the **General** tab. The **External Table Name** field is on the **Advanced** tab. The **Collector** field, **Process Tool** field, and **External Name** field operate as follows:
   - For tables that SAS IT Resource Management supplies, do not change the collector name, tool name, or external name.

     The collector name and tool name are documented in the table in the description of the COLLECTR= parameter in the topic %CPPROCES in the SAS IT Resource Management macro reference documentation.

     The only place that the external name is documented is in the table definition.

     When you process data into this table with a process macro, specify

     ```
     %CxPROCES(...COLLECTR=collector-name,TOOLNM=tool-name,...);
     ```

     where *x*=M, P, S, or W.

     For more information about the %CMPROCES macro, see the topic %CMPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CSPROCES macro, see the topic %CSPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CWPROCES macro, see the topic %CWPROCES in the SAS IT Resource Management macro reference documentation.

- For user-created tables, if you *have* created the table definition as part of a set of collector-support entities for installation and use with %CPPROCES,
  - the collector name and tool name should be the same as the ones that you have used or will use to "tag" the package. For more information about collector-support packages and installation, see "Creating and Installing a Collector Support Package" on page 426.

    When you process data into this table with a process macro, specify

    ```
    %CPPROCES(...COLLECTR=collector-name,TOOLNM=tool-name,...);
    ```

  - the external name should be the name of the SAS data set or view in the COLLECT library that represents the staged data.

    For example, if the staging code stages the data to *COLLECT.XYZ*, then *XYZ* is the external name.
- For user-created tables, if you *have not* created the table definition as part of a set of collector-support entities for installation and use with %CPPROCES,
  - use *GENERIC* as the collector name, and use *SASDS* or *CHARDELIM* as the tool name. (If you want to use the supplied staging code for character-delimited data, specify *CHARDELIM*. Otherwise, specify *SASDS*.)

    When you process data into this table with a process macro, specify

    ```
    %CxPROCES(...COLLECTR=GENERIC,TOOLNM=tool-name,...)
    ```

    where *x*=M, P, S, or W.
  - the external name should be the name of the SAS data set or view (in the COLLECT library) that represents the staged data.

    For example, if the staging code stages the data to *COLLECT.XYZ*, then *XYZ* is the external name.

To change the value in the **Collector** field, overwrite the existing value with the new value.

To change the value in the **Process Tool** field, overwrite the existing value with the new value.

To change the value in the **External Table Name** field, select the **Advanced** tab and overwrite the existing value with the new value.

14 Select the **Advanced** tab, if it is not already selected. The **Archive** field specifies whether you want incoming data to the table to be archived.

If you want to change the value, select the other radio button.

15 Select the **General** tab. In the **Description** field, type a long description of the data that is to be in the table.

16 Select the **General** tab. The value in the **Label** field is a brief description of the data that is in the table. The value in the **Description** field is a long description of the data that is in the table.

If you want to update the value in the **Label** field, type over it. If you want to update the value in the **Description** field, type over it.

17 Select the **Table Status** tab. The age limit that is to be used for each level is specified under the **Age Limit** header.

If you want to change any of the existing values, overwrite them with the values that you prefer.

18 Review the variables in your table to verify that their Kept status and the statistics to be calculated are appropriate for your site. Within your tables, the

variables that are generally most useful for analysis are the ones whose default value for the Kept status is *Yes*. If you at first decide on a Kept status of *Yes*, you can to change it to *No* later. If you change the Kept status to *No*, the existing values are deleted and new values are not processed into the detail level from that time forward. Similarly, if you decide on a Kept status of *No*, you can decide later to change it to *Yes*, in which case values are processed into the detail level from that time forward.

Note: If in the SAS log you received messages similar to

```
NOTE: CSIIPDI not initialized
```

while running %CSPROCES on the exported data, then some of the variables that are defined in the table definition for this device are not being logged. To enable logging of these attributes, see "Cabletron Spectrum Appendix 5: Editing Model Types for SPECTRUM" in the *SAS IT Resource Management Server Setup Guide*. △

19 From the **General** tab, you can create, delete, edit, and view the table's variables by selecting **Variables**. The View Variables window opens and displays a list of the variables in the table. Select **Close** to return to the main window for table properties.

For information about creating, deleting, editing and viewing variables, see "Overview of Variables" on page 378

20 Select the **Structure Variables** tab. If you want to view the list of BY or CLASS variables for each level in the table, select **By/Class**. The list of BY/CLASS variables for each level is displayed under the **Variables** heading.

- If you want to edit the list for the detail level, select **Detail**. The Edit By-Variable Order window opens and displays the list of the table's BY variables for the detail level (with numbers indicating order), followed by a list of all other variables in the detail level that are eligible to be BY variables.

  To add a variable to the BY variables list, select that variable from the lower list and select the up arrow. To remove a variable from the BY variables list, select that variable from the upper list and select the down arrow.

  When you finish specifying the list of BY variables (in which the last three variables are typically DATETIME HOUR SHIFT), select **OK**. The tab displays the new list.

- If you want to edit the list for the day level, select **Day**. The Edit Class Variable Order window opens and displays the list of the table's CLASS variables for the day level (with numbers indicating order), followed by a list of all other variables in the day level that are eligible to be CLASS variables.

  To add a variable to the CLASS variables list, select that variable from the lower list and select the up arrow. To remove a variable from the CLASS variables list, select that variable from the upper list and select the down arrow.

  When you finish specifying the list of the CLASS variables for the day level (in which the last three variables are typically DATETIME HOUR SHIFT), select **OK**. The tab displays the new list.

- If you want to edit the value for the week level, select **Week** and specify the table's CLASS variables list for the week level. Use the technique that is described above for day level.

- If you want to edit the value for the month level, select **Month** and specify the table's CLASS variables list for the month level. Use the technique that is described above for day level.

- If you want to edit the value for the year level, select **`Year`** and specify the table's CLASS variables list for the year level. Use the technique that is described above for day level.

21 If you want to view the list of ID variables for each level in the table (except detail level, which does not have a list of ID variables), select **`ID`**. The list of ID variables for each level is displayed under the **`Variables`** heading.

If you want to edit the list for any level, use the technique that is described above for editing the BY/CLASS variables lists.

22 If you want to view the list of Index variables for each level in the table, select **`Index`**. The list of Index variables for each level is displayed under the **`Variables`** heading.

If you want to edit the list for any level, use the techniques that is described above for editing the BY/CLASS variables lists.

Note that there is no order to the index variables because each index is simple (that is, each index variable defines a different index).

23 Select **OK ► Close**

Viewing/Editing Defaults for Table Definitions

You can create one or more table definitions in batch mode by using the %CPDDUTL macro. Any table properties that you do not specify are defined by default values. You can edit those default values if you want to use values that are different from the built-in values.

You can display and edit the defaults for table definitions

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET DEFAULTS control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET DEFAULTS control statement, see the topic SET DEFAULTS in the SAS IT Resource Management macro reference documentation.

Note: The values that you define expire at the end of the current SAS IT Resource Management session. △

Working with Tables: Related Topics

Naming Conventions for z/OS Tables and Variables

Naming Conventions for Supplied Tables and Variables

One set of supplied tables is for MXG data from the SMF collector. Those tables are named according to one of the following alterations of the corresponding MXG name.

XTY*ttss* where *tt* is the type and *ss* is the subtype, if any, of the data. This is used for SMF data that is assigned to specific SMF record types. For example, SMF type 70 records are stored in table XTY70, and SMF type 30 subtype 5 records are stored in table XTY30_5.

X*ffllll* where *ff* is the first two characters of the MXG name and *llll* is the last four characters of the MXG name. If the MXG name is less than seven characters long, then *ffllll* is the MXG name padded with blanks on the right.

Variables supplied by SAS IT Resource Management are named according to the following scheme:

- If the MXG name is one to seven characters long, the SAS IT Resource Management name at detail level is the same as the MXG name.

  At the summary levels, a statistic suffix (for example, S for sum) is added to the name as the eighth character. If the name at the detail level contains less than seven characters, underscores supply padding so that the statistic suffix is the eighth character.

  For example, if the MXG name is ABCDE, the SAS IT Resource Management name at the detail level is ABCDE, and the name of the sum for ABCDE at the summary levels is ABCDE__S.
- If the MXG name is eight characters long, the SAS IT Resource Management name at the detail level is the MXG name without the third character of the MXG name (or with some other change if necessary to make the SAS IT Resource Management variable name unique).

  At the summary levels, a statistic suffix (for example, S for sum) is added to the name as the eighth character.

  For example, if the MXG name is ABCDEFGH, the SAS IT Resource Management name at detail level is ABDEFGH, and the name of the sum for ABDEFGH at the summary levels is ABDEFGHS.

For more information about the statistic suffix, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

Another set of supplied tables uses a different naming convention:

- ACC (for UNIX accton command data)
- ETE (for End-to-End Watch by Candle)
- NT*x* (for NTSMF data)
- NWC (for NetFlow FlowCollector data from Cisco)
- PCS (for HP OpenView Performance Agent and HP OpenView Reporter data)
- PT*x* (PATROL data from BMC software)
- R*xx* (for Rolm PBX data)
- SAP (for SAP R/3 data)
- SAR (for UNIX **sar** command data)
- SS (SiteScope data from Mercury Interactive software)
- VZ (for BMC Visualizer data)
- XDC (for IBM DCOLLECT data)
- XER (for IBM EREP data)
- XIM (for Boole and Babbage / BMC Software IMS Measurement Facility (IMF) data)
- XQA (for IBM AS/400 data)
- XTC2 (for The Monitor for CICS from Landmark version 2 or later)
- XTCC (for The Monitor for CICS from Landmark version 1.3 data)
- XTCnon-2-or-C (for The Monitor for CICS from Landmark version 8.0 data)
- XTMD (for The Monitor for DB2 from Landmark data)
- XTMS and XDSNREC (for Computer Associates Tape Management System data)

- XTPF (for IBM Transaction Processing Facility data)
- WEB (for Web log data).

Naming Conventions for User-Defined Tables and Variables

The name of a user-defined table

- must start with a letter. The letter *U* is strongly recommended for user-defined tables.
- has a maximum length of seven characters (U plus zero to six letters, numbers, and/or underscores).

  The name is not case sensitive; thus, NAME=utable1 and NAME=UTABLE1 are equivalent.
- must be unique within the PDB.

  In case you want to use the same table in another PDB later, you might also want the name to be unique within all the PDBs at your site.

  Similarly, in case you want to install the table in the master data dictionary later, you might also want the name not to match the name of any supplied table.
- must avoid the name space of any other table within the PDB. The name space of a table consists of the name itself, the name with *D* appended, and the name with *P* appended.

  For example, if the table name is *CPU*, then the table uses the names *CPU*, *CPUD*, and *CPUP*. Thus, all three of these names need to be avoided in choosing the name for a new table.

  For the reasons that are mentioned above, you might also want to avoid the name space of tables in other PDBs at your site and the name space of supplied tables.

  For more information about the table name with *D* and *P* appended, see "Overview of Tables" on page 353.

For user-defined variables in supplied tables or user-defined tables:

- A table must have a variable named DATETIME. For more information about the variable DATETIME, see "Variable Indicating Datetime, Day, Week, Month, Year" on page 350.
- A table must have a variable named DURATION, if the table type is *Interval*. DURATION is the length of the interval, in seconds.
- The name of a user-defined variable
  - must start with a letter.
  - has a maximum length of seven characters (letters and numbers only; do not use underscores).

    The name is not case sensitive; thus, NAME=var1 and NAME=VAR1 are equivalent.

    Note: Underscores are not allowed in names of user-defined variables; however, they might be added automatically, as padding (so that the statistic suffix is in the eighth character) in the variable names that are generated for you when you request predefined statistics at the day, week, month, and year levels on the user-defined variables. △
  - must be unique within the table.

    In case you want to use the same variable in another table later, you might also want the name to be unique within all tables in the PDB.

Similarly, in case you want to install the table in the master data dictionary later, you might also want the name not to match the name of any supplied variable.

- must not be COUNT or TYPE.

Naming Conventions for UNIX and Windows Tables and Variables

Naming Conventions for Supplied Tables

Tables supplied by SAS IT Resource Management are named according to the following scheme: *CCCnnn*, or *cmmnnn*, where the following definitions apply:

CCC is the three-character Collector, which is defined as follows:

ACC	UNIX accounting enhanced (ACCUNX) based on the UNIX accton command
ETE	End-to-End Watch by Candle
LS*x*	Landmark Performance Works
NT*x*	NTSMF
NWC	Network Conversation tables (NetFlow FlowCollector by Cisco Systems)
PCS	OpenView Performance Collection Software and MeasureWare by HP
PRX	Probe/X by Landmark Systems
PT*x*	BMC Patrol
R*xx*	Rolm PBX
SAP	SAP R/3
SAR	Based on the UNIX **sar** command
SNM	SunNet Mgr RPC-Agents by Sun/Connect
SS	SiteScope by Mercury Interactive
TKR	TRAKKER by Concord Communication
TMN	Landmark TMON for UNIX
XTYttss or Xfflll	SMF data staged by MXG. For more details, see “Naming Conventions for z/OS Tables and Variables” on page 371.
WEB	Web Log Server

c is the one-character Collector, which is defined as follows:

C	SPECTRUM by Aprisma, Inc.
H	HP OpenView Node Mgr or IBM Tivoli NetView for AIX
S	SunNet Manager by Sun/Connect
M	MIB-based Data

mm is the two-character Creator. Examples for coding this field are as follows:

N2	is the abbreviation to use for NB MIB-II

VZ	is the abbreviation to use for BMC Visualizer
WF	is the abbreviation to use for Wellfleet MIB

nnn is the three-character Name, which is an abbreviation for the group of metrics in this table. It is defined as follows:

ICM	nb-mibII.icmp
NIX	nb-mibII.non-indexed metrics.

Naming Conventions for User-Defined Tables and Variables

The name of a user-defined table

- must start with a letter. The letter *U* is strongly recommended for user-defined tables.
- has a maximum length of seven characters (U plus no more than six letters, numbers, and/or underscores).

  The name is not case sensitive; thus, NAME=utable1 and NAME=UTABLE1 are equivalent.
- must be unique within the PDB.

  In case you want to use the same table in another PDB later, you might also want the name to be unique within all the PDBs at your site.

  Similarly, in case you want to install the table in the master data dictionary later, you might also want the name not to match the name of any supplied table.
- must avoid the name space of any table within the PDB. The name space of a table consists of the name itself, the name with *D* appended, and the name with *P* appended. For example, if a table name is *CPU*, then the table uses the names *CPU*, *CPUD*, and *CPUP*. Thus, all three of these names need to be avoided in choosing the name for a new table.

  For the reasons that are mentioned above, you might also want to avoid the name space of tables in other PDBs at your site and the name space of supplied tables.

  For more information about the table name with *D* and P appended, see "Overview of Tables" on page 353.

For user-defined variables in supplied tables or user-defined tables:

- A table must have a variable named DATETIME. For more information about the variable DATETIME, see "Variable Indicating Datetime, Day, Week, Month, Year" on page 350.
- A table must have a variable named DURATION, if the table type is *Interval*. DURATION is the length of the interval, in seconds.
- The name of a user-defined variable
  - must start with a letter.
  - has a maximum length of seven characters (letters and numbers only; do not use underscores).

    The name is not case sensitive; thus, NAME=var1 and NAME=VAR1 are equivalent.

    Note: Underscores are not allowed in names of user-defined variables; however, they might be added automatically, as padding (so that the statistic suffix is in the eighth character) in the variable names that are generated for you when you request predefined statistics at the day, week, month, and year levels on the user-defined variables. △

- must be unique within the table.

  In case you want to use the same variable in another table later, you might also want the name to be unique within all tables in the PDB.

  Similarly, in case you want to install the table in the master data dictionary later, you might also want the name not to match the name of any supplied variable.
- must not be COUNT or TYPE.

CHAPTER

18

Administration: Working with Variables

Working with Variables: Main Topics

Overview of Variables

Within a table and level, performance data exists as columns and rows. A variable is equivalent to a column. An observation is equivalent to a row.

There are three kinds of variables: regular, derived, and formula.

- *Regular variables* are variables whose values are obtained from the raw data (or calculated by the staging code; for example, the value of SHIFT is calculated by the staging code if the value is not provided in the raw data).
- *Derived variables* are variables whose values are calculated, during processing, from the values of constants, regular variables, and any derived variables that occur earlier in the ascending alphanumeric sequence of derived variables.
- *Formula variables* are variables whose values are calculated (when a view is requested) in ascending alphanumeric order from the values of regular variables, derived variables, constants, any formula variables that occur earlier in the ascending alphanumeric sequence of formula variables, and/or the _LEVEL_ variable.

At the detail level of the table in which the variable is located, typically there is one observation for each record in the raw data. In the detail-level observation,

- the values of regular variables are stored in the observation in the data set and accessible by means of the view of the data set.

  If the view is requested, no time is required to calculate the value. But in the meantime, space is required to store the value in the data set in case the view is requested.
- the values of derived variables are stored in the observations in the data set and accessible by means of the view of the data set.

  If the view is requested, no time is required to calculate the value. But in the meantime, space is required to store the value in the data set in case the view is requested.
- the values of formula variables are not stored (that is, are not physically present) in the observations in the data set, but are accessible (logically present) by means of the view of the data set.

  If the view is requested, run time is required to calculate the value. But in the meantime, space is not required to store the value in the data set in case the view is requested.

At the summary levels of the table in which the variable is located, there is one observation for each unique combination of class variables. For example, suppose the table's week level has DATETIME and HOUR as the class variables. Then, there is one observation for each hour of each week. On a week-level observation, as seen in the WEEK.*table-name* view,

- the values of class variables are available (regular and/or derived variables can be class variables). Also, the values of ID variables, if any, are available (regular and/or derived variables can be ID variables).

  The names of the class variables on the summary-level observation are the same as their names in the detail-level observations. Also, the names of the ID variables, if any, in the summary-level observation are the same as their names in the detail-level observations.

Thus, in the preceding example, a week-level observation has a value for DATETIME (whose contents at week level specify the week) and a value for HOUR (whose contents at week level specify the hour).

- the values of requested statistics for regular and/or derived variables are available.

  For example, suppose that R is a regular variable, D is a derived variable, and the sum, count, and average are to be calculated for each of them at month level. Then, in the week-level observation, as seen in the view, there is a value for R's sum statistic, a value for R's count statistic, a value for R's average statistic, a value for D's sum statistic, a value for D's count statistic, and a value for D's average statistic. Notice that averages are formulas (*average=sum/count*), so the view *does* have R's average statistic and D's average statistic, but the data set on which the view is based *does not* have R's average statistic and D's average statistic.

  The names of the statistics in a summary-level observation are constructed from the names in the detail-level observations. For example, R's sum statistic is called *R_____S*, R's count statistic is called *R_____N*, and R's average statistic is called *R*. D's sum statistic is called *D_____S*, D's count statistic is called *D_____N*, and D's average statistic is called *D*. (Note that the *_____A* is dropped from the names of the average statistics.) For more about the names of statistic variables, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

- the values of formula variables are available.

  For example, suppose that F is a formula variable that has been defined at week level as the ratio of the average statistic of R and the average statistic of D. That is, *F=R/D*. The week-level observation, as seen in the view, *does* contain the value of F, but the data set on which the view is based *does not* contain the value of F.

In addition to values, a variable also has properties (together the properties are called the variable's definition or the variable's attributes). The properties consist of data type (numeric or character), internal storage length, external presentation format, and so on. The variable's properties are stored in the data dictionary, along with the properties of the table in which the variable is located.

When you *add* a supplied table definition from the master data dictionary to the active PDB's data dictionary, the variable definitions for that table are also added. When you *create* a new table definition in the active PDB, you must also create its variable definitions, then or later. In both cases (added table definitions and created table definitions), you can edit the variable definitions, delete variable definitions, copy variable definitions, and so on, at a later time. Also, in both cases (added table definitions and created table definitions), you can edit the table definitions, delete table definitions, and so on, at a later time.

The following are additional topics (in alphabetical order) that are related to variables:

- "Copying a Regular Variable" on page 380
- "Creating/Editing/Viewing a Derived Variable" on page 381
- "Creating/Editing/Viewing a Formula Variable" on page 388
- "Creating/Editing/Viewing a Regular Variable" on page 394
- "Deleting a Derived Variable" on page 400
- "Deleting a Formula Variable" on page 401
- "Deleting a Regular Variable" on page 403
- "Exploring Supplied Variable Definitions" on page 404
- "Listing Derived Variables" on page 405
- "Listing Formula Variables" on page 406

□ “Listing Regular Variables” on page 406
□ “Viewing/Editing Defaults for Variable Definitions” on page 406.

Copying a Regular Variable

You can create a regular variable that has the same attributes as an existing variable in the master data dictionary and/or an existing variable in the active PDB

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE and CREATE VARIABLE control statements) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see “Chapter 4: Data Dictionary Macro and Control Statements” in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the LIKE= parameter on the %CPDDUTL CREATE VARIABLE control statement, see the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Copying a Regular Variable (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can copy the properties of a regular variable in order to create a new regular variable with the same properties (and same name) in the active table in the active PDB by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.
2. In the **`PDB Access Mode`** field, select **`Write`** if it is not already selected.
3. Select **`OK`**. SAS IT Resource Management returns you to the main menu.
4. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Config Active PDB Dictionary`**. The Dictionary Tables window opens.
5. Select the table in which you want to create the new variable. Then, select **`ItemActions`**. A menu opens. From the menu, select **`Edit Definition`**. The Edit Table window opens.
6. Select **`Variable List`**. The Variables window opens and displays a list of the variables in the table.
7. Select **`Actions`**. A menu opens. From the menu, select **`Copy Variables`**. The Copy Variables from Supplied Definitions window opens.
8. In the **`Collector`** field, leave the value at *All* or select the right arrow to specify the collector whose variable you want to copy.
9. In the **`Select Table(s)`** field, select the right arrow to specify the table that contains the variable that you want to copy.
10. In the **`Select Variable(s)`** field, select the variable that you want to copy and then select **`OK`**.
11. To return to the main menu, select **File ▶ End ▶ File ▶ End ▶ OK**

Copying a Regular Variable (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can copy the properties of a regular variable in order to create a new regular variable with the same

properties (and same name) in the active table in the active PDB by following these steps:

1. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2. Select **Properties**. The Set Active PDB Options window opens.
3. On the **Access** tab, select **WRITE** if it is not already selected.
4. Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.
5. Select **Close**. SAS IT Resource Management returns you to the main window.
6. On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens.
7. Select the name of the table in which you want to create the new regular variable. Then select **File** and, from the menu that opens, select **Properties**. The Table Properties window opens.
8. Select **Variables**. The View Variables window opens.
9. Select **Tools** and, from the menu that opens, select **Copy Variable**.
10. In the **Collector** field, specify whether you want to search for the variable (whose properties you want to copy) in the tables for a selected collector or in all tables.
11. In the **Scope** field, specify whether you want to search in the master data dictionary (**All Supplied**) or in the active PDB (**In active PDB**).
12. In the **Choose** field, select **Variables**, if it is not already selected.
13. Select **OK** and **Yes**. The Explore Tables/Variables window opens.
14. Select the variable whose properties you want to copy. Then select **Add** and **Close**. The View Variables window shows the new variable on its list of variables in the table.
15. To return to the main window, select **Close ▶ OK ▶ Close ▶ OK**

Creating/Editing/Viewing a Derived Variable

You can create, edit, and/or view a derived variable

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE control statement and the CREATE DERIVED, UPDATE DERIVED, and/or PRINT TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL CREATE DERIVED control statement, see the topic CREATE DERIVED in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL UPDATE DERIVED control statement, see the topic UPDATE DERIVED in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL PRINT TABLE control statement, see the topic PRINT TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Creating/Editing/Viewing a Derived Variable (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can create, edit, and/or view a derived variable in a table in the active PDB by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2. In the **PDB Access Mode** field, select **Write** if it is not already selected and you want to create or edit a derived variable.
3. Select **OK**. SAS IT Resource Management returns you to the main menu.
4. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens and displays a list of the tables in the active PDB.
5. Select the name of the table in which you want to create, update, or view a derived variable.
6. Select **ItemActions**. A menu opens. From the menu, select
   - **Edit Definition**, if you want to create or edit a derived variable. The Edit Table window opens.
   - **Browse Definition**, if you want to view a derived variable. The Browse Table window opens.
7. Select **Variable List**. The Variables or Browse Variables window opens and lists the existing variables in the table.
8. Choose one of the following steps, depending on your purpose:
   - If you want to *create* a derived variable, select **Actions**. A menu opens. From the menu, select **Create Derived Variable**. The Create Derived window opens. In the **New Variable** field, type the name that you want the new derived variable to have. (For information about variable-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for z/OS Tables and Variables" on page 371.) Select **OK**. The Create Derived Variable window opens.
   - If you want to *edit* a derived variable, select the variable that you want to edit. Then select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Derived Variable window opens.
   - If you want to *view* an existing derived variable, select the variable that you want to view. Then select **ItemActions**. A menu opens. From the menu, select **Browse Definition**. The Browse Variable window opens.

   Note: There are only two differences in the content and functionality of the Create Derived Variable window and the Edit Derived Variable window:
   - In the Create Derived Variable window, the values are the defaults. In the Edit Derived Variable window, the values are the ones that were specified when the variable was created or most recently edited.
   - In the Create Derived Variable window, the name of the variable and the data type (value of the **Stored As** field) of the variable can be changed. In the Edit Derived Variable window, the name and data type of the variable cannot be changed.

   △

   The Browse Variable window has the same information as the Edit Derived Variable window, but does not have write access.
9. The value of the **Keep in PDB** field specifies whether the variable's definition is used or ignored by SAS IT Resource Management.

If the value is *Yes*, the variable's definition will be saved (when you select **OK**) and used. If the value is *No*, the variable's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

To specify the other value, select its radio button.

Note: If you have no current use for the variable, but do anticipate future use, you might want to set the variable's Kept status to *No*. When the variable's Kept status is set to *No*, the variable's definition is retained but ignored, and the variable's values, if any, are discarded. (There will be missing values in all existing observations for the variables that had their Kept status changed from *Yes* to *No*.) △

10 The value of the **Stored as** field represents the variable's data type.

If you are creating the variable's definition, you can specify the other data type by selecting it. If you are editing the variable's definition, you cannot change the data type.

11 The value of the **Interpretation** field represents a "fine-tuning" of the data type.

To use a different interpretation type, select **Interpretation**. SAS IT Resource Management displays a list of the interpretation types that are consistent with the data type. Use the scroll bar on the right of the list to scroll to the interpretation type that you want to use, and select that interpretation type. SAS IT Resource Management returns you to the variable's main definition window.

For more about interpretation type, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

12 The setting of the **Length** field represents the maximum length in bytes that is required to store a value of the variable.

If you want to specify a different length, overwrite the length.

13 The setting of the **Format** field represents the format that is to be used to display a value of the variable.

If you want to use a different format, overwrite the format with any valid SAS format that is appropriate for the data type.

For more information about formats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14 .

14 The setting of the **Informat** field represents the informat that is to be used to acquire a value of the variable.

If you want to use a different informat, overwrite the informat with any valid SAS informat that is appropriate for the data type.

For more information about informats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

15 The value of the **Label** field is a brief description of the variable. The label is used in reports to identify the variable.

If you want to use a different label, overwrite the label.

16 The value of the **Description** field is a long description of the variable. The description is used within the SAS IT Resource Management GUI and underlying data dictionary to provide a more complete description of the variable.

If you want to use a different description, overwrite the description.

17 The **Subject** field is not used.

18 The value of the **Source Code** field is the set of SAS DATA step statements that define how the value of the derived variable is to be built from constants, regular variables, and/or derived variables that are earlier in the alphanumeric order of derived variables.

Select **Source Code**. SAS IT Resource Management opens the BUILD window. To create or edit the source code, type or edit one or more SAS DATA step statements. (For more information about SAS statements, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.) For example, if the derived variable C is to be the sum of the regular variables A and B, then you could write a statement like this one:

```
C = SUM(A,B) ;
```

When you finish creating, editing, or viewing the statement(s), select **File**. A menu opens. From the menu, select **Close**. (You can do a separate save, but the close also does a save.) SAS IT Resource Management saves the statements and then returns you to the variable's main definition window.

Note: For more information about how to define a derived variable, see "How to Define Derived and Formula Variables" on page 407. △

19 If the variable's data type is numeric, the value of the **Weight Variable** field displays the name of the alternate weight variable. Specifying an alternate weight variable is optional. The regular weight variable is DURATION for tables of type Interval; there is no regular weight variable for tables of type Event. If the name of the alternate weight variable is specified, the alternate weight variable overrides the regular weight variable.

To specify an alternate weight variable, select **Weight Variable**. SAS IT Resource Management displays a list of the other variables in the table that have a data type of Numeric. Page down to the one that you want to use and select it. Then select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

If you later decide not to specify an alternate weight variable, select **Weight Variable** and then **OK**. By not specifying a choice, you indicate to SAS IT Resource Management that you want to use the default weight variable (DURATION). SAS IT Resource Management returns you to the variable's main definition window and displays -NONE- to the right of the button.

20 If the variable's data type is Numeric, select **Statistics**. The Edit Variable Statistics window opens and displays the status of the "automatic" statistics (that is, statistics whose calculation mechanism is already defined by SAS IT Resource Management) at day, week, month, and year levels.

An intersection of a statistic row and level column represents that statistic for that level. If you see an X at an intersection, that statistic *is* selected at that level (that is, that statistic is requested at that level, or to say it still another way, the calculation status of that statistic at that level is *Yes*). If you see an empty box at an intersection, that statistic *is not* selected at that level.

The statistics that are selected by default are a set that is based on the data type and interpretation type of the variable and the table type of the table. If you want to change the status of a statistic for a given level, click on the intersection of the statistic's row and the level's column.

Some selections imply others. For example, if you select Average, then Count and Sum are (implicitly) selected. Thus, if you select Average, you do not need to select Count and Sum unless you want to display their values in the view. (For more information about the implied selections, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.)

When you finish selecting the statistics that you want, select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

21 (The specification of age limits for the variable's data and the use of the variable in BY, CLASS, ID, and/or Index variables lists are parts of the table's definition,

not parts of the variable's definition. For more information, see "Overview of Levels" on page 337.)

22 Select **OK ▶ File ▶ End ▶ OK ▶ File ▶ End ▶ OK**

Creating/Editing/Viewing a Derived Variable (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create, edit, and/or view a derived variable in a table in the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

2 Select **Properties**. The Set Active PDB Options window opens.

3 On the **Access** tab, select **WRITE** if it is not already selected and if you want to create or edit a derived variable.

4 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.

5 Select **Close**. SAS IT Resource Management returns you to the main window.

6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.

7 Select the name of the table in which you want to create, update, or view a derived variable.

8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

9 On the **General** tab, select **Variables**. The View Variables window opens. The window lists the variables in the table.

10 Choose one of the following steps, depending on your purpose:

- If you want to *create* a derived variable, select **Tools**. A menu opens. From the menu, select **Create Derived Variable**. The Create New Derived Variable window opens. Type the name that you want to give to the new derived variable. (For information about variable-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for UNIX and Windows Tables and Variables" on page 374.) Select **OK**. The Variable Properties window opens.
- If you want to *edit* an existing derived variable, select the variable that you want to edit. Then select **File**. A menu opens. From the menu, select **Properties**. The Variable Properties window opens.
- If you want to *view* an existing derived variable, select the variable that you want to view. Then select **File**. A menu opens. From the menu, select **Properties**. The Variable Properties window opens.

Note: There are only two differences in the content and functionality of the Variable Properties window between create and edit/view:

- For create, the values are the defaults. For edit/view, the values are the ones that were specified when the variable was created or most recently edited.
- For create, the name of the variable and the data type (value of the **Stored As** field) of the variable can be changed. For edit/view, the name and data type of the variable cannot be changed.

△

11 Select the **General** tab if it is not already selected.

12 The value of the **Kept in PDB** field specifies whether the variable's definition is to be used or ignored by SAS IT Resource Management.

If the value is *Yes*, the variable's definition will be saved (when you select **OK**) and used. If the value is *No*, the variable's definition will be saved (when you

select **OK**) and ignored (except for being available for viewing and, optionally, for using later). Also, if the Kept status was *Yes* and changes to *No*, any values of the variable are discarded. (There will be missing values in all existing observations for the variables that had their Kept status changed from *Yes* to *No*.)

To specify the other value, select its radio button.

13 The value of the **Stored as** field represents the variable's data type.

If you are creating the variable, you can use the other data type by selecting it. If you are editing the variable, you cannot change the data type.

14 The value of the **Interpretation** field represents a "fine-tuning" of the data type.

To use a different interpretation type, select the arrow to the right of the **Interpretation** field. SAS IT Resource Management displays a list of the interpretation types that are consistent with the data type. Select the interpretation type that you want to use, and then select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

For more about interpretation type, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

15 The setting of the **Length** field represents the maximum length in bytes that is required to store a value of the variable.

If you want to specify a different length, overwrite the length.

16 Select the **Advanced** tab.

17 The setting of the **Format** field represents the format that is to be used to display a value of the variable.

If you want to use a different format, select the arrow to the right of the **Format** field. SAS IT Resource Management lists formats that are consistent with the data type. If one of those formats is what you want or similar to what you want, select it and select **OK**; otherwise, select **Cancel**. If necessary, overwrite the value with any valid SAS format that is appropriate for the data type.

For more information about formats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

18 The setting of the **Informat** field represents the information that is to be used to acquire a value of the variable.

If you want to use a different informat, select the arrow that is to the right of the **Informat** field. SAS IT Resource Management lists informats that are consistent with the data type. If one of those informats is what you want or similar to what you want, select it and select **OK**; otherwise, select **Cancel**. If necessary, overwrite the value with any valid SAS informat that is appropriate for the data type.

For more information about informats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14

19 Select the **General** tab.

20 The value of the **Label** field is a brief description of the variable. The label is used in reports to identify the variable.

If you want to use a different label, overwrite the label.

21 The value of the **Description** field is a long description of the variable. The description is used within the SAS IT Resource Management GUI and underlying data dictionary to provide a more complete description of the variable.

If you want to use a different description, overwrite the description.

22 Select the **Advanced** tab.

23 The **Subject** field is not used.

24 The value of the **Source Code** field is the set of SAS DATA step statements that define how the value of the derived variable is to be built from the values of one or

more constants, regular variables and/or derived variables that are earlier in the alphanumeric order of derived variables.

If you want to create or edit the code, type or edit one or more SAS DATA step statements. (For more information about SAS statements, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.) For example, if the derived variable C is to be the sum of the regular variables A and B, then you could write a statement like this one:

```
C = SUM (A,B);
```

Note: For more information about how to define a derived variable, see "How to Define Derived and Formula Variables" on page 407. △

25 Select **Check Code**. If the message

```
NOTE: Formula has no syntax errors
```

is displayed in the SAS message area at the bottom of the SAS IT Resource Management window, go to the next field. Otherwise, adjust the statement(s) until it passes the code check.

26 If the variable's data type is numeric, the value of the **Weight By** field displays the name of the alternate weight variable. Specifying an alternate weight variable is optional. The regular weight variable is DURATION for tables of type Interval; there is no regular weight variable for tables of type Event. The regular weight variable is overridden by the alternate weight variable, if any.

To specify an alternate weight variable, select the arrow to the right of the **Weight By** field. SAS IT Resource Management displays a list of the other variables in the table that have a data type of Numeric. Select the one that you want to use. Then select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

If you later decide not to specify an alternate weight variable, delete the characters in the **Weight By** field.

27 If the variable's data type is Numeric, select the **Statistics** tab. The tab displays the status of the "automatic" statistics (that is, statistics whose calculation mechanism is already defined by SAS IT Resource Management) at day, week, month, and year levels.

An intersection of a statistic row and level column represents that statistic for that level. If you see a check in the box at an intersection, that statistic *is* selected at that level (that is, that statistic is requested at that level, or in other words, the calculation status of that statistic at that level is *Yes*). If you see an empty box at an intersection, that statistic *is not* selected at that level.

The statistics that are selected by default are a set that is based on the data type and interpretation type of the variable and the table type of the table. If you want to change the status of a statistic for a given level, click on the intersection of the statistic's row and the level's column.

Some selections imply others. For example, if you select Average, then Count and Sum are (implicitly) selected. Thus, if you select Average, you do not need to select Count and Sum unless you want to display their values in the view. (For more information about the implied selections and about the suffix that identifies every statistic, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.)

28 (The specification of age limits for the variable's data and the use of the variable in BY, CLASS, ID, and/or Index variables lists are parts of the table's definition, not parts of the variable's definition. For more information, see "Overview of Levels" on page 337.)

29 Select **OK ▶ Close ▶ OK ▶ Close ▶ OK**

Creating/Editing/Viewing a Formula Variable

You can create, edit, and/or view a formula variable

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE control statement and the CREATE FORMULA, UPDATE FORMULA, and/or PRINT TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL control statement SET TABLE, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL CREATE FORMULA control statement, see the topic CREATE FORMULA in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL UPDATE FORMULA control statement, see the topic UPDATE FORMULA in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL PRINT TABLE control statement, see the topic PRINT TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: If the table that you are working with has DATETIME but does not have DATE and TIME, you can add DATE and TIME as formula variables if you think that they would be useful in reporting.

- The properties for DATE are: Level (*Detail*, *Day*, *Week*, *Month*, *Year*), Length (*6*), Type (*Numeric*), Kept (*Yes*), Label (*Date*), Description (*Date*), Formula (*date=datepart(datetime);*), Format (*DATE9.*).
- The properties for TIME are: Level (*Detail*, *Day*, *Week*, *Month*, *Year*), Length (*6*), Type (*Numeric*), Kept (*Yes*), Label (*Time*), Description (*Time*), Formula (*time=timepart(datetime);*), Format (*TIME.*).

△

Creating/Editing/Viewing a Formula Variable (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can create, edit, and/or view the definition of a formula variable in a table in the active PDB by following these steps:

1. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.
2. In the **`PDB Access Mode`** field, select **`Write`** if it is not already selected and you want to create or update a formula variable.
3. Select **`OK`**. SAS IT Resource Management returns you to the main menu.
4. From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Config Active PDB Dictionary`**. The Dictionary Tables window opens. The window displays a list of the tables in the active PDB.
5. Select the name of the table in which you want to create, update, or view a formula variable.
6. Select **`ItemActions`**. A menu opens. From the menu, select

- **Edit Definition**, if you want to create or edit a formula variable. The Edit Table window opens.
- **Browse Definition**, if you want to view a formula variable. The Browse Table window opens.

7 Select **Variable List**. The Variables window opens and lists the existing variables in the table.

8 Choose one of the following steps, depending on your purpose:

- If you want to *create* a formula variable, select **Actions**. A menu opens. From the menu, select **Create Formula Variable**. The Create Formula window opens. In the **New Variable** field, type the name that you want to give to the new derived variable. (For information about variable-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for z/OS Tables and Variables" on page 371.) Select **OK**. The Make Formula Variable window opens.
- If you want to *edit* an existing formula variable, select the variable that you want to edit. Then select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Formula Variable window opens.
- If you want to *view* an existing formula variable, select the variable that you want to view. Then select **ItemActions**. A menu opens. From the menu, select **Browse Definition**. The Browse Formula Variable window opens.

Note: There are only two differences in the content and functionality of the Make Formula Variable window and the Edit Formula Variable window:

- In the Make Formula Variable window, the values are the defaults. In the Edit Formula Variable window, the values are the ones that were specified when the variable was created or most recently edited.
- In the Make Formula Variable window, the name of the variable and the data type (value of the **Type** field) can be changed. In the Edit Formula Variable window, the name and data type of the variable cannot be changed.

△

The browse window has the same information as the edit window, but does not have write access.

9 The value of the **Keep in PDB** field specifies whether the variable's definition is used or ignored by SAS IT Resource Management.

If the value is *Yes*, the variable's definition will be saved (when you select **OK**) and used. If the value is *No*, the variable's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

To specify the other value, select its radio button.

Note: If you have no current use for the variable, but do anticipate future use, you might want to set the variable's Kept status to *No*. When the variable's Kept status is set to *No*, the variable's definition is retained but ignored, and the variable's values, if any, are discarded. (There will be missing values in all existing observations for the variables that had their Kept status changed from *Yes* to *No*.) △

10 The value of the **Type** field represents the variable's data type.

If you are creating the variable, you can use the other data type by selecting it.

11 There is no Interpretation field for formula variables. As a result, double-check whether the values in the **Length** field and **Format** field are appropriate for the variable.

12 The setting of the **Length** field represents the maximum length in bytes that is required to store a value of the variable.

If you want to specify a different length, overwrite the length.

13 The setting of the **Format** field represents the format that is to be used to display a value of the variable.

If you want to use a different format, overwrite the format with any valid SAS format that is appropriate for the data type. For more information about formats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

14 The value of the **Label** field is a brief description of the variable. The label is used in reports to identify the variable.

If you want to use a different label, overwrite the label.

15 The value of the **Description** field is a long description of the variable. The description is used within the SAS IT Resource Management GUI and underlying data dictionary to provide a more complete description of the variable.

If you want to use a different description, overwrite the description.

16 The **Subject** field is not used.

17 The value of the **Edit Formula Source Code** field is the set of SAS DATA step statements that define how the value of the formula variable is to be built from the values of one or more regular variables, derived variables, constants, formula variables earlier in the alphanumeric order of formula variables, and/or the _LEVEL_ variable.

Note: The _LEVEL_ variable exists automatically in the context of a formula variable, but it does not appear on the list of variables. You can test the _LEVEL_ variable to provide specific formulas for specific levels. The valid values for the _LEVEL_ variable are *DETAIL*, *DAY*, *WEEK*, *MONTH*, and *YEAR*.

For example, you can use a statement similar to this in your source code for the formula:

```
IF _LEVEL_ = 'MONTH' ... ;
```

△

Select **Edit Formula Source Code** or **Browse Formula Source Code**. SAS IT Resource Management opens the BUILD window. To create or edit the source code, type or edit one or more SAS DATA step statements. (For more information about SAS statements, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.) For example, if the formula variable C is to be the sum of a regular variable A and a derived variable B, then you could write a statement like this:

```
C = SUM(A,B) ;
```

When you finish creating, editing, or viewing the statement(s), select **File**. A menu opens. From the menu, select **Close**. (You can save separately, but when you close, the work that you did is saved automatically.) SAS IT Resource Management saves the statements and then returns you to the variable's main definition window.

If you created or edited the formula variable, select **Test Syntax**. If the message "NOTE: Formula has no syntax errors." is displayed at the top of the window, go to the next field; otherwise, edit the source code statements so they are acceptable to the syntax tester.

Note: For more information about how to define a formula variable, see "How to Define Derived and Formula Variables" on page 407. △

18 The **Apply to Level** field specifies the level or levels at which the values of the formula are to be available (by calculation, if requested by a report). An X preceding a level indicates that values *are* to be available at that level (in other words, the calculation status at that level is *Yes*). An empty box preceding a level indicates that values *are not* to be calculated at that level.

Note that the values of the formula variable at the detail, day, week, month, or year level are based on the values of its underlying variables *at that same level*. For example, if the formula variable C is to be the sum of the regular variables A and B, then you could write a statement like this one:

```
C = SUM(A,B) ;
```

In this case, here are examples of the meaning of the formula:

- at detail level, A is the name of the value of variable A on the detail-level observation, B is the name of the value of variable B on that same detail-level observation, and thus C for that observation is the sum of the value of variable A on that observation and the value of variable B on that observation.
- at day level, A is the name of the average statistic of variable A on the day-level observation, B is the name of the average statistic of variable B on that day-level observation, and C for that observation is the sum of the average statistic of variable A and the average statistic of variable B on that observation.

For more information about the suffix that identifies each statistic, see "Appendix 5: Variable Types" in the SAS IT Resource Management macro reference documentation.

19 Select **OK ▶ File ▶ End ▶ OK ▶ File ▶ End ▶ OK**

Creating/Editing/Viewing a Formula Variable (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create, edit, and/or view the definition of a formula variable in a table in the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

2 Select **Properties**. The Set Active PDB Options window opens.

3 On the **Access** tab, select **WRITE** if it is not already selected and if you want to create or update a formula variable.

4 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.

5 Select **Close**. SAS IT Resource Management returns you to the main window.

6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.

7 Select the name of the table in which you want to create, update, or view a formula variable.

8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

9 On the **General** tab, select **Variables**. The View Variables window opens. The window lists the variables in the table.

10 Choose one of the following steps, depending on your purpose:

- If you want to *create* a formula variable, select **Tools**. A menu opens. From the menu, select **Create Formula Variable**. The Create New Formula Variable window opens. Type the name that you want to give to the new

formula variable. (For information about variable-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for UNIX and Windows Tables and Variables" on page 374.) Select **OK**. The Make/Edit Formula window opens.

- If you want to *edit* an existing formula variable, select the variable that you want to edit. Then select **File**. A menu opens. From the menu, select **Properties**. The Make/Edit Formula window opens.
- If you want to *view* an existing formula variable, select the variable that you want to view. Then select **File**. A menu opens. From the menu, select **Properties**. The Make/Edit Formula window opens.

Note: There are only two differences in the content and functionality between create and edit/view:

- For create, the values are the defaults. For edit/view, the values are the ones that were specified when the variable was created or most recently edited.
- For create, the name of the variable and the data type (value of the **Type** field) of the variable can be changed. For edit/view, the name and data type of the variable cannot be changed.

△

11 Select the **General** tab if it is not already selected.

12 The value of the **Kept in PDB** field specifies whether the variable's definition is to be used or ignored by SAS IT Resource Management.

If the value is *Yes*, the variable's definition will be saved (when you select **OK**) and used (that is, values will be calculated when requested). If the value is *No*, the variable's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

To specify the other value, select its radio button.

13 The value of the **Type** field represents the variable's data type.

If you are creating the variable, you can use the other data type by selecting it. If you are editing the variable, you cannot change the data type.

14 There is no **Interpretation** field for formula variables. (The interpretation type for formula variables is *Formula*.) As a result, double-check whether the values in the **Length** field and **Format** field are appropriate for the variable.

15 The setting of the **Length** field represents the maximum length in bytes that is required to store a value of the variable.

If you want to specify a different length, overwrite the length.

16 Select the **Advanced** tab.

17 The setting of the **Format** field represents the format that is to be used to display a value of the variable.

If you want to use a different format, select the arrow to the right of the **Format** field. SAS IT Resource Management lists some formats that are appropriate for the data type. If one of those formats is what you want or similar to what you want, select it and select **OK**; otherwise, select **Cancel**. If necessary, overwrite the format with any valid SAS format that is appropriate for the data type.

For more information about formats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

18 Select the **General** tab.

19 The value of the **Label** field is a brief description of the variable. The label is used in reports to identify the variable.

If you want to use a different label, overwrite the label.

20 The value of the **Description** field is a long description of the variable. The description is used within the SAS IT Resource Management GUI and underlying data dictionary to provide a more complete description of the variable.

If you want to use a different description, overwrite the description.

21 Select the **Advanced** tab.

22 The **Subject** field is not used.

23 Select the **Formula Source** tab.

24 The value of the field on the tab is the set of SAS DATA step statements that define how the value of the formula variable is to be built from the values of one or more regular variables, derived variables, constants, formula variables earlier in the alphanumeric order of formula variables, and/or the _LEVEL_ variable.

Note: The _LEVEL_ variable exists automatically in the context of a formula variable, but it does not appear in the list of variables. You can test the _LEVEL_ variable to provide specific formulas for specific levels. The valid values for the _LEVEL_ variable are *DETAIL*, *DAY*, *WEEK*, *MONTH*, and *YEAR*.

For example, you can use a statement similar to this in your source code for the formula:

```
IF _LEVEL_ = 'MONTH' ... ;
```

△

Type or edit one or more SAS DATA step statements. (For more information about SAS statements, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.) For example, if the formula variable C is to be the sum of a regular variable A and a derived variable B, then you could write a statement like this one:

```
C = SUM(A,B) ;
```

Note: For more information about how to define a formula variable, see "How to Define Derived and Formula Variables" on page 407. △

25 Select the **Check Code** button. If the SAS message area (at the bottom of the SAS IT Resource Management window) displays the message

```
NOTE: Formula has no syntax errors.
```

then go on to the next field; otherwise, adjust the formula variable's statements until they pass the code check.

26 Select the **General** tab.

27 The **Apply to Level** field indicates the level or levels at which the values of the formula are to be available (by calculation, if requested by a report). A box with a check mark preceding a level indicates that values *are* to be available at that level (in other words, the calculation status at that level is *Yes*). An empty box preceding a level indicates that values *are not* to be calculated at that level.

Note that the values of the formula variable at the detail, day, week, month, or year level are based on the values of its underlying variables *at that same level*. For example, if the formula variable C is to be the sum of the regular variables A and B, then you could write a statement like this one:

```
C = SUM(A,B) ;
```

In this case, here are examples of the meaning of the formula:

- at detail level, A is the name of the value of variable A on the detail-level observation, B is the name of the value of variable B on that same detail-level observation, and thus C for that observation is the sum of the

value of variable A on that observation and the value of variable B on that observation.

- at day level, A is the name of the average statistic of variable A on the day-level observation, B is the name of the average statistic of variable B on that day-level observation, and C for that observation is the sum of the average statistic of variable A and the average statistic of variable B on that observation.

For more information about the suffix that identifies each statistic, see "Appendix 5: Variable Types" in the SAS IT Resource Management macro reference documentation.

28 Select **OK ▶ Close ▶ OK ▶ Close ▶ OK**

Creating/Editing/Viewing a Regular Variable

You can create, edit, and/or view the definition of a regular variable

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE control statement and the CREATE VARIABLE, UPDATE VARIABLE, and/or PRINT TABLE control statement) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL CREATE VARIABLE control statement, see the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL UPDATE VARIABLE control statement, see the topic UPDATE VARIABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL PRINT TABLE control statement, see the topic PRINT TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Creating/Editing/Viewing a Regular Variable (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can create, edit, and/or view the definition of a regular variable in a table in the active PDB by following these steps:

1 From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.

2 In the **`PDB Access Mode`** field, select **`Write`** if it is not already selected and if you want to create or update a regular variable.

3 Select **`OK`**. SAS IT Resource Management returns you to the main menu.

4 From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Config Active PDB Dictionary`**. The Dictionary Tables window opens. The window displays a list of the tables in the active PDB.

5 Select the name of the table in which you want to create, update, or view a regular variable.

6 Select **`ItemActions`**. A menu opens. From the menu, select

- **Edit Definition**, if you want to create or edit a regular variable. The Edit Table window opens.
- **Browse Definition**, if you want to view a regular variable. The Browse Table window opens.

7 Select **Variable List**. The Variables window opens. The window lists the existing variables in the table.

8 Choose one of the following steps, depending on your purpose:

- If you want to *create* a regular variable, select **Actions**. A menu opens. From the menu, select **Create Variable**. The preliminary Create Variable window opens. In the **New Variable** field, type the name that you want to give to the new regular variable. (For information about variable-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for z/OS Tables and Variables" on page 371.) Select **OK**. The Create Variable window opens.
- If you want to *edit* an existing regular variable, select the variable that you want to edit. Then select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Variable window opens.
- If you want to *view* an existing regular variable, select the variable that you want to view. Then select **ItemActions**. A menu opens. From the menu, select **Browse Definition**. The Browse Variable window opens.

Note: There are only two differences in the content and functionality of the Create Variable window and the Edit Variable window:

- In the Create Variable window, the values are the defaults. In the Edit Variable window, the values are the ones that were specified when the variable was created or most recently edited.
- In the Create Variable window, the name of the variable and the data type (value of the **Stored As** field) of the variable can be changed. In the Edit Variable window, the name and data type of the variable cannot be changed.

△

The Browse Variable window has the same information as the Edit Variable window, but it does not have write access.

9 The value of the **Keep in PDB** field specifies whether the variable's definition is used or ignored by SAS IT Resource Management.

If the value is *Yes*, the variable's definition will be saved (when you select **OK**) and used. If the value is *No*, the variable's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later).

To specify the other value, select its radio button.

10 The value of the **Stored as** field represents the variable's data type.

If you are creating the variable, you can use the other data type by selecting it. If you are editing the variable, you cannot change the data type.

11 The value of the **Interpretation** field represents the "fine-tuning" of the data type.

To use a different interpretation type, select **Interpretation**. SAS IT Resource Management displays a list of the interpretation types that are consistent with the data type. Use the scroll bar to the right of the list to scroll to the interpretation type that you want to use, and select that interpretation type. SAS IT Resource Management returns you to the variable's main definition window.

For more about interpretation type, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

12 The setting of the **Length** field represents the maximum length in bytes that is required to store a value of the variable.

If you want to specify a different length, overwrite the length.

13 The setting of the **Format** field represents the format that is to be used to display a value of the variable.

If you want to use a different format, overwrite the value with any valid SAS format that is appropriate for the data type.

For more information about formats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

14 The setting of the **Informat** field represents the informat that is to be used to acquire a value of the variable.

If you want to use a different informat, overwrite the informat with any valid SAS informat that is appropriate for the data type.

For more information about informats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

15 The value of the **Label** field is a brief description of the variable. The label is used in reports to identify the variable.

If you want to use a different label, overwrite the label.

16 The value of the **Description** field is a long description of the variable. The description is used within the SAS IT Resource Management GUI and underlying data dictionary to provide a more complete description of the variable.

If you want to use a different description, overwrite the description.

17 The **Subject** field is not used.

18 If the variable's data type is numeric, the value of the **Weight Variable** field displays the name of the alternate weight variable. Specifying an alternate weight variable is optional. The regular weight variable is DURATION for tables of type Interval; there is no regular weight variable for tables of type Event. If the name of the alternate weight variable is specified, the alternate weight variable overrides the regular weight variable.

To specify an alternate weight variable, select **Weight Variable**. SAS IT Resource Management displays a list of the other variables in the table that have a data type of Numeric. Page down to the one that you want to use and select it. Then select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

If you later decide not to specify an alternate weight variable, select **Weight Variable** and then **OK**. By not specifying a choice, you indicate to SAS IT Resource Management that you want to use the default weight variable (DURATION). SAS IT Resource Management returns you to the variable's main definition window and displays -NONE- to the right of the button.

19 If the variable's data type is numeric, select **Statistics**. The Edit Variable Statistics window opens and displays the status of the "automatic" statistics (that is, statistics whose calculation mechanism is already defined by SAS IT Resource Management) at day, week, month, and year levels.

An intersection of a statistic row and level column represents that statistic for that level. If you see an X at an intersection, that statistic *is* selected at that level (that is, that statistic is requested at that level, or in other words, the calculation status of that statistic at that level is *Yes*). If you see an empty box at an intersection, that statistic *is not* selected at that level.

The statistics that are selected by default are a set that is based on the data type and interpretation type of the variable and the table type of the table. If you

want to change the status of a statistic for a given level, click on the intersection of the statistic's row and the level's column.

Some selections imply others. For example, if you select Average, then Count and Sum are (implicitly) selected. Thus, in this case you do not need to select Count and Sum unless you want to display their values in the view. (For more information about the implied selections, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.)

When you finish selecting the statistics that you want, select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

20 (The specification of age limits for the variable's data and the use of the variable in BY, CLASS, ID, and/or Index variables lists are parts of the table's definition, not parts of the variable's definition. For more information, see "Overview of Levels" on page 337.)

21 Select **OK ▶ File ▶ End ▶ OK ▶ File ▶ End ▶ OK**

Creating/Editing/Viewing a Regular Variable (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create, edit, and/or view the definition of a regular variable in a table in the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

2 Select **Properties**. The Set Active PDB Options window opens.

3 On the **Access** tab, select **WRITE** if it is not already selected and if you want to create or update a regular variable.

4 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.

5 Select **Close**. SAS IT Resource Management returns you to the main window.

6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.

7 Select the name of the table in which you want to create, update, or view a regular variable.

8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

9 Select **Variables**. The View Variables window opens. The window lists the variables in the table.

10 Choose one of the following steps, depending on your purpose:

- If you want to *create* a regular variable, select **Tools**. A menu opens. From the menu, select **Create Variable**. The Create New Variable window opens. Type the name that you want to give to the new regular variable. (For information about variable-naming conventions, see "Naming Conventions for User-Defined Tables and Variables" in "Naming Conventions for UNIX and Windows Tables and Variables" on page 374.) Select **OK**. The Variable Properties window opens.
- If you want to *edit* an existing regular variable, select the variable that you want to edit. Then select **File**. A menu opens. From the menu, select **Properties**. The Variable Properties window opens.
- If you want to *view* an existing regular variable, select the variable that you want to view. Then select **File**. A menu opens. From the menu, select **Properties**. The Variable Properties window opens.

Note: There are only two differences in the content and functionality between create and edit/view:

- For create, the values are the defaults. For edit/view, the values are the ones that were specified when the variable was created or most recently edited.
- For create, the name of the variable and the data type (value of the **Stored As** field) of the variable can be changed. For edit/view, the name and data type of the variable cannot be changed.

△

11 Select the **General** tab if it is not already selected.

12 The value of the **Kept in PDB** field specifies whether the variable's definition is to be used or ignored by SAS IT Resource Management.

If the value is *Yes*, the variable's definition will be saved (when you select **OK**) and used. If the value is *No*, the variable's definition will be saved (when you select **OK**) and ignored (except for being available for viewing and, optionally, for using later). Also, if the Kept status was *Yes* and changes to *No*, any values of the variable are discarded. (There will be missing values in all existing observations for the variables that had their Kept status changed from *Yes* to *No*.)

To specify the other value, select its radio button.

Note: If you have no current use for the variable, but do anticipate future use, you might want to set the variable's Kept status to *No*. When the variable's Kept status is set to *No*, the variable's definition is retained but ignored, and the variable's values, if any, are discarded. (There will be missing values in all existing observations for the variables that had their Kept status changed from *Yes* to *No*.) △

13 The **External Var Name** field represents the name of the variable (column) in the staged data.

14 The value of the **Stored as** field represents the variable's data type.

If you are creating the variable, you can use the other data type by selecting it. If you are editing the variable, you cannot change the data type.

15 The value of the **Interpretation** field represents the "fine-tuning" of the data type.

To use a different interpretation type, select the arrow to the right of the **Interpretation** field. SAS IT Resource Management displays a list of the interpretation types that are consistent with the data type. Select the interpretation type that you want to use, and then select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

For more about interpretation type, see "Appendix 5: Variable Interpretation Types" in the SAS IT Resource Management macro reference documentation.

16 The setting of the **Length** field represents the maximum length in bytes that is required to store a value of the variable.

If you want to specify a different length, overwrite the length.

17 Select the **Advanced** tab.

18 The setting of the **Format** field represents the format that is to be used to display a value of the variable.

If you want to use a different format, select the arrow to the right of the **Format** field. SAS IT Resource Management lists some formats that are appropriate for the data type. If one of those formats is what you want or similar to what you want, select it and select **OK**; otherwise, select **Cancel**. If necessary, overwrite the value with any valid SAS format that is appropriate for the data type.

For more information about formats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

19 The setting of the **Informat** field represents the information that is to be used to acquire a value of the variable.

If you want to use a different informat, select the arrow to the right of the **Informat** field. SAS IT Resource Management lists informats that are consistent with the data type. If one of those informats is what you want or similar to what you want, select it and select **OK**; otherwise, select **Cancel**. If necessary, overwrite the value with any valid SAS informat that is appropriate for the data type.

For more information about informats, which are a feature of the SAS language, see "Additional SAS Documentation" on page 14.

20 Select the **General** tab.

21 The value of the **Label** field is a brief description of the variable. The label is used in reports to identify the variable.

If you want to use a different label, overwrite the label.

22 The value of the **Description** field is a long description of the variable. The description is used within the SAS IT Resource Management GUI and underlying data dictionary to provide a more complete description of the variable.

If you want to use a different description, overwrite the description.

23 Select the **Advanced** tab.

24 The **Varidnum** field and **Varoid** field are collector-specific and are used by only a few collectors. To see if these fields are used for the collector that collected the data for this variable, see "Collector-Specific Setup Documentation" on page 12.

25 The **Subject** field is not used.

26 If the variable's data type is Numeric, the value of the **Weight By** field displays the name of the alternate weight variable. Specifying an alternate weight variable is optional. The regular weight variable is DURATION for tables of type Interval; there is no regular weight variable for tables of type Event. The regular weight variable is overridden by the alternate weight variable, if any.

To specify an alternate weight variable, select the arrow to the right of the **Weight By** field. SAS IT Resource Management displays a list of the other variables in the table that have a data type of Numeric. Select the one that you want to use. Then select **OK**. SAS IT Resource Management returns you to the variable's main definition window.

If you later decide not to specify an alternate weight variable, overwrite the name of the alternate weight variable with blanks.

27 If the variable's data type is Numeric, select **Statistics**. The tab displays the status of the "automatic" statistics (that is, statistics whose calculation mechanism is already defined by SAS IT Resource Management) at day, week, month, and year levels.

An intersection of a statistic row and level column represents that statistic for that level. If you see a check in the box at an intersection, that statistic *is* selected at that level (that is, that statistic is requested at that level, or in other words, the calculation status of that statistic at that level is *Yes*). If you see an empty box at an intersection, that statistic *is not* selected at that level.

The statistics that are selected by default are a set that is based on the data type and interpretation type of the variable and the table type of the table. If you want to change the status of a statistic for a given level, click on the intersection of the statistic's row and the level's column.

Some selections imply other selections. For example, if you select Average, then Count and Sum are (implicitly) selected. Thus, in this case you do not need to select Count and Sum unless you want to display their values in the view. (For more information about the implied selections and about the suffix that identifies

every statistic, see “Appendix 5: Variable Interpretation Types” in the SAS IT Resource Management macro reference documentation.)

28 (The specification of age limits for the variable’s data and the use of the variable in BY, CLASS, ID, and/or Index variables lists are parts of the table’s definition, not parts of the variable’s definition. For more information, see “Overview of Levels” on page 337.)

29 Select **OK ▶ Close ▶ OK ▶ Close ▶ OK**

Deleting a Derived Variable

After you delete a derived variable, it is no longer available itself or for use as an underlying variable for a another derived variable or a formula variable, no longer available as a variable to test in a WHERE clause, no longer available as a BY, CLASS, ID, or Index variable, and so on.

Note: If you have no current use for the variable, but do anticipate future use, you might want *not* to delete the variable but instead to set the variable’s Kept status to *No*. When the variable’s Kept status is set to *No*, the variable’s definition is retained but ignored, and the variables’ values are discarded. (There will be missing values for all existing observations for the variables that had their Kept status set to *No*.) For more information about changing the variable’s Kept status, see “Creating/Editing/Viewing a Derived Variable” on page 381. △

You can delete the definition and data for a derived variable

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE and DELETE DERIVED control statements) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see “Chapter 4: Data Dictionary Macro and Control Statements” in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL DELETE DERIVED control statement, see the topic DELETE DERIVED in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: You cannot delete a derived variable if it is currently in use as a BY, CLASS, or weight variable. △

Note: To delete values (data) from a variable in a PDB but not delete the variable’s definition, see “Purging Data” on page 415. △

Deleting a Derived Variable (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can delete the definition and data of a derived variable from a table in the active PDB by following these steps:

1 From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Set Active PDB Options`**. The Display/Modify PDB Options window opens.

2 In the **`PDB Access Mode`** field, select **`Write`** if it is not already selected.

3 Select **`OK`**. SAS IT Resource Management returns you to the main menu.

4 From the main menu, select **`PDB Admin`**. A menu opens. From the menu, select **`Config Active PDB Dictionary`**. The Dictionary Tables window opens and displays a list of the tables in the active PDB.

5 Select the name of the table from which you want to delete the derived variable.
6 Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.
7 Select **Variable List**. The Variables window opens and lists the existing variables in the table.
8 Select the name of the derived variable whose definition and data you want to delete.
9 Select **ItemActions**. A menu opens. From the menu, select **Delete Definition**. The Delete Variable window opens.
10 Select **OK**. SAS IT Resource Management returns you to the Variables window, and the variable that you deleted is no longer on the list.
11 Select **File ▶ End ▶ OK ▶ File ▶ End ▶ OK**

Deleting a Derived Variable (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete the definition and data of a derived variable from a table in the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The **Manage PDBs** window opens.
2 Select **Properties**. The Set Active PDB Options window opens.
3 On the **Access** tab, select **WRITE** if it is not already selected.
4 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.
5 Select **Close**. SAS IT Resource Management returns you to the main window.
6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.
7 Select the name of the table from which you want to delete the derived variable.
8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.
9 On the **General** tab, select **Variables**. The View Variables window opens. The window lists the variables in the table.
10 Select the derived variable whose definition and data you want to delete.
11 Click your right mouse button. A menu opens. From the menu, select **Delete**. A confirmation dialog box opens.
12 Select **Yes ▶ OK** SAS IT Resource Management returns you to the View Variables window, and the variable that you deleted is no longer on the list.
13 Select **Close ▶ OK ▶ Close ▶ OK**

Deleting a Formula Variable

After you delete a formula variable, it is no longer available itself or for use as an underlying variable for another formula variable.

Note: If you have no current use for the variable, but do anticipate future use, you might want *not* to delete the variable but instead to set the variable's Kept (that is, calculated) status to *No*. When the variable's Kept status is set to *No*, the variable's definition is retained but ignored, and the values of the variable's data are not calculated. For more information about changing the variable's Kept status, see "Creating/Editing/Viewing a Formula Variable" on page 388. △

You can delete the definition for a formula variable

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE and DELETE FORMULA control statements) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL DELETE FORMULA control statement, see the topic DELETE FORMULA in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: You cannot delete a formula variable if it is referenced by another formula variable. △

Deleting a Formula Variable (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can delete the definition of a formula variable from a table in the active PDB by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2. In the **PDB Access Mode** field, select **Write** if it is not already selected.
3. Select **OK**. SAS IT Resource Management returns you to the main menu.
4. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens and displays a list of the tables in the active PDB.
5. Select the name of the table from which you want to delete the formula variable.
6. Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.
7. Select **Variable List**. The Variables window opens and lists the existing variables in the table.
8. Select the name of the formula variable whose definition you want to delete.
9. Select **ItemActions**. A menu opens. From the menu, select **Delete Definition**. The Delete Variable window opens.
10. Select **OK**. SAS IT Resource Management returns you to the Variables window, and the variable that you deleted is no longer on the list.
11. Select **File ▶ End ▶ OK ▶ File ▶ End ▶ OK**

Deleting a Formula Variable (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete the definition of a formula variable from a table in the active PDB by following these steps:

1. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2. Select **Properties**. The Set Active PDB Options window opens.
3. On the **Access** tab, select **WRITE** if it is not already selected.
4. Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.
5. Select **Close**. SAS IT Resource Management returns you to the main window.

6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.

7 Select the name of the table from which you want to delete the formula variable.

8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

9 On the **General** tab, select **Variables**. The View Variables window opens. The window lists the variables in the table.

10 Select the formula variable whose definition you want to delete.

11 Click your right mouse button. A menu opens. From the menu, select **Delete**. A confirmation dialog box opens.

12 Select **Yes ▶ OK** SAS IT Resource Management returns you to the View Variables window, and the variable that you deleted is no longer on the list.

13 Select **Close ▶ OK ▶ Close ▶ OK**

Deleting a Regular Variable

After you delete a regular variable, it is no longer available itself or for use as an underlying variable for a derived or formula variable, no longer available as a variable to test in a WHERE clause, no longer available as a BY, CLASS, ID, or Index variable, and so on.

Note: If you have no current use for the variable, but do anticipate future use, you might want *not* to delete the variable but instead to set the variable's Kept status to *No*. When the variable's Kept status is set to *No*, the variable's definition is retained but ignored, and the variable's values are discarded. (There will be missing values in all existing observations for the variables that had their Kept status changed from *Yes* to *No*.) For more information about changing the variable's Kept status, see "Creating/Editing/Viewing a Regular Variable" on page 394. △

You can delete the definition and data for a regular variable

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE and DELETE VARIABLE control statements) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statements, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL DELETE VARIABLE control statement, see the topic CREATE VARIABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: You cannot delete a regular variable if it is currently in use as a BY, CLASS, or weight variable. △

Note: To delete values (data) from a variable in a PDB but not delete the variable's definition, see "Purging Data" on page 415. △

Deleting a Regular Variable (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can delete the definition and data of a regular variable from a table in the active PDB by following these steps:

1 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.
2 In the **PDB Access Mode** field, select **Write** if it is not already selected.
3 Select **OK**. SAS IT Resource Management returns you to the main menu.
4 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens and displays a list of the tables in the active PDB.
5 Select the name of the table from which you want to delete the regular variable.
6 Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.
7 Select **Variable List**. The Variables window opens and lists the existing variables in the table.
8 Select the name of the regular variable whose definition and data you want to delete.
9 Select **ItemActions**. A menu opens. From the menu, select **Delete Definition**. The Delete Variable window opens.
10 Select **OK**. SAS IT Resource Management returns you to the Variables window, and the variable that you deleted is no longer on the list.
11 Select **File ▶ End ▶ OK ▶ File ▶ End ▶ OK**

Deleting a Regular Variable (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete the definition and data of a regular variable from a table in the active PDB by following these steps:

1 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2 Select **Properties**. The Set Active PDB Options window opens.
3 On the **Access** tab, select **WRITE** if it is not already selected.
4 Select **OK**. SAS IT Resource Management returns you to the Manage PDBs window.
5 Select **Close**. SAS IT Resource Management returns you to the main window.
6 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens. The window lists the tables in the active PDB.
7 Select the name of the table from which you want to delete the regular variable.
8 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.
9 On the **General** tab, select **Variables**. The View Variables window opens. The window lists the variables in the table.
10 Select the regular variable whose definition and data you want to delete.
11 Click your right mouse button. A menu opens. From the menu, select **Delete**. A confirmation dialog box opens.
12 Select **Yes ▶ OK**

SAS IT Resource Management returns you to the View Variables window, and the variable that you deleted is no longer on the list.

13 Select **Close ▶ OK ▶ Close ▶ OK**

Exploring Supplied Variable Definitions

You can explore supplied variable definitions in the active PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Write access or update access to the PDB is not required for you to view or search the list of supplied variable definitions.

For more information about variable definitions, see "Overview of Variables" on page 378.

Exploring Supplied Variable Definitions (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can explore supplied variable definitions in the active PDB by following these steps:

1. Select **Explore** from the SAS IT Resource Management main menu. A menu opens.
2. Select **All Supplied Variables**. The Explore All Supplied Variables window opens.
3. Select **Edit ▶ Find**

   Next, in the **Search** field type the name of the variable definition that you want to explore, and select the other search characteristics. Then, select **OK**.

   SAS IT Resource Management searches the variable description fields, highlights the variables whose descriptions contain the string that you entered in the **Search** field, and scrolls to the first highlighted variable. (If there is more than one variable, you might need to scroll to find the others.)
4. Select the variable that you want to know more about, and then select **ItemActions ▶ Browse Variable Definition**

Exploring Supplied Variable Definitions (UNIX and Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can explore supplied variable definitions in the active PDB by following these steps:

1. On the main window's Administration tab, select **Explore Tables/Variables**. The Select Tables and Variables window opens.
2. In the **Choose** field, select **Variables**.
3. Select and/or type the remaining criteria that describe which variables you want to view or explore, and then select **OK**. A list displays the variables that match your criteria.

Listing Derived Variables

You can list the names and labels of the derived variables in a given table

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE and LIST DERIVEDS control statements) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL LIST DERIVEDS control statement, see the topic LIST DERIVEDS in the SAS IT Resource Management macro reference documentation.

Note: If you want to list any other properties of the derived variables in a given table, use the %CPDDUTL PRINT TABLE control statement instead of the LIST DERIVEDS control statement. For more information about the %CPDDUTL PRINT TABLE control statement, see the topic PRINT TABLE in the SAS IT Resource Management macro reference documentation. △

Listing Formula Variables

You can list the names and labels of the formula variables in a given table

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE and LIST FORMULA control statements) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL LIST FORMULAS control statement, see the topic LIST FORMULAS in the SAS IT Resource Management macro reference documentation.

Note: If you want to list any other properties of the formula variables in a given table, use the %CPDDUTL PRINT TABLE control statement instead of the LIST FORMULAS control statement. For more information about the %CPDDUTL PRINT TABLE control statement, see the topic PRINT TABLE in the SAS IT Resource Management macro reference documentation. △

Listing Regular Variables

You can list the names and labels of the regular variables in a given table

- by using the %CPSTART macro and the %CPDDUTL macro (with the SET TABLE and LIST VARIABLES control statements) in batch mode.

  For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL SET TABLE control statement, see the topic SET TABLE in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL LIST VARIABLES control statement, see the topic LIST VARIABLES in the SAS IT Resource Management macro reference documentation.

Note: If you want to list any other properties of the regular variables in a given table, use the %CPDDUTL PRINT TABLE control statement instead of the LIST VARIABLES control statement. For more information about the %CPDDUTL PRINT TABLE control statement, see the topic PRINT TABLE in the SAS IT Resource Management macro reference documentation. △

Viewing/Editing Defaults for Variable Definitions

You can create one or more variable definitions in batch mode by using the %CPDDUTL macro. Any variable properties that you do not specify are defined by

default values. You can edit those default values if you want to use values that are different from the built-in values.

You can display and edit the defaults for variable definitions

- by using the SET DEFAULTS control statement with the %CPDDUTL macro in batch mode. For more information about the %CPDDUTL macro, see "Chapter 4: Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL control statement SET DEFAULTS, see the topic SET DEFAULTS in the SAS IT Resource Management macro reference documentation.

Note: The values that you define expire at the end of the current SAS IT Resource Management session. △

Working with Variables: Related Topics

How to Define Derived and Formula Variables

The order in which variables are calculated is significant because code that bases the calculation of one variable on another variable that has not yet been created generates errors. The order of creation is consistent across all variable types and within each type and can be summarized as follows:

- The values of regular variables are assigned first, the values of derived variables are calculated next, and the values of formula variables are calculated last.
- The values of derived and formula variables are always calculated in alphanumeric order.

1 Regular Variables

Regular variables are assigned values first when %C*x*PROCES executes (where *x*=M, P, S, or W). Regular variables are assigned values before the values of any other variables are calculated.

2 Derived Variables

Derived variables are assigned values after %C*x* PROCES (where *x*=M, P, S, or W) assigns values to the regular variables. The values of derived variables are calculated in ascending alphanumeric order. A derived variable can be based directly on the values of constants, regular variables, and any derived variables that are earlier than this derived variable in the ascending alphanumeric sequence of derived variables.

Example:

- Assume that A and B are regular variables and that C and D are derived variables.
- Assume that C = SUM(A,B).
- Assume that D = A / SUM(A,B).

The calculation for D could also be coded as D = A/C for derived variables, because derived variables are defined in ascending alphanumeric order. At the time that derived variable D is being calculated, derived variable C already has its value.

3 Formula Variables

Formula variables are calculations in one or more views of a table and are assigned values when the view for a table is read. Formula variables are

calculated in ascending alphanumeric order. A formula variable can be based directly on the values of constants, regular variables, derived variables, any formula variables that are earlier than this formula variable in the ascending alphanumeric sequence of formula variables, and the _LEVEL_ variable.

Example:

- Assume that A and B are regular variables and that C and D are formula variables.
- Assume that C = SUM(A,B).
- Assume that D = A / SUM(A,B).

The calculation for D could also be coded as D = A/C for formula variables, because formula variables are defined in ascending alphanumeric order. At the time that formula variable D is being calculated, formula variable C already has its value.

Summary

The calculations of specific derived and regular variables must be based on values that are calculated before the specific variable is calculated.

Example:

- Assume that X and Y are regular variables.
- Assume that P and Q are derived variables.
- Assume that G is a formula variable.

The calculation for P can be coded P = SUM(X,Y) because the regular variables, X and Y, exist before any derived variables exist.

The calculation for Q can be coded Q = X/P for the following reasons: X, a regular variable, exists before any derived variables exist; P, a derived variable, precedes Q in ascending alphabetical order; therefore, the value of P exists before the value of Q is calculated.

The calculation for G can be coded G = P+Q because all derived variables exist before any formula variables exist.

Variable Interpretation Types

For more information about variable interpretation types, see “Appendix 5: Variable Types” in the SAS IT Resource Management macro reference documentation.

CHAPTER
19

Administration: Working with Data

Working with Data: Main Topics

Overview of Data

Typically the data flows through the following steps:

1 The data is *logged*.

That is, the software that collects the raw data writes it to a log. If necessary, the log is then transferred to a host where SAS IT Resource Management is installed.

2 The data is *processed*. Processing consists of the following phases:

   a The data is *staged*.

      That is, the data is read from the log and written to one or more SAS data sets. Or one or more SAS views are generated that make the data look as if it is in SAS data sets. Or in some other way the data is made available as if it is in SAS data sets.

   b The data is *pre-summarized* (if the volume of data is large).

      That is, the SAS SUMMARY procedure (PROC SUMMARY) reads the staged event data, does a preliminary summary (typically by HOUR and SHIFT), and writes the output as interval data to one or more SAS data sets.

      In the staged observations, the time in the DATETIME variable indicates the time of the event. In the pre-summarized observations, the time in the DATETIME variable indicates the beginning of the interval that is represented by the summary statistics. For example, if the preliminary summary is by HOUR and SHIFT, then the time in DATETIME (and the hour in HOUR) is set to the beginning of the hour that is represented by the summary statistics. For example, if the time in DATETIME is 17:00:00, then the value of HOUR is 17.

      For each staged variable whose values are pre-summarized, one or more preliminary summary statistics are created. For each preliminary summary statistic, its label indicates the content of the staged variable and the type of statistic (for example, Sum or Count). The variable name is seven or fewer characters in length and does not have a suffix that indicates the type of statistic.

   c The data is *merged* (if the volume of data is manageable).

      That is, the incoming data (in the staged or pre-summarized SAS data sets) is merged with the existing data in the detail level of the PDB.

      On the same pass, a copy of the incoming data is placed in an archive, if archiving has been requested.

      Also on the same pass, if any of the existing data in the detail level is older than the SAS IT Resource Management administrator that is specified for the PDB, the older data is removed from the PDB.

3 The data is *reduced*.

   That is, the data is read from detail level (or pre-summarized SAS data sets or staged SAS data sets), statistics are calculated, and the output is assimilated into one or more summary levels of the PDB.

Depending on the particular data source, the data goes through some or all of the above steps.

The following additional topics (in alphabetical order) are related to data:

Archiving Incoming Data

An archive is a set of archive libraries.

If archiving is activated, each time the process step runs, it generates an archive library. In the archive library is a copy of the incoming data to the detail level of the tables in the PDB for which archiving is enabled. For an overview of archiving, see "Appendix 2: Archive Data" in the SAS IT Resource Management macro reference documentation.

You can specify whether to archive and how to archive the data that is incoming to the detail level of the active PDB

- by using the %CPPDBOPT macro to specify the location and type of archive, and by using the %CPDDUTL macro with one or more %CPDDUTL UPDATE TABLE control statements to activate archiving of incoming data for one or more tables in the active PDB.

  For more information about the %CPPDBOPT macro, see the topic %CPPDBOPT in the SAS IT Resource Management macro reference documentation. For more information about the %CPDDUTL macro, see the topic %CPDDUTL in the SAS IT Resource Management macro reference documentation. For more information about the UPDATE TABLE control statements, see the topic UPDATE TABLE in the SAS IT Resource Management macro reference documentation.
- by using the SAS IT Resource Management GUI for z/OS.
- by using the SAS IT Resource Management GUI for UNIX and Windows.

The archiving takes place during the process step. For more information about the process step, see "Processing Data" on page 413.

For more information about archiving, see

- "Viewing/Editing Your Active Site Library's Archive Information" on page 284
- "Viewing/Editing Your Active PDB's Archive Information (z/OS)" on page 327
- "Retrieving Archived Data" on page 418
- "Creating a Table" on page 357 or "Viewing/Editing a Table" on page 364
- the topic %CPARCRET in the SAS IT Resource Management macro reference documentation.
- the section "Guidelines for Allocating Space in the Archive" in the Archive topic in "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation and the section "Guidelines for Revising Space Allocations in the Archive" in "Appendix 1: Recovery Procedures" in the SAS IT Resource Management macro reference documentation.
- the section "Archiving Overview" in "Chapter 2: Administrative Macros" in the SAS IT Resource Management macro reference documentation.

Archiving Incoming Data (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can specify whether to archive and how to archive the data that is incoming to the detail level by following these steps:

1. Activate the PDB whose incoming data you want to archive, and request write access to the PDB.

   For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about changing your access to the active PDB, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.

2 From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Set Active PDB Options**. The Display/Modify PDB Options window opens.

3 In the **Archive Type** field, select whether you want a **DASD** archive or a **Tape** archive.

4 In the **Archive Unit** field, type the value that you want to be used for each archive library's UNIT= parameter.

5 In the **Archive Root** field, the high-level qualifiers that identify the PDB are the default location of the archive. That is, by default, the archive is "under" the PDB. If you want a different location, overwrite the current value with another location.

6 In the **Archive Library Allocation Parameter String** field, type any additional options that you want to be used when an archive library is allocated. For example, if you want your archives to be on tape and to be retained for one year, your parameter string would look like this: *LABEL=(1,SL,,,RETPD=365)*.

7 Select **OK**. SAS IT Resource Management returns you to the main menu.

8 Select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens and displays a list of the tables in the active PDB.

9 Select a table for which you want to enable archiving.

10 Select **ItemActions**. A menu opens. From the menu, select **Edit Definition**. The Edit Table window opens.

11 In the **Archive** field, select **On**.

12 Select **OK**. SAS IT Resource Management returns you to the list of tables.

13 Repeat the steps to enable archiving (steps 9 through 12) for any other tables whose incoming data you want to archive.

14 When you are finished enabling the archiving, select **File**. A menu opens. From the menu, select **End**. SAS IT Resource Management returns you to the main menu.

Archiving Incoming Data (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can specify whether to archive and how to archive the data that is incoming to the detail level by following these steps:

1 Activate the PDB whose incoming data you want to archive, and request write access to the PDB.

For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about changing your access to the active PDB, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.

2 On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.

3 Select **Properties**. The Set Active PDB Options window opens.

4 Select the **Archive** tab.

5 In the **Archive Type** field, select whether you want a **Disk** archive or a **Tape** archive.

6 In the **Archive Device** field, type the device where you want the archive to reside. For example, type *c:*.

7 In the **Archive Root** field, type the directory that you want to use as the archive. If you leave this field blank, the default value is the location of the PDB. That is, by default, the archive is "under" the PDB. If you want a different location, provide a different value for the field.

8 In the **Archive Library Allocation Parameter String** field, type any additional SAS LIBNAME options that you want to be used when an archive library is allocated.

9 Select **OK** and then **Close**. SAS IT Resource Management returns you to the main window.

10 On the main window's **Administration** tab, select **Manage Tables**. The Manage Tables window opens and displays a list of the tables in the active PDB.

11 Select a table for which you want to enable archiving.

12 Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.

13 Select the **Advanced** tab.

14 In the **Archive** field, select **Yes**.

15 Select **OK**. SAS IT Resource Management returns you to the list of tables.

16 Repeat the steps to enable archiving (steps 11 through 15) for any other tables whose incoming data you want to archive.

17 When you are finished enabling the archiving, select **Close**. SAS IT Resource Management returns you to the main window.

Checking for Duplicate Data

For information about duplicate-data checking, see "Appendix 4: Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation.

Collecting Data

The performance data read by SAS IT Resource Management is collected by performance data *collectors*. These data collectors gather data about your information technology systems, such as phone systems, networks, Web servers, and more.

The collector's data is read, processed into the SAS IT Resource Management performance data warehouse (PDB), and reduced into summary levels in the PDB (based on statistics that you select).

For more information about collectors and how they can be set up to work with SAS IT Resource Management, see "Overview of Setup" on page 37 and especially "Setting Up the Server, with Setup Cases" on page 38.

Processing Data

After your site collects data about your information systems, the data is passed from the data collector to a PDB. You should not begin processing until you have logged data for at least one day (preferably several days). Moving the data into a SAS IT Resource Management PDB is referred to as *processing* the data. When you process data into the PDB, it goes through the following stages:

1 The raw data is read into staging data sets.

2 The data in the staging data sets is sorted based on your list of BY variables (variables used for sorting) stored in the corresponding table definition in the PDB's data dictionary. BY variables are defined when you create a table definition in the PDB's data dictionary. As *new* data is processed and reduced into the PDB, *existing* detail level observations are aged out or deleted if they are older than the age limit that is specified in the PDB's data dictionary.

3 The staged data sets are then merged with the data sets in the detail level of the PDB. (For information about this topic, see "Tables, Variables, and Definitions in a PDB" in "What Is a PDB?" on page 7.) The data set is referred to as the detail level data set because it contains all of the detailed information from your raw data.

4 At the end of the process task, a report is generated about the status of the detail level of the PDB.

The tasks of processing data into the PDB must be performed on the host where the SAS IT Resource Management server license is installed. Also, you must have write access to the PDB to perform this task. The tasks of processing and reducing data are usually performed or automated by the SAS IT Resource Management administrator. For additional information, see "General-Purpose Server Setup Documentation" on page 12.

When you process data into the PDB, the YEARCUTOFF SAS system option is automatically set to the year 1950. The YEARCUTOFF option is used to determine the century in which a two-digit year value belongs. By setting YEARCUTOFF to 1950, a two-digit year from 50 to 99 will be assigned 19 as the century and a two-digit year from 00 to 49 will be assigned 20 as the century. Additionally, the YEARCUTOFF option enables SAS software to use and support both two- and four-digit date values. Therefore, you can specify and process a date that includes a year value of 2001 as 01 or as 2001.

Note: For information about using CPFUTURE to process data with future datetimes, see "Global and Local Macro Variables" in Chapter 1 of the SAS IT Resource Management macro reference documentation. △

You can process data into the active PDB

- *in batch mode* by using the %CPPROCES macro or an older macro (%CMPROCES for z/OS, %CSPROCES for UNIX, and %CWPROCES for Windows) in batch mode.

  Note: For more information about using macros in batch mode, see Chapter 1 in the SAS IT Resource Management macro reference documentation and "Working with the Interface for Batch Mode" on page 18. For more information about the %CPPROCES macro, see the topic %CPPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CMPROCES macro, see the topic %CMPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CSPROCES macro, see the topic %CSPROCES in the SAS IT Resource Management macro reference documentation. For more information about the %CWPROCES macro, see the topic %CWPROCES in the SAS IT Resource Management macro reference documentation. △

- *interactively* by using the Process/Reduce Wizard in the SAS IT Resource Management GUI for UNIX and Windows or by selecting PDB Data on the main menu of the SAS IT Resource Management GUI for z/OS. For more information about using the Process/Reduce Wizard, see the *Getting Started with SAS IT Resource Management* documentation.

For medium to large volumes of data, the batch method is recommended.

Processing Data (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can process data by following these steps:

1 Activate the PDB into which you want to process data, and obtain write access to the PDB. For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about obtaining access to the PDB, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.

2 From the main menu, select **PDB Data**. A menu opens. From the menu, select **Process Raw Data**. The Process Raw Files window opens.

3 Select **Actions**. A menu opens. Select **Set Raw File**. The Raw File window opens.

4 Type the fully qualified name of the file that contains the data that is to be processed into the PDB.

5 Select **OK**.

6 Select **Actions**. A menu opens. Select **Select Collector**.

7 Select the name of the collector that acquired the data.

8 Select **OK**.

9 Select the name of the table or tables into which you want the data processed.

10 Select **ItemActions**. A menu opens. Select **Run Process**. The Migrate in Progress window opens. Select **OK**.

11 When processing is finished, select **File**. A menu opens. From the menu, select **End**. SAS IT Resource Management returns you to the main menu.

Processing Data (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can process data by following these steps:

1 Activate the PDB into which you want to process data, and obtain write access to the PDB. For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about obtaining write access, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.

2 On the main window's **Administration** tab, select **Process/Reduce Wizard**.

3 Select **Process Data into PDB** if it is not already selected, and then select **Next**.

4 Follow the directions in the wizard.

5 When processing is finished, select **Close ▶ Close ▶ Return to Main**

SAS IT Resource Management returns you to the main window.

Purging Data

You can delete the data from the detail level in all tables in the active PDB or delete the data from the summary levels in all tables in the active PDB or both

- by using the %CPDBPURG macro in batch mode. For more information about the %CPDBPURG macro, see the topic %CPDBPURG in the SAS IT Resource Management macro reference documentation.

You can delete data from a specific table in the active PDB (but keep the table and its definition)

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: If you want to delete data and also delete the table definition and table that contains the data, see "Deleting a Table" on page 361. △

If you want to delete data and also delete the variable definition and variable that contains the data, see "Deleting a Regular Variable" on page 403 or "Deleting a Derived Variable" on page 400.

Purging Data (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can purge data by following these steps:

1. Activate the PDB from which you want to purge data and obtain write access to the PDB. For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about obtaining write (OLD) access, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.
2. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens and displays a list of the tables in the active PDB.
3. Select the table from which you want to delete data.
4. Select **ItemActions**. A menu opens. From the menu, select **Delete Table**. The Delete Table window opens.
5. Select **Purge Table** if it is not already selected.
6. Select **OK**. SAS IT Resource Management deletes the data but not the table, and returns you to the Dictionary Tables window.
7. Select **File**. A menu opens. From the menu, select **End**. SAS IT Resource Management returns you to the main menu.

Purging Data (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can purge data by following these steps:

1. Activate the PDB from which you want to delete data and obtain write access to the PDB. For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about obtaining write access, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.
2. From the main menu, select **Manage Tables**. The Manage Tables window opens and displays a list of the tables in the active PDB.
3. Select the table whose data you want to delete (without removing the table itself).
4. Select **File**. A menu opens. From the menu, select **Delete Table Definition**. The Delete Table window opens.
5. Select **Delete Data in PDB Only** if it is not already selected.
6. Select **OK**. SAS IT Resource Management deletes the data without deleting the table.
7. SAS IT Resource Management displays a "data deleted" message. Select **OK**. SAS IT Resource Management returns you to the Manage Tables window.
8. Select **Close**. SAS IT Resource Management returns you to the main window.

Reducing Data

To *reduce* data means to read data from the detail level of the PDB, calculate summary statistics, and write the summary statistics to the day, week, month, and year levels of the PDB. By reducing the data and providing summary statistics at each level, you can analyze your service information in different levels of summarization, over short or long periods of time.

Data reduction is not only an effective way to view all of your data, but it also saves you time and space. For example, suppose that the day level contains one observation per hour per disk drive. If one of the fields on that observation is the minimum amount of free space on that drive at that hour, the value for that one observation in the day

level is based on the corresponding values for an hour's worth of observations in the detail level. If you want to use the minimum value in a report, it is faster to read it from one observation than to calculate it from many observations. Furthermore, after that value (and any other summary statistic) has been calculated and you do not have any other use for the detail data, the detail data can be deleted.

When your data is reduced from the detail level to the summary levels (day, week, month, and year), it goes through the following stages:

1 For each table
   a observations from the detail level are summarized into temporary files, based on grouping or class variables
   b summarized observations are incorporated into the day, week, month, and year levels
   c observations from the detail level and summarized reduction levels are deleted if the age limit for the data in the detail level is zero days.
2 At the end of the reduce task, a report is generated that indicates the status of all of the levels of the PDB.

   For more information, see "Tables, Variables, and Definitions in a PDB" in "What Is a PDB?" on page 7.

The task of reducing data in the PDB must be performed on the host where the SAS IT Resource Management server license is installed. You also must have write access to the PDB to perform this task. The tasks of processing and reducing data are usually performed or automated by an administrator at your site. For additional information about setting up your PDB to process and reduce data for a specific collector, refer to "Setting Up the Server, with Setup Cases" on page 38.

You can reduce data into one or more summary levels in a PDB

- interactively
  - by selecting **PDB Data** on the main menu in the SAS IT Resource Management GUI for z/OS
  - or by using the Process/Reduce Wizard in the SAS IT Resource Management GUI for UNIX and Windows.

    Note: For more information about using the Process/Reduce Wizard, see the *Getting Started with SAS IT Resource Management* documentation. △
- in batch mode
  - by using the %CMPROCES, %CPPROCES, %CSPROCES, or %CWPROCES macro and the %CPREDUCE macro.

    Note: If the data is staged by MXG and if you want to bypass the detail level, you need to use the %CMEXTDET macro in batch mode before you use the %CPREDUCE macro in batch mode. For more information about the %CMEXTDET macro, see the topic %CMEXTDET in the SAS IT Resource Management macro reference documentation. △

    Note: For more information about the %CPREDUCE macro, see the topic %CPREDUCE in the SAS IT Resource Management macro reference documentation. For more information about using macros in batch mode, see Chapter 1 of the SAS IT Resource Management macro reference documentation . △

For medium to large volumes of data, the batch method is recommended.

If you have an extremely large volume of data, you can bypass processing and reducing the data from the detail level of the PDB, and read the data directly into the summary (reduction) levels. For more information, see the DETAIL= parameter on the

%CPREDUCE macro (and, for z/OS, see the %CMEXTDET macro). For more information about the %CPREDUCE macro, see the topic %CPREDUCE in the SAS IT Resource Management macro reference documentation . For more information about the %CMEXTDET macro, see the topic %CMEXTDET in the SAS IT Resource Management macro reference documentation.

Reducing Data (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can reduce data by following these steps:

1. Activate the PDB in which you want to reduce data, and obtain write (OLD) access. For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about obtaining access to the PDB, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.
2. From the main menu, select **PDB Data**. A menu opens. From the menu, select **Reduce PDB Data**. The Reduce PDB window opens and displays a list of the tables in the active PDB.
3. Select the table(s) whose data you want to reduce.
4. Select **ItemActions**. A menu opens. Select **Run Reduce**.
5. When reduction ends, select **File**. A menu opens. From the menu, select **End**. SAS IT Resource Management returns you to the main menu.

Reducing Data (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can reduce data by following these steps:

1. Activate the PDB in which you want to reduce data, and obtain write access to the PDB. For more information about activating a PDB, see "Activating a PDB" on page 299. For more information about obtaining write access, see "Viewing/Editing Your Active PDB's Access Mode" on page 325.
2. On the main window's **Administration** tab, select **Process/Reduce Wizard**.
3. Select **Reduce PDB Data** if it is not already selected, and then select **Next**.
4. Follow the directions in the wizard.
5. When reduction is complete, select **Close**. SAS IT Resource Management returns you to the main window.

Retrieving Archived Data

You can retrieve data that was archived for a PDB

- by using the %CPARCRET macro in batch mode.

  Note: The %CPARCRET macro does not check for the possibility that some of the archive libraries are on tape and some are on disk. It assumes that all are on the same media as currently defined in the dictionary of the ARCHPDB. △

  Note: Archive libraries are retrieved in their entirety. That is, if you request a time span starting at noon, and the archive library starts at 8:00 a.m, the data from 8:00 a.m. until noon will also be retrieved. △

  For more information about the %CPARCRET macro, see the topic %CPARCRET in the SAS IT Resource Management macro reference documentation.

Staging Data

For many data sources, SAS IT Resource Management supplies code that stages the data and supplies table (and variable) definitions that the detail level and summary levels of the PDB use to store the data. If you want to use data from a source for which staging code and table definitions are not supplied, you can process the data into your PDB by using the Generic Collector Facility.

For more information about the Generic Collector Facility, see "Using the Generic Collector Facility" on page 158. For additional information, see "Overview of Setup" on page 37.

Viewing Amount of and Datetime Range of Data

You can view the amount of data and the datetime range of the data in the active PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing Amount of and Datetime Range of Data (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can view the amount of data and the datetime range of the data by following these steps:

1. From the main menu, select **PDB Admin**. A menu opens. From the menu, select **Config Active PDB Dictionary**. The Dictionary Tables window opens and displays a list of the tables in the active PDB.
2. Select the table for which you want to know the amount of data and the datetime range of the data.
3. Select **ItemActions**. A menu opens. From the menu, select **Table Status**. The Table Status window opens.
4. The columns **DateTime of Oldest Data** and **DateTime of Newest Data** display the datetime range for the data in each level of the table. Each row displays information for a different level. The column **Number of Observations** displays the amount of data in each level of the table.
5. To return to the Dictionary Tables window, select **Goback**.
6. To return to the main menu, select **File**. A menu opens. From the menu, select **End**.

Viewing Amount of and Datetime Range of Data (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view the amount of data and the datetime range of the data by following these steps:

1. On the main window's **Administration** tab, select **Manage Tables**. The **Manage Tables** window opens and displays a list of the tables in the active PDB.
2. Select the table for which you want to know the amount of data and the datetime range of the data.
3. Select **File**. A menu opens. From the menu, select **Properties**. The Table Properties window opens.
4. Select the **Table Status** tab.
5. The columns **DateTime of Oldest Data** and **DateTime of Newest Data** display the datetime range for the data in each level of the table. Each row displays

information for a different level. The column **# of Obs (Number of Observations)** displays the amount of data in each level of the table.

6 To return to the Manage Tables window, select **OK**.

7 To return to the main window, select **Close**.

Viewing Data Values

You can view the data in the active PDB

- by using the SAS IT Resource Management GUI for z/OS
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing Data Values (z/OS)

In the SAS IT Resource Management GUI for z/OS, you can view the data by following these steps:

1 From the main menu, select **PDB Data**. A menu opens. From the menu, select **Examine PDB Data**. The Query PDB Data window opens.

2 Each row identifies a table/level pair, which represents a view named *level-name.table-name*. The row also displays the number of observations in that view and a description of the table for which this is a view.

3 Select the row whose data you want to view.

4 Select **ItemActions**. A menu opens. From the menu, select **View Data** or **Browse Data**.

- If you select **View Data**, the Viewing Table window opens. Each row represents an observation. (Use **page down** and **page up** to scroll through the observations). Each column represents a variable. (Issue the commands **right** and **left** to scroll through the variables.)
- If you select **Browse Data**, the Browsing Table window opens. Each subsequent window represents one observation. The observation number is in the upper-right corner of the window. For each variable, there is a label followed by a value. (Issue the commands **forward** and **back** to scroll through the observations.)

5 When you are finished looking at the data, select **File**. A menu opens. From the menu, select **Close**. SAS IT Resource Management returns you to the Query PDB Data window.

6 To return to the main menu, select **File**. A menu opens. From the menu, select **End**.

Viewing Data (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view the data by following these steps:

1 From the main window's **Administration** tab, select **Examine PDB Data**. The Examine Data window opens.

2 On the left, select the PDB level that you want to view.

3 On the right, select the view that you want to see. (For example, DAY.HN2IFT will enable you to see day-level data in the HN2IFT table.)

4 Select **Locals**. A menu opens. From that menu, select **View Data**. A window displays the day data in the HN2IFT view. The rows represent observations

(records). The columns represent variables. The intersection of a row and column represents that row's value of that variable.

Use the horizontal and vertical scroll bars to reach rows and columns that are not initially visible.

5 When you are finished viewing the data, select **Close**. SAS IT Resource Management returns you to the Examine Data window.

6 To return to the main window, select **Close**.

Working with Data: Related Topics

Handling Time Stamps for Coordinated Universal Time

Some performance data contains time stamps expressed as Coordinated Universal Time (Greenwich Mean Time) rather than local time values. Currently, the only data sources that are supported by SAS IT Resource Management data that use Universal time in this way are SunNet Manager and Sun Enterprise Manager. The SAS IT Resource Management version %CSPROCES macro translates the Universal time stamp to local time before storing the data in the PDB. %CSPROCES uses two PDB options to derive local time: the offset of the local time zone from Universal time, and the Daylight Saving Time expression for the local time zone.

Both of these values can be set by following this path in the SAS IT Resource Management GUI for UNIX and Windows: **Manage PDBs ► select the PDB ► Properties ► Time Zone**

The offset of the local time zone is expressed as the difference in hours of the local time zone from Universal time. For example, Washington DC, USA, is +5; Sydney, Australia, is -10; Wellington, New Zealand, is -12; Calcutta, India, is -5.5; and London, England, is 0.

The calculation of local time is adjusted for Daylight Saving Time based on an expression that is evaluated as true when Daylight Saving Time is in effect (between 2:00 a.m. on the first Sunday in April and 2:00 a.m. on the last Sunday in October). If Daylight Saving Time is not appropriate for your data, you should blank out the expression that determines DST on the PDB Properties screen that is identified above and the calculation will not be adjusted.

Changes to Daylight Saving Time Expression

If you have PDBs that were created by using any release of SAS/CPE software or SAS IT Resource Management prior to SAS IT Resource Management 1, you should update the PDBs to contain the following expression to enable Daylight Saving Time (unlike previous expressions, this expression is not date dependent):

```
* Convert DST expression in a PDB to correct expression;

%CPSTART( pdb=/my/pdb, mode=batch, access=write );
data dictlib.cppdbopt;
  modify dictlib.cppdbopt;
  if cpdstime^=' ' then do;
         put 'Was:'  cpdstime;
         cpdstime='dhms(mdy(4,8-weekday(mdy4,7,year(datepart
```

```
            (datetime))),year(datepart(datetime))),3,0,0)
            <datetime<dhms(mdy(10,32-weekday(mdy(10,31,year(datepart
            (datetime))),year(datepart(datetime))),2,0,0)';
        put 'Is now:' cpdstime;
        end;
run;
```

Using the YEARCUTOFF SAS System Option

When you process data with dates that use a two-digit year value, the value of the YEARCUTOFF option is used to determine the century for the two-digit date value. In other words, YEARCUTOFF is used to map the two-digit value to a four-digit value. For example, if you are using a date such as 120101 (MMDDYY), would this date represent December 1, 1901 or 2001? When you process data into the PDB, the value of the YEARCUTOFF option determines the century for that date value.

The YEARCUTOFF option defines the starting point for a 100-year time span and, therefore, enables SAS and SAS IT Resource Management to determine the appropriate year for the two-digit value. For example, if YEARCUTOFF=1900, then a two-digit year between 00 and 99 will be assigned 19 as the century because all 100 years fall within the 19th century. If YEARCUTOFF=1950, then a two-digit year between 50 and 99 is assigned 19 as the century and a two-digit year between 00 and 49 is assigned 20 as the century. This is because the 100-year span starts with 1950, so a year of 00 in the 100-year span beginning with 1950 would be the year 2000.

The following table provides examples of how three different two-digit year values are interpreted based on different YEARCUTOFF values.

Table 19.1 Century Interpretation Based on YEARCUTOFF Value

	Incoming Date Value: 120100 (mmddyy)	Incoming Date Value: 120198 (mmddyy)	Incoming Date Value: 120133 (mmddyy)
Year Cutoff Value is 1900 (SAS 6 default)	Dec. 1, 1900	Dec. 1, 1998	Dec. 1, 1933
Year Cutoff Value is 1920 (SAS 7 and SAS 8 default)	Dec. 1, 2000	Dec. 1, 1998	Dec. 1, 1933
Year Cutoff Value is 1950 (SAS IT Resource Management default)	Dec. 1, 2000	Dec. 1, 1998	Dec. 1, 2033

When you process data into the PDB, SAS IT Resource Management checks the current setting of the YEARCUTOFF SAS system option. The default value for the YEARCUTOFF SAS system option is 1900 for SAS Version 6, 1920 for SAS Version 7 and SAS Version 8, and 1950 for SAS IT Resource Management. Your site can change this default option within SAS, or you can override it when you process data into a SAS IT Resource Management PDB. You can also change the default option within SAS. SAS IT Resource Management automatically changes the value of YEARCUTOFF when you process data into a SAS IT Resource Management PDB, in the following situations:

- If the %CMPROCES, %CPPROCES, %CSPROCES, or %CWPROCES macro determines that the current value of YEARCUTOFF is lower than the SAS IT Resource Management default value of 1950, then SAS IT Resource Management

automatically sets YEARCUTOFF=1950 to prevent possible problems in processing raw data with two-digit year values in the date.

- When you set CPFUTURE=ACCEPT, the %CPPROCES or %CMPROCES macro accepts data with datetimes more than two days in the future. To achieve this functionality with the %CSPROCES macro or the %CWPROCES macro, you can set the MAXDATE= parameter to a future date. This parameter defines the greatest or highest date in the future for which data can be processed into the PDB.

  If you set CPFUTURE=ACCEPT to permit future dates, the YEARCUTOFF value is automatically set to 1950 in order to avoid errors in processing the two-digit year values.

  If you specify MAXDATE=*some-future-date*, where *some-future-date* is a future date in a valid SAS date format, SAS IT Resource Management subtracts 99 (years) from that specified date to determine the minimum value that can be used for YEARCUTOFF. If the existing value of YEARCUTOFF is less than this new calculated date value, then the new calculated date is used as the value for YEARCUTOFF until the process task completes.

The default SAS IT Resource Management cutoff date of 1950 is used because it prevents problems in processing data for the year 2000, and it is unlikely that you will be processing data with datetimes older than this date. The YEARCUTOFF option is automatically reset to your previous value after your data has been processed into the PDB.

For more information about using the CPFUTURE macro variable, see the section "Global and Local Macro Variables" in Chapter 1 of the SAS IT Resource Management macro reference documentation .

For more information about using the MINDATE= or MAXDATE= parameters when processing data, see the topic %CSPROCES in the SAS IT Resource Management macro reference documentation or see the topic %CWPROCES in the SAS IT Resource Management macro reference documentation.

CHAPTER

20

Administration: Extensions to SAS IT Resource Management

Extensions to SAS IT Resource Management Including Duplicate-Data Checking

Overview of Extensions

Extensions enable you to add optional capabilities within SAS IT Resource Management. For example, you can use extensions to

- specify how duplicate data is handled in your PDB(s).
- package, distribute, and install user-developed collector support.
- alter normal processing of raw data with exit points that pass control to user-written code that filters or creates data or meets other custom requirements.

For more information about extensions to SAS IT Resource Management, see the following topics:

- "Checking for Duplicate Data" on page 426
- "Creating and Installing a Collector Support Package" on page 426
- "Overview of Using Process Exits" on page 442.

Checking for Duplicate Data

If you want to understand and/or change how the process task handles duplicate data, see Appendix 4: "Working with Duplicate Data" in the SAS IT Resource Management macro reference documentation.

Collector Support Packages

Creating and Installing a Collector Support Package

Note: This document describes the requirements and methods that are used to package collector-support entities and then to install them in another SAS IT Resource Management location. Although it makes recommendations on how to create your collector-support entities, this document is not intended to explain the collector-support writing process. Where possible the reader will be directed to other SAS IT Resource Management documentation for more detailed information. △

For more information, see the following macros: the topic %CPPKGCOL in Chapter 4, "Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation, the topic %CPRPTPKG in Chapter 4, "Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation, and the topic %CPINSPKG in Chapter 4, "Data Dictionary Macro and Control Statements" in the SAS IT Resource Management macro reference documentation.

The following topics describe how to create and install a collector package:

Introduction

SAS IT Resource Management has always provided the ability for users to write their own support for data collectors or data sources when support is not supplied as part of the product. Once written, these collector-support entities did not have the "feel" of being an integrated part of SAS IT Resource Management; also, it was difficult to share or distribute them.

The package functionality enables users to package their collector-support entities and to install them into a SAS IT Resource Management location. This is beneficial because the collector-support entities will appear and behave like the supplied collector support; this includes having the ability to use the Process/Reduce Wizard.

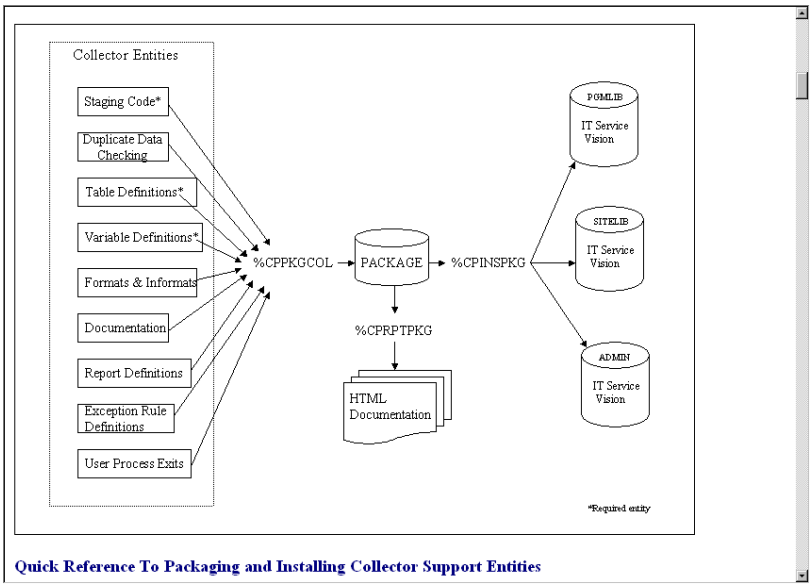
For more information about the macros that are mentioned here, see the SAS IT Resource Management Software macro reference documentation .

Overview

This section provides an overview of the packaging process from writing your own collector-support entities through packaging and installing the collector-support entities. The intention is that the overview will provide sufficient information for those users who have experience in writing collector-support entities and have an in-depth knowledge of SAS IT Resource Management. For users new to this process, it provides a more concise version of the "Details" section that follows.

Process Flow

The following diagram briefly explains the process flow of generating a collector package through to installation.



Quick Reference to Packaging and Installing Collector-Support Entities

The following sections give a brief example of how the process works. The "Details" section that follows provides more information about each stage.

Creating Your Collector-Support Entities

For detailed instructions about how to write support for your own collector, see the Generic Collector Facility documentation in the "Setup the Server" topic of the "Administration: Setup" section of the online help for SAS IT Resource Management. If you already have collector support that you use with the Generic Collector Facility, then converting it to a package and installing it is a relatively easy process. The following table shows a list of SAS IT Resource Management entities that are packaged as part of collector support. If the required field is set to *Yes*, then that SAS IT Resource Management entity must exist in order for a valid package to be constructed.

Table 20.1 Collector-Support Entities

Collector Entity	Description	Required	Restriction(s)
Staging Code	This is the user written SAS code that converts raw data into a SAS data set.	Yes	□ The code should reside in ADMIN.collector.toolnm.SOURCE (or SCL if you have written your staging code in SCL). When %CPPROCES runs after installation, it will use the COLLECTR= and TOOLNM= values that you supply in order to locate the appropriate staging code in the ADMIN library. □ The SAS data set(s)/view(s) that is produced by the staging code should reside in COLLECT.*tablenm* or COLLECT.t*able-external-name*. □ If the Process Wizard is to be used after installation, then the staging code should point to a fileref of 'RAWDATA' in order to access the raw data. %CPPROCES will automatically allocate the fileref of 'RAWDATA' when past the location of the raw data.
Formats and Informats	User-written formats and informats that are used in the staging code or for variables.	No	All formats and informats should be stored in ADMIN.CPFMTS.
Report Definitions	User-written report definitions (including source for the %CPSRCRPT reports).	No	All user-written report definitions should be stored in the ADMIN.ITSVRPT folder. If any of these reports use the %CPSRCRPT macro and have associated source entries, the location of the source entries does not matter as long as they are available when the package is built.
Exception Rule Definitions	Exception rule definitions that apply to this collector support.	No	Exception rule definitions for this collector support should be stored in the folder ADMIN.CPXRULE.
CPDUPCHK entry	This SAS Catalog source entry contains a default specification for the %CPDUPCHK macro and is required only if you have implemented duplicate-data checking by means of the %CPDUP* macros in your staging code.	No	This source entry should be stored in ADMIN.*collector*.CPDUPCHK.SOURCE.

Collector Entity	Description	Required	Restriction(s)
User Exits	Any user exits that are written for and apply to this collector support.	No	User exits should be stored in ADMIN.EXITSR*C.exitpoint*.SOURCE.
Table and Variable Definitions	SAS IT Resource Management table and variable definitions for this collector support.	Yes	Ensure that the tables that you want to be packaged with this collector support are added to the active PDB. Any formula and derived variables are automatically handled by the packaging process.
Documentation	User-written documentation for this new collector. By default the documentation is produced in HTML format.	No	□ Although this is optional, if you are packaging and sharing this user-written collector support, it is recommended that you include documentation on how to use it. □ If you decide to add documentation, then at a minimum, you should place some valid HTML in ADMIN.*collector*.UHTMDFLT.SOURCE. When installing a package, some automatic documentation is produced that will link to your HTML file if it exists. □ If you want to include or link to additional HTML files, you should also place them in the ADMIN.*collector* SAS catalog as source entries and ensure that they all start with 'UHTM'. □ When adding HTML to these source entries, do not to go past column 68, since this can cause problems when installing the package on z/OS.

SAS IT Resource Management enables you to process and reduce data, and to generate reports and exceptions from this ADMIN location. This, prior to packaging, means that you can ensure that your collector-support entities are built, tested, and documented and are ready to package.

Packaging Your Collector-Support Entities

Ensure that SAS IT Resource Management is active, and ensure that the allocated PDB contains the table definitions for your collector support and that the various components that are listed in the previous table are in their correct locations. Then, submit code like the following example, which directs the package to *c:\package.xpt*. (For more information about the %CPPKGCOL macro, see the topic %CPPKGCOL in the SAS IT Resource Management macro reference documentation.)

```
filename pkgfile 'c:\package.xpt';
%CPPKGCOL(PACKFILE=PKGFILE
         ,COLLECTR=collector
         ,TOOLNM=toolnm
         ,_RC=pkgrc);
```

```
%put Return code is &pkgrc
filename pkgfile clear;
```

After this macro runs successfully, you will have a single package file that contains all the collector-support entities. This package can now be used to install your collector-support entities on either this SAS IT Resource Management installation or another. If the SAS IT Resource Management installation is in a different operating environment, then the package should be moved to that operating environment by using binary FTP.

Listing Contents of the Package

To list the contents of the package and see any documentation that has been included, you should ensure that SAS IT Resource Management is active and then submit the %CPRPTPKG macro from the SAS Program Editor window. Here is an example:

```
filename pkgfile 'c:\package.xpt';

%CPRPTPKG(PACKFILE=PKGFILE
         ,DOCLOC=output-location
         ,_RC=pkgrc);

%put Return code is &pkgrc

filename pkgfile clear;
```

After this macro runs successfully, the output location will contain HTML files (see note on z/OS, below) of which one is default.htm. Open this file in a browser. It will contain information about the package, and it will also have a link to any user documentation that might have been included in the package.

For more information about the %CPRPTPKG macro, see the topic %CPRPTPKG in the SAS IT Resource Management macro reference documentation.

Note: On z/OS, the HTML files can be written to the z/OS UNIX File System or to a partitioned data set. If the HTML files have been written to a partitioned data set, then they need to be FTP'd to an alternative location in order to be viewed by means of a Web browser. △

Installing Your Collector-Support Entities

Note: When you install a user-written collector, it is recommended that you make a full backup of your SAS IT Resource Management installation, particularly the destination libref that is specified on the %CPINSPKG macro. △

This document assumes that you will install the package into PGMLIB.

In order to perform the installation successfully, you should ensure that the person who performs the installation has "write" authority to the destination libref that is specified on the DEST= parameter and to the location that is specified in the DOCLOC= parameter. Start SAS IT Resource Management interactively and then choose to 'Exit SAS IT Resource Management Only.' Then, you can run the %CPINSPKG macro from the SAS Program Editor window. (For more information about the %CPINSPKG macro, see the topic %CPINSPKG in the SAS IT Resource Management macro reference documentation.) Here is an example:

```
filename pkgfile 'c:\package.xpt';
%CPINSPKG(PACKFILE=PKGFILE
         ,DEST=PGMLIB
         ,REPLACE=NOREPLACE
         ,DOCLOC=output-location
        ,_RC=pkgrc);
```

```
%put Return code is &pkgrc
filename pkgfile clear;
```

After this macro runs successfully, your collector-support entities will be installed to the destination and ready for processing data and for reporting.

Details

There are four stages in generating and installing a collector package. The most complicated task is actually writing and setting up the collector-support entities. The degree of difficulty depends on the features that the collector-support writer decides to implement and the complexity of the data source format.

Creating Your Collector-Support Entities

The following describes the collector-support entities that can be packaged by the %CPPKGCOL macro. Some of the entities are required; that is, they must exist before the package will be created. The others are optional. At the start of each entity description, a small summary discusses whether the entity is required and where it should be stored in order to be packaged by the %CPPKGCOL macro.

Staging Code

- Required
- Location - ADMIN.*collector*.*toolnm*.SOURCE | SCL

For information about writing staging code, see the "Use the Generic Collector Facility" topic of the "Administration: Extensions to SAS IT Resource Management" section of the online help for SAS IT Resource Management. Although that document discusses writing a collector support that uses the Generic Collector Facility, the processes are almost identical. The following table lists the differences.

Table 20.2 Staging Code Differences

Staging Code Feature	Collector Support Using the Generic Collector Facility	Collector Support Not Using the Generic Collector Facility
Collector Value	Restricted to GENERIC	User defined, although must begin with a 'U' and should be no longer than 8 characters (letters and numbers).
Tool Name Value	SASDS	User defined, although normally this is left as SASDS.
Rawdata Reference	The file reference to the raw data can be anything because the user writes and controls the submission of this code.	%CPPROCES automatically sets the value of the &CPRAWDAT macro variable to the path and name of the external file as specified in the RAWDATA= parameter. This macro variable can be used to obtain and set the RAWDATA fileref.
Location of staging code	No fixed location as long as it is executed prior to %CPPROCES.	Should be created in ADMIN.*collector*.*toolnm*.SOURCE. %CPPROCES will use the COLLECTR= and TOOLNM= parameters to locate the staging code automatically.

Staging Code Feature	Collector Support Using the Generic Collector Facility	Collector Support Not Using the Generic Collector Facility
Execution of staging code	The staging code has to be included and run prior to %CPPROCES.	The staging code is automatically included by %CPPROCES by using the COLLECTR= and TOOLNM= parameters to locate the staging code.
Process Wizard Functionality	Restricted to using default functionality for GENERIC collectors.	When packaging user-written collector support, you can set which options are made available via the Process Wizard for this COLLECTR/ TOOLNM pair. When installed, that functionality will be available in the Process Wizard.

Defining Tables and Variables

- Required
- DICTLIB

For information about defining tables and variables, see the "Use the Generic Collector Facility" topic of the "Administration: Extensions to SAS IT Resource Management" section of the online help for SAS IT Resource Management.

Formats Optional

- Optional
- ADMIN.CPFMTS

User-written formats and informats that are used in either staging code, variable definitions, report definitions, etc., for this collector should be placed in ADMIN.CPFMTS.

For information about creating user-defined formats and informats, see the "Use the Generic Collector Facility" topic of the "Administration: Extensions to SAS IT Resource Management" section of the online help for SAS IT Resource Management.

Duplicate-Data Checking

- Optional
- ADMIN.*collector*.CPDUPCHK.SOURCE

A good understanding of how duplicate-data checking works (as implemented by the %CPDUP* macros) in %CPPROCES is required. For information about this, see the "User-Generated Table Definitions and Non-MXG-Based Staging Code" topic in "Appendix 4: Working with Duplicate Data" in the macro reference.

There are two stages to implementing duplicate-data checking:

1. Create an ADMIN.*collector*.CPDUPCHK.SOURCE SAS catalog source entry that contains a %CPDUPCHK macro with your recommended default values.
2. Modify your staging code to incorporate the duplicate data macros and functionality. For detailed information about this, see "Using the Generic Collector Facility" on page 158.

Reports

- Optional
- ADMIN.ITSVRPT report folder

Ensure that all report definitions that you write for this user-written collector support are stored in the ADMIN.ITSVRPT report folder, because this is the location that will be searched when creating the package.

If you define any source reports, then store that source code in a SAS catalog source entry. It does not matter where these source entries are stored as long as they are available when generating the reports and running the %CPPKGCOL macro.

Exception Rules

- Optional
- ADMIN.CPXRULE report folder

Ensure that all exception rule definitions that you write for this user-written collector support are stored in the ADMIN.CPXRULE exception folder, because this is the location that will be searched when creating the package.

Documentation

- Optional
- ADMIN.*collector*.UHTMDFLT.SOURCE (and possibly other UHTM*.SOURCE entries)

When the %CPRPTPKG or %CPINSPKG macros run, they automatically generate HTML documentation that provides information about the contents of the package. In addition, the authors who created the collector-support entities can optionally supply their own HTML documentation that will be packaged by the %CPPKGCOL macro and generated along with the automatic documentation.

If you decide to add documentation, then at a minimum you should place some valid HTML in ADMIN.*collector*.UHTMDFLT.SOURCE. When the automatic documentation is generated, it will automatically link to this file. The contents of this UHTMDFLT HTML file are up to you. The only requirement is that they must contain valid HTML. If you want to add other HTML files that are linked to from this default location, then create them in ADMIN.*collector*.UHTM*xxxx*.SOURCE, where *xxxx* is any character valid for a SAS catalog source entry name. All the HTML files that are packaged will be placed in a single directory by the %CPRPTPKG or %CPINSPKG macros.

User Exits

- Optional
- ADMIN.EXITSRC.*exit*.SOURCE

All user exit routines should be stored in ADMIN.EXITSRC.exitpoint.SOURCE. For more information about process exits, see “Overview of Using Process Exits” on page 442.

Packaging the Collector-Support Entities

SAS IT Resource Management enables you to run and use the collector support that you have written while it is in the ADMIN library. However, you might want to package these collector-support entities in order to install them at other locations. Alternatively, you might want to install them into SITELIB or PGMLIB so that other PDB’s can easily be created. Additionally, after installation by the %CPINSPKG macro, your collector support will be inserted into the Process Wizard functionality.

Before packaging the entities that you have written for this collector support, you should test them in their ADMIN location to be certain that everything is working.

The packaging macro %CPPKGCOL knows where to look for the collector-support entities that you have written. It will automatically package the staging code, formats and informats, table and variable definitions (including formula and derived variables), exception rule and report definitions, and documentation and user exits. In addition to this information, you can specify several other parameters that, after the package is installed, will dictate the options/functions that will be available by means of the Process Wizard and also what operating environments the user-written collector

support will run on. For more information about the %CPPKGCOL macro and its parameters, see the topic %CPPKGCOL in the SAS IT Resource Management macro reference documentation.

Package Example

```
filename pkgfile 'c:\package.xpt';    (1)

  %CPPKGCOL(PACKFILE=PKGFILE          (2)
           ,COLLECTR=UMYDATA          (3)
           ,TOOLNM=SASDS              (4)
           ,OPSYS=WIN OSYS MVS        (5)
           ,RAWDIR=YES                (6)
           ,DUPMODE=OPTIONAL          (7)
           ,_RC=pkgrc);               (8)

  %put Return code is &pkgrc

  filename pkgfile clear;             (9)
```

Explanation of the above lines of code:

1 This first statement allocates a file that will contain the package. Internally, this file will be a SAS transport file that is created by the PROC CPORT macro; however, it contains additional information and is constructed in a format that is used by the %CPINSPKG macro to install the collector-support entities in the specified destination.

2 PACKFILE= refers to the fileref that is already allocated.

3 COLLECTR= specifies the value that you used for your collector support. This value, along with the value of the TOOLNM= parameter, assists in extracting the appropriate collector-support entities and inserting them into the package file.

4 TOOLNM= this is the value for tool name that you used for your collector support. This value, along with the value of the COLLECTR= parameter, assists in extracting the appropriate collector-support entities and inserting them into the package file.

5 OPSYS= is optional and provides the package with information regarding the operating environments on which this collector support has been tested. In the example shown, this user collector support has been written and tested to run on Windows operating environments, OSYS (UNIX operating environments), and MVS (z/OS mainframe). These three values are the only valid options. If not specified, the default value that is passed to this parameter is based on the operating system where the package was created. This means that the collector support will run ONLY on that system.

6 RAWDIR= is one of several parameters that determine the behavior of the Process Wizard for this user-written collector support. The RAWDIR= parameter determines whether the Process Wizard window that is used to point to the raw data enables the user to point to a directory rather than an individual file. By default, this option is set to NO, which means that the Process Wizard will allow only an individual file to be specified. However, if the staging code has been written in such a way that it can process multiple files from a directory, then this option should be set to YES.

7 DUPMODE=OPTIONAL also determines the behavior of the Process Wizard. If the user-written collector support has been written to implement the duplicate-data-checking macros, then setting this value to OPTIONAL means that the Process Wizard will enable the user to set the DUPMODE= parameter.

8 _RC= enables the user to specify the name of a macro variable that will contain the value of the return code from this macro.

9 This final statement releases the fileref that was originally allocated for the package.

Note: After this macro has been run, the package file can be FTP'd in binary mode to any operating environment and installed. (See "Installing Package" below.) △

Checking the Contents of the Package

If you are provided with a package file and you are not familiar with its contents, then you can use the %CPRPTPKG macro to generate HTML documentation that describes the contents of the package.

At a minimum, the package will produce a default.htm file that contains information about the contents of the package and when the package was created. This information is generated automatically by the %CPRPTPKG macro. If the user who created the package also supplied documentation, that documentation is generated and is linked to automatically from the default.htm page.

This macro does NOT install any of the collector-support entities. This macro merely unloads the documentation from the package.

Example of Reporting on Package

```
filename pkgfile 'c:\package.xpt';  (1)

  %CPRPTPKG(PACKFILE=PKGFILE        (2)
           ,DOCLOC=C:\DOCLOC        (3)
           ,_RC=rptrc);             (4)

  %put Return code is &rptrc

  filename pkgfile clear;           (5)
```

Explanation of the above lines of code:

1 This first statement allocates a file that will contains the package.

2 PACKFILE= refers to the fileref that is already allocated to the package file.

3 DOCLOC= points to an existing directory (UNIX, Windows, z/OS (the z/OS UNIX File System)) or partitioned data set (z/OS) in which this macro will place the HTML documentation.

4 _RC= enables the user to specify the name of a macro variable that will contain the value of the return code from this macro.

5 This final statement releases the fileref that was originally allocated for the package.

Installing the Package

Before installing any package to a SAS IT Resource Management installation, make sure that you have a current backup of that installation that can be restored, in case there are any problems. Also, you must have write access to the destination library.

One of the checks that are performed by the %CPINSPKG macro is to compare the SAS IT Resource Management software version and dictionary version on which the package was created with that of the target installation. If they differ, then the package will not be installed unless the appropriate FORCE= option is specified. A package that has different software or dictionary version cannot be guaranteed to be installed correctly (or even work correctly, if installed).

Prior to installing the package, you need to decide where in the SAS IT Resource Management installation you want the collector support to be installed. At present, you can install to ADMIN, SITELIB, or PGMLIB. Each of these locations has its advantages and disadvantages, which are covered later in this document.

Example of Installing a Package

```
filename pkgfile 'c:\package.xpt'; (1)

  %CPINSPKG(PACKFILE=PKGFILE       (2)
           ,DEST=PGMLIB            (3)
           ,REPLACE=NOREPLACE      (4)
           ,DOCLOC=output-location (5)
           ,_RC=pkgrc);            (6)

  %put Return code is &pkgrc

  filename pkgfile clear;          (7)
```

Explanation of the above lines of code:

1. This first statement allocates a file that will contain the package. Internally, this file is a SAS transport file that is created by the %CPPKGCOL macro. However, it contains additional information and is constructed in a format that is used by the %CPINSPKG macro to install the collector-support entities to a destination.
2. PACKFILE= refers to the fileref that is already allocated to the package file.
3. DEST= points to the destination libref into which the collector-support entities will be installed.
4. REPLACE= determines what should happen if any collector-support entity for this collector support already exists. See the discussion of this option, below, for more information.
5. DOCLOC= points to an existing directory or folder (on UNIX, Windows, or the z/OS UNIX File System) or a partitioned data set (on z/OS) in which this macro will place the HTML documentation.
6. _RC= enables the user to specify the name of a macro variable that will contain the value of the return code from this macro.
7. This final statement releases the fileref that was originally allocated for the package.

Installed Location of Collector-Support Entities

The following table shows the source location of the collector-support entities; that is, it shows the location in which they should be created. It also shows the destination of the collector-support entities, when installed by the %CPINSPKG macro, based on the different values of the DEST= parameter.

Table 20.3 Source Locations for Collector-Support Entities

Entity	Source	Destination
Staging Code	ADMIN.*collector*.*toolnm*.SOURCE|SCL	ADMIN: ADMIN.*collector*.*toolnm*.SOURCE|SCL SITELIB: SITELIB.*collector*.*toolnm*.SOURCE|SCL PGMLIB: PGMLIB.*collector*.*toolnm*.SOURCE|SCL
Duplicate-Data Checking	ADMIN.*collector*.CPDUPCHK.SOURCE	ADMIN: ADMIN.*collector*.CPDUPCHK.SOURCE SITELIB: SITELIB.*collector*.CPDUPCHK.SOURCE|SCL PGMLIB: SITELIB.*collector*.CPDUPCHK.SOURCE|SCL
Table Definitions	DICTLIB.TABINFO	ADMIN: DICTLIB.TABINFO SITELIB: DICTLIB.TABINFO PGMLIB: DICTLIB.TABINFO PGMLIB.TABINFO
Variable Definitions	DICTLIB.VARINFO	ADMIN: DICTLIB.VARINFO SITELIB: DICTLIB.VARINFO PGMLIB: DICTLIB.VARINFO PGMLIB.VARINFO
Formats and Informats	ADMIN.CPFMTS	ADMIN: ADMIN.CPFMTS SITELIB: SITELIB.CPFMTS PGMLIB: PGMLIB.CPFMTS
Reports	ADMIN.ITSVRPT	ADMIN: ADMIN.ITSVRPT SITELIB: SITELIB.ITSVRPT PGMLIB: PGMLIB.ITSVRPT
Exception Rules	ADMIN.CPXRULE	ADMIN: ADMIN.CPXRULE SITELIB: SITELIB.CPXRULE PGMLIB: PGMLIB.CPXRULE
User Process Exits	ADMIN.EXITSRC	ADMIN: ADMIN.EXITSRC SITELIB.EXITSRC PGMLIB: PGMLIB.EXITSRC

Discussion of the REPLACE= Option on %CPINSPKG

The REPLACE= option has four possible settings: REPLACE, NOREPLACE, APPEND, and MERGE. NOREPLACE is the default. The behavior of each of the REPLACE= options depends on the type of entity that is being installed to a destination. For example, a SAS data set type will be handled differently from a SAS catalog source entry. The REPLACE= options facilitate the ability to install collector-support entities from a newer version of a package that might have been updated or to which additional entities might have been added.

The following table describes how each of the collector-support entities is affected by the different REPLACE= options.

Table 20.4 Effect of REPLACE= Option on Collector-Support Entities

Entity	Entity Type	REPLACE= Option	Description of Behavior
Staging Code, CPDUPCHK Source Entry	SAS Catalog Source Entry	NOREPLACE	If the destination catalog source entry does not exist, then it is created. If the destination catalog source entry already exists, then the %CPINSPKG macro stops and does not perform the installation.
		REPLACE, MERGE	If the destination catalog source entry already exists, it is removed and the new one from the package is installed in its place.
		APPEND	If the destination catalog source entry already exists, it is NOT overwritten, and the new one is NOT installed, but the installation continues.
Report Definitions, Exception Rule Definitions, PDB Table and Variable Definitions	SAS Data Set	NOREPLACE	If the destination SAS data set does not exist, it is created. If the destination SAS data set exists but has no conflicting definitions, then the new definitions are appended. If there are conflicting definitions, then processing stops.
		REPLACE	If the destination SAS data set exists and already contains definitions that could conflict with the package, then they are removed before the new definitions from the package are added.
		APPEND	If the destination SAS data set exists and already contains definitions that conflict with those that are coming from the package, only the non-conflicting ones are added from the package.*
		MERGE	If the destination SAS data set exists and contains definitions that conflict with those that are coming from the package, all new definitions are added, overwriting any existing ones if necessary.*

Entity	Entity Type	REPLACE= Option	Description of Behavior
Exit Source Catalog, Format and Informats	SAS Catalog	NOREPLACE	If the destination SAS catalog does not exist or it does exist but contains no conflicting entries, then the %CPINSPKG macro installs the new entities. If there are any conflicting SAS catalog source entries, then the installation stops.
		REPLACE	If the destination SAS catalog already exists and contains SAS catalog source entries that were previously installed for this collector support, then they are removed from the destination and %CPINSPKG installs all the SAS catalog source entries from the package.
		APPEND	If the destination SAS catalog already exists and contains SAS catalog source entries that conflict with those in the package, then the conflicting entries from the package are NOT installed by the %CPINSPKG macro.
		MERGE	If the destination SAS catalog already exists and contains SAS catalog source entries that conflict with those in the package, then only the conflicting entries are removed from the destination and %CPINSPKG installs all the SAS catalog source entries from the package.

* Conflicting observations in SAS data sets are determined by using either unique index values or, if the SAS data set does not have an index, the sorted-by variables.

Installing to PGMLIB

Installing to PGMLIB will cause the collector-support entities to appear as though they were supplied with the product.

- Advantages
  - The Process Wizard will work with this new collector support.
  - Anyone who uses this SAS IT Resource Management installation will be able to process and reduce data for this collector support.
  - Anyone who uses this SAS IT Resource Management installation will be able to create their own PDB by using this collector support.
  - SAS IT Resource Management tables for this new collector support will be available to anyone who uses this installation of SAS IT Resource Management through the ADD TABLE functionality through either the GUI or the %CPDDUTL macro.
  - Collector-support entities such as formats, informats, reports, and exception rules will be accessible to anyone who uses this installation of SAS IT Resource Management.

- Read-only access to PGMLIB by users prevents the collector-support entities from being modified.

- Disadvantages
  - When SAS IT Resource Management is reinstalled or upgraded, then the package will have to be reapplied.
  - Modifying the collector-support entities after installation is more difficult because they were placed in PGMLIB; however, if modifications are likely, then PGMLIB is probably not the appropriate destination.

Installing to SITELIB

Installing to SITELIB will enable anyone with read access to this SITELIB to use these collector-support entities.

- Advantages
  - The Process Wizard will work with this new collector support.
  - Anyone who uses this SAS IT Resource Management installation with this SITELIB will be able to process and reduce data for this collector support.
  - Collector-support entities such as formats, informats, reports and exception rules will be accessible to anyone who uses this SITELIB when invoking SAS IT Resource Management.
  - Read-only access to SITELIB by users prevents the collector-support entities from being modified, although this can be changed by users through the %CPSTART macro.

- Disadvantages
  - To construct other PDBs for data from this collector, you would have to make a %CPDDUTL GENERATE SOURCE run to create the necessary %CPDDUTL control statements for the tables and variables. These control statements could then be used to add the tables to other PDBs. These PDBs would have to use the same SITELIB in order to pick up the other collector-support entities.
  - SAS IT Resource Management tables for this new collector support will NOT be available through the ADD TABLE functionality through either the GUI or the %CPDDUTL macro.
  - When SAS IT Resource Management is reinstalled or upgraded, then the package will have to be reapplied (or the %CPSITEUP macro can be used to copy the entities from the old SITELIB).

Installing to ADMIN

Installing to ADMIN will enable anyone with read access to this PDB to use these collector-support entities.

- Advantages
  - The Process Wizard will work with this new collector support.
  - ADMIN is useful when you are in the early stages of modifying or upgrading the collector-support entities and repackaging.
  - When SAS IT Resource Management is reinstalled or upgraded, then the package will NOT have to be reapplied.

- Disadvantages
  - To construct other PDBs for data from this collector, you would have run the %CPINSPKG macro and install the collector-support entities to a different ADMIN, or to SITELIB or PGMLIB.

- Only when this PDB is active will anyone be able to use this collector's support entities.
- SAS IT Resource Management tables for this new collector support will NOT be available to anyone who uses this installation of SAS IT Resource Management through the ADD TABLE functionality through either the GUI or the %CPDDUTL macro.

Using Process Exits

Overview of Using Process Exits

The %CMPROCES, %CPPROCES, %CSPROCES, and %CWPROCES macros have exit points for which you can provide exit routines by using the EXITSRC= parameter. For more information, see the following topics:

- "General Information on Exit Points" on page 442
- "Exit Points for %CPPROCES and %CMPROCES" on page 447
- "Exit Points for %CSPROCES and %CWPROCES" on page 457.

General Information on Exit Points

This section explains the exit points available during the process phase of SAS IT Resource Management and guides you in selecting and coding exit points, if you want to use them.

Exit points (sometimes referred to as exits) are specific points, during the processing of data, at which control is transferred to a user-written routine, if a user-written routine is available. The process subsystem used by SAS IT Resource Management allows for "exit" processing. What this means is that during the processing of raw data into the detail level of the PDB, there are specific points where control is given to user-written code. Users can use exit points to filter or create data or to meet other custom requirements.

Process exits can be used to provide various types of functionality, such as

- to create additional data sets (such as accounting charges) on the same pass of the data
- to add various reporting capabilities in the midst of the process run
- to extend the processing power of the installed software.

CAUTION:

In any situation where you have the ability to affect and alter the intended flow and execution of code, there is the possibility of error. As such, SAS Institute issues the following warnings and disclaimers with regard to the use of exits with SAS IT Resource Management. △

Warnings and Disclaimers:

- The sample code shown in these "exit" files, while accurate, is subject to change. Any changes to this code that affect user exits will be reported to you by means of the normal usage notes that are packaged with maintenance.
- It is impossible for SAS IT Resource Management to anticipate or to be responsible for the processing that occurs within an exit. As such, exit code that

you provide, if any, is simply included at the documented points. Any generation of reports during the actual execution of the exits is left up to you.

- Additionally, it is possible to put code in exits that causes the data that will be stored in the SAS IT Resource Management PDB to be invalid. SAS cannot be responsible for invalid PDB data that is caused by user-written exits.

Note: Use exits with great caution. △

Overview

Before the details of the individual exit points are discussed, some general questions will be addressed:

Each of these topics will be explored, and then more detailed information about the individual exit points will be presented.

How Do I Tell SAS IT Resource Management to Use Process Exits?

- An optional parameter, EXITSRC=, is available on the %CMPROCES, %CPPROCES, %CWPROCES, and %CWPROCES macros. The value to which EXITSRC= is set is the location of the exit routine (source code). The location can be either an external file/directory or the name of a SAS catalog.
- If EXITSRC= is not specified on the %CMPROCES, %CPPROCES, %CSPROCES, or %CWPROCES macro, then the process step is executed without regard to exit points.
- If EXITSRC= is specified on the %CMPROCES, %CPPROCES, %CSPROCES, or %CWPROCES macro, SAS IT Resource Management first checks to see if the specified value is a valid external library (partitioned data set for z/OS; directory for UNIX and Windows). If not, it then checks to see if the specified value is a SAS catalog. If the specified value cannot be verified as either an external library or a SAS catalog, then the process macro terminates with an appropriate error message.
- The SAS IT Resource Management GUI enables you to specify the EXITSRC= parameter as an advanced feature of the Process/Reduce Wizard on UNIX and Windows, and in the Process Rawfiles window of the SAS IT Resource Management GUI for z/OS.

Which Exit Points Should I Use?

This is not a simple question because the answer depends upon what you want to accomplish. The best way to determine the exit points to use is to consult the table that lists all of the exit points by name, includes a description of the locations in the supplied code, and includes the purpose that the exit point could serve.

You should then also look at the relevant code example to make sure that you understand the context in which the exit point occurs, the purpose that you want to accomplish, and any implications that this might have on the regular processing of data into the PDB.

How Do I Write My Exit Routines?

- Exits exist as SAS source code, either in an external library member or as a .SOURCE member in a SAS catalog. The naming convention is proc*nnn*, where *nnn* is the exit point number (with leading zeros). For a complete list of portable exit names and descriptive information, see below. The exits are numbered by 5's so that any future exit points that are added can still remain in numerical order.

  Specific examples, based on the code for exit point proc005, are as follows:

  - *SAS Catalog*

    Assume that the exits are stored in the catalog SASUSER.CPEXITS. Then the source for the first exit point is in the member:

    ```
    SASUSER.CPEXITS.PROC005.SOURCE
    ```

    In this case, the value for EXITSRC= is *SASUSER.CPEXITS*.

  - *External Library*

    Clearly this is a host-specific issue. SAS IT Resource Management handles this in the following manner:

    z/OS: The exits must reside in a PDS. Assuming that the exits are stored in USERID.CPE.EXITS, then the location of the source code for the first exit point is USERID.CPE.EXITS(PROC005).

    In this case, the value for EXITSRC= is *USERID.CPE.EXITS*.

    UNIX: The exits must reside as flat files in a directory. The filename that is used for the SAS IT Resource Management exit source should be entirely in lowercase. Assuming that the exits are stored in */u/cpe/exits*, then the location of the source code for the first exit point is */u/cpe/exits/proc005.sas*.

    In this case, the value for EXITSRC= is */u/cpe/exits*.

    Windows: The exits must reside as flat files in a directory. The filename that is used for the SAS IT Resource Management exit source should be entirely in lowercase. Assuming that the exits are stored in *c:\cpe\exits*, then the location of the source code for the first exit point is *c:\cpe\exits\proc005.sas*.

    In this case, the value for EXITSRC= *c:\cpe\exits*.

- Inside the process step, SAS IT Resource Management verifies whether or not an exit exists in the location that is specified by EXITSRC=. If so, SAS IT Resource Management includes the code for the exit when the process code is built in the preview buffer (and later submits the combined process code).
- The actual code within an exit is nothing more than SAS language statements, as appropriate to the context in which the exit source code is included. Thus for some exits, the only valid contents might be data set options. For other exits, almost any valid SAS statement is valid. Again, each exit point needs to be taken in context to determine what is valid.
- It is important to remember that the exit routines are treated as source code that is located and included for each exit point. SAS IT Resource Management does not keep track of any activity with regard to the actual execution (or non-execution) of the exit.
- When the process step ends, a report is generated that lists the exit points that were included.
- It is strongly recommended that each exit be coded as a macro and that the macro be given the same name as the exit point and then invoked. By doing so, it is easier to determine where any errors may have occurred (for example, if there is a

syntax error in an exit). The following is a simple example of the recommended way to code the proc005 exit point:

```
%macro proc005;
   /* actual exit-specific code goes here */
%mend;
%proc005
```

CAUTION:
Do not place a semicolon after the invocation of the macro (in this example, the %proc005 that follows the %mend;). △

- If an exit needs to test the value of a macro variable (using %IF) or otherwise needs to be conditionally invoked, it is the responsibility of the exit to handle these matters. Again, the recommended way of doing this is to code the exit as a macro that has the same name as the exit point and then to invoke the macro immediately.

  Here is an example that shows conditional invocation of the exit routine:

```
%macro proc005;
     %if &SYSENV=BACK %then %do;
       /* actual exit-specific code goes here */
       %end;
%mend;
%proc005
```

CAUTION:
Do not place a semicolon after the invocation of the macro (in this example, the %proc005 that follows the %mend;). △

Are There Any Special Variables Available to My Exits?

- The global macro variable CPTABLES is available to all of the exit points. The value of CPTABLES is a list of the names of all of the tables that are being processed in this invocation of the process macro. The table names are separated by white space.
- CPARCTAB - a macro variable that contains a space-delimited list of all tables whose data is to be archived in the process run.
- The global macro variable CPTABLE is available to each exit that occurs on a per-table or per-observation basis. The value of CPTABLE is the name of the table whose data is currently being processed.
- CPARCTB - as each table is being processed, the CPARCTB macro variable is set to *Y* (for Yes) or *N* (for No), depending upon whether or not this specific table is to be archived.
- The global macro variable CPBYVAR is available to each exit that occurs on a per-table or per-observation basis. The value of CPBYVAR is the list of BY variables for the table whose data is currently being processed. The BY-variable names are separated by white space.
- The global macro variable CPCUTDT is available to exit points proc150 through proc225. The value of CPCUTDT is the datetime constant that is used for the age-limit trimming of the detail-level data set. That is, the value of CPCUTDT is the earliest date that is allowed for observations that remain in the detail-level data set.

Are There Any Considerations That I Need to Be Aware Of?

- If you want to delete observations in the final DATA step, it is recommended that you use exit point proc040. If you perform this activity at a later point (such as at exit point proc200), you run the risk of corrupting the data dictionary because accumulator values might have been incremented to include these (now deleted) observations.
- Any macro variables that are created in an exit point will be *local* macro variables and will not be accessible to other exit points. If an exit needs to set a macro variable for use in a later exit point, then it is your responsibility to declare the macro variable as GLOBAL (by using a %GLOBAL macro statement).
- The use of DATA step views has several important implications:
  - Whenever a DATA step view is defined, any additional data sets that are created in the same data step are not actually built until the view is executed. Therefore, exit points proc010 through proc085 are not actually executed until the view is executed. For the Generic Collector Facility, this means that the view is actually executed when the staged data is sorted.
  - As a side effect of this, if you were to create an additional output data set by using proc020 in the cases that were outlined previously, the data set would not exist at the time that exit point proc090 is encountered.
  - For more information about working with DATA step views, see the section about SAS DATA step views in the *SAS: Language Reference: Concepts* book for your release of SAS.
- The use of exit points proc035 and proc165 to add additional data sets for SET processing can have several subtle implications:
  - For proc165, because the data sets in the SET statement are combined by using the BY statement that follows, it is important that if you add any additional SET data sets, they should already be sorted by using this BY-variables list. The BY list can be found in the macro variable CPBYVAR.
  - For proc165, counts of total observations, counts of observations with duplicate BY variables, counts of old observations, and counts of new observations, as well as the earliest and latest datetime values in the table, will be affected by the data in data sets that you might add to the SET statement. If you do not want the additional data sets to affect the counts, use the exit points preceding these specific points in processing, along with the IN= variables for the table and new (staged) data, to limit the scope to the desired input data sets.
  - For proc035 and proc165, the data that is written to the table will include observations from all data sets in the SET statement. If you add additional data sets to the SET statement and do not want the additional data to be written to the table, use the exit point immediately before the OUTPUT statement to limit the scope to the desired input data sets.

Information about Specific Exit Points

General information:
The general flow of the generated code for process is as follows:

MACRO setup section	Macro variables are defined as global, and macro variable assignments are performed for the SAS IT Resource Management macro variables shown at the beginning of the sample code.
DATA step section	To perform variable renames, to add variables SHIFT and HOUR, and to assign values to SHIFT, HOUR, and LSTPDATE. This could

	be referred to as the final staging of the data. For z/OS COLLECTR=GENERIC setting, the output is a SAS DATA step view.
PROC SORT section	The new (staged) data is sorted by using the BY variables for this table. Any duplicate observations are deleted.
PROC SORT section	If the detail-level data set is not sorted by using the BY variables, then this PROC SORT step is built and executed in order to sort the detail-level data set by using the BY variables from the data dictionary.
DATA step section	To "merge" (actually SET-BY) the new data and the detail-level data. This updates the detail level of the PDB.

Exit Points for %CPPROCES and %CMPROCES

This section covers the process exit points for the collectors that use the %CPPROCES and %CMPROCES macros.

Table of Process Exit Points

Note: For each exit point that is mentioned in this table, the information in the table is designed to be read in conjunction with the exit point's position in the sample code that follows the table.

In this table, "open" means not in the middle of a statement, DATA step, or PROC step. △

Table 20.5 Process Exit Points

Exit Number in Code Example	Exit Point Name	Placement	Purpose	Frequency of Execution
#1	proc005	The very first exit point, before the %LET CPTABLE statement but after the %LET CPTABLES= statement	Enables any initial one-time open code processing	Once per process run
#2	proc010	Directly before the first DATA step	Enables setting of options and unconditional pre-processing before the final staging	Once per table
#3	proc015	Directly after the KEEP= list for the output staged data set	Enables the user to specify data set options	Once per table
#4	proc020	Directly before the closing semicolon in the DATA statement	Enables the user to add his/her own output data set	Once per table
#5	proc025	Directly after the ATTRIB statement in the first DATA step	Enables user-specified attributes and open DATA step code	Once per observation

Exit Number in Code Example	Exit Point Name	Placement	Purpose	Frequency of Execution
#6	proc030	Directly before the closing parenthesis of the data set OPTIONS clause in the SET statement for the staged data set	Enables user to override data set options	Once per observation
#7	proc035	Directly after the closing parenthesis of the data set OPTIONS clause in the SET statement for the staged data set	Enables the user to specify additional data sets for SET processing	Once per observation
#8	proc040	Directly after the SET statement in the first DATA step	Open DATA step code point for user	Once per observation
#9	proc045	Directly before the DELETE statement for observations that have missing DATETIME values in the first DATA step	Enables user to handle observations that have missing DATETIME values	Once per observation that has a missing DATETIME value
#10	proc050	Directly after the HOUR variable has been set in the first DATA step	Enables user to override the setting of HOUR	Once per observation
#11	proc055	Directly after LSTPDATE has been set to the current datetime stamp in the first DATA step, if LSTPDATE is currently missing	Enables user to override setting of LSTPDATE	Once per observation in which LSTPDATE has a missing LSTPDATE value
#12	proc060	Directly before the SHIFT variable calculations begin	Open DATA step code point for user	Once per observation
#13	proc065	Directly after the SHIFT variable has been set by using the shift matrix, which happens only if the SHIFT variable is not set in the staged data set	Enables user to override the shift matrix	Once per observation that has a blank SHIFT variable value
#14	proc070	After the SHIFT variable value has been set, regardless of whether it was set in the staged data set; this occurs before the Holiday list	Enables user to override the SHIFT variable value, prior to calculation that uses the Holiday list	Once per observation
#15	proc075	After the SHIFT value has been set, if the date matches an entry in the Holiday list	Enables user to SHIFT variable value, if set by the Holiday list	Once per observation that has been set by Holiday list

Exit Number in Code Example	Exit Point Name	Placement	Purpose	Frequency of Execution
#16	proc077	Inside the IF _LAST clause, directly before the macro variable CPLDATM is populated, with the latest value of DATETIME in the incoming data	Enables user to perform any final actions in the staging view DATA step	Once per table
#17	proc080	Directly before the OUTPUT statement for the staged data set / DATA step view in the first DATA step	Final open DATA step code point available before the observation is written to the staged data set / DATA step view	Once per observation
#18	proc085	Directly after the OUTPUT statement for the staged data set / DATA step view in the first DATA step	Enables user OUTPUT statements and open DATA step code	Once per observation
#19	proc090	Directly before the PROC SORT of the staged data	Enables pre-sort processing of the staged data	Once per table
#20	proc095	Directly before the NODUP clause of the PROC SORT statement for the staged data	Enables user to specify additional sort options	Once per table
#21	proc097	Directly before the BY statement of the PROC SORT statement for the staged data	Enables user to override the default BY list for the sort of the staged data	Once per table
#22	proc100	Directly after the RUN statement for the PROC SORT of the staged data	Enables post-sort processing of the staged data	Once per table
#23	proc105	Directly after the PROC SORT of the staged data, if there is no PROC SORT of the detail data	Only invoked if the detail data set is not being sorted	Once per table, if detail data will not be sorted
#24	proc110	Directly after the PROC SORT of the staged data, if there is a PROC SORT of the detail data	Only invoked if the detail data set is being sorted; enables pre-sort code for detail	Once per table, if detail data will be sorted
#25	proc115	Directly before the closing semicolon in the PROC SORT statement for the detail data	Enables user to specify additional sort options	Once per table

Exit Number in Code Example	Exit Point Name	Placement	Purpose	Frequency of Execution
#26	proc120	Directly after the PROC SORT of the detail data, if there is a PROC SORT of the detail data	Invoked only if the detail-level data set was sorted; enables post-sort code for detail	Once per table, if detail data was sorted
#27	proc125	Directly before the "merge" DATA step	Enables setting of options and open DATA step code prior to the merge of the detail and staged data	Once per table
#28	proc130	Directly before the closing parenthesis on the data set options for the detail data set	Enables user to override data set options	Once per table
#29	proc135	Directly before the closing semicolon in the DATA statement	Enables the user to add his/her own output data set	Once per table
#30	proc140	Directly before the END statement in the IF _N_=1 block	Enables first-pass-only code	Once, during the first pass of the DATA step
#31	proc145	Directly after the last ATTRIB statement in the "merge" DATA step, and before the SET statement for the detail and staged data sets	Enables user-specified attributes and open DATA step code	Once per observation
#32	proc150	Directly after the datetime (age-limit trimming) value in the WHERE clause for detail	Enables user to override/ extend age-limit trimming	Once per observation
#33	proc155	Directly before the closing parenthesis in the data set OPTIONS statement in the SET statement for the detail data set	Enables user to override data set options	Once per observation
#34	proc160	Directly before the closing parenthesis in the data set OPTIONS statement in the SET statement for the staged data set	Enables user to override data set options	Once per observation
#35	proc165	Directly after the SET clause for the staged data	Enables user to include his/her own data set in the SET data sets for BY (merge) processing	Once per observation

Exit Number in Code Example	Exit Point Name	Placement	Purpose	Frequency of Execution
#36	proc170	Before the closing semicolon in the SET statement	Enables the user to override or add SET options	Once per observation
#37	proc175	After the BY statement	Open code point; this is the recommended exit point to use for deleting observations	Once per observation
#38	proc180	Duplicate BY variable observation encountered	Enables user to handle observations that have duplicate BY variable values	Once per observation encountered with duplicate BY variable values
#39	proc185	Earlier datetime encountered	Enables user code to be executed each time a new (earliest) datetime stamp is found in the DATA step. (The earliest datetime stamp value is stored in the variable _oet.)	Once per observation that has an earlier datetime stamp than any previous observations
#40	proc190	Later datetime encountered	Enables user code to be executed each time a new (latest) datetime stamp is found in the DATA step. (The latest datetime stamp value is stored in the variable _olt.)	Once per observation that has a later datetime stamp than any previous observations
#41	proc195	When the observation is found to have come from the detail-level data set	Enables the user code to be executed each time an output observation is found to have come from the detail data set. (A count of these observations is accumulated in the variable _oldobs.)	Once per observation read from the detail-level data set
#42	proc200	When the observation is found to have come from the new (staged) data set	Enables user code to be executed each time an observation is found to have come from the new (staged) data set. (A count of these observations is accumulated in the variable _addobs.)	Once per observation read from the staged data set

Exit Number in Code Example	Exit Point Name	Placement	Purpose	Frequency of Execution
#43	proc205	Directly before the OUTPUT statement in the "merge" DATA step	Enables the user to modify the observation before it is finally written to the detail level	Once per observation
#44	proc210	Directly after the OUTPUT statement in the "merge" DATA step	Enables the OUTPUT statements for any user-created data sets	Once per observation
#45	proc 215	After duplicate warning message has been issued	Enables user to perform his/her own end-of-table duplicate reporting. This exit is called only when observations have been found that have duplicate BY variable values.	Once per table, if any observations have been determined to have duplicate BY variable values
#46	proc220	After symbol values have been set for end-of-table reporting	Enables the user to perform end-of-table reporting	Once per table, on the last pass of the DATA step
#47	proc225	After the RUN statement in the "merge" DATA step	Open DATA step code point to enable the user to work with the new merged data, and/or any user data sets that might have been created.	Once per table
#48	proc230	After processing has been completed for all tables	Enables return-code setting or other final operations	Once per run of process

Specific Locations of Process Exit Points

The following is an example of the code that is submitted by the process step when the value of the COLLECTR= parameter on the %C*x*PROCES macro is set to GENERIC, where *x*=M, P, S, or W. The location of an exit point is indicated by #*n*, where *n* is the number of the item in the above table.

```
/* list of global macro variables that are needed */
 %GLOBAL CPCUTDT CPTABLES CPTABLE CPBYVAR CPVWNM;
 %GLOBAL CPARCTBS CPARCTB;
 %GLOBAL CPTOTOBS CPADDOBS CPOLDOBS CPDUPOBS CPEDATM CPLDATM ;
 %LET CPVWNM = WORK.EJLDPCV;
/* unique name per process run */
 %LET CPTABLES = XDCVOLS XDCDSET ;
 %LET CPARCTBS = ;
/* list of tables to archive while processed */
 #1
/* name of table to process */
 %LET CPTABLE = XDCVOLS;
```

```
 %LET CPARCTB = N;  /* Archive this table? Y or N */
/* list of BY variables for this table */
 %LET CPBYVAR = MACHINE  DCVSGTC  DCVVOLS  DCVDVNU  DATETIME ;
 #2
/*************************************************************/
/*  DATA step view created from the staged data.  This is    */
/*  to add and set variables, such as SHIFT, HOUR, and       */
/*  DATETIME, that will be needed for sorting.  The          */
/*  bizarre-looking name for the view is generated as a      */
/*  unique name, to avoid any sort of namespace conflict.    */
/*************************************************************/
data WORK.EJLDPCV(keep=
          DCUTMSTP
          . . .
          ZDATE
 #3 )
 #4 /view=WORK.EJLDPCV;
/* Output: TEMP View */
/* Assign labels to the BY-variables.  This forces the  */
/* variables used for sorting to the first part         */
/* of each observation for improved sort efficiency.    */
attrib MACHINE      label='Machine';
attrib DCVSGTC      label='Storage Class';
attrib DCVVOLS      label='Volume Label';
attrib DCVDVNU      label='Unit number';
attrib DATETIME     label='Datetime';
attrib shift length=$1
         _odate format=date9.
         _hshift length=$1
         _charval length=$10 ;
#5
set GENLIB.DCOLVOLS(in=_new
               rename=(
         DCUTMSTP = DATETIME
         . . .
         DCUSYSID = MACHINE
  )
#6)
#7 end=_last;
#8
  retain _odate _hshift ;

  if datetime=. then do ;
    put 'WARNING: Invalid value for DATETIME on obs '_n_ ' ,obs deleted.' ;
    put 'NOTE: Data for obs. '_n_ ' follows:' ;
    put _all_;
#9
  if _last then _dellast = 1;
  else delete;
 end ;
 hour = hour(datetime) ;
#10
 if lstpdate = . then do;
   lstpdate = '22MAR1999:13:42:22.15'dt ;
```

```
#11
  end;
#12
/* if shift not set in input data make it now */
 if shift = ' ' then do ;
   _date = datepart(datetime) ;
   shift = substr("&cpshift",
           ((weekday(_date)-1)*24+
           hour(datetime)+1),1) ;
#13
 end ;
#14
 if _odate ^= _date then do ;
    _charval=put(_date,CPHDAY.) ;
    if _charval ^= ' ' then do;
    _hshift = put(0,CPHDAY.) ;
    end;
    else do;
      _hshift = ' ' ;
    end;
  end ;
  if _hshift ^= ' ' then do;
     shift = _hshift ;
 #15
  end;
  _odate = _date;
  if datetime > _olt  or  _olt = . and not(_dellast)
     then _olt = datetime;
  if _last
     then do;
#16
 call symput('cpldatm',put(_olt,best32.2));
 if _dellast then delete;
 end;
#17
 output WORK.EJLDPCFV;
#18
 run;
/*************************************************************/
/*  PROC SORT of the staged data by the BY variables.  This */
/*  prepares the data to be "merged" with the detail-level  */
/*  data set.  Here the view is sorted out into a real data */
/*  set in the WORK library; again the data set has a       */
/*  unique name.                                            */
/*************************************************************/
#19
 proc sort
    data=WORK.EJLDPCFV     /* Input:  VIEW DS      */
    out=WORK.EJLDPCFC      /* Output: WORK DS      */
#20
 nodup ;     /* eliminate duplicates from new data */
#21
 by    MACHINE  DCVSGTC  DCVVOLS  DCVDVNU  DATETIME ;
 run ;
```

```
/*********************************************************/
/*  If the BY-variable list has changed since the last      */
/*  run of the process macro, then you would see a PROC     */
/*  SORT of the detail-level data set onto itself, sorted   */
/*  by the current list of BY variables.                    */
/*********************************************************/
#22
#23   /* only if there is no PROC SORT of detail level */
#24   /* only if there is a PROC SORT of detail level  */
/*********************************************************/
/*  For illustration purposes, the sort of the DETAIL level */
/*  is shown.  This code would be generated and submitted   */
/*  only if the detail-level data set required sorting.     */
/*********************************************************/
 proc sort
   data=DETAIL.XDCVOLSD   /* Input/Output:  DETAIL DS       */
#25
   ;
   by
    MACHINE  DCVSGTC  DCVVOLS  DCVDVNU  DATETIME ;
   run ;
#26   /* only if there was a PROC SORT of detail level */
/*********************************************************/
/*  DATA step that does a SET-BY ("merge") of the new data  */
/*  with the active detail-level data set, storing the     */
/*  output into the detail-level data set.  SET statement   */
/*  age limit trimming is performed only on the detail-     */
/*  level data set, so that new data can be reduced before  */
/*  the data is aged out of detail level.  Also,           */
/*  statistics are accumulated for the running of the       */
/*  process macro on the data in this table.                */
/*********************************************************/
#27
 data DETAIL.XDCVOLSD   /* Output: DETAIL */
    (compress=yes
    keep=
      DATETIME
      . . .
      ZDATE
    sortedby=
      MACHINE  DCVSGTC  DCVVOLS  DCVDVNU  DATETIME
#28 )
#29 ;
/* Assign attributes to the BY variables.  This forces the */
/* variables used for merging to the first part of    */
/* each observation for efficiency in later reporting and  */
/* subsequent runs of process.                             */
 attrib MACHINE label='Machine'  length = $8;
 format MACHINE; informat MACHINE;
   . . .
 attrib DATETIME   label='Datetime' length = 8
                 format=DATETIME21.2;
 informat DATETIME;
```

```
 retain _totobs 0 ;
 retain _oldobs 0 ;
 retain _addobs 0 ;
 retain _dups 0 ;
 retain _oet . _olt . _oest ._olst .;
 if _n_ = 1 then do ;
   call symput('cpsedatm','0') ;
   call symput('cpsldatm','0') ;
   call symput('cpedatm','0') ;
   call symput('cpldatm','0') ;
   call symput('cptotobs','0') ;
   call symput('cpoldobs','0') ;
   call symput('cpaddobs','0') ;
   call symput('cpdupobs','0') ;
#30
 end ;
#31
/* Concatenate DETAIL and NEW and select DETAIL */
/* data with dates newer than 12MAR1999:16:32:41.78 (Age= 10 days) */
      %let CPCUTDT = '12MAR1999:16:32:41.78'dt ;
      set DETAIL.XDCVOLSD (in=_detail
          where=(datetime>'12MAR1999:16:32:41.78'dt
#32)
#33)
     WORK.EJLDPCFC (in=_new
#34)
#35
     end=_last
#36 ;
  by
    MACHINE  DCVSGTC  DCVVOLS  DCVDVNU  DATETIME  ;
#37
 /* count and keep duplicate obs. */
 if not first.DATETIME then do;
#38
   _dups = _dups + 1 ;
 end;
 if _oet= . or datetime < _oet then do;
#39
  _oet = datetime ;
 end ;
 if _olt= . or datetime > _olt then do;
#40
  _olt = datetime ;
 end ;
 if _detail then do;
#41
   _oldobs = _oldobs + 1 ;
  end;
  else if _new then do;
#42
  _addobs = _addobs + 1 ;
  if _olst= . or datetime > _olst then do;
    _olst = datetime ;
```

```
  end;
  if _oest= . or datetime < _oest then do;
    _oest = datetime ;
  end;
end;

#43
 output DETAIL.XDCVOLSD;
 _totobs = _totobs + 1;
#44
 if _last then do;
     /* notify number of duplicates */
   if _dups > 0 then do;
     put 'WARNING: Table XDCVOLS has ' _dups
         ' duplicate observations by key.' ;
#45
  end;
  call symput('cpldatm',put(_olt,best32.2)) ;
  call symput('cpedatm',put(_oet,best32.2)) ;
  call symput('cpsldatm',put(_olst,best32.2)) ;
  call symput('cpsedatm',put(_oest,best32.2)) ;
  call symput('cptotobs',put(_totobs,best32.)) ;
  call symput('cpdupobs',put(_dups,best32.)) ;
  call symput('cpaddobs',put(_addobs,best32.)) ;
  call symput('cpoldobs',put(_oldobs,best32.)) ;
#46
  end ;
 run ;
#47  /* per-table basis */
  data _null_;
  run;  /* just to keep things clean */
#48  /* only once per-process run */
  data _null_;
  run;  /* just to keep things clean */
```

Exit Points for %CSPROCES and %CWPROCES

This section covers the process exit points for the UNIX and Windows collectors that use the %CSPROCES and %CWPROCES macros.

Index to Topics in this File:

Old-Style Process Exits

This topic contains information about the %CSPROCES and %CWPROCES "old-style" exit facility.

Note: Although the functionality of the "old-style" exit points might be made available in a future release by using the "new style" of exits points, the older form that is described here will continue to be supported in SAS IT Resource Management indefinitely. △

□ Overview of Old-Style Process Exits

Three exit points exist for processing data with %CSPROCES and %CWPROCES. These points allow the inclusion of user-developed SAS statements at a point directly before an observation that is read from the raw input file is output to the staging data set (COLLECT.*tablenm*).

These exit points are included for compatibility with older releases of SAS IT Resource Management and, though they will continue to be supported, should not be preferred to the newer exit-point strategy that utilizes the EXITLIB= parameter on the %CSPROCES and %CWPROCES macros. For information about the newer style exit points, see "New-Style Process Exits" on page 459.

At the old-style exit point for each of these collectors, catalog entries named

SITELIB.CSEXITS.CS.SOURCE then

DICTLIB.CSEXITS.CS.SOURCE

are included (if they exist) in the SAS code that is generated by %CSPROCES. If both entries exist, both are included. Any syntactical or semantic errors that are introduced into %CSPROCES or %CWPROCES by the statements in these entries are entirely the responsibility of the user, as is the upkeep and transport of the entries, if the PDB is copied.

□ Usage of Old-Style Process Exits

The exits generally should be used to set data values in the observation from information available in that one observation (such as the value of a user-added variable). The data has not gone through standard %CSPROCES or %CWPROCES data cleanup at this point. This has little effect for COLLECTOR=HP-OV, but for COLLECTOR=SPECTRUM and COLLECTOR=SUNETMGR this means that C2RATE variables have not been converted to rates and the value of DURATION has not been set. The exits should not be used to remove observations from the input data. Rather, the SUBSET= parameter in the %CSPROCES or %CWPROCES macro should be used for that purpose.

□ Future of Old-Style Process Exits

The function of this exit facility will be replaced in a future release by one that includes more exit points and the ability to specify a personal exit library. Though the old-style exit points will continue to be supported, they will not be updated to exploit new functionality in SAS IT Resource Management.

□ Example of Old-Style Process Exits

```
*******************************************************;
* This example adds an exit for HP-OV data            ;
*******************************************************;
* Define new PDB with added variable for department code;
x "rm -r /usr/tmp/pdb-exit";
%CPSTART( pdb=/usr/tmp/pdb-exit, mode=batch );
%CPCAT; cards4;
  delete table name=hn2sys noerror;
  add table name=hn2sys ;
  create variable name=udept extname=DEPT type=character
       interpret=string  length=16
       kept=yes label='Dept' description='Department' ;
  update table name=hn2sys
/* Do not change BY list for HP-OV tables */
    day =( classvars=   'udept machine datetime hour shift' )
    week =( classvars=  'udept machine datetime hour shift' )
    month  =( classvars= 'udept machine datetime hour shift' )
```

```
    year   =( classvars= 'udept machine datetime hour shift' );
;;;;
   %CPCAT( cat=work.temp.exitdutl.source );
   %CPDDUTL( entrynam=work.temp.exitdutl.source, list=y );

* Save the exit statements in dictlib with the proper name;
%CPCAT; cards4;

 *-------------------------------------;
 * User exit to set UDEPT from MACHINE.;
 *-------------------------------------;
 retain _u_cnt_ 0;   * This risks varname collision!;;
 select ;
    when (scan(machine,1) in ('dali' 'trinity'))
          udept = 'Personnel';
    when (machine =: 'R' )
          udept = 'Research';
    when (machine = 'henry')
          udept = 'Facilities';
    otherwise do;
       _u_cnt_ + 1;    * Warn of unknown dept only 25 times;
       if _u_cnt_<=25
         then put "WARNING: Unrecognized " machine=;
       udept = 'Unknown';
    end;
   end;
   *-------------------------------------;
   * End user exit code.                ;
   *-------------------------------------;


;;;;
%CPCAT( cat=dictlib.csexits.cshn2sys.source );

* Process HP-OV data;
%let csdelcol=0;   * Opt for debug: Suppress COLLECT cleanup;
%let cp_nmsg=2;    * Opt for debug: Show generated DATA step;
%CSPROCES( <rawdata>,
            hn2sys,
            collectr=HP-OV,
            subset=udept^="Unknown" );
%let csdelcol=2;   * Reset to default;
%let cp_nmsg=1;    * Reset to default;

%CPREDUCE;
```

New-Style Process Exits

Table 20.6 Location and Purpose of Process Exit Points

Exit Point Name	Placement	Purpose	Frequency of Execution
proc100	Directly after the RUN statement for the PROC SORT of the staged data	Enables post-sort processing of the staged data	Once per table
proc125	Directly before the 'merge' DATA step	Enables setting of options and open DATA step code prior to the merge of the detail level and staged data	Once per table
proc135	Directly before the closing semicolon in the DATA statement	Enables the user to add his/her own output data sets	Once per table
proc180	Duplicate observation already in detail	Enables user to prevent the automatic deletion of observations that duplicate those already in DETAIL. (Set variable _delobs to 0 to prevent deletion of duplicate observations. By default, _delobs is set to 1.)	Open data step code and is run once per observation
proc200	When the observation is found to have come from the new (staged) data set	Enables user code to be executed each time an output observation is found to have come from the new (staged) data set. (A count of these observations is accumulated in the variable _addobs.)	Once per observation read from the staged data set
proc205	Directly before the OUTPUT statement on the 'merge' DATA step	Enables the user to modify the observation before it is finally written to the detail level	Once per observation
proc230	After processing has been completed for all tables	Enables return code setting or other final operations	Once per table

□ Process Exits for UNIX and Windows

The following is an overview of the logic of %CSPROCES with the locations of the four exit points inserted. Exit points are indicated by <<< proc*nnn* >>>, where *nnn* is the three-digit exit number.

Note: For %CSPROCES and %CWPROCES, SAS IT Resource Management supports exits 100, 125, 135, 200, 205, and 230 only. △

* Read raw data into staging data set;

```
FILENAME RAWDATA ... ;
DATA COLLECT.<table>...;
RUN ;
```

* Sort staged data to prepare for merge with DETAIL;

```
PROC SORT DATA=COLLECT.<table>D...;
RUN ;
<<< proc100 >>>    * Exit 100 is after the sort ;
<<< proc125 >>>    * Exit 125 is before merge ;
```

* Merge DETAIL with COLLECT;

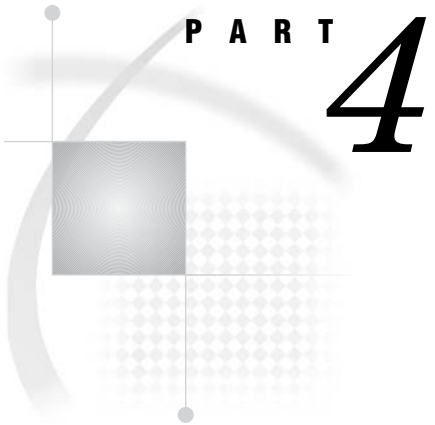
```
DATA DETAIL.<table>D (KEEP= <all vars to keep in the table>)
    <<< proc135 >>>   /* Exit 135 is before end of DATA stmt */
    ;
    SET COLLECT.<table>_ (IN=FROMNEW) ... ;
    ...
    <<< proc180 >>>    * Exit 180 allows duplicate obs to be kept;
    ...
    IF FROMNEW THEN DO;
    <<< proc200 >>> * Exit 200 knows observation from COLLECT;
    END;
    ...

    <<< proc205 >>>   * Exit 205 is directly before OUTPUT DETAIL;
    OUTPUT DETAIL.<table>D;
    ...
    RUN;
    <<< proc230 >>> * Exit 230 is after all tables have
                      been processed ;
```

□ Example (Proc200 Exit)

The following is an example of creating and invoking the proc200 exit:

```
/* list of global macro variables that are needed */
%global CPTABLE;
* This example converts metrics from an older version
  of the performance data collector (HP PCS) into their
  equivalent values in a newer version of the collector
  (HP OpenView Performance Agent);
 * Allocate the PDB;
 %CPSTART( pdb=/my/pdb, mode=batch, access=write );
 * Create exit code for PCSGLB that maps PCS metrics
   to their HPOVA equivalents;
 %CPCAT ; cards4 ;
 * Define a macro to convert certain PCS global metrics
   to HPOVA equivalent;
   %macro FIXPCS ;
      %* Only do this for table PCSGLB;
      %if %UPCASE(&CPTABLE)=PCSGLB %then %do ;
       * Calculate the sum of these 6 metrics
        minus the (alleged) total;
       _d_i_f_f = abs( sum( GLBCSW,  GLBINT,  GLBSYS,
                           GLBRTIM, GLBNICE, GLBUSER )
                           - GLBCPTO );
        * This detects old-style data;
        if (_d_i_f_f<0.00001 or
           (GLBCPTO=1 and _d_i_f_f<0.1)) and
           GLBSCUT=. and GLBNMUT=. then
           do ;
```

```
              * Convert PCS metrics into their HPOVA form ;
              GLBSCUT=GLBSYS ;
              GLBNMUT=GLBUSER ;
              GLBSYS=GLBCSW + GLBINT + GLBSCUT ;
              GLBUSER=GLBRTIM + GLBNICE + GLBNMUT ;
              end ;
            drop _d_i_f_f ;
          %end ;
   %mend FIXPCS ;
     %FIXPCS ;
;;;;
 * Store the exit source in catalog WORK.PROCRED in
   entry PROC200.SOURCE;
 %CPCAT( cat=work.procred.proc200.source ) ;
* Process the data;
 %CSPROCES( /my/extracted/data,
            *,
            exitsrc=work.procred,
            collectr=HP-OVPA );
```



PART 4

Reporting

CHAPTER
21

Report Concepts and Tools

Report Concepts and Tools

Overview of Report Concepts and Tools

SAS IT Resource Management provides a variety of reporting tools on "The Reporting Tab" on page 30 in the SAS IT Resource Management GUI for UNIX and Windows, as well as report macros that you can run in batch. (For information about using report macros, see "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.) Some of the report tools enable you to interactively perform ad hoc analysis, drill down, and more, against a selected group of data. Other report tools are more suited for production reports because they enable you to create and save a *report definition* that you can run on a regular basis against your production data. The actual report is created when you create and run the report definition against data in your PDB.

A report definition identifies which data to use for your report, what variables to calculate and display on the report, the type of report to produce (for example, a bar or pie chart or a plot), and many other report options.

You can create a report definition interactively, through the Manage Report Definitions window, or in batch using the report macros, which you write and submit through the Program Editor window or in batch mode. For information about the report macros, see "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.

From the Manage Report Definitions window, you can create and run report definitions several ways:

- Run the supplied report definitions that are provided with SAS IT Resource Management. You can run these on sample data or data that is collected at your site.
- Modify a supplied report definition and run it with a demonstration PDB or with your data. You can also save the modified definition as a new custom report definition.
- Create a new report definition by using the Manage Report Definitions window.

In batch, you can use SAS IT Resource Management macros to create and run report definitions in the following ways:

- Specify parameters in a report macro and submit the macro to run in batch. (For information about how to use report macros to analyze data, see "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.)
- Create a report definition through the Manage Report Definitions window, save the definition in a folder, and run the definition in batch by using the %CPRUNRPT macro. For information about using macros in batch, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation.
- Create a report definition by using the Manage Report Definitions window, save the macro source for that definition, and run the macro source in batch.

For more information about using macros, see the SAS IT Resource Management macro reference documentation .

Note: Before you can run custom or supplied reports on your own data, you or the SAS IT Resource Management administrator at your site must set up the SAS IT Resource Management client and server and your data collector. Your site administrator can set up the collection, process, and reduction tasks for you and can also create any necessary report definitions at your site. Contact your site administrator for details about the data that you want to use in your report, such as the PDB name, the physical location of the data, and the names of relevant tables and variables. △

Here are several general topics that you might want to understand before you proceed to more specific topics:

- "Using the Data Visualization Tool" on page 466
- "Using the Graph Data Tool" on page 467
- "Using the Manage Report Definitions Tool" on page 468
- "Using the Perform Exception Analysis Tool" on page 468.

Here are the more specific topics that are related to reporting from the client GUI:

- "Overview of Galleries" on page 551
- "Overview of Palette Definition Folders" on page 517
- "Overview of Palette Definitions" on page 525
- "Overview of Remote Server Profiles" on page 571
- "Overview of Report Definition Folders" on page 471
- "Overview of Report Definitions" on page 480
- "Overview of Reports" on page 511
- "Overview of Rule Definition Folders" on page 531
- "Overview of Rule Definitions" on page 535
- "Overview of Rule Results" on page 547.

Using the Data Visualization Tool

The **`Data Visualization`** tool enables you to explore and analyze data through interactive plots and charts. Using its point-and-click interface, you can quickly examine relationships between performance metrics, perform trend analysis, and visualize your performance data in one or more dimensions. You can analyze univariate distributions, investigate multivariate distributions, and fit explanatory models by using analysis of variance, regression, and the generalized linear model. This tool is

integrated with SAS IT Resource Management's dynamic graphics and advanced data analysis tools, providing a valuable addition to your performance analysis needs. You can use it to explore trends in your data, which you can analyze on a regular basis by using reports that you create in the Manage Report Definitions window.

To start the data visualization utility from your SAS IT Resource Management session, select the **Data Visualization** item on the **Reporting** tab in the SAS IT Resource Management main window. When prompted, select a table to use for the current session. When the Data Visualization window opens, the number of observations and variables is displayed in the top-left corner of the table. The variable names are displayed across the top of the table as column headings.

From this window you can also perform the following tasks:

Switch PDBs - select the **Activate PDB** icon to open the Manage PDBs window and activate a different PDB. The directory path or location of the active PDB is displayed next to the icon in the Data Visualization window.

Options - enables you to record your actions or selections during this Data Visualization session so that you can use or replay the actions later.

Start - starts your Data Visualization session and opens the data table that you selected.

When the Data Visualization window opens, you can perform the following tasks:

1. Browse the list of available tables by selecting the arrow to the right of the **Table** text box. Select a table from the list or type the name of the table that you want to use for this session.
2. Move around in the data table, find a record, sort the data, select variables, and more, by selecting the arrow in the top-left corner of the data table to display a menu of choices.
3. Assign a plotting symbol to a specific group of observations by selecting a variable from one of the column headings. While the variable is highlighted, select a color from the color palette, and then from the available list select the group of data to include.
4. Analyze the data with a graph or chart by selecting one of the items from the **Analyze** menu.

Note: For more information, see the documentation for SAS/INSIGHT. △

Using the Graph Data Tool

The Graph Data tool offers a variety of exploratory data analysis graphs and charts. This analysis tool is very useful for ad hoc exploration when you are analyzing unexpected trends, patterns, or extreme outliers in your IT service data. The graphs and charts can be used with data from your active PDB or with data from other SAS data sets.

To use the Graph Data tool,

1. From the SAS IT Resource Management main window, select the **Reporting** tab.
2. On the **Reporting** tab, select **Graph Data**. The Select Table for Graph-N-Go Application window opens. (The Graph Data tool is based on the SAS Graph-N-Go Application.)
3. Specify the table that you want to work with and also specify any subset of the data that you want to focus on.
4. Select the **Start** button. The Graph-N-Go window opens.

   The Graph-N-Go window consists of two panes with icons along the left side of the window.

- The upper pane enables you to specify the data that you want to model.

  The upper pane initially contains an icon for the data that you selected in the earlier window.

  To add other data, select the upper icon next to the top pane. In the **Member Type** field, select **SQL/Data Step/DBI Views (VIEW)**, if it is not already selected, and then select the data's library name and view name. Or select the lower icon next to the top pane to select data from a multidimensional database (MDDB). The data model icon for the data that you selected will appear in the upper pane.

  Double-click on a data icon to view the properties of your data or to subset the data further.
- The icons next to the lower pane represent viewers. To select a viewer, click on its icon. (You can select bar charts, pie charts, and so on.) Drag and drop the viewer to an empty part of the lower pane.

  Next, drag and drop a data model from the upper pane onto the viewer. The data is displayed according to the defaults for that viewer. You can modify the display by putting your mouse pointer over the graph, right-clicking, and choosing from the selection menu.

  Multiple graphs and/or tables can be displayed in the lower pane.

You can save any report that you develop to SAS library for future editing (by default, the report will be saved in SASUSER._GNGMODEL_.ITSV.GFORM). And/or you can export the report to other file types such as HTML (on the File pull-down menu within the Graph-N-Go application, see the export function).

For more information about this application, refer to the online help within the application.

Using the Perform Exception Analysis Tool

The Exception Analysis window enables you to create exception rules to evaluate exception conditions in your data. For example, you might want to identify conditions when disk usage was high or when queue length or wait times were long.

You can create folders in which to store the exception rules, and you can evaluate a single rule or all rules in a folder. When you evaluate all rules in a folder, the rules that apply to tables in the active PDB are evaluated. Also, the rules are evaluated that apply to other data for which the libref(s) has been defined in the current SAS session. When you evaluate the exception rules, you can view reports that summarize only the data that meets your exception criteria.

To create an Exception Analysis report, follow these steps.

Exception rules must be defined within the SAS IT Resource Management GUI for UNIX and Windows, but can then be evaluated interactively, by using the interface, or in batch, by using the %CPEXCEPT macro. In batch, you can use the %CPXHTML macro to create exception reports that you can display by using a Web browser.

Using the Manage Report Definitions Tool

From the Manage Report Definitions window, you can create, edit, view and run new or existing report definitions (see "Creating/Editing/Viewing a Report Definition" on page 481 and "Running a Report Definition" on page 488), as well as save a report (see "Saving a Report" on page 512) and save the report definition (see "Saving a Report Definition" on page 489). You can also access SAS IT Resource Management supplied report definitions (see "Exploring Supplied Report Definitions (UNIX, Windows)" on page 488).

To open the Manage Report Definitions window, start SAS IT Resource Management and select the **Reporting** tab from the SAS IT Resource Management main window. On the **Reporting** tab, select **Manage Report Definitions** from the list of reporting tasks.

For more information about the Manage Report Definitions Tool, see the following chapters:

- "Overview of Report Definition Folders" on page 471
- "Overview of Report Definitions" on page 480
- "Overview of Reports" on page 511
- "Overview of Palette Definition Folders" on page 517
- "Overview of Palette Definitions" on page 525
- "Overview of Rule Definition Folders" on page 531
- "Overview of Rule Definitions" on page 535
- "Overview of Galleries" on page 551
- "Overview of Remote Server Profiles" on page 571.

CHAPTER

22

Reporting: Working with Report Definition Folders

Working with Report Definition Folders

Overview of Report Definition Folders

Report definition folders contain report definitions.

You have a list of known report definition folders. Originally, your list contains the names of the report definition folders that have the supplied report definitions.

If you create one or more report definition folders, their names are automatically added to your list. You can also add the names of report definition folders that other people create.

Similarly, you can remove the names of report definition folders from your list. (The names of the folders are deleted from your list, but the actual folders are not deleted.)

The following report folders are typically available in your list of known report definition folders. As you create additional folders and add additional folders, those folders will also be on the list.

SITELIB.ITSVRPT
: contains report definitions that have been placed in your organization's site library, which has a SAS libref of SITELIB. These report definitions can be used by all users who have "readonly" or "read/write" access to this library.

PGMLIB.ITSVRPT
: contains supplied report definitions that have been provided with SAS IT Resource Management.

SASUSER.ITSVRPT
: contains report definitions that you have stored in your *sasuser* directory. These definitions are referred to as your *custom* report definitions, and they include

report definitions that you created by using the Manage Report Definitions window. To make your own custom reports available to others at your site, you can move them to the site library.

ADMIN.ITSVRPT
contains report definitions that you and others have created and saved with the active PDB.

The PGMLIB.CPRPTS and SASUSER.CPUPGMS folders might also appear in the folders list. These folders contain supplied and custom reports from a previous version of this product.

For more information about the list of report definition folders and subsets of the list, see "Viewing Your List of Known Report Definition Folders" on page 476.

For more information about report definition folders, see the following topics:

- "Adding a Report Definition Folder" on page 472
- "Creating a Report Definition Folder" on page 473
- "Removing a Report Definition Folder" on page 475
- "Viewing the List of Report Definitions in a Report Definition Folder" on page 475
- "Viewing Your List of Known Report Definition Folders" on page 476.

Adding a Report Definition Folder

You can add the name of an existing report definition folder (for example, a report definition folder that someone else created) to your list of known report definition folders

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Adding a Report Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can add the name of an existing report definition folder to your list of known report definition folders by following these steps:

1. If you want to add the name of an existing report definition folder that is on a remote server, first activate a PDB on that server.

   For more information about activating a PDB, see "Activating a PDB" on page 299.

2. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens. The Folders list displays the names of all the folders for the current SAS session.

3. Select **File**. A menu opens. From the menu, select **Add Report Folders**. The Add Report Definition Folders window opens.

   The **Report Definition Folders** list displays the names of your known report definition folders.

4. Select the arrow to the right of the **Library** field. A menu opens. It contains a list of the libraries for which a libref has already been defined. Select the library that contains the report definition folder whose name you want to add.

   If the library that you want to use is not on that list, select **Define Library**. The **Define a SAS Library** window opens.

   a. In the **Libref** field, type the libref that you want to use for the library. The libref should begin with an alphabetic character. The remaining characters can be alphabetic or numeric.
   b. In the **Path** field, type the full path and name of the library's directory (UNIX, Windows) or PDS (on z/OS).

c In the **Access** field, select the access mode that you want to use (assuming that you have permission from the owner or security system to use that mode).
d In the **Location** field, select **Local** or **Remote**. If you select **Remote**, the **Server ID** field opens. The value in that field defaults to the server ID of the server to which you most recently connected.
e Select **OK**. The value in the **Library** field is the libref that you defined.

5 In the **Folder Name** field, type the name of the folder.
6 In the **Description** field, type a description of the folder.
7 Select **Apply** ▶ **OK** You are returned to the Manage Report Definitions window. The folder whose name you added is now on your list of known report definition folders.
8 When you are ready to return to the main window of SAS IT Resource Management, in the Manage Report Definitions window select **Close**.

Note: 1. If you have a "write" lock on the library to which you added the folder, as soon as you finish using the folder for "write" purposes (for example, saving report definitions to it), remember to free the library fully or partially so that other people and tasks can use it. One way to free the library is to exit from SAS IT Resource Management and SAS. Another way is to follow these steps:

□ From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
□ Select **File**. A menu opens. From the menu, select **Add Report Folders**. The Add Report Definition Folders window opens.
□ Select **Define Library**. The **Define a SAS Library** window opens.

1 In the **Libref** field, type the libref of the library to which you added the folder.
2 If the libref was one that you created, leave the **Path** field blank (if you do not intend to use that library again during your current SAS session). That will entirely free the library.

If the libref was one that you did not create or one that you did create but now can use in "read" mode, in the **Access** field select **Readonly**. That will partially free the library; people will be able to use it for read access, but not for write access.

3 Select **OK**.

Note: If you set a path to blank for a libref that you defined, the libref will be de-assigned and (in the SAS message area at the bottom of the window) you will get a confirmation message that says it is now "not assigned." (This is a NOTE message. Ignore the word ERROR.) △

□ Select **OK**.

△

Note: 2. If the library to which you added the folder was remote, in addition to freeing that library (see Note 1), you also need to release the lock on the PDB so that other people and tasks (such as the process task and reduce task) can use it. For more information about releasing the lock, see "Releasing the Lock on the Active PDB" on page 319. △

Creating a Report Definition Folder

You can create a new report definition folder and automatically add its name to your list of known report definition folders

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Creating a Report Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create a new report definition folder and automatically add its name to your list of known report definition folders by following these steps:

1. If you want to create the report definition folder on a remote server, first activate a PDB on that server. For more information about activating a PDB, see "Overview of Whole PDBs" on page 296.
2. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
3. Select **File**. A menu opens. From the menu, select **Add Report Folders**. The Add Report Definition Folders window opens.

   The top of the window displays your list of known report definition folders.
4. Select the arrow to the right of the **Library** field. A menu opens. It contains a list of the libraries for which a libref has already been defined. Select the library in which you want to place your new report definition folder.

   If the library that you want to use is not on that list, select **Define Library**. The **Define a SAS Library** window opens.

   a. In the **Libref** field, type the libref that you want to use for the library. The libref should begin with an alphabetic character. The remaining characters can be alphabetic or numeric.
   b. In the **Path** field, type the full path and name of the library's directory (UNIX, Windows) or PDS (on z/OS). On UNIX or Windows, the library will be created if it does not exist; on z/OS, the library must already exist.
   c. In the **Access** field, select **Write** if it is not already selected.

      Note: You must have prior permission from the owner of this library in order to use "write" access. △

   d. In the **Location** field, select **Local** or **Remote**. If you select **Remote**, the **Server ID** field opens. The value in that field defaults to the server ID of the server to which you most recently connected.
   e. Select **OK**. The value of the **Library** field is the libref that you defined.
5. In the **Folder Name** field, type the name of the folder.
6. In the **Description** field, type a description of the folder.
7. Select **OK**. You are returned to the Manage Report Definitions window. The folder's name is on your list of known report definition folders.

Note: 1. If you have a "write" lock on the library to which you added the folder, as soon as you finish using the folder for "write" purposes (for example, saving report definitions to it), remember to free the library fully or partially so that other people and tasks can use it. One way to free the library is to exit from SAS IT Resource Management and SAS. Another way is to follow these steps:

□ From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.

□ Select **File**. A menu opens. From the menu, select **Add Report Folders**. The Add Report Definition Folders window opens.

□ Select **Define Library**. The **Define a SAS Library** window opens.

   1 In the **Libref** field, type the libref of the library to which you added the folder.

2 If the libref was one that you created, leave the **Path** field blank (if you do not intend to use that library again during your current SAS session). That will entirely free the library.

   If the libref was one that you did not create or one that you did create but now can use in "read" mode, in the **Access** field select **Readonly**. That will partially free the library; people will be able to use it for read access, but not for write access.

3 Select **OK**.

Note: If you set a path to blank for a libref that you defined, the libref will be de-assigned and (in the SAS message area at the bottom of the window) you will get a confirmation message that says "not assigned." (This is a NOTE message. Ignore the word ERROR.) △

□ Select **OK**.

△

Note: 2. If the library to which you added the folder was remote, in addition to freeing that library (see Note 1), you also need to release the lock on the PDB so that other people and tasks (such as the process task and reduce task) can use it. For more information about releasing the lock, see "Releasing the Lock on the Active PDB" on page 319. △

Removing a Report Definition Folder

You can remove the name of a report definition folder from your list of known report definition folders

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Removing the name of the folder from the list does not delete the folder. The folder continues to exist. △

Removing a Report Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can remove the name of a report definition folder from your list of known report definition folders by following these steps:

1 From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.

2 Select **File**. A menu opens. From the menu, select **Remove Report Folders**. The Remove Report Definition Folders window opens. A list of the names of the report definition folders is displayed at the top of the window.

3 Select the name of the report definition folder that you want to remove from your list.

4 Select **Remove**.

5 Select **OK**.

   Notice that the name of that report definition folder is no longer on the your list of known folders. (The report folder itself was not deleted; it still exists.)

6 When you are ready to return to the main window of SAS IT Resource Management, select **Close** in the Manage Report Definitions window.

Viewing the List of Report Definitions in a Report Definition Folder

You can view the names of the report definitions in a report definition folder

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing the List of Report Definitions in a Report Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view the list of report definitions in a report definition folder by following these steps:

1. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens. Your list of known report definition folders displays in **Folders**.
2. Select the name of the report definition folder whose contents you want to know.
3. The list of the report definitions in that folder is displayed in **Current PDB Reports**.

   Note: By default, the list includes the names of the report definitions that are based on the tables in the active PDB and excludes the names of the report definitions that are based on other tables.

   - To change the filter so that all of the folder's report definitions are displayed, regardless of the location of the tables on which they are based, select **Tools ▶ Filter Report Definitions ▶ Show All Report Definitions ▶ OK**
   - To restore the default, so that the list of report definitions displays only the ones that are based on the tables in the active PDB, select **Tools ▶ Filter Report Definitions ▶ Only Show Report Definitions Available for This PDB ▶ OK**

   △

For more about report definitions, see "Overview of Report Definitions" on page 480.

Viewing Your List of Known Report Definition Folders

There are three versions of your list of known report definition folders:

- The complete list, which contains all of these sets of names:
  - the names of report definition folders that SAS IT Resource Management looks for automatically
  - the names of report definition folders that you created and have not deleted
  - the names of report definition folders that someone else created and that you added and have not removed.

  Notice that the complete list might contain the names of folders to which you do not currently have access. For example, the names might refer to folders that have librefs that are not currently defined. Or the names might refer to folders that have librefs that are currently defined, but the librefs points to different libraries than they pointed to earlier. (The library might be in a PDB that is not the active PDB, or a site library that is not the active site library, or the program library that is not the active program library, or the remote program library that is not the active remote program library.)
- The complete list's subset for librefs that are currently defined with "write" access.

  Notice that this subset of the complete list might contain the names of folders to which you do not currently have access, for the reasons that are described above. For example, suppose the folder named ADMIN.MYRDEFS is in this subset. The ADMIN libref is currently defined (the libref points to the active PDB's ADMIN library). And if you are accessing the PDB with "write" access, you have "write"

access to the active PDB's ADMIN library. But the folder might be in some other PDB's ADMIN library.

- The complete list's subset to which you currently have access.

  These are the report definition folders for which the libref is currently defined. A folder of that name is found in the library to which that libref currently points.

You can view your list of known report definition folders and the subsets of that list

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing Your List of Known Report Definition Folders (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view your list of known report definition folders and the subsets of that list by following these steps:

1. From the main window's **`Reporting`** tab, select **`Manage Report Definitions`**. The Manage Report Definitions window opens.
2. The complete list's subset of folders to which you currently have access is displayed in **`Folders`**.
3. Select **File ▶ Add Report Folders** or select **File ▶ Remove Report Folders** The complete list is displayed in **`Report Definition Folders`**.

   Select **`OK`** or **`Cancel`** to return to the Manage Report Definitions window.
4. Select a report definition folder and select a report definition. Then select **File ▶ Save Report Definition** The complete list's subset of folders with librefs that are currently defined with "write" access is displayed in **`Report Folders`**. (If you select a folder that does not currently exist and you save a report definition to it, the folder will be created and then the report definition will be saved to it.)

   Select **`OK`** or **`Cancel`** to return to the Manage Report Definitions window.
5. When you are ready to return to the main window of SAS IT Resource Management, select **`Close`** in the Manage Report Definitions window.

CHAPTER 23

Reporting: Working with Report Definitions

Working with Report Definitions: Main Topics

Overview of Report Definitions

A report definition contains the specification for a report. You run a report definition to generate a report.

For more information about report definitions, see the following topics:

Copying a Report Definition

You can copy a report definition to another report definition folder

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Copying a Report Definition

In the SAS IT Resource Management GUI for UNIX, you can copy a report definition to another report definition folder by following these steps:

1. Select (view) the report definition from the first folder.

   For more information, see "Selecting a Report Definition" on page 491.

2. Save the report definition with a new name in the same folder, or save the report definition (with the same name or a new name) in a different folder. Notice that you must have "write" access to this folder.

   For more information, see "Saving a Report Definition" on page 489.

However, if you have a large number of report definitions to copy, you might want to do the following:

1. On the main window's **`Reporting`** tab, select **`Manage Report Definitions`**. The Manage Report Definitions window opens.
2. Select **File ▶ Copy Report Definitions** The Copy Report Definitions window opens.
3. To select the "From" folder, select the down arrow to the right of the **`From`** field and then select the folder that you want to copy report definitions from.

   If the folder that you want to copy from is not on that list, select the right arrow to the right of the **`From`** field and define the libref for the folder that you want to copy from. For more information about defining a libref for a folder, see "Adding a Report Definition Folder" on page 472.
4. To select the "To" folder, select the down arrow to the right of the **`To`** field and then select the folder that you want to copy report definitions to. Notice that you must have "write" access to this folder.

If the folder that you want to copy to is not on that list, select the right arrow to the right of the **To** field and define the libref for the folder that you want to copy to. For more information about defining a libref for a folder, see "Adding a Report Definition Folder" on page 472.

5 If you want to copy all the report definitions in the From folder, make sure that there is a check in the box to the left of the **Copy All Reports** field. (To toggle between checked and unchecked, click on "Copy All Reports.")

If you want to copy some but not all of the report definition in the From folder, make sure that the check box to the left of the **Copy All Reports** field is unselected. (To toggle between checked and unchecked, click on "Copy All Reports.") In the list below, select the report definitions that you want to copy. (A single click selects a single report definition. To select a range, hold down the Shift key while you are selecting. To select multiple reports but not a range, hold down the Ctrl key while you are selecting.)

Note: When you use this method of copying, you cannot change the name of report definitions while you are copying them. △

6 Select **OK**. You return to the Manage Report Definitions window. The new report definitions are listed in **Current PDB Reports**.

Note: If you do not see the new report definitions in that list, select **Tools** ▶ **Filter Report Definitions** ▶ **Show All Report Definitions** ▶ **OK** △

7 When you are ready to return to the main window of SAS IT Resource Management, select **Close** in the Manage Report Definitions window.

Creating/Editing/Viewing a Report Definition

SAS IT Resource Management provides supplied report definitions. You can use a supplied report definition as-is, you can modify a supplied report definition and save it as a customized report definition, and/or you can create an entirely new report definition.

You can create, edit, or view a report definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: If your server is running in a z/OS operating environment and you are using a supplied report definition that is based on the XJOBS table, it is important that you understand the values in that table in order to correctly interpret your output. For an explanation of the values in that table, refer to RMF Appendix 1 in the SAS IT Resource Management Setup documentation. (See "General-Purpose Server Setup Documentation" on page 12.) △

Creating/Editing/Viewing a Report Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create, edit, or view a report definition by following these steps:

1 From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.

2 The name of the active PDB is in the upper-right corner of the window. If the data on which you want to base the report definition is in a different PDB, select the icon to the left of the PDB name and activate that PDB. For more information, see "Activating a PDB" on page 299.

Note: If you are unsure about the location of your data, check with your SAS IT Resource Management site administrator and determine where the data for your report is located (the server name, the PDB, and the level and tables in the PDB). △

3 If you want to create a new report definition, select **File**. A menu opens. From the menu, select **New Report Definition**.

If you want to edit or view an existing report definition, list the existing report definitions and select the one that you want to view or modify. For more information, see "Viewing the List of Report Definitions in a Report Definition Folder" on page 475.

4 *View (level and table):* The **Select Table** field specifies the view (that is, the level of the table) on which the report definition is based.

To see more about the view, modify the view, or specify the view:

a On the right side of the window, select the arrow to the right of the **Select Table** field. The Select Table window opens.

b If the view that you want to use is in the active PDB, in the **Levels** field select the level of the table and in the **Level.Table** list, select the view.

If the view that you want to use is not in the active PDB, in the **Levels** field select **OTHERS**. The SAS Data Library window opens. Select the libref of the library that contains the view that you want to use. In the **Level.Table** list, select the view.

c Select **OK**. You return to the Manage Report Definitions window. The view that you selected is listed in the **Select Table** field.

5 *Image (report style and sub-style):* In the lower-right corner of the window, the image illustrates the report style and sub-style on which the report definition is based.

Reports are grouped into two major categories: graph and text reports. Graph reports include a variety of styles, ranging from basic plots to more complex spectrograph or spectrum plot reports. Graphs are useful for reviewing trends over time and relationships between variables. Text reports include styles such as listings and tabular reports. These report styles are useful for gathering summary statistics on your data.

If you have not selected a report style for your current report definition, the most recently selected style and sub-style are used.

To see more about the report style and sub-style (and advanced options), to modify them, or to specify them:

a Select the image. The Report Style Selection window opens.

b From the Report Style list, select the report style that you want to use. If the report style has sub-styles, the sub-styles are displayed in the **Report Sub-Style** list. Select the sub-style that you want to use.

For more information about the report styles and sub-styles, select the **Help** button.

c If the style or sub-style has advanced options, you can see or specify the advanced options by selecting **Advanced**. The Advanced Options window opens. View or change the values of the advanced options, and select **OK**.

For more information about the advanced options, select the **Help** button in the Advanced Options window.

d Select **OK**. The image below **Current Report Style** illustrates the report style and sub-style that you selected.

For more information about choosing a report style, see "Report Styles" on page 499.

6 *Variables (roles and attributes):* The **Select Variables** field specifies the variables on which the report definition is based. (If more variables are selected than can fit in the field, some of the selected variables are not displayed.)

To view, modify, or specify the selected variables and their roles (types), labels, and formats:

a Select the arrow to the right of the **Select Variables** field. The Select Variables window opens.

b In the **Variable Types** list select **Show All**. The selected variables, their roles, and their descriptions are displayed in the **All Selected Variables** list.

To see the attributes of the selected variables, select **Attributes** and then select the variable whose attribute you want to see. To return to the Select Variables window, select **OK**.

Note: The Variable Type *Statistics* does *not* represent variables. It represents one or more statistics that will be calculated for the variables that are assigned to the *Analysis* role (type). (If you do not select a statistic, the default statistic is MEAN.) In graph reports (3D Graph, Chart, Plot, and Spectrum reports) if the Summary Period is not AS IS, the statistic is used to transform the Analysis variables, and the transformed Analysis variables are displayed in the report. In tabular reports, the statistic or statistics are used to transform the Analysis variables, and the transformed Analysis variables are displayed in the report. △

Note: For additional information about variable types see "Variable Types" on page 501. △

c The unselected variables and their descriptions are displayed in the **Available Variables** list.

d To move a variable from the **All Selected Variables** list to the **Available Variables** list, select the variable in the **All Selected Variables** list, select its variable type in the **Variable Types** list, and then select **<-Remove**. The variable moves to the **Available Variables** list.

e To move a variable from the **Available Variables** list to a particular type's list (which then is displayed in the **All Selected Variables** list when you select **Show All**), select the variable in the **Available Variables** list, select the variable type to which you want to assign it in the **Variables Types** list, and then select **Add->**. The variable moves to the right column, whose title depends on the variable type that you selected, and is assigned to the role that is specified by the selection in the **Variable Types** list.

Note: For some report styles, you can select multiple variables for a single variable type. △

Note: In the **Variable Types** list, variables are coded as *Req* (required for the selected report style), *Opt* (optional for the selected report style), or *N/A* (not applicable for the selected report style). If you select a variable that is marked *N/A*, the variable type will not be used for the current report, but the variable and type information will be saved. If you later change your report style and this variable type is applicable to the new type, it will be used for that report. △

f To move a variable up or down in the list of selected variables, select the variable that you want to move and then select **Move Up** or **Move Down**.

For analysis variables (Variable Type Analysis), the order in which the variables are displayed in the Analysis Variables list and Show All list is the order in which they are graphed or plotted on the report.

For statistics (Variable Type Statistics), the order that the statistics are displayed in the Statistics list and Show All list is the order that they will be displayed on your report. For example, this determines the order in which they would appear in columns or rows on a tabular report.

g To change the role (that is, the type) that is assigned to a selected variable, you must first remove the variable from the variable type that it is in and then add it again as a different variable type.

h You can change a variable's attributes while it is in the **All Selected Variables** list. To change the attributes of a selected variable, select **Attributes**. The Variable Attributes window opens. Select the variable whose attributes you want to change. The **Variable Label**, **Variable Format**, and **Variable Description** fields display the current values. Overwrite them with values that you prefer. Select **OK** to return to the Select Variables window.

Note: If you change one or more attributes in this way, the change affects this report definition only. The change is saved with the report definition (if the report definition is saved), but the change is not saved in the data dictionary. To change the attributes in a way that affects all report definitions, change the properties of the variable(s) in the data dictionary. For more information about changing the properties of variables in the data dictionary, see "Creating/Editing/Viewing a Derived Variable" on page 381, "Creating/Editing/Viewing a Formula Variable" on page 388, or "Creating/Editing/Viewing a Regular Variable" on page 394. △

i Select **OK**. You return to the Manage Report Definitions window. At least some of the variables that you selected are listed in the **Select Variables** field.

7 *Subsetting (datetime and WHERE):* You can subset data; that is, you can use a portion of your data that meets specified criteria. The criteria can be based on a datetime range or on a WHERE clause or on both.

To view, modify, or specify the subsetting criteria:

a Select the arrow to the right of the **Subset Data** field.

b To see any subsetting that is based on a datetime range, select the **Start/End Dates/Times** tab. The subsetting criteria are listed at the top of the tab. If the subsetting is for this report definition only, its criteria are in the **Override Default Values For This Report** field. If the subsetting is for all report definitions, its criteria are in the **Set Default Values For All Reports** field. For more information about datetime ranges, select **Help** in this window.

When a report is generated by a report definition that specifies subsetting "for this report definition only," any subsetting "for all report definitions" is ignored in that report.

c To see any subsetting that is based on a WHERE clause, select the **Query/Where Clause Builder** tab. The subsetting criteria are listed at the top of the tab. If the subsetting is for this report definition only, its criteria are in the **Local Query/Where Clause For This Report** field. If the subsetting is for all report definitions, its criteria are in the **Global Query/Where Clause For All Reports** field. If *Instead of* is selected, only the local version is used. If *And* is selected, the data will satisfy both the local and global versions. For more information about WHERE clauses, select **Help** in this window.

Note: In these WHERE fields, underscores are used to indicate lines and are interpreted as blanks. Thus, if you type an underscore, it will be interpreted as a blank. △

Note: When the global WHERE expression is joined to the local WHERE expression with AND, the WHERE expressions must be compatible for a report to be generated. For example, no report is generated if one WHERE expression has MACHINE="Alpha" and the other WHERE expression has MACHINE="Beta". (Joining these two expressions with AND does not result in the condition *True*.) △

Note: Do not use the ampersand (&) to mean AND in WHERE expressions. △

d If both datetime range criteria and WHERE criteria are specified, the data will satisfy *both* the range criteria and the WHERE criteria.

To specify the criteria or to change the criteria:

a On the **Start/End Dates/Times** tab, you can specify values or change values for the ends of the ranges by selecting the arrow to the right of a field. A Select window opens. If you select **Relative date**, you can specify a relative date and time. If you select **Absolute date**, you can specify an absolute date and time. For more information about specifying the ends of the ranges, select **Help** in the Select window.

Note: To remove a value, put your cursor in a field and delete the characters in its specification. △

b To see any subsetting that is based on a WHERE clause, select the **Query/ Where Clause Builder** tab. The subsetting criteria are listed at the top of the tab. If the subsetting is for this report definition only, its criteria are in the **Local Query/Where Clause For This Report** field. If the subsetting is for all report definition, its criteria are in the **Global Query/Where Clause For All Reports** field. If *Instead of* is selected, only the local version is used. If *And* is selected, the data will satisfy both the local and global versions.

Note: To remove a value, select the **Clear** button to the right of the value. △

To return to the Manage Report Definitions window, select **OK**.

Note: The WHERE criteria are displayed in the **Subset Data** field. The datetime range criteria are not displayed in the **Subset Data** field. △

8 *Summarization (ASIS and period):* You can summarize the data. Summarization is particularly useful when there is a large amount of data. The default is to have no summarization, which is represented by a blank field or by *ASIS*.

If you want to collapse your data into groups based on the date and time value for each observation, you can do so by selecting a summary time period, such as **15 Min**, **Hour**, or **24 Hour**. (When you select a summary time period, your data is collapsed before the analysis statistic is applied.)

When you select a summary time period, the data in each summary group is combined to form a single point on a plot, a bar on a chart, or a row or column in a table. For example, if you want to summarize your data at 15-minute intervals, you would select a summary time period of *15 MIN*. The *datetime* value for each observation is modified to group the observations into 15-minute summary groups and your report displays a point, bar, or row for data that is summarized every 15 minutes.

To view, modify, or specify the summarization period:

a Select the arrow to the right of the **Summary Time Period** field. The Summary Time Period window opens and displays a list of the periods that you can select.
b Select the period that you want to use. For more information about periods, select the **Help** button.
c Select **OK** to return to the Manage Report Definitions window.

9 *Palette definition (specific and default):* You can specify the palette definition that you want to use for the report. A palette definition is a stored set of SAS and SAS/GRAPH options, such as colors, patterns, symbols, legends, and axis definitions, that can be associated with one or more report definitions. If you do not specify a particular palette definition, the default palette definition is used.

To view, modify, or select a palette definition, see "Creating/Editing/Viewing a Palette Definition" on page 527.

For more information about the default palette definition, see "Specifying/Editing/Viewing the Default Palette Definition" on page 529.

10 *Titles and footnotes:* You can specify the title and footnote that you want to use with the report definition.

To view, modify, or specify the title and footnote:

a Select **Locals**. A menu opens. From the menu, select **Set Titles and Footnotes**. The Set Titles and Footnotes window opens.
b You can specify four lines in the title. The lines will be centered when they are displayed in the report.

Note: If you later use the report definition in batch mode, you can have as many as 10 lines in the title. △

c You can specify two lines in the footnote. The lines will be centered when they are displayed in the report.

Note: If you later use the report definition in batch mode, you can have as many as 10 lines in the footnote. △

d Select **OK** (to accept changes, if any) or **Cancel** (to ignore changes, if any) to return to the Manage Report Definitions window.

In the title and footnote lines, to display the date that the report is produced, use *%str(&SYSDATE)* instead of *&SYSDATE*. You can also use other features of the SAS TITLE and FOOTNOTE statements, such as color, height, font, and justification. Additionally, you can use macro functions. For example, suppose that your report definition has MACHINE as a BY variable and the title is

```
For Server #BYVAL(MACHINE) on %STR(%SYSDATE)
```

Then the current value of MACHINE and the current value of system datetime are retried and inserted. (Also, the default BY line after the title is suppressed.)

11 *Formatting and direction:* You can direct where you want the report to be produced and format the report definition appropriately for that direction. One copy is available in the Results window. You can direct whether the other copy goes to the default window (Graph for graph reports, Output for text reports) or elsewhere.

To view, modify, or specify where you want the report to be generated:

a Select **Locals**. A menu opens. From the menu, select **Report Output Options**. The Report Output Options window opens.
b If **SAS Window** is selected, the report is formatted for and directed to a SAS window (Graph for graph reports, Output for text reports).

c If **SAS Catalog** is selected, the report is formatted for and directed to a SAS catalog. To view the location, see the **Current Settings** list.

To modify or specify the location, select **Set Attributes**. The Output to a SAS Catalog window opens. To specify values, type them in. To modify values, type over the current values. Select **OK** (to accept changes, if any) or **Cancel** (to ignore changes, if any).

d If External File is selected, the report is formatted for and directed to an external file. To view the location, see the **Current Settings** list.

To modify or specify the location, select **Set Attributes**. The Output to an External File window opens. To specify values, type them in. To modify values, type over the current values. Select **OK** (to accept changes, if any) or **Cancel** (to ignore changes, if any).

e If *Web* is selected, the report is formatted for and directed to GIF files and/or HTML files. To view the location and other attributes, see the **Current Settings** list.

To modify or specify the location and other attributes, select **Set Attributes**. The Generate Web Output window opens. To specify values, type them in or make selections. To modify values, type over the current values or change the selections. Select **OK** (to accept changes, if any) or **Cancel** (to ignore changes, if any).

For more information, see "Directing a Report to the Web" on page 491.

f Select **OK** (to accept changes, if any) or **Cancel** (to ignore changes, if any) and return to the Manage Report Definitions window.

If you make and accept changes and later save the report definition, the changes are saved in the report definition.

For more information about directing reports (in batch mode or the GUI) to the above destinations, see the topic "How the OUT*= Parameters Work Together" in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.

To run the report definition, see "Running a Report Definition" on page 488.
To save the report definition, see "Saving a Report Definition" on page 489.

Deleting a Report Definition

You can delete a report definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Deleting a Report Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete a report definition by following these steps:

1 From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.

2 The name of the active PDB is in the upper-right corner of the window. If the folder that contains the report definition that you want to delete is in a different PDB, select the icon to the left of the PDB name and activate that PDB. For more information, see "Activating a PDB" on page 299.

3 In the **Folders** list, select the folder that contains the report definition that you want to delete. The report definitions that are in that folder are displayed in the **Current PDB Reports** list.

4 In the **Current PDB Reports** list, select the report definition that you want to delete.

5 Select **File ► Delete Report Definition ► Yes**

6 When you want to return to the main window of SAS IT Resource Management, select **`Close`** in the Manage Report Definitions window.

Exploring Supplied Report Definitions (UNIX, Windows)

You can explore supplied report definitions

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Exploring Supplied Report Definitions (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can explore supplied report definitions by following these steps:

1 View the list of report definitions in the *PGMLIB.ITSVRPT* folder. For more information, see "Viewing the List of Report Definitions in a Report Definition Folder" on page 475.

2 For each report definition that interests you, view the report definition. For more information, see "Creating/Editing/Viewing a Report Definition" on page 481.

Note: If you do not see the supplied report definition that you are looking for, check the filter by selecting **Tools ► Filter Report Definition** △

3 If the active PDB contains the table on which the report definition is based, you can run the report definition and generate a report. For more information, see "Running a Report Definition" on page 488.

Running a Report Definition

When you run a report definition, a report is generated.
You can run a report definition

□ by using the %CPSTART macro (to invoke SAS IT Resource Management) and the reporting macros.

For more information about the %CPSTART macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation.

For more information about the reporting macros, see "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Running a Report Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can run a report definition by following these steps:

1 From the main window's **`Reporting`** tab, select **`Manage Report Definitions`**. The Manage Report Definitions window opens.

2 Create, edit, or view the report definition. For more information, see "Creating/Editing/Viewing a Report Definition" on page 481.

3 In the Manage Report Definitions window, select the **`Run`** button. (Or, if you have several reports to run and want to run them all at the same time, select **Locals ► Run Multiple Reports** Then select the report definitions that you want to run at the same time, and select the **`Run`** button.)

If the active PDB is remote, a menu opens after you select the **Run** button. From that menu, select **Local** if you want the data to be downloaded to the local host and the report definition to run by using the CPU on the local host. Or, from that menu, select **Remote** if you want the report definition to run by using the CPU on the remote host and the report to be downloaded to the local host. (Typically, select **Remote**, because less network traffic is caused by downloading the report than by downloading the data. However, if you are directing the report to the Web (by selecting, in the Manage Report Definitions window, **Locals ▶ Report Output Options ▶ Web ▶ Set attributes** and providing the requested information), then select **Local**, not **Remote**, if the remote host is running z/OS.)

If the active PDB contains the table on which the report definition is based, the report definition runs and generates a report.

Note: If your report definition displays unexpected results, see "Troubleshooting Report Problems" on page 513. △

4 The report definition specifies where the report is to be directed.

By default, graph reports are directed to the SAS GRAPH window, which opens automatically and displays the report automatically when a report is directed to it. Remember that a report might have multiple parts. If so, scroll down in the window to see all the parts. When you are finished looking at the report, iconize the window.

By default, text reports are directed to the SAS OUTPUT window, which does not open automatically when a report is directed to it. To open the SAS OUTPUT window, select **Window**. A menu opens. From the menu, select **Output**. The report is displayed automatically. Remember that a report might have multiple parts. If so, scroll down in the window to see all the parts. When you are finished looking at the report, iconize the window.

After you have directed more than one report to a SAS window, scrolling up and down becomes complicated because you scroll not only through the parts of a single report but through multiple reports. At that point, you might prefer to use the other method of viewing the reports, whether or not the reports are directed to SAS windows: Open the SAS Results window, which is available in an iconized form when you invoke SAS IT Resource Management. A list is displayed of the reports that were generated. Select the report that you want to view. When you are finished looking at the reports, iconize the window.

5 Select **Close** in the Manage Report Definitions window to return to the main window of SAS IT Resource Management.

Saving a Report Definition

You can save a report definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Saving a Report Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can save a report definition by following these steps:

1 Create, edit, view, or run the report definition.

For more information about creating, editing, or viewing a report definition, see "Creating/Editing/Viewing a Report Definition" on page 481.

For more information about running a report definition, see "Running a Report Definition" on page 488.

2 You can save the report definition in interactive form or in batch form or in both. Notice that if you save it in interactive form, you can still run it in batch mode by using the %CPRUNRPT macro. Thus, typically report definitions are saved in interactive form.

To save the report definition in interactive form:

a With the Manage Report Definitions window as the active window, select **File**. A menu opens. From the menu, select **Save Report Definition**. The Save Report Definition window opens.

b The **Report Folders** list displays the complete list's subset of folders to which you currently have write access.

If you want to save your report definition to one of these folders, select the name of the folder to which you want to save your report. The name that you selected is displayed in the **Folder** field. (If the folder does not currently exist, it will be created when you save the report.)

If you want to save your report definition to another folder, select **Add Report Folder**. You can then define a libref to point to the library in which you want to add the folder, or you can change an existing libref from "readonly" access to "write" access. For more information, see "Adding a Report Definition Folder" on page 472.

c In the **Report Name** field, type a new name or edit an existing name.

The name can contain letters, numbers, and underscores. The name does not need to begin with a letter. You might want to start the name of your report definition with U in order to identify that this is a user-defined report definition.

The length of a report name must be between one and eight characters.

Note: If the name that you use is identical to a name that is already in the folder, your report definition will overwrite the previous report definition that had that name. △

Note: Do not specify OUTNAME= as a member of a z/OS PDS; that is, do not use the form OUTNAME=DATASET(MEMBER). △

d In the **Description** field, type a new description or edit an existing description.

e Select **OK**. SAS IT Resource Management returns you to the Manage Report Definition window. The report definition that you saved is now on the list of report definitions in the selected folder.

To save the batch form of the report definition:

a Select **Locals**. A menu opens. Select **Preview Macro Source**. The PREVIEW window opens and displays the report definition in the form of a call to a reporting macro (plus SAS statements that specify features such as the title and footnote, if needed).

Select **File ▶ Save as File**

The Save As window opens. Select the directory (UNIX) or folder (Windows) in which you want to save the file. Type the name of the file in which you want to save the report definition, and select **Save**. If the file does not exist, it will be created.

Note: If you edit the statements in this window, they will affect what is saved (if you do a save), but they will not affect what runs interactively (because this is a *copy* of what runs interactively). △

b Select **File ▶ Close** to return to the Manage Report Definitions window.

3 When you want to return to the main window of SAS IT Resource Management, select **Close**.

Selecting a Report Definition

When you select a Report Definition, its values are displayed and you can view them.

For more information about selecting (viewing) a report definition, see "Creating/Editing/Viewing a Report Definition" on page 481.

Working with Report Definitions: Related Topics

Directing a Report to the Web

When you are working interactively, you can direct a report to the Web by specifying values in one or more fields in a GUI. When you are working in batch mode, you can direct a report to the Web by specifying values for one or more parameters when you call a report macro.

Each field is similar (but not identical) to a parameter. If you work in both modes (interactive and batch), you might find it helpful to know which fields correspond *approximately* to which parameters. For that information, see the following table. The table is in three sections, one for each of the windows. (For more information about the windows, see "Directing a Report to the Web (UNIX, Windows)" below.

Table 23.1 Approximate Correspondence of Fields and Parameters

Window: Report Output Options	
Field (in GUI)	Parameter (on macro call)
Web radio button	OUTMODE=WEB

Window: Generate Web Output	
Field (in GUI)	Parameter (on macro call)
HTML Directory	HTMLDIR=<*value*>
SAS Catalog Libref Catalog Name	OUTLOC=<*libref.catalog_name*>
Report Group	OUTDESC=<*description*>
Use Interactive Graphics when possible Java ActiveX	LARGEDEV=JAVA LARGEDEV=ACTIVEX
Clear Gallery	WEBCLR=YES or call %CPWEBINI

Window: Secondary Options for Web Output	
Field (in GUI)	Parameter (on macro call)
HTML URL	HTMLURL=<*value*>

Image Directory	IMAGEDIR=<*value*>
Image URL	IMAGEURL=<*value*>
Web Style	WEBSTYLE=<*value*>
Thumbnail Size	SMALLDEV=<*value*>
Enlarged Size	LARGEDEV=<*value*>
(not available, but there are default names)	OUTNAME=<*value*>

You can direct a report to the Web

- by using OUTMODE=WEB (and, optionally, other Web-related parameters) on your call to the report macro in batch mode. For more information about "the big picture" related to OUTMODE=WEB, see the topic "How the OUT*= Parameters Work Together" in Chapter 3: Report Macros in the SAS IT Resource Management macro reference documentation.
  - For more information about the OUTMODE= parameter, see the OUTMODE= parameter, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.
  - For information about the HTMLDIR= parameter, see the HTMLDIR= parameter, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.
  - For information about the HTMLURL= parameter, see HTMLURL=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  - For information about the IMAGEDIR= parameter, see IMAGEDIR=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  - For information about the IMAGEURL= parameter, see IMAGEURL=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  - For information about the LARGEDEV= parameter, see LARGEDEV=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  - For information about the OUTDESC= parameter, see OUTDESC=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  - For information about the OUTLOC= parameter, see OUTLOC=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  - For information about the OUTNAME= parameter, see OUTNAME=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  - For information about the SMALLDEV= parameter, see SMALLDEV=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation
  - For information about the WEBCLR= parameter, see WEBCLR=, which is in the %CPRUNRPT and %CPXHTML macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.
  - For information about the WEBSTYLE= parameter, see WEBSTYLE=, which is in most of the macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Directing a Report to the Web (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can direct a report to the Web by specifying fields in the Report Output Options window, Generate Web Output window, and Secondary Options for Web Output window. These windows open in sequence.

1. The Report Output Options window:
   a. On the **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
   b. From the **Folders** list, select the folder where the report definition is stored.
   c. From the **Current PDB Reports** list, select the report definition that you want to run.
   d. Select **Locals ▶ Report Output Options** The Report Output Options window opens.
   e. Select **Web**.

      This selection indicates that you want the report to be directed to the Web.

      For the report to actually be directed to the Web, you must *also* specify where the Web report is to be stored. You can specify that location in the **HTML Directory** field in the Generate Web Output window (and also, optionally, in the **Image Directory** field in the Secondary Options for Web Output window).
2. The Generate Web Output window:
   a. In the Report Output Options window, select **Set Attributes**. The Generate Web Output window opens.
   b. The following field is required.
      - **HTML Directory** field

        In this field, specify the directory or folder where you want to store the HTML files that are created for your report (*welcome.htm* and its associated .htm files; an .htm file if the report is a graph report and you plan to use a Java applet or an ActiveX control for the enlarged image of the report; and an .htm file if the report is a text report). The directory or folder must already exist and you must have write access to it. When you want to browse the report, you can start by pointing your Web browser to the *welcome.htm* file in this directory or folder and opening that file. *To create Web output, a value is required in this field*.

        For example, to direct the Web output to a folder named *c:\reports\daily*, type **c:\reports\daily** in the **HTML Directory** field and run the report definition.

        Note: If you prefer to use a macro variable for the value, you can do so. For example, suppose that you want to make a macro variable named mydir and have its value be *c:\reports\daily*.

        - In the body of the SAS Program Editor window, type the code

          ```
          %let mydir=c:\reports\daily ;
          ```

          Then select **Run ▶ Submit** This creates the macro variable, names it, and assigns a value to it.

          For more information about submitting SAS code, see "Submitting a SAS Program through the SAS GUI" on page 18.
        - Type **%str(&mydir)** in the **HTML directory** field. (The **&** obtains the value of the macro variable, and the **%str()** is required so that

the macro variable is evaluated at the appropriate time during the report production.)

- Then, run the report definition. When the report is sent to the Web, the macro processor replaces *%str(&mydir)* with the value *c:\reports\daily*.

△

The value of this field works with the value of the **SAS Catalog Name** field to determine which reports you can view from one Web page. For each HTML directory, all reports that are stored in the same HTML directory and that have the same SAS catalog name are accessible from one welcome.htm Web page.

Note: If the value of **Image Directory** and the value of **HTML Directory** point to the same location, then you do not need to specify the value of HTML URL or the value of **Image URL**, because everything will be stored in one directory. This is convenient and also enables you to easily move the directory as needed. △

Note: If you omit the values of **Image Directory, Image URL**, and **HTML URL** (for more information about these fields, see below), then all of the links in your HTML files are relative to the HTML directory. This combination provides maximum portability. All of your HTML files and image files will be stored together, and the HTML files will search for the images in that same directory. △

c The following fields are optional:

- **SAS Catalog** field (built from the **Libref** field and the **Catalog Name** field)

  In this field, you can specify the name of the SAS catalog (in the format libref.catalog_name) where the SAS IT Resource Management report is stored. This location is used as internal storage for the report. A report is then converted to one or more files that you can display on the Web. You must have a separate HTML directory (and, optionally, a separate image directory) for each catalog.

  This value is not required. If you do not specify this value, then the default location is WORK.CPWEB_GR, which is temporary storage space for your current SAS session. When you exit from your current SAS session, all items in temporary storage are deleted.

- **Report Group** field

  In this field, you can specify a string of text that describes the report group. The string can be a maximum of 40 characters and should not contain double quotation marks.

  - If you have one or more graph reports in your HTML directory, this value is optional but, if specified, adds a report grouping functionality to your Web page.
  - If you have one text report in your HTML directory, this value is optional, but is helpful if you are using this field for other reports on this Web page.
  - If you have more than one text report in your HTML directory, this value is required, and each text report must have a unique value, because text reports cannot be grouped.

  When you display the *welcome.htm* page by using your Web browser, all reports that have the same description and the same SAS catalog

name are displayed in the same report group. The description is used as the name of the report group.

- **`Use Interactive Graphics When Possible`** field

  For a graph report, you can specify whether the report is to be static or the report is to be interactive (if possible).

  - If this box is empty, the report is to be static. Both the "thumbnail" and "enlarged" versions of the report will use GIF images, and the size of the images will be based on the value of the **`Thumbnail Size`** field and the value of the **`Enlarged Size`** field, as specified (or defaulted) in the Secondary Options for Web Output window.
  - If this box has an X in it, the report is to be interactive (if possible). The report is to use Java or ActiveX, as selected. The size of the small image will be based on the value of the **`Thumbnail Size`** field in the Secondary Options for Web Output window. Do not change the value (Java or ActiveX) in the **`Enlarged Size`** field in the **`Secondary Options for Web Output`** window. If possible, the "enlarged" version of the report will use a Java applet or ActiveX control.

    Note: Support for ActiveX is experimental. △

    Note: "If possible" refers to the report style. Some report styles and sub-styles cannot be interactive. For example, Spectrum Graphs cannot be interactive. △

- **`Clear Gallery`** button

  This button opens the Clear Web Output window, through which you can clear reports from the SAS catalog, HTML directory, and, if used, image directory.

  Select **`Yes`** or **`No`**, and then select **`OK`**.

3 The Secondary Options for Web Output window

  a In the Generate Web Output window, select **`Advanced Options`**. The Secondary Options for Web Output window opens.

  b All the fields in this window are optional.

  - **`HTML URL`** field

    In this field, you can specify a relative URL (relative to the location of *welcome.htm*) or an absolute URL. *This value is not required.*

    The value that you specify is used as a prefix for all references to HTML files (*welcome.htm* and its associated .htm files). For example, if your reports are stored in the directory *webrpts\reports* on your *c:* drive (that is, the value of **`HTML directory`** is *c:\webrpts\reports*) on a Windows server named *www.reporter.com*, then you might specify *http://www.reporter.com/reports* as the URL for the HTML files, if *webrpts* is the "root" for the server. (If you prefer to use a macro variable for the value, you can do so. See the **`HTML Directory`** field, above, for an example.)

    A URL is an Internet Web address that identifies where a file is located. If you do not specify this value, then the links or file addresses in your HTML files are relative to the directory in which the HTML files reside. In other words, when a URL is not coded in your HTML file, then the HTML file expects to find the linked file or image in the same directory where that HTML file is stored. When you store all your images and HTML files in one location, you do not need to specify the

HTML URL and you can easily move the entire directory without breaking links.

If you specify this value, then the URL is hard-coded in your HTML source. You can use this value when you want to hard-code a URL in the HTML files that others can access remotely. When this is the case, you must be careful when moving these files so that you do not break any of the links. The HTML file will look for the linked files only in the location that is specified in this URL.

Typically, the value of **HTML URL** is not specified.

□ **Image Directory** field

In this field, you can specify the directory in which to store the GIF image files for your report. If welcome.htm and its associated .htm files use icons, then the GIF files for the icons are also stored here. *This value is not required. If you do not specify this value, then the value of* **HTML Directory** *is used by default.*

The directory must already exist and you must have write access to it.

When a graph report is created, at least one GIF image (the "thumbnail" image) is created. The other image (the "enlarged" image) may be a GIF image, a Java applet (with an extension of .htm), or an ActiveX control (with an extension of .htm). The GIF images are stored in this directory.

If you are creating the report on UNIX or Windows and want to use the image directory, the value of **Image Directory** must be the complete path and name of the directory or folder. For example, on Windows you might specify the value *c:\webrpts\reports* for **Image Directory** to store your images in the *\webrpts\reports* folder on your *c:* drive. (If you prefer to use a macro variable for the value, you can do so. See the **HTML Directory** field, above, for an example.)

Typically, the value of the **Image Directory** field is not specified.

□ **Image URL** field

In this field, you can specify the location (URL address) that is used in your HTML output to locate your report images. In *welcome.htm* and its associated .htm files, the value that you specify is used as a prefix for all references to GIF files. You can specify a relative URL (relative to the location of welcome.htm) or an absolute URL.

This value is required if you do not specify the same value or location for the **HTML Directory** *and the* **Image Directory**. If you do not specify this value, the HTML files look for the images in the directory where your HTML files are stored. If the value of **Image Directory** and the value of **HTML Directory** are different, the HTML files cannot display the images unless you provide the location of the images by specifying the image URL.

A URL is an Internet Web address that identifies where a file is located. If you do not specify this option, then the links or file addresses in your HTML files (that link to images) are relative to the directory in which the HTML files are placed. This option enables you to identify a specific location or address where the images are stored.

For example, if your report images are stored in the folder *webrpts\reports* on your *c:* drive (that is, the value of Image Directory is *c:\webrpts\reports*) on the Windows server named *www.reporter.com*, then you might specify *IMAGEURL=http://www.reporter.com/reports* as the URL for the GIF files, if *webrpts* is the "root" for the server. (If

you prefer to use a macro variable for the value, you can do so. See the **HTML Directory** field, above, for an example.)

If the value of **Image Directory** is the same as the value of **HTML Directory** (that is, you store all report files in the same location), then you can easily move all files to a new location without breaking any of the "links" that are coded in the HTML files. If you store your HTML files and IMAGE files in separate directories, your links from the HTML to the image files might not work if you move the files.

Typically, the value of **Image URL** is not specified.

- **Web Style** field

  Select the arrow for this field. A list opens. From the list, you can select the layout (style) for the gallery of reports. The Web style affects the type of frames, the location of titles, and the control options.

  Valid values are *GALLERY*, *GALLERY2*, and *DYNAMIC*. *The default value is GALLERY2.*

- **Thumbnail Size** field

  Select the arrow for this field. A list opens. From the list, you can specify the name of the SAS/GRAPH device driver to use in order to create the small ("thumbnail") GIF image of your report. *The default is SMALLDEV=GIF160 when WEBSTYLE=GALLERY. The default value is GIF260 when WEBSTYLE=GALLERY2 or WEBSTYLE=DYNAMIC.*

  GIF160 produces a GIF image of 160x120 pixels, and *GIF260* produces an image whose size is 260x195 pixels. Other choices and image sizes include *GIF373* (373x280), *GIF570* (570x480), and *GIF* (800x600).

  The selected device driver reads the report from the SAS catalog that is specified by the SAS catalog name and writes the "thumbnail" image file of the report to the image directory.

- **Enlarged Size** field

  If you selected Use Interactive Graphics When Possible, a value (Java or ActiveX) is already specified in this field.

  Otherwise, you can select the arrow for this field. A list opens. From the list, you can specify the name of the SAS/GRAPH device driver to use in order to create the enlarged version of your report. *The default value is LARGEDEV=GIF733, which produces an image whose size is 733x550 pixels.*

  Other choices (and their corresponding image sizes) include *GIF160* (160x120), *GIF260* (260x195), *GIF373* (373x280), *GIF570* (570x480), and *GIF* (800x600).

  If the **Use Interactive Graphics When Possible** field is not selected, then the "enlarged" report is not interactive. If the **Use Interactive Graphics When Possible** field is selected, then the "enlarged" report is interactive and, depending on whether **Java** or **ActiveX** is selected, the mechanism for displaying the report uses a Java applet or ActiveX controls.

  - If **Java** is selected, then when users click on the thumbnail report in their Web browser, the enlarged report is displayed by using a Java applet. The applet provides some ability to dynamically manipulate the graph, including drill-down capability if the report definition includes subgroups. (For additional customization options, the user can access the shortcut menu by clicking the right mouse button.)

For information that affects the user's ability to display the report by using a Java applet, see the note about LARGEDEV=JAVA in the description of the LARGEDEV= parameter, which is in most macros in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.

- If **ActiveX** is selected, then when users click on the thumbnail report in the Web browser, the enlarged report is displayed by using ActiveX controls. The ActiveX controls provide some ability to dynamically manipulate the graph, if the report definition includes subgroups. (For additional customization options, the user can access the shortcut menu by clicking the right mouse button.)

- Report Name

  In the GUI, you cannot specify a value for report name.

  For a graph report, the report name is created automatically, as follows:

  - if the enlarged report image is not generated by Java or ActiveX, the following pair of report names is used: *S000000n.gif* (for the thumbnail image), and *L000000n.gif* (for the enlarged image).
  - if the enlarged report image is generated by Java or ActiveX, the following pair of report names is used: *S000000n.gif* (for the thumbnail image), and *Ln.htm* (for the enlarged image).

  For a text report, the report name is created automatically, as follows:

  - if the report definition was saved, the report is given the name that you specified for the report definition
  - if the report definition was not saved, the report is given the name *PRTRPTn.htm* (for a print-style report) or *TABRPTn.htm* (for a tabular-style report).

  Note: If you specify one or more BY values in a print-style report or you specified one or more PRINTBY= values in a tabular-style report, the report is broken into *m* small files. The characters *_page1* through *_pagem* are appended to the names of the smaller files. △

4 To return to the Manage Report Definitions window, select **OK ▶ OK ▶ OK**

Naming Conventions for Report Definitions

When you create and save a *custom report definition*, you must select the folder in which to save the report definition, and you must provide a name for the report definition.

The report definition name must follow SAS naming conventions. The name can be between one and eight characters long and can contain letters, numbers, and the underscore (_) character. The name must begin with a letter and cannot contain blank spaces or any other special characters. There are no other rules that apply to user-defined reports or report definitions, but you might find it convenient to use a naming convention similar to the one that is used for some SAS IT Resource Management supplied report definitions.

Some SAS IT Resource Management *supplied report definitions* use a naming convention that provides you with information about the report and the data that will be used for the report. The supplied reports use a name format of *pprrrtnn*. The first position identifies the operating environment and the second position represents the

data source, collector, and measurement source. The third, fourth, and fifth positions (*rrr*) identify the resource type. The sixth space (*t*) represents the type of report, and the seventh and eighth positions (*nn*) represent a unique identifier, which can be any alphanumeric combination.

Report Styles

1 The following report styles are available:
   - □ "Print Report" on page 499
   - □ "Plot Report" on page 499
   - □ "Chart Report" on page 500
   - □ "Three-Dimensional Graph Report" on page 500
   - □ "Spectrum Report" on page 500
   - □ "Source Report" on page 500
   - □ "Tabular Report" on page 501.
2 To set additional style options such as the number of bars or slices of pie to display on the report, select **`Advanced`**.

Other report options such as variables and colors can be selected from the Manage Report Definitions window.

Print Report

If you want to print a basic report of your data file, select the Print report style in the Report Style Selection window. This report style prints up to 15 selected variables from your data file, in a simple columnar format.

Note: If you are running the %CPPRINT macro from batch, you can select up to 30 variables to be printed. △

The report does not contain row or column lines and does not analyze the data in any way unless you select summarization for each column of data. This style provides a very simple method of printing your data.

Plot Report

Plots are one of the most common forms of graphical analysis. With plots, you can plot a continuous variable such as time (the DATETIME variable) on the horizontal or X axis and one or more analysis variables on the vertical or Y axis. You can also compare two analysis variables with different data ranges, using separate vertical (Y) axes, the left and right.

For Plot reports, you can select a maximum of 15 analysis (Y) variables.

If a class variable or multiple analysis (Y) variables are selected, multiple lines are drawn on the same graph, each of which represents either a specific class or a specific analysis (Y) variable. If you select a class variable, you can select only one analysis variable.

If you select one or more BY variables for a Plot report, one graph is generated for each combination of values for the BY variables.

Chart Report

Charts are very useful for displaying summary statistics on your data. You can select a single analysis variable categorized by a single class variable, and each portion of the chart represents the sum or mean of the analysis variable in that category. You can also select a maximum of 15 analysis variables to display each variable as a category or group of data. You can also select a Stack variable and produce a separate portion of the chart for each value of the stack variable.

If you select a single analysis variable, you can select a class variable, and for each value of the class variable, a separate portion is produced on the chart.

If you select one or more BY variables for the report, one graph is generated for each combination of values for the BY variables.

When you select one or more analysis variable, you can further categorize the variable(s) by a GROUP or SUBGROUP variable. If you select a GROUP variable, the chart displays groups of bars (for bar charts) or a separate chart (for pie or star charts) for each value of the group variable. If you select a SUBGROUP variable, each bar (or portion on the chart), which represents the analysis variable, is subdivided for each value of the subgroup variable. You cannot select a subgroup variable if your chart sub-style is Pie or Star. You cannot select a stack variable and a subgroup variable in the same report.

Three-Dimensional Graph Report

The 3D Graph produces a three-dimensional scatter plot that displays multidimensional trends or concentrations in data. This type of report plots one or more variables against X and Y axis variables.

You can select a maximum of 15 analysis variables. If you select multiple analysis variables, the variable names are displayed on the Y axis. The values of those variables are displayed on the Z axis and the X variable is displayed on the X axis.

You can select a class variable that is represented by color to add a fourth dimension to the plot. If you select a class variable, then you can select only one analysis variable. The class variable is displayed on the Y axis, the analysis variable is displayed on the Z axis, and the X variable is displayed on the X axis.

Spectrum Report

The Spectrum report generates a three-dimensional graph with colors that represent the relationship among three variables on a two-dimensional surface. The surface is typically defined by DATETIME on the X axis and a class variable on the Y axis.

By default, the colors on the graph represent the value of the analysis (Z) variable at a given time, using a cool-to-hot color spectrum. The colors range from cool (blue) to warm (green, yellow, orange) to hot (red) when the variable's value changes. Blue represents cool or low numbers and red represents hot or high numbers.

For this report style, you can specify a maximum of 15 analysis variables. If you specify a class variable by which to group the analysis variable on the plot, then you can specify only one analysis variable.

Source Report

The Source report style enables you to create complex reports that can require multiple steps to manipulate your data, while still using the global values of variables such as begin and end dates and WHERE expressions. You can use this report style to run an existing SAS program, such as an existing report definition, that reads and further summarizes a specified group of data.

When you select the report type of Source, you specify the location of the source that you want to run in this report definition. This can be any source code that you want to

run, such as a sample report that you have saved as source or a report definition that you have created and saved as source code. You can then specify an existing data set to read into this report. For example, you might have an existing SAS data set that you want to read and further summarize. When you select variables for your Source report definition, those variables are used to further summarize the existing data. The newly summarized data is saved in a new data set, then read by your existing program, and that program is submitted to create your final report.

Tabular Report

The Tabular report is displayed in a table the classification variables and a summary of the data in each class. The variables are summarized based on your selected statistics and the table displays summary information about one of six table styles. This report is an interface to the TABULATE procedure in Base SAS software.

This report provides six sub-styles. The first sub-style summarizes a list of analysis variables with selected statistics, by listing each variable on a separate row in the chart. Sub-styles two, three, and four place both the analysis variables and the statistics across the top of the table and have a class variable and its summary in separate table rows. Sub-styles five and six place both analysis variables and statistics across the table and have nested class summarization of multiple variables in table rows.

Variable Types

The following variables are available for your reports.

Across
: specifies the variables to use for column headings for tabular report types.

Analysis
: specifies the variables that you want to analyze in your report. If you select statistics for your report, those are applied to this analysis variable. If you specify multiple analysis variables, the unit of measure for all of the variables should be the same and the range of values should be similar.

By
: specifies the variables in the order that you want them sorted for your report. A separate graph or chart is produced in the report, for each value of the By variable or for each combination of By variable values, if you have multiple By variables. For example, if you have four *diskid* variables on three machines and you select the variables *machine* and *diskid* as By variables, you will receive 12 reports, one for each unique combination of the two variables. If you do not select a By variable, observations are sorted in the order that the variables are listed in the By variable or Class variable list in your PDB.

Class
: specifies a group or category variable on which calculations are performed. For each unique value of the class variable, a separate portion is created on the graph. For example, a portion might represent a portion of pie on a pie chart, but all values of the class variable are included in a single graph in the report.

Group
: is the name of the variable that subsets the observations. If you select multiple analysis variables, they are displayed in groups with one group for each value of the group variable. The presentation of the group depends on your report type and subtype, such as a report style of chart and a sub-style of star. If you select a Group variable, you can select either a Stack variable or Subgroup variable, but not both.

ID
is the variable that, for the Print report style, can be used to replace observation numbers. For example, if MACHINE and DATETIME are ID variables and you run a Print report, the observations are identified by the values of MACHINE and DATETIME and not the observation number (1, 2, 3, ...).

Stack
is the name of the variable by which your data is subdivided. A separate bar or block is produced for each value of the Stack variable. Each bar (for a bar chart) is divided into sections, one for each analysis variable.

Statistics
specifies the statistic to use to summarize data. The available statistics vary depending on your report type.

Subgroup
specifies a variable by which to subdivide the groups of variables that are specified in the Group variable. Each subdivision of the Group variable becomes a portion of the graph, such as a section on a bar.

X Variable
specifies the variable to display on the X axis for report styles that use an X axis.

Weight
is the variable by which to weight statistical calculations for Interval data. The default weight variable is Duration (for tables that have a Type of Interval).

Appendices for z/OS

A Collection of Tips for z/OS

Here are some tips that might be useful to you as you generate reports with SAS IT Resource Management.

- Test report definitions on a small subset of your data so that the testing proceeds quickly.

  For details, see “Tips for Subsetting Data on z/OS” on page 504 in this appendix.
- If running your report does not seem to produce any results, be sure that you are looking for the report in the correct window. The default window for graph reports is the SAS/GRAPH window. The default window for text reports is the SAS OUTPUT window. (For more about accessing windows, see “Using the z/OS GUI” on page 21.)
- If running your report definition produces unexpected results, examine the SAS log for messages.
- If you get the message

  ```
  No observations were selected
  ```

  check the following:
  - Does the SAS log display the report definition that you ran? Check to see that you are running at the PDB level (REDLVL) at which you have data. For more information about the SAS log, see “Using SAS Windows” on page 24.
  - Check to see that you are running with the date range (BEGIN and END) in which you have data. For more about date range subsetting, see “Subsetting via DATETIME” on page 504.

- Check to see that you are running with a subsetting WHERE expression (local WHERE or global WHERE) for which you have data. If not, and if you are running a supplied report definition that requires setup, the lack of a subsetting WHERE expression could easily be the cause. Check the Explanation for the report definition.

  For more about a subsetting WHERE expression, see "Subsetting by a Simple Local WHERE Expression" on page 504 and "Subsetting by a Simple Global WHERE Expression" on page 505.

- Check to see that there really is data in the table at the level in that date range for the variables in the report definition. From the main menu in the SAS IT Resource Management server GUI for z/OS, follow this path: **PDB Data ▶ Examine PDB Data ▶** ***select the table*** **▶ ItemActions ▶ Browse Data** ***or*** **View Data**

  Then, if you are looking for specific information, you might want to subset the data by using a WHERE expression. For instance, if you are looking for observations on machine 41TS because a graph shows an outlier value for that machine, you could follow this path from the FSBROWSE window: **Search ▶ Where ▶** ***type machine='41TS'*** **▶ OK**

- If the data that you see in your table by using **`Browse Data`** or **`View Data`** is unexpected, check the table's definition. From the SAS IT Resource Management main menu, follow this path: **PDB Admin ▶ Config Active PDB Dictionary ▶** ***select the table*** **▶ ItemActions ▶ Browse Definition**

  Check that the Kept status of the appropriate table is set to *Yes* and the appropriate levels are set to nonzero durations.

  In the table definition window, select **`Variable List`** to check the variables' definitions. Check that the Kept status of the appropriate variables is set to *Yes*. For the summary levels of regular and derived variables, check that the appropriate statistics are selected. For a formula variable, check that the variable is defined at the appropriate level and that the table and variables on which it is based are defined appropriately. To check the levels for which it is defined, select ***the formula variable*** **▶ ItemActions ▶ Edit Definition**

  Note: A formula variable does not have selectable statistics. If you want statistics on a formula variable, you must define the statistics as formula variables at the desired summary levels. △

- Check that the collector was configured correctly to provide the metrics.

- If you do not get the "No observations" window but you get no report, perhaps all data values are missing. That is, the observation might exist but the value on the observation might be set to a SAS missing-value code.

  At detail level, this can occur if you have not set up data logging properly. At all levels, this can occur for a formula variable if the underlying variables have a Kept status of *No*.

- For the type of report definition that you are running, select the report type from the pull-down menu under **`DesignGraphRpt`** or **`DesignTextRpt`**. Then read the online documentation which contains tips on generating reports.

- The most probable cause of the message

  ```
  ERROR: The left vertical axis labeled could not be fit as specified.
  ```

  in a graph report is that you have more classes than will fit in the legend on the graph. A solution to this problem is to increase the height of the GRAPH window. From the report's main definition screen, select **`Output Options`**. Edit the

GOptions field from DEVICE=value to DEVICE=value VPOS=100, where VPOS=100 means a vertical height of 100 rows. Then rerun the report definition.

- Add formula variables (variables that are calculated only when accessed) to simplify reporting. For details, see "Creating/Editing/Viewing a Formula Variable" on page 388.

Tips for Reporting on z/OS

The following topics are discussed in this document:

Tips for Subsetting Data on z/OS

Subsetting via DATETIME

If your report displays no observations (you get an empty report) or if your report displays observations in only one part of the date range of the processed or reduced data, you might need to reset the report definition's datetime range.

Follow this path from the report's main report definition window: **DateTime Range ▶ Reset from PDB ▶ OK**

SAS IT Resource Management looks in the PDB's data dictionary to find the earliest and latest DATETIME values at the level that is specified on the main report definition window.

If you want to use less of that range, you can type over the values in the **`Begin`** and **`End`** fields.

Note: The need to reset might occur when you are designing or modifying report definitions and switch from one table to another or one level to another. The begin and end points of the range are global variables that retain their values until you change them or switch the active PDB. △

Subsetting by a Simple Local WHERE Expression

A local WHERE expression applies to a single report definition.

If your report definition generates a report that includes observations in which you are not interested, you can use the local WHERE to restrict the observations to the ones in which you are interested. For instance, suppose you have a report definition with a BY variables list that includes MACHINE, and you have many machines but you are only interested in a few of them for this report definition. Rather than get dozens of graphs and ignore all but a few, you can generate just the ones that you want to see.

From the report's main definition screen, select **`Local Where`**. Under **`Local subsetting`**, follow this path: **Varname ▶ MACHINE ▶ Operator ▶ IN ▶ Value(s) ▶ *select the machines on which you want to report* ▶ OK ▶ OK**

This method is not restricted to subsetting by MACHINE. You can subset by any variable that is in the specified level of the specified table. For instance, you can report on only first shift data by using this WHERE expression:

```
SHIFT = '1'
```

If you want to subset by more than a few values, you might want to create a user format.

Note: By default, if there is an expression in the local WHERE, it overrides the expression, if any, in the global WHERE. If you want the report definition to use both the local WHERE and the global WHERE, insert the words SAME AND in front of the WHERE expression in the local WHERE. △

Note: Do not use the ampersand (&) to mean AND in WHERE expressions. △

Subsetting by a Simple Global WHERE Expression

A global WHERE expression applies to all report definitions.

You might want to generate a number of reports with the same subsetting; for this, use a global WHERE expression. For instance, you might be having trouble with a few machines and want to run a number of reports for just those machines. In this case, rather than changing the local WHERE for each report, you can set the global WHERE (on any report definition) and it will apply to all report definitions until you reset it to blank.

From the report's main definition screen, select **`Global Where`**. Under **`Global subsetting`**, follow this path: **Varname ▶ MACHINE ▶ Operator ▶ IN ▶ Value(s) ▶** ***select the machines on which you want to report*** **▶ OK ▶ OK**

This method is not restricted to subsetting by MACHINE. You can subset by any variable that is in the specified level of the specified table. For instance, you can report on only first shift data by using this WHERE expression:

```
SHIFT = '1'
```

If you want to subset by more than a few values, you might want to create a user format.

Note: By default, if there is an expression in the local WHERE, the global WHERE is ignored (unless the local WHERE starts with the words SAME AND). △

Note: Do not use the ampersand (&) to mean AND in WHERE expressions. △

Appendices for UNIX and Windows

A Collection of Tips for UNIX and Windows

Here are some tips that might be useful to you as you generate reports with the SAS IT Resource Management GUI.

- If running your report definition does not seem to produce any results, be sure that you are looking for the report in the correct window. The default window for graph reports is the SAS/GRAPH window. The default window for text reports is the SAS Output window. You can check your report setting by selecting **Locals ▶ Report Output Options**

  One of four output options is available:

  - SAS Window
  - SAS Catalog
  - External File
  - Web.

- If running your report definition produces unexpected results, examine the SAS log for messages.
- Some SAS IT Resource Management supplied report definitions require setup to run properly.

  To be certain that you have met all the setup requirements for a report definition, follow this path from the SAS IT Resource Management main window: **Reporting ▶ Explore Supplied Reports ▶** ***select the report definition*** **▶** ***click right mouse button*** **▶ Show Report Explanation**
- If you get the message

  ```
  No observations were selected
  ```

  check the following:
  - Does the SAS log display the report definition that you ran? Check to see that you are running at the (REDLVL) at which you have data.
  - Check to see that you are running with the date range (BEGIN and END) in which you have data. For more about date range subsetting, see "Subsetting via DATETIME" on page 507 .
  - Check to see that you are running with a subsetting WHERE expression (local WHERE or global WHERE) for which you have data. If not, and if you are running a supplied report definition that requires setup, the lack of a subsetting WHERE expression could easily be the cause. Check the Explanation for the report definition.

    For more about a subsetting WHERE expression, see "Subsetting by a Simple Local WHERE Expression" on page 508 and "Subsetting by a Simple Global WHERE Expression" on page 508 .
  - Check to see that there really is data in that table at that level in that date range for the variables in the report definition. From the SAS IT Resource Management Administration tab, follow this path: **Examine PDB Data ▶** ***select the level*** **▶** ***select the table*** **▶** ***click right mouse button*** **▶ View Data**

    Note: Use can also use %CPPRINT to browse the data. See the macro reference documentation for SAS IT Resource Management for more information. △
  - If the data that you see in your table by using Browse Data or View Data is unexpected, check the table's definition. From the SAS IT Resource Management Administration tab, follow this path: **Manage Tables ▶** ***select the table*** **▶** ***click right mouse button*** **▶ Properties**

    Check that the Kept status of the table is set to *Yes*. Then select **`Variables`** and check that the Kept status of the appropriate variables is set to *Yes*.

    To check this, follow this path: ***select a variable*** **▶** ***click right mouse button*** **▶ Properties**

    You should also check that the appropriate levels are set to nonzero durations. For the summary levels, check that the appropriate statistics are selected. (For a formula variable, check that the formula is applied to the appropriate level and that the table and variables on which it is based are defined appropriately.)

    Note: A formula variable does not have selectable statistics. If you want statistics on a formula variable, you must define the statistics as formula variables at the desired summary levels. △
  - Check that the collector was configured correctly to provide the metrics.

- If you do not get the "No observations" window but you get no report, perhaps all data values are missing. That is, the observation might exist but the value on the observation is set to the missing value code.

  At the detail level, this can occur if you have not set up data logging properly. At all levels, this can occur for a formula variable if the underlying variables have a Kept status of *No*.

  Note: A formula variable does not have selectable statistics. If you want statistics on a formula variable, you must define the statistics as formula variables at the desired summary levels. △

- Read the online documentation on Report Styles. From the **`Reporting`** tab select: **Manage Report Definitions ▶** ***select the Current Report Style image*** **▶ Help**
- The most probable cause of the message

  ```
  ERROR: The left vertical axis labeled could not be fit as specified.
  ```

  is that you have more classes than will fit in the legend on the graph.

  A solution to this problem is to increase the height of the Graph window. Type the following statement in the body of the Program Editor window:

  ```
  goptions device=value vpos=100 ;
  ```

  where VPOS=100 means a vertical height of 100 rows. Then submit it and rerun the report definition.
- Test report definitions on a subset of your data so that the testing proceeds quickly. For details, see “Tips for Subsetting Data on UNIX and Windows” on page 507.
- If the data that you see in your table by using **`Browse Data`** or **`View Data`** are unexpected, check the table’s definition. From the SAS IT Resource Management Administration tab, follow this path: **Manage Tables ▶** ***select the table*** **▶** ***click right mouse button*** **▶ Properties**
- Add formula variables (variables that are calculated only when accessed) to simplify reporting. For more information, see “Creating/Editing/Viewing a Formula Variable” on page 388.

Tips for Reporting on UNIX and Windows

The following topics are discussed in this document:

Tips for Subsetting Data on UNIX and Windows

Subsetting via DATETIME

If your report displays no observations (that is, you get an empty report) or if your report displays observations in only one part of the date range of the processed or reduced data, you might need to reset the report definition’s datetime range.

To subset the data by datetime, from the Manage Report Definitions window, select the report that you want to work with. Then, select the arrow to the right of the

Subset Data field and the **Start/End Dates/Time** tab opens. You can select start dates and times and/or end dates and times for this report and for all reports. To select dates and times, click the arrow to the right of the appropriate field.

You can choose either relative or absolute dates, by clicking the check box next to the **Relative date** or **Absolute date** field. Use the control arrows to adjust the dates and times that you require. The relative date section uses keywords like "number of days before latest" or "number of weeks after earliest." This enables you to limit the reported data to a moving window of time, relative to the latest or earliest date and time stamps in the actual data. For more information about this, select **help** in this window.

Note: The need to reset might occur when you are designing or modifying report definitions and switch from one table to another or one level to another. The begin and end points of the range are global variables that retain their values until you change them or switch the active PDB. △

Subsetting by a Simple Local WHERE Expression

A local WHERE expression applies to a single report definition.

If your report definition generates a report that includes observations in which you are not interested, you can use the local WHERE to restrict the observations to the ones in which you are interested. For instance, suppose you have a report definition with a BY variables list that includes MACHINE, and you have many machines but you are only interested in a few of them for this report definition. Rather than get dozens of graphs and ignore all but a few, you can generate just the ones that you want to see.

In the main Manage Report Definitions window, select a report from the list of **Current PDB Reports**. Select the arrow to the right of the **Subset Data** field and then select **Query/Where Clause Builder**. For the local query, follow this path: **Variable ▶ *select MACHINE* ▶ OK ▶ Operator ▶ *select IN* ▶ OK ▶ Value ▶ *select the machines on which you want to report* ▶ OK**

This method is not restricted to subsetting by MACHINE. You can subset by any variable that is in the specified level of the specified table. For instance, you can report on only first shift data by using this expression:

```
SHIFT = '1'
```

Note: By default, if there is an expression in the local WHERE, it overrides the expression, if any, in the global WHERE. If you want the report definition to use both the local WHERE and the global WHERE, insert the words SAME AND in front of the expression in the local WHERE. △

Subsetting by a Simple Global WHERE Expression

A global WHERE expression applies to all report definitions.

You might want to generate a number of reports with the same subsetting; for this, use a global WHERE expression. For instance, you might be having trouble with a few machines and want to generate a number of reports for just those machines. In this case, rather than using the local WHERE for each report, you can set the global WHERE (on any of the report definitions) and it will apply to all report definitions until you reset it to blank.

In the main Manage Report Definitions window, select a report from the list of **Current PDB Reports**. Select the arrow to the right of the **Subset Data** field and then select **Query/Where Clause Builder**. For the global query, follow this path: **Variable ▶ *select MACHINE* ▶ OK ▶ Operator ▶ *select IN* ▶ OK ▶ Value ▶ *select the machines on which you want to report* ▶ OK**

This method is not restricted to subsetting by the variable MACHINE. You can subset by any variable that is in the specified level of the specified table. For instance, you can report on only first shift data by using this expression:

```
SHIFT = '1'
```

Note: By default, if there is an expression in the Local Query, the Global Query is ignored (unless **And** is selected between the two queries). △

CHAPTER

24

Reporting: Working with Reports

Working with Reports: Main Topics

Overview of Reports

When a report definition runs, it generates a report.
For more information about reports, see the following topics:

- "Deleting a Report" on page 513
- "Generating a Report" on page 511
- "Printing a Report" on page 511
- "Saving a Report" on page 512
- "Troubleshooting Report Problems" on page 513.

Generating a Report

When a report definition runs, a report is generated. For more information, see "Running a Report Definition" on page 488.

Note: In a drill-down report that is based on a graph applet, one way to back up from a drill-down operation is to press Ctrl + Shift while holding the left mouse button. △

Printing a Report

Direct your report to the SAS Graph or SAS Output window. From these windows, you can print your report.

1 If the Graph or Output window is not already the active window, make it the active window by making the Manage Report Definitions window active and selecting **Globals ▶ Graph** or **Globals ▶ Output**

2 From the Graph window or Output window, select **File ▶ Print ▶ OK**

Saving a Report

You can save a report

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Saving a Report (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can save a report by following these steps:

1 First, generate the report to the default location (the Graph window, for graph reports, and the Output window, for text reports).

For more information, see "Generating a Report" on page 511.

2 If the report is in the Graph window, you can save it to a SAS catalog or to an external file.

□ To save the report to a SAS catalog, select **File ▶ Save as Image**

The Save As window opens. In the left frame, double-click on the name of the library that you want to use, and then select the catalog in which you want to save the image. (If you do not want to use an existing catalog and want to make a new catalog in that library, then right-click in the right frame, and follow this path: **New ▶ Catalog ▶ OK ▶** ***type the name of the catalog*** **▶ OK**

Then, select the new catalog.) The right frame displays the names of the images that are in that catalog now. In the **Entry Name** field, type a name for the new image, and then select **Save**.

To view the report later, from the Manage Report Definitions window, select **Globals ▶ Graph**

The Graph window opens. Then select **Tools ▶ Image Editor ▶ File ▶ Open Image Entry**

The Open window opens. In the left frame, double-click on the name of the library that contains the catalog, and then double-click on the name of the catalog. In the right frame, select the name of the report, and select **Open**. The Image Editor window opens and displays the report.

□ To save the report to an external file, select **File ▶ Export as Image**

The Export window opens. In the **Save In** field, select the directory or folder in which you want to save the image. In the **File Name** field, type a name for the new image. Select the arrow to the right of the **Save as type** field. A menu opens. Select the format in which you want the image to be saved, and then select **Save**.

To view the image later, from the Manage Report Definitions window, select **Globals ▶ Graph** The Graph window opens. Then select **Tools ▶ Image Editor ▶ File ▶ Open**

The Import window opens. In the **Look In** field, select the directory that contains the file. In the **Files of Type** field, select the type of the file. Select the name of the file, and select **Open**. The Image Editor window opens and displays the report.

3 If the report is in the Output window, you can save the report to a SAS catalog or to an external file.

□ To save the report to a SAS catalog, select **File ▶ Save As Object**

The Save As window opens. In the left frame, double-click on the name of the library that you want to use, and then select the catalog in which you want to save the image. (If you do not want to use an existing catalog and want to make a new catalog in that library, right-click in the right frame, and select **New ▶ Catalog ▶ OK ▶** ***type the name of the catalog*** **▶ OK**

Then, select the new catalog.) The right frame displays the names of the reports that are in that catalog now. In the **`Entry Name`** field, type a name for the report. In the **`Entry Description`** field, type a description for the report. Select the arrow to the right of the **`Entry Type`** field, and select **`OUTPUT entry(OUTPUT)`**. Then, select **`Save`**.

To view the report later, from the Manage Report Definitions window, select **Globals ▶ Output** or double-click on the **`Output`** icon. The Output window opens. Then select **View ▶ Explorer**

The Explorer window opens. Double-click on the name of the library that contains the catalog. Double-click on the name of the catalog that contains the report. In the right frame, double-click on the name of the report. The Build window opens and displays the report. If you do not want to see line numbers, type *numbers off* in the command field in the SAS IT Resource Management window, and then press ENTER or Return.

□ To save the report to an external file, select **File ▶ Save As**

The Save As window opens. In the **`Save In`** field, select the directory or folder in which you want to save the report. In the **`File Name`** field, type a name for the report. Select the arrow to the right of the **`Save as type`** field. A menu opens. Select **`List Files`**, and then select **`Save`**.

To view the report later, from the Manage Report Definitions window, select **Globals ▶ Output**

The Output window opens. Then select **Tools ▶ Text Editor ▶ File ▶ Open**

The Open window opens. In the **`Look In`** field, select the directory that contains the file. In the **`Files of Type`** field, select **`List Files`**. Select the name of the file, and select **`Open`**. The report is displayed.

Note: If some of the border characters in the report are incorrect, you can edit the report with an ASCII editor and replace those characters with the correct characters. For example, you can replace the hex83 derived character (which sometimes displays as an italic “f”) with an underscore.

4 Iconize the SAS Graph or Output window. The Results window becomes active.

5 Select the Manage Report Definitions window.

6 To return to the main window of SAS IT Resource Management, select **`Close`** in the Manage Report Definitions window.

Deleting a Report

To delete an output file that you have saved, use the remove command that is specific to the system on which you have installed the SAS IT Resource Management client.

Troubleshooting Report Problems

If your report displays no observations (it is empty) or if your report displays observations from only one part of the data that you specified in your date/time range criteria, review the following tips for assistance correcting your report.

- Verify that your date and time ranges that are specified in the Subset Data window are valid for the reduction levels in the PDB that you are using for this report.
- Check the SAS LOG window for possible errors or warnings.
- If you receive strange error messages or no messages at all, you might have selected variables that are not available in the selected table or PDB. Verify that the tables and variables for your report definition are available in the active PDB.
- If the report definition that you are running is a supplied report, make sure that you have fulfilled all setup requirements for the report definition.

To reset the datetime range for your report, follow these steps from the Manage Report Definitions window (see "Using the Manage Report Definitions Tool" on page 468 for more information about this window):

1. From the **Folders** list box, select the name of the folder that contains the report definition that you want to modify.
2. From the **Reports** list box, select the name of the report definition that you want to modify.
3. Select the arrow beside the **Subset Data** text box. The Subset Data window opens.
4. Select the **Table Status** tab to review the date and time ranges for data at each reduction level.
5. Select the **Start/End Dates/Times** tab and modify the date/time ranges to reflect the correct range of data. You can also remove the date/time ranges so that your report includes the complete range of data at the specified level of the PDB.
6. Select OK to close the window and save the criteria.
7. Rerun the report definition.

If you are reporting from the client GUI and directing a report to the Web, the report might go to a SAS window instead of to the Web. If that happens, do not run the report remotely. Run the report locally by selecting **Run ▶ Local**

If you are running a tabular-style report definition in batch mode (by calling %CPTABRPT) and you are using the macro's BY= parameter and you are running on z/OS, then you might prefer to set the HTMLDIR= parameter to direct the reports to a directory in the z/OS UNIX File System area. (Typically, this directory is one of the directories that are created by calling the %CPHTREE macro.) For more information, see the BY= parameter for the topic %CPTABRPT in the SAS IT Resource Management macro reference documentation.

Similarly, if you are running a print-style report definition in batch mode (by calling %CPPRINT) and you are using the macro's PAGEBY= parameter and you are running on z/OS, then you might prefer to set the HTMLDIR= parameter to direct the report to a directory in the z/OS UNIX File System area. (Typically, this directory is one of the directories that are created by calling the %CPHTREE macro.) For more information, see the PAGEBY= parameter for the topic %CPPRINT in the SAS IT Resource Management macro reference documentation.

If you use the vertical bar chart style to generate reports that are based on very low data values, some of the resulting reports might display only a horizontal row of zeros across the center of the page and no vertical bars. This problem occurs because the data values are so low that they translate to a bar height of zero, so no bars are actually displayed; however, the values of the bars are still displayed. The data values for each bar in a vertical bar chart are ordinarily centered above the bars; however, if the numerals are wider than the width of the bars, the numerals are not displayed. If a data value is zero, it will always be displayed because the numeral zero is never wider than the width that is allowed for a vertical bar. Therefore, if each data value in a report is low enough to translate to a bar height of zero, then no bars are displayed, but zeros are displayed in each position where a bar would ordinarily appear. If this

problem occurs, select a horizontal bar chart style. The data values in horizontal bar charts are always displayed at the far right of the chart, no matter how high or low the values might be, so that all the reports that are generated have a similar appearance.

CHAPTER 25

Reporting: Working with Palette Definition Folders

Working with Palette Definition Folders

Overview of Palette Definition Folders

Palette definition folders contain palette definitions.

You have a list of known palette definition folders. Originally, your list contains the names of the palette definition folders that have the supplied palette definitions.

If you create one or more palette definition folders, their names are automatically added to your list. You can also add the names of palette definition folders that other people create.

Similarly, you can remove the names of palette definition folders from your list. (The names of the folders are deleted from your list, but the actual folders are not deleted.)

For more information about palette definition folders, see the following topics:

- “Adding a Palette Definition Folder” on page 518
- “Creating a Palette Definition Folder” on page 519
- “Deleting a Palette Definition Folder” on page 521
- “Removing a Palette Definition Folder” on page 521
- “Viewing the List of Palette Definitions in a Palette Definition Folder” on page 522
- “Viewing Your List of Known Palette Definition Folders” on page 522.

Adding a Palette Definition Folder

You can add the name of an existing palette definition folder (for example, a palette definition folder that someone else created) to your list of known palette definition folders

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Adding a Palette Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can add the name of an existing palette definition folder to your list of known palette definition folders by following these steps:

1. If you want to add the name of an existing palette definition folder that is on a remote server, first activate a PDB on that server.

   For more information about activating a PDB, see "Activating a PDB" on page 299.
2. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
3. Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens. The top of the window displays your list of known palette definition folders.
4. Select **File**. A menu opens. Select **Add folder**. The Add or Create New Folder window opens.
5. Select the arrow to the right of the **Libref** field. A menu opens. It contains a list of the libraries for which a libref has already been defined. Select the library that contains the palette definition folder whose name you want to add.

   If the library that you want to use is not on that list, select **Define Library**. The **Define a SAS Library** window opens.

   a. In the **Libref** field, type the libref that you want to use for the library. The libref should begin with an alphabetic character. The remaining characters can be alphabetic or numeric.
   b. In the **Path** field, type the full path and name of the library's directory (UNIX, Windows) or PDS (on z/OS).
   c. In the **Access** field, select the access mode that you want to use (assuming that you have permission from the owner or security system to use that mode).
   d. In the **Location** field, select **Local** or **Remote**. If you select **Remote**, the **Server ID** field opens. The value in that field defaults to the server ID of the server to which you most recently connected.
   e. Select **OK**. The value of the **Library** field is the libref that you defined.
6. In the **Catalog** field, type the name of the catalog that contains the folder.
7. Select **Add Folder to List**. You are returned to the Manage Report Definitions window. The folder whose name you added is now on your list of known palette definition folders.
8. Select **Close**. You are returned to the Manage Report Definitions window.
9. When you want to return to the main window of SAS IT Resource Management, in the Manage Report Definitions window select **Close**.

Note: 1. If you have a "write" lock on the library to which you added the folder, as soon as you finish using the folder for "write" purposes (for example, saving palette definitions to it), remember to free the library fully or partially so that other people and

tasks can use it. One way to free the library is to exit from SAS IT Resource Management and SAS. Another ways is to follow these steps:

- From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
- Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens.
- Select **File**. A menu opens. From the menu, select **Add Folder**. The Add or Create New Folder window opens.
- Select **Define Library**. The **Define a SAS Library** window opens.
  1. In the **Libref** field, type the libref of the library to which you added the folder.
  2. If the libref was one that you created, leave the **Path** field blank (if you do not intend to use that library again during your current SAS session). That will entirely free the library.

     If the libref was one that you did not create or one that you did create but now can use in "read" mode, in the **Access** field select **Readonly**. That will partially free the library: people will be able to use it for read access, but not for write access.
  3. Select **OK**.

     Note: If you set a path to blank for a libref that you defined, the libref will be de-assigned and (in the SAS message area at the bottom of the window) you will get a confirmation message that says "not assigned." (This is a NOTE level message. Ignore the word ERROR.) △
- Select **OK**.

△

Note: 2. If the library to which you added the folder was remote, in addition to freeing that library (see Note 1), you also need to release the lock on the PDB so that other people and tasks (such as the process task and reduce task) can use it. For more information about releasing the lock, see "Releasing the Lock on the Active PDB" on page 319. △

Creating a Palette Definition Folder

You can create a new palette definition folder and automatically add its name to your list of known palette definition folder

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Creating a Palette Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create a new palette definition folder and automatically add its name to your list of known palette definition folders by following these steps:

1. If you want to create the palette definition folder on a remote server, first activate a PDB on that server. For more information about activating a PDB, see "Activating a PDB" on page 299.
2. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
3. Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens. The top of the window displays your list of known palette definition folders.

4 Select **File**. A menu opens. Select **Add folder**. The Add or Create New Folder window opens.

5 Select the arrow to the right of the **Libref** field. A menu opens. It contains a list of the libraries for which a libref has already been defined. Select the library where you want to place the new palette definition folder.

   If the library that you want to use is not on that list, select **Define Library**. The **Define a SAS Library** window opens.

   a In the **Libref** field, type the libref that you want to use for the library. The libref should begin with an alphabetic character. The remaining characters can be alphabetic or numeric.
   b In the **Path** field, type the full path and name of the library's directory (UNIX, Windows) or PDS (on z/OS). On UNIX or Windows, the library will be created if it does not exist; on z/OS, the library must already exist.
   c In the **Access** field, select **Write** if it is not already selected.

      Note: You must have prior permission from the owner of this library to use "write" access. △

   d In the **Location** field, select **Local** or **Remote**. If you select **Remote**, the **Server ID** field opens. The value in that field defaults to the server ID of the server to which you most recently connected.
   e Select **OK**. The value of the **Library** field is the libref that you defined.

6 In the **Catalog** field, type the name of the folder.

7 Select **Add Folder to List**. You are returned to the Manage Report Definitions window. The folder that you created is now on your list of known palette definition folders.

8 Select **Close**. You are returned to the Manage Report Definitions window.

9 When you want to return to the main window of SAS IT Resource Management, in the Manage Report Definitions window select **Close**.

Note: 1. If you have a "write" lock on the library to which you added the folder, as soon as you finish using the folder for "write" purposes (for example, saving palette definitions to it), remember to free the library fully or partially so that other people and tasks can use it. One way to free the library is to exit from SAS IT Resource Management and SAS. Another ways is to follow these steps:

- From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
- Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens.
- Select **File**. A menu opens. From the menu, select **Add Folder**. The Add or Create New Folder window opens.
- Select **Define Library**. The **Define a SAS Library** window opens.

  1 In the **Libref** field, type the libref of the library to which you added the folder.
  2 If the libref was one that you created, leave the **Path** field blank (if you do not intend to use that library again during your current SAS session). That will entirely free the library.

     If the libref was one that you did not create or one that you did create but now can use in "read" mode, in the **Access** field select **Readonly**. That will partially free the library: people will be able to use it for read access, but not for write access.
  3 Select **OK**.

     Note: If you set a path to blank for a libref that you defined, the libref will be de-assigned and (in the SAS message area at the bottom of the window)

you will get a confirmation message that says "not assigned." (This is a NOTE level message. Ignore the word ERROR.) △

□ Select **OK**.

△

Note: 2. If the library to which you added the folder was remote, in addition to freeing that library (see Note 1), you also need to release the lock on the PDB so that other people and tasks (such as the process task and reduce task) can use it. For more information about releasing the lock, see "Releasing the Lock on the Active PDB" on page 319. △

Deleting a Palette Definition Folder

You can delete a palette definition folder (which also automatically removes its name from your list of known palette definition folders)

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: If you do not want to delete the folder but only remove it from your list, see "Removing a Palette Definition Folder" on page 521. △

Deleting a Palette Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete a palette definition folder (and automatically remove its name from your list of known palette definition folders) by following these steps:

1. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
2. Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens and displays, at the top, the list of the names of the known palette definition folders.
3. Select the name of the palette definition folder that you want to delete.
4. Select **File**. A menu opens. Select **Delete Folder**.
5. The Palette Management window redisplays the list of known palette definition folders. The one that you selected is no longer on the list (and also no longer exists).
6. Select **Close**. You return to the Manage Report Definitions window.
7. When you are finished using the Manage Report Definitions window, select **Close** to return to the main window.

Removing a Palette Definition Folder

You can remove the name of a palette definition folder from your list of known palette definition folders

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Removing the name of the folder from the list does not delete the folder. The folder continues to exist. If you want both to remove the name and delete the folder, see "Deleting a Palette Definition Folder" on page 521. △

Removing a Palette Definition Folders (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can remove the name of a palette definition folder from your list of known palette definition folders by following these steps:

1 From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
2 Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens and, at the top, displays the list of known palette definition folders.
3 Select the name of the palette definition folder that you want to remove from the list.
4 Select **File**. A menu opens. From the menu, select **Remove Folder**.
5 The Palette Management window redisplays the list of known palette definition folders. The one that you selected is no longer on the list.
6 Select **Close**. You return to the Manage Report Definitions window.
7 When you are finished using the Manage Report Definitions window, select **Close** to return to the main window.

Viewing the List of Palette Definitions in a Palette Definition Folder

You can view a report definition folder

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing the List of Palette Definitions in a Palette Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view the list of report definitions in a report definition folder by following these steps:

1 From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
2 Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens and, at the top, displays a list of the known palette definition folders.
3 If the folder whose contents you want to see is not on the list, add it to the list. For more information, see "Adding a Palette Definition Folder" on page 518.
4 From the list of the known palette definition folders, select the folder whose contents you want to see. The list of palette definitions in that folder is displayed in the lower half of the window.
5 If you want to work with a palette definition, see "Overview of Palette Definitions" on page 525.
6 When you want to return to the Manage Report Definitions window, select **Close**.
7 When you want to return to the main window of SAS IT Resource Management, select **Close** in the Manage Report Definitions window.

Viewing Your List of Known Palette Definition Folders

You can view your list of known palette definition folders

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Viewing Your List of Known Palette Definition Folders (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view your list of known report definition folders by following these steps:

1. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
2. Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens and, at the top, displays your list of known palette definition folders.
3. When you want to return to the Manage Report Definitions window, select **Close**.
4. When you want to return to the main window of SAS IT Resource Management, select **Close** in the Manage Report Definitions window.

CHAPTER

26

Reporting: Working with Palette Definitions

Working with Palette Definitions: Main Topics

Overview of Palette Definitions

A palette definition contains the specification for a palette. A palette definition enables you to specify and save a group of graphic options and SAS options, and then to use those options for a report by associating the palette definition with the report definition.

For example, you might want to create a palette definition

- to set display options that you want to use whenever you run reports on a specific platform
- to set specific options for a report type such as reports that you want to display on the Web
- to set default options that you frequently use for your reports.

By using a palette definition, you can set options such as symbols, colors, legends, patterns, axes options, fonts, line width, and so on.

For more information about palette definitions, see the following topics:

- "Associating a Palette Definition with a Report Definition" on page 526
- "Copying a Palette Definition" on page 526
- "Creating/Editing/Viewing a Palette Definition" on page 527

Associating a Palette Definition with a Report Definition

For any particular report definition, you can specify the palette definition that you want to be used with the report definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Any number of report definitions can be associated with a particular palette definition. △

Note: If you do not associate a particular palette definition with the report definition, the default palette definition will be used with the report definition. For more information about the default palette definition, see “Specifying/Editing/Viewing the Default Palette Definition” on page 529. △

Associating a Palette Definition with a Report Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can associate a palette definition with a report definition by following these steps:

1 Create, edit, or view the report definition.

   For more information, see “Creating/Editing/Viewing a Report Definition” on page 481.

2 Create, edit, or view the palette definition.

   For more information, see “Creating/Editing/Viewing a Palette Definition” on page 527.

3 Select **`Close`** to return to the Manage Report Definitions window.

4 You can confirm the palette definition that is associated with the report definition by selecting **Locals ▶ Preview Macro Source**

   The value of the PALETTE= parameter is the name of the palette definition that is associated with the report definition that displays in the Preview Macro Source window.

5 You can run the report definition now to see the effect of the palette definition. If you prefer, you can save the report definition with the associated palette definition.

   For more information, see “Running a Report Definition” on page 488 and “Saving a Report Definition” on page 489.

6 When you are ready to return to the main window, select **`Close`** in the Manage Report Definitions window.

Copying a Palette Definition

You can copy a palette definition to the same palette definition folder (if you use another name for the definition) or to another palette definition folder

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Palettes definitions can be created only by using Manage Report Definitions on the **`Reporting`** tab in the SAS IT Resource Management main window. Therefore, if you want to run report definitions (interactively or in batch mode) on a server host and

you want to use a palette definition that you created on a client host, you must copy the palette definition to a folder on the server. △

Copying a Palette Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can copy one or more palette definitions to the same folder or another folder by following these steps:

1 View the palette definition that you want to copy.

For more information, see "Creating/Editing/Viewing a Palette Definition" on page 527.

2 Use the **Save as** button to save the palette definition to another folder.

For more information, see "Saving a Palette Definition" on page 529.

Creating/Editing/Viewing a Palette Definition

You can create, edit, or view a palette definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Creating/Editing/Viewing a Palette Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create, edit, or view a palette definition by following these steps:

1 From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.

2 Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens.

3 If you want to create a new palette definition, select **File**. A menu opens. From the menu, select **New Palette**. The Graphics Palette Properties window opens.

If you want to edit or view an existing report definition, list the existing palette definitions and select the one that you want to view or modify. The Graphics Palette Properties window opens. For more information about listing palette definitions, see "Viewing the List of Palette Definitions in a Palette Definition Folder" on page 522.

4 To specify or edit the palette definition, click on each tab and provide or modify the values of the fields on the tabs. (Because so much information needs to be added if you are specifying a new palette definition, it might be easier to start a new palette definition by copying the palette definition that is the most similar to the new definition that you want to create. For more information about copying a palette definition, see "Copying a Palette Definition" on page 526.)

To view the palette definition, click on each tab and look at the values of the fields on the tabs.

For more information about the fields, select **Help** in the Graphics Palette Properties window.

5 When you finish creating or editing the palette definition, select **Save** (to write the palette definition to its previous location) or **Save as** (to write the palette definition to a new location).

When you finish viewing the palette definition, select **Cancel**.

6 When you finish working in the Palette Management window, select **Close**.

7 When you finish working in the Manage Report Definitions window, select **Close**.

Deleting a Palette Definition

You can delete a palette definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Deleting a Palette Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete a palette definition by following these steps:

1. From the main window's **Reporting** tab, select **Manage Report Definitions**. The Manage Report Definitions window opens.
2. Select **Locals**. A menu opens. From the menu, select **Select Palette**. The Palette Management window opens.
3. From the **Folders** list, select the folder that contains the palette definition that you want to delete.
4. From the **Palettes** list, select the palette definition that you want to delete.
5. Select **File**. A menu opens. From the menu, select **Delete Palette**. The palette that you selected is deleted and is removed from the **Palettes** list.
6. When you are ready to return to the Manage Report Definitions window, select **Close**.
7. When you are ready to return to the main window, select **Close** in the Manage Report Definitions window.

Exploring Supplied Palette Definitions

You can explore supplied palette definitions

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Exploring Supplied Palette Definitions (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can explore supplied palette definitions by following these steps:

1. View the list of palette definitions in the *PGMLIB.PALETTE* palette definition folder. For more information, see "Viewing the List of Palette Definitions in a Palette Definition Folder" on page 522.
2. For each palette definition that interests you, view the palette definition. For more information, see "Creating/Editing/Viewing a Palette Definition" on page 527.
3. If you want to see the effect of a particular palette definition, associate the palette definition with a report definition, and run the report definition. For more information, see "Associating a Palette Definition with a Report Definition" on page 526 and "Running a Report Definition" on page 488.

   Note: Before you try different palette definitions, submit

   ```
   options source;
   ```

   in the SAS Program Editor window in order to write the source code to the log as it is executed. The name of each palette that you apply will then be recorded in the log. You can refer to the log to find the name of any palette that you want to reapply. △

Saving a Palette Definition

You can save a palette definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Saving a Report Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can save a report definition by following these steps:

1 Create, edit, or view a palette definition.

For more information about creating, editing, or viewing a palette definition, see "Creating/Editing/Viewing a Palette Definition" on page 527.

2 In the Graphics Palette Properties window, select **Save** to write to the currently selected palette definition folder, using the palette definition name on the **General** tab.

Or select **Save as** to have a choice of palette definition folders and palette definition names.

3 When you want to return to the Manage Report Definitions window, select **Close**.

4 When you want to return to the main window, select **Close** in the Manage Report Definitions window.

Specifying/Editing/Viewing the Default Palette Definition

When you select a default palette, this palette will be used anytime you create and run a report definition without specifying a palette for that specific report definition. Unless you override the default for an individual report, the default palette is used, regardless of the platform on which you are running the reports or the type of report you are running. For example, if you select a default palette and you create a report that you want to display on the Web, your default palette is used unless you select the Web palette (PGMLIB.PALETTE.WEB) or another palette for that report.

You must set the default palette in the SAS IT Resource Management GUI, but it is used both in the GUI and in batch, unless you select another palette when you run the report definition.

If you do not specify a default palette and you create and run a report definition without selecting a palette for that report, the default palette for your current host is used. For example, on Windows, the PGMLIB.PALETTE.WIN palette is the default palette for that operating environment.

You can specify one palette definition to use as the default for any report that does not have an associated palette definition (or edit the specification so that there is no default palette definition)

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Specifying/Editing/Viewing the Default Palette Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can specify the palette definition that is used by default (or edit the specification so that there is no default palette definition) by following these steps:

1 View your list of known palette definition folders.

For more information, see "Viewing Your List of Known Palette Definition Folders" on page 522.

2 At the bottom of the Palette Management window, the value in the **Default Palette** field is the name of the palette definition that is used by default.

3 If you want to use a different palette definition as the default palette definition, select the folder that has the palette definition that you do want to use.

4 Select the palette definition that you want to use by default. Click the right mouse button. A menu opens. Select **Make Default**. At the bottom of the Palette Management window, the **Default Palette** field displays the name of the palette definition that you selected.

Note: If you do not want a default palette definition, you can click on the value in the **Default Palette** field and then delete the characters, so that the result is an empty field. △

5 When you finish with the Palette Management window, select **Close**. You are returned to the Manage Report Definitions window.

6 When you finish with the Manage Report Definitions window, select **Close**. You are returned to the main window of SAS IT Resource Management.

CHAPTER

27

Reporting: Working with Rule Definition Folders

Working with Rule Definition Folders: Main Topics

Overview of Rule Definition Folders

A rule is a specification for detecting exceptions. A rule definition includes information such as the table and level, the expression, the message, and other attributes. In addition, a rule definition can include subsetting information. It can also contain a recommendation, perhaps for follow-up reports. Rule definitions reside in rule definition folders.

Some exception rules are supplied with your SAS IT Resource Management software. These rules reside in a (read-only) folder called PGMLIB.CPXRULE. You can also create customized rules and add them to any location (that is, folder and library) to which you have write access (for example, SASUSER.CPXRULE).

Note: You can organize your rules into multiple rules folders, as needed. (You might want to separate your rules folders by data source. Alternatively, the rules that are specific to a PDB can reside in a folder called ADMIN.CPXRULE. Rules that apply to your site can reside in the SITELIB.CPXRULE definition folder.) △

When you run the rule, the exceptions that are detected are called the results. By default, the results are placed in SASUSER.XRESULTS. You can direct the results to another location, such as ADMIN.RESULTS, if you have write access to the library and if you specify a unique name for the results data set.

For more information about rule definition folders, see the following topics:

- “Adding/Creating a Rule Definition Folder” on page 532
- “Deleting a Rule Definition Folder” on page 532

Adding/Creating a Rule Definition Folder

You can create a new rule definition folder and add it to your list of rule definition folders

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Creating/Adding a New Rule Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create a new rule definition folder and add it to your list of known rule definition folders by following these steps:

1 From the main window's **Reporting** tab, select **Perform Exception Analysis**. A list of rule definition folders opens. (You can view the rules that are contained in a folder by clicking a rule definition folder.)

2 Right-click in the folder area to open the menu of actions that you can perform. Click **New folder**.

Alternatively, you can select **File ▶ Folders ▶ Create/Add**

3 When the Create or Add Folder window opens, enter the name of the library where you want the folder to reside.

Alternatively, you can click the right arrow to list the libraries that are available. The Select a Library window opens. Select the library that you want to use and then click **OK** to return to the Create or Add a Folder window.

If you want to use a library that is not listed, click the **Define Library** button. The Define a SAS Library window opens. Enter the path and select the access to and location of the library. Click **OK**.

4 Enter the name of the folder that you want to create. The name of the rule definition folder must be between one and eight characters and can contain numbers, letters, and underscores. (The name cannot begin with a number; it can begin with a letter or an underscore.)

Alternatively, you can click the right arrow to list the folders that are available. The Select a Library window opens. Select the folder you want to use and then click **OK** to return to the Create or Add a Folder window.

The new folder name must be unique. Click the right arrow to list the folders that are already available. Be sure that the name you have chosen is not on that list of available folders.

5 Enter the description.

6 Click **OK** to create the new folder.

7 Click **Close** to return to the main **Reporting** tab.

Deleting a Rule Definition Folder

You can delete a rule definition folder

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Deleting a Rule Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete a rule definition folder by following these steps:

1. From the main window's **Reporting** tab, select **Perform Exception Analysis**. A list of rule definition folders opens for you to view.
2. Click to select the folder that you want to delete. A list of the rules that reside in that folder opens.
3. Right-click on that folder. A menu opens that lists the actions that you can perform. Select **Delete folder**.

   Alternatively, you can select **File ▶ Folders ▶ Delete**
4. You are asked if you are sure that you want to delete the rule definition folder. Click **Yes** to delete.

   Note: When the rule definition folder is deleted, all the rules in that rule definition folder are deleted. △
5. Click **Close** to return to the main **Reporting** tab.

Evaluating (Running) a Rule Definition Folder

You can evaluate (run) a rule definition folder

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Evaluating (Running) a Rule Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can evaluate (run) a rule definition folder by following these steps:

1. From the main window's **Reporting** tab, select **Perform Exception Analysis**. A list of folders opens.
2. To specify the location of the results data set, select **Locals ▶ Options**

   The Options for Rule Evaluation window opens. From the **Output** tab of this window, you can specify the library name and the data set that you want to use for the results. Click **OK** to return to the Exception Reporting window.

   To specify the date and time period of the data that you want to evaluate, select **Locals ▶ Options**

   From the **Time** tab of this window, you can specify the date and time period that you want to include in the results. Click **OK** to return to the Exception Reporting window.
3. From Exception Reporting window, click the rule definition folder that you want to evaluate. Right-click on that folder. A menu opens that lists the actions that you can perform. Select **Evaluate folder**.

   Alternatively, you can highlight the folder that you want to evaluate and select **File ▶ Folders ▶ Evaluate**
4. For information about viewing the results of the evaluation, see "Overview of Rule Results" on page 547.

Viewing the List of Rule Definitions in a Rule Definition Folder

You can view the list of rule definitions in a rule definition folder

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Viewing the List of Rule Definitions in a Rule Definition Folder (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view the list of rule definitions in a rule definition folder by following these steps:

1. From the main window's **`Reporting`** tab, select **`Perform Exception Analysis`**. The Exception Reporting window opens and displays a list of folders. This list consists of the supplied rule definition folders as well as the rule definition folders that you have created and added.
2. Select the rule that you want to view. The rules that are contained in that folder are listed in the lower frame of the window.
3. Click **`Close`** to return to the main **`Reporting`** tab.

Viewing Your List of Known Rule Definition Folders

You can view your list of known rule definition folders

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Viewing Your List of Known Rule Definition Folders (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can view your list of known rule definition folders by following these steps:

1. From the main window's **`Reporting`** tab, select **`Perform Exception Analysis`**. The Exception Reporting window opens and, at the top, displays your list of known rule definition folders.
2. Click **`Close`** to return to the main **`Reporting`** tab.

CHAPTER 28

Reporting: Working with Rule Definitions

Working with Rule Definitions: Main Topics

Overview of Rule Definitions

A rule definition (also known as a rule) describes how to detect the presence of exceptional data. A rule definition includes information such as the table and level, the expression, the recommendation, and other attributes.

For more information about rule definitions, see the following topics:

- "Copying a Rule Definition" on page 545
- "Creating a Rule Definition" on page 536
- "Deleting a Rule Definition" on page 545
- "Evaluating (Running) a Rule Definition" on page 546
- "Opening a Rule Definition" on page 544
- "Viewing/Editing a Rule Definition" on page 547.

Creating a Rule Definition

You can create a rule definition

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Creating a Rule Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create a rule definition by following these steps:

1. From the main window's **Reporting** tab, select **Perform Exception Analysis**. The **Exception Reporting** window opens and displays a list of rule definition folders.
2. Select the folder where you want the new rule to be stored. (You must have write access to this folder.) Then, select **File ▶ Rules ▶ New**
3. The Properties for Exception Rule window opens. Enter the information that specifies the new rule on these six tabs:
   - **General** tab
     - **Name** field

       Enter the name of the rule. The name must be unique within its rule definition folder. The name can have as many as eight characters, must begin with an alphabetic character, and can contain letters, numbers, and underscores. The name is automatically uppercased.
     - **Is This Rule Active?** field

       When this box has an X in it, this rule is active. When this box is empty, this rule is ignored.
     - **Description** field

       Enter text that explains or describes the rule.
   - **Data** tab
     - **Level** field

       Select the arrow next to the field. This selection opens a menu that lists (from the active PDB) the librefs of the libraries that have the level of data that you want to use (these librefs are identical to the level names and to the library names: DETAIL, DAY, WEEK, MONTH, and YEAR) and also lists the item **Other**.

       If the data that you want to evaluate is in the detail, day, week, month, or year level of the active PDB, select the libref that corresponds to the level of data that you want to use.

       If the data that you want to evaluate is not in the detail, day, week, month, or year level of the active PDB, select **Other**. A window opens that lists librefs, and for each libref, lists the library that the libref currently points to. Select the libref that corresponds to the library whose data you want to use.
     - **Table** field

       In order to specify the table name or data set name that contains the data that you want to evaluate with this rule, select the arrow next to the field. This selection opens a list of names of the tables and/or data

sets that are in the library whose libref you selected. Select the name of the table or data set whose data you want to use.

- **`By`** field

  Select the arrow next to the field. A window opens. Each item in the window is a list of BY variables. Select the list that you want to use or select **`<none>`** if you do not want to use a list.

  - *With a list:* When the rule runs, each unique set of values in the list defines a separate group of data that is evaluated. For example, if you select *MACHINE RELDEVN* and MACHINE has 3 values and RELDEVN has 4 values, then 12 groups of data are evaluated. In that same example, the result is zero to 12 exceptions. An exception is "owned" by a group and indicates that one or more observations in the group satisfy the rule.

    Note: The BY groups that are available are subsets (up to but not including the DATETIME variable) of the BY list or CLASS list that is associated with the libref and table name (or libref and data set name) that you specified. For example, if DETAIL.DISKS has BY list of *MACHINE CTLR DISK DATETIME SHIFT*, choices of BY grouping for exception analysis are: MACHINE, MACHINE CTLR, MACHINE CTLR DISK, and none. △

  - *Without a list:* A BY group value of <*none*>, means that no subsetting will be done by BY group. (Other subsetting will be done if other subsetting conditions are specified.) All the data (the whole data) is in one group; thus, no more than one exception will be generated for the rule.

- **`Subsetting`** tab

  You can subset the data by date, by time of day, and/or by using a WHERE expression.

  - *Date*

    You can specify a range of dates by which you want to subset the data. You can also set the cutoff time for defining the value of LATEST.

    - **`Cutoff Point for LATEST`** field

      You can specify a cutoff time for LATEST. On the last date for which there is data, if data is available from a time later than the cutoff time, then the last date for which there is data is used as the value of LATEST. However, on the last date for which there is data, if data is not available from a time later than the cutoff time, then the date that is *one day before* the last date for which there is data is used as the value of LATEST.

      If *no* cutoff time is specified for LATEST (or if the cutoff time is 0:00), then the value of LATEST is the last date for which there is data.

      Note: The LTCUTOFF= value is used only to determine the date that is represented by LATEST. Thus, the cutoff time for LATEST has no effect on the begin time or end time. △

    - Begin date

      In the **`Date Begin`** field, the begin date is LATEST-0 by default. To use another begin date, change the number. For example, LATEST-1 starts one day before the date that is specified by the value of LATEST.

- End date

  By default, the end date is LATEST. If you want to change the default, you can change it after you finish creating and saving the rule. Then, from the Exception Reporting window, select **Locals ▶ Options ▶ Time ▶ Specified date**

  Enter an end date and select **OK**.

  Note: The change to the end date is a global change that affects all rules in all rule definition folders. △

- *Time of Day*

  You can specify a range of times that are used within the date range. The time range is specified by the times that you enter in the Time of Day **Begin** and **End** fields.

  The default is to evaluate data from the entire 24-hour period (from BEGIN=0:00 to END=24:00) on each date in the date range.

- *Where*

  If you specify a WHERE expression, the data that is evaluated must satisfy this expression.

  Enter the expression. You can construct the expression by typing it into the field that is provided. Omit the word WHERE, because it will be added automatically.

  You can also construct the expression (if the mouse pointer is positioned over the expression field) by right-clicking. Right-clicking opens a pop-up menu.

  The pop-up menu displays lists of variables, operators, and values that you can use to construct the expression for this rule definition. (Notice that you must select a variable from the list of variables before you select a value for that variable.) Follow these steps to construct your expression:

  a From the pop-up menu, select **Variable List**. The Select Table Variables window opens.
  b In the right column (Selected), select a variable that you *do not* want to use, and in the middle column select the left arrow. This arrow moves the variable to the "available" column.
     Repeat until the Selected column is empty.
  c In the left column (Available), select the first variable that you *do* want to use in the expression, and in the middle column select the right arrow. This arrow moves the variable to the Selected column.
  d Follow the same steps to add operators, values, and other variables to your expression.

  Note: The pop-up menu is available to you only if you have already specified (on the **Data** tab) a value in the **Level** field and a value in the **Table** field. (The GUI constructs the list of variables and the list of values by opening the table or data set and examining the data.) △

  Note: Do not use the ampersand (&) to mean AND in WHERE expressions. △

Note: If you subset the data in more than one way, the data that is evaluated must satisfy all of the subsetting criteria. For example, if you specify a date range and a time range and a WHERE expression, the data that is evaluated is in that date range *and* in that time range *and* satisfies that WHERE expression. △

- **`Expression`** tab

  The expression and condition are combined as follows: After the data is subset, then the unique BY group (or the data as a whole, if BY groups have not been specified) has an exception if the expression is true AND the related condition (see the **`Is True For`** field) is true.

  The top of this tab enables you to define the expression. The bottom of this tab enables you to define the related condition.

  - *Expression*

    Enter the expression. You can construct the expression by typing it into the field that is provided.

    You can also construct the expression (if the mouse pointer is positioned over the expression field) by right-clicking. Right-clicking opens a pop-up menu.

    The pop-up menu displays lists of variables, operators, and values that you can use to construct the expression for this rule definition. (Notice that you must select a variable from the list of variables before you select a value for that variable.) Follow these steps to construct your expression:

    a From the pop-up menu, select **`Variable List`**. The Select Table Variables window opens.

    b In the right column (Selected), select a variable that you *do not* want to use, and in the middle column select the left arrow. This arrow moves the variable to the "available" column.

    Repeat until the Selected column is empty.

    c In the left column (Available), select the first variable that you *do* want to use in the expression, and in the middle column select the right arrow. This arrow moves the variable to the "selected" column.

    d Follow the same steps to add operators, values, and other variables to your expression.

    Note: The pop-up menu is available to you only if you have already specified (on the **`Data`** tab) a value in the **`Level`** field and a value in the **`Table`** field. (The GUI constructs the list of variables and the list of values by opening the table or data set and examining the data.) △

  - **`Is True For`** field (Related Condition)

    This field enables you to select the related condition. Choose one of the following conditions, and then (in the case of all conditions except the first one) modify the operator and/or the value of N to tailor the condition to your needs. (If you do not change the operator and/or the value of N, the second, third, and fourth conditions are logically equivalent to the first condition.)

    - *Any Observation*: After the data is subset, if the expression is true for one or more observations in the unique BY group (or in the data as a whole if BY groups are not defined), then that unique BY group (or all the data) has an exception.
    - *Number of Observations*: After the data is subset, if the expression is true for greater than (or less than, equal to, greater than or equal to, less than or equal to, or not equal to) *N* observations in the unique BY group (or in the data as a whole if BY groups are not defined), then that unique BY group (or all the data) has an exception.

- *Percent of Observations*: After the data is subset, if the expression is true for greater than (or less than, equal to, greater than or equal to, less than or equal to, or not equal to) N percent of the observations in the unique BY group (or in the data as a whole if BY groups are not defined), then that unique BY group (or all the data) has an exception.
- *Observations in a Row*: After the data is subset, if the expression is true for greater than (or less than, equal to, greater than or equal to, less than or equal to, or not equal to) N observations in a row in the unique BY group (or in the data as a whole if BY groups are not defined), then that unique BY group (or all the data) has an exception.

- **`Messages`** tab

  If you want to provide an exception message, use the fields on this tab.

  - Select the **`Display Message When Rule is True`** box. When the box has an X in it, the **`Messages`** tab is active. When the box is empty, this tab is ignored.
  - Enter a brief message. For example, you might want to have a message like this one: "The xxx rule has encountered an exception; give this exception a yyy priority," where *xxx* is a description of the rule and *yyy* is *high*, *medium*, or *low*.
  - Location where the message is displayed:
    - When the rule definition runs interactively: If there are exceptions, then an exception report is produced. For each exception, there is one page in the report, and the message is displayed near the bottom of the page.
    - After the rule definition runs in batch mode: View the Exceptions Summary page on the Web. If the cell for that rule lists one or more exceptions, select the cell. A page is displayed and the message is displayed near the top of the page.

- **`Recommendation`** tab

  When a user encounters an exception, if you want to provide a recommendation and/or suggest follow-up reports, then use the fields on this tab.

  - Select the **`Display Recommendation When Rule is True`** box. When the box has an X in it, the **`Recommendation`** tab is active. When the box is empty, this tab is ignored.
  - Enter the text of the recommendation in the field. (The field is 10 lines long and 70 characters wide.) You might want to provide a detailed explanation of the exception and/or you might want to suggest an appropriate course of action to take if an exception is detected.
  - Location where the recommendation is displayed:
    - When the rule definition runs interactively, if there are exceptions, then an exception report is produced. For each exception, there is one page in the report, and the recommendation is displayed when you select the **`Recommendation`** button on that page.
    - After the rule definition runs in batch mode, view the Exceptions Summary page on the Web. If the cell for that rule lists one or more exceptions, select the cell. A page is displayed and the recommendation is displayed near the top of the page.

- A *follow-up report* is a report that the person who creates the rule recommends to a user who wants to understand more about an exception. For each exception, a default follow-up report is available automatically, unless you suppress it.

  Optionally, you can suggest that the user look at alternate or additional follow-up reports by entering a RUN_REPORT= command after the text of the recommendation. The main purpose of the RUN_REPORT= command is to specify these reports so that they are produced in batch mode. However, you can produce these reports interactively.

  For more information about the RUN_REPORT= command, see "Using the RUN_REPORT= Command (UNIX, Windows)" on page 541.

4 Click **OK** to create the new rule and return to the Exception Reporting window. The new rule will be listed in **Exception Rules**, which is the lower frame of this window.

Using the RUN_REPORT= Command (UNIX, Windows)

The RUN_REPORT= command specifies how the follow-up reports are to be produced.

Issuing the Command:

Issue the RUN_REPORT= command after the rest of the recommendation. There can be only one RUN_REPORT= command in the recommendation. The command must begin on a new line. If it is too long for one line, it can wrap to one or more additional lines or it can break at one or more new-line characters or carriage-return characters. However, the command must not extend beyond the 10th line of the recommendation field. The command can be uppercase or lowercase. Do not insert a blank space before or after the = sign.

Syntax of the Command:

The RUN_REPORT= command has the following syntax (which is broken here for clarity, but can be on one line or on multiple lines):

```
RUN_REPORT=report-name--1 <... report-name-n>
 , FOLDER=libref.folder-name
 <, BEGIN=SAS-datetime-value>
 <, END=SAS-datetime-value>
```

where

- *report-name-1* through *report-name-n* are names of report definitions in the folder that is specified by the FOLDER= parameter. If more than one report name is specified, separate each report name from the next with one or more spaces. This parameter is required.

  The report definition(s) must be report definitions that are available in the Manage Report Definitions window. The specifications of the report definition(s) do not need to match the specifications of this rule. (For example, the rule definition and the report definition can specify different tables.) However, both should refer to data that is in the same PDB or the same *Other* library.

  You can use report definitions that already exist or report definitions that you create especially for use with this rule.

  For more information about Manage Report Definitions, see "Using the Manage Report Definitions Tool" on page 468. For more information about report definitions, see "Overview of Report Definitions" on page 480.

- *libref.folder-name* is the name of the report definition folder where the report definitions are stored. This parameter is required.

The folder must be a folder that is available in Manage Report Definitions. The libref of the folder does not need to be the same as the libref that is specified in the rule's definition.

For more information about report definition folders, see "Overview of Report Definition Folders" on page 471.

- The *BEGIN*= and *END*= parameters subset the data that is used when the follow-up reports are produced. Each of these parameters is optional. For BEGIN= and for END=:
  - if the value is provided on the rule's **Recommendation** tab, that value is used;
  - otherwise, if the value is provided in the report definition, that value is used;
  - otherwise, the default value is used. The default value for BEGIN= is EARLIEST. The default value for END= is LATEST.

  The value of the LTCUTOFF= parameter is used to choose which date is used as the value for LATEST when the follow-up reports are produced by %CPRUNRPT. For LTCUTOFF=:
  - if the value is provided (that is, is not a missing value) on the rule's **Subsetting** tab, that value is used;
  - otherwise, if the value is provided in the report definition, that value is used;
  - otherwise, the default value is used. The default value for LTCUTOFF= is 0:00.

  For more information about the BEGIN= parameter, see the Help in this rule definition or the BEGIN= parameter for the %CPRUNRPT macro in the SAS IT Resource Management macro reference documentation. For more information about the END= parameter, see the Help in this rule definition or the END= parameter for the %CPRUNRPT macro in the SAS IT Resource Management macro reference documentation. For more information about the LTCUTOFF= parameter, see the Help in this rule definition or the LTCUTOFF= parameter for the %CPRUNRPT macro in the SAS IT Resource Management macro reference documentation.

Viewing the specified follow-up reports:

- *When you run the rule in batch mode:* When the rule definition is evaluated (runs) in batch mode (by a call to the %CPEXCEPT macro and then a call to the %CPXHTML macro)
  1. %CPEXCEPT searches for exceptions and writes the exceptions to the RESULTS= folder.
  2. %CPXHTML reads the exceptions from the RESULTS= folder and, depending on the value of GENERATE= parameter that was specified in the call to %CPXHTML, produces *welcome.htm*, *excwelco.htm*, *allwelco.htm*, *srvwelco.htm*, and/or any follow-up reports or default follow-up reports.

     For the follow-up reports and the default follow-up reports, if GENERATE=FOLLOWUP (or GENERATE=ALL or if GENERATE= a list of values and one of the values is FOLLOWUP) and if the RUN_REPORT= command is specified, then the follow-up reports (or default follow-up reports) are automatically produced and directed to the Web.

     For each exception, if the rule specifies follow-up reports:

     a. %CPXHTML creates a temporary global macro variable that describes the exception's By group.
     b. To produce the follow-up reports, %CPXHTML calls %CPRUNRPT. In the call, the follow-up reports and their folders are specified, and OUTMODE=WEB is specified. For the following parameters, the

parameter values in the call to %CPXHTML are copied to the call to %CPRUNRPT: HTMLDIR=, HTMLURL=, IMAGEDIR=, IMAGEURL=, LARGEDEV=, OUTDESC=, OUTLOC=, PALETTE=, SMALLDEV=, and WEBSTYLE=.

c %CPRUNRPT uses the values of its parameters and the value in the temporary global macro variable to produce the exception's follow-up reports and direct them to the Web.

Note: When the global WHERE expression (from the temporary global macro variable) is joined with an AND to the local WHERE expression (from the report definition),

- the WHERE expressions must be compatible in order for a report to be generated. For example, no report is generated if one WHERE expression has MACHINE="Alpha" and the other WHERE expression has MACHINE="Beta". (Joining these two expressions with an AND does not result in the condition *True*.)
- the WHERE expressions must be compatible in order for the global WHERE expression to affect the report. For example, if the local WHERE expression specifies INSTEAD OF, the global WHERE expression is ignored. (In the client GUI, *INSTEAD OF* on the Query/Where Clause Builder tab in the report definition is equivalent to, in batch mode, omitting *SAME AND* in the value of the WHERE= parameter in the call to the report macro.)

△

For each exception, if the rule does not specify follow-up reports: The same steps take place as the ones that are described above for the specified follow-up reports, except that %CPXHTML calls %CPPLOT1 (not %CPRUNRPT) to produce the default (best-guess) follow-up report.

3 Use your Web browser to view the Exceptions Summary page on the Web. If the cell for that rule lists one or more exceptions, select the cell. A page is displayed. The RUN_REPORT= command is displayed in the middle of the page (in the **Actions** field). At the bottom of the page is a list of one or more follow-up reports. To view a follow-up report, select its **Report** cell.

For more information about the %CPEXCEPT macro, see the topic %CPEXCEPT in the SAS IT Resource Management macro reference documentation. For more information about the %CPXHTML macro, see the topic %CPXHTML in the SAS IT Resource Management macro reference documentation. For more information about the %CPRUNRPT macro, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation.

- *When you run the rule by using the GUI:* When the rule definition is evaluated (runs) interactively

1 If there are exceptions, then a set of pages is produced. For each exception, there is one page. On that page, the **BY Group** field displays the values that identify the BY group in which this exception was detected.

2 On that same page, select the **Recommendation** button. A window opens that displays the recommendation and the RUN_REPORT= command, if it was specified.

3 In the Manage Report Definitions tool, run the report definitions as specified in the RUN_REPORT= command.

If you want the data in the follow-up reports to be only from the BY group, you can specify a global WHERE that describes the BY group. For example,

suppose the **BY Group** field contains *MACHINE='ABC', NODE='DEF'*. In the SAS Program Editor window, type

```
CPWHERE= MACHINE='ABC' AND NODE='DEF' ;
```

and then select **Run ▶ Submit** All report definitions that you run until you submit another value for CPWHERE or exit from SAS will subset data as if

```
MACHINE='ABC' AND NODE='DEF'
```

were in the **Global Query/Where Clause for All Reports** field on the tab **Query/Where Clause Builder** in the window that opens when you select the arrow next to the **Subset Data** field in the report definition.

Note: When the global WHERE expression (from CPWHERE) is joined with an AND to the local WHERE expression (from the report definition),

- the WHERE expressions must be compatible in order for a report to be generated. For example, no report is generated if one WHERE expression has MACHINE="Alpha" and the other WHERE expression has MACHINE="Beta". (Joining these two expressions with an AND does not result in the condition *True*.)
- the WHERE expressions must be compatible in order for the global WHERE expression to affect the report. For example, if there is a local WHERE expression and it specifies *INSTEAD OF*, the global WHERE expression is ignored.

△

For additional information about the exception and its context, see "Overview of Rule Results" on page 547. For more information about the global macro variable *CPWHERE*, see the topic "Global and Local Macro Variables" in Chapter 1 of the SAS IT Resource Management macro reference documentation.

Viewing the default follow-up reports:
A default follow-up report is a best-guess line plot that is based on the first analysis variable in the expression.

1. *When you run the rule in batch mode:* A default follow-up report is produced for each exception. The data in each follow-up report is from the BY group that "owns" the exception.

   You can suppress the default follow-up reports by setting the DEFRPT= parameter to *NO* in the call to %CPXHTML or by issuing a RUN_REPORT= command in the rule definition.

2. *When you run the rule by using the GUI:* A default follow-up report is produced when you select the Graph button on an exception's page in the exception report. The data in the default follow-up report is from the BY group that "owns" the exception.

   A default follow-up report can be produced for each exception, without regard to whether you specified the RUN_REPORT= command in the rule definition.

Opening a Rule Definition

You can open and then edit a rule definition

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Opening a Rule Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can open a rule definition by following these steps:

1. From the main window's **Reporting** tab, select **Perform Exception Analysis**. A list of folders opens.
2. Select each folder to determine which one contains the rule that you want to open. (You must have write access to that folder in order to edit any rule within it.) Then, click to select that rule.
3. Select **File ▶ Rules ▶ Open** The Properties for Exception Rule opens, which displays the details about the rule.

   You can also display the details about a rule by right-clicking on the rule and selecting **Properties**, or alternatively, you can select the rule and then double-click on it.
4. At this point, you can edit the rule.

   Note: If you change the name of the rule (on the **General** tab), then you can create another copy of the rule. You can revise this new rule as needed. △

   Click **OK** to save the newly edited (or copied) rule.
5. Click **OK** to return to the previous window.

Deleting a Rule Definition

You can delete a rule definition

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Deleting a Rule Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete a rule definition by following these steps:

1. From the main window's **Reporting** tab, select **Perform Exception Analysis**. A list of folders opens.
2. Select the folder that contains the rule that you want to delete. (You must have write access to the folder in order to delete a rule.) A list of the rules that reside in that folder opens.
3. Right-click on the rule that you want to delete. A menu opens that lists the actions that you can perform. Select **Delete rule**.

   Alternatively, after highlighting the rule, you can select **File ▶ Rules ▶ Delete**
4. You are asked if you are sure that you want to delete the rule definition. Click **Yes** to delete.
5. Click **Close** to return to the main **Reporting** tab.

Copying a Rule Definition

Copying a Rule Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can copy a rule definition by following these steps:

1 From the main window's **Reporting** tab, select **Perform Exception Analysis**. A list of folders opens.
2 Select the folder that contains the rule that you want to copy. (You must have write access to the folder in order to copy a rule.) A list of the rules that reside in that folder opens.
3 Right-click on the rule that you want to copy. A menu opens that lists the actions that you can perform. Select **Copy rule**.

   Alternatively, after highlighting the rule, you can select **File ▶ Rules ▶ Copy**
4 A copy of the rule you selected is added to the list of rules in the folder. The name of the new rule is constructed from the first seven characters of the original name and appended to a unique number, for example the original "RULEABC" is copied to "RULEABC1."

   All the components of the original rule appear in the newly copied rule. To edit or change these components, double-click the rule. This opens the Properties for Exception Rule window, which displays the details about the rule. Click **OK** to accept the changes and return to the list of exception rules.
5 Click **Close** to return to the main **Reporting** tab.

You can also copy a rule by another method. For further details about this, see "Opening a Rule Definition" on page 544.

Evaluating (Running) a Rule Definition

You can evaluate (run) a rule definition

- by using the %CPSTART macro (to start SAS IT Resource Management), followed by the %CPEXCEPT macro (to run the rule definition to generate results) and, optionally, the %CPXHTML macro (to Web-enable the results).

  For more information about the %CPSTART macro, see the %CPSTART topic in the SAS IT Resource Management macro reference documentation . For more information about the %CPEXCEPT macro, see the %CPEXCEPT topic in the SAS IT Resource Management macro reference documentation. For more information about the %CPXHTML macro, see the %CPXHTML topic in the SAS IT Resource Management macro reference documentation .
- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Evaluating (Running) a Rule Definition (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can evaluate (run) a rule definition by following these steps:

1 From the main window's **Reporting** tab, select **Perform Exception Analysis**. A list of folders opens.
2 To specify the location of the results data set, select **Locals ▶ Options** The Options for rule evaluation window opens. From the **Output** tab of this window, you can specify the Library name and the Data Set that you want to use for the results. Click **OK** to return to the Exception Reporting window.

   To specify the date and time period of the data that you want to evaluate, select **Locals ▶ Options** From the **Time** tab of this window, you can specify the date and time period that you want to include in the results. Click **OK** to return to the Exception Reporting window.

3 From the Exception Reporting window, select the folder that contains the rule that you want to evaluate. Double-click the rule to open the Properties for Exception Rule window.

   Then, click **Run** to evaluate the rule.

4 Alternatively, you can click to highlight the rule that you want to evaluate and then click the right mouse button. From the pop-up menu, click **Evaluate rule**.

5 The results of the rule evaluation will be displayed. If there are no exceptions detected, a message to that effect will be displayed. For information about viewing the results, see "Overview of Rule Results" on page 547.

Viewing/Editing a Rule Definition

You can view/edit a rule definition

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

Viewing/Editing a Rule Definition (UNIX, Windows)

For information about viewing and/or editing a rule definition, see "Opening a Rule Definition" on page 544.

Overview of Rule Results

The results of running a rule definition consist of zero, one, or more exceptions. An exception is an observation that satisfies the conditions that are described in a rule. The exceptions are written to the results data set. (The total number of exceptions indicates the number of groups for which the rule is true, not the number of individual observations for which the rule is true.)

For more information, see "Results of Rule Evaluations" on page 547.

Results of Rule Evaluations

Results are generated when a rule definition or a rule definition folder runs. Rule definitions must be defined through the SAS IT Resource Management GUI for UNIX and Windows. Then, you must generate rule exceptions (or results). You can evaluate the rule

- on a periodic (once each day, once each week, and so on) basis in batch mode. For information about this, refer to "Viewing Rule Exceptions (Using Macros in Batch)" on page 548.
- interactively through the SAS IT Resource Management GUI. For information about this method, refer to "Viewing Rule Exceptions (UNIX, Windows)" on page 548.

Generating Rule Exceptions (Using Macros in Batch)

You can also view an exception by following these steps

- Use the %CPSTART macro to start SAS IT Resource Management. For information about this macro, see the topic %CPSTART in the SAS IT Resource Management macro reference documentation.

- Run the %CPEXCEPT macro. (This runs the rule definition and generates exception results.) For information about this macro, see the topic %CPEXCEPT in the SAS IT Resource Management macro reference documentation.

Viewing Rule Exceptions (Using Macros in Batch)

To view rule exceptions, run the %CPXHTML macro (to Web-enable the results), and then point your Web browser to the Web-enabled results. For information about how to view the Web-enabled results with your Web browser, see the topic %CPXHTML in the SAS IT Resource Management macro reference documentation.

Generating Rule Exceptions (UNIX, Windows)

Before you can view the results, you must generate rule exceptions (or results)

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client and/or server license. △

To generate results from the evaluation of a single rule, see "Evaluating (Running) a Rule Definition" on page 546. To generate results from the evaluation of an entire rule definition folder, see "Evaluating (Running) a Rule Definition Folder" on page 533.

Viewing Rule Exceptions (UNIX, Windows)

After the rule (or rule definition folder) has been evaluated, the View Results of Exception Evaluation window opens. This window contains the details of the exception that was detected:

Exception: n of x — the numerical identifier of the exception. This is the *nth* exception of the total *x* number of exceptions.

Name: — the name of the rule that is responsible for the exception. The name identifies the rule within the rule definition folder. The name can be up to 8 bytes long and it must begin with an alphabetic character. It is automatically uppercased.

Level: — the level of the PDB in which the exception was detected.

Table: — the name of the table, data set, or view in the PDB in which the exception was detected.

Description: — the description of the rule. It can be up to 40 characters long.

BY Group: — the identifier of the group that is responsible for the exception. The BY group is a group of observations or rows that have the same value for a variable that is specified in a BY statement. (Click the right arrow to open the Display BY-Group for This Exception window.)

Actual Count: — the number of times the exception has been detected in this instance of the rule evaluation.

Note: The **Actual Count** field is present only if multiple exceptions are detected. △

Threshold: — the number of times the exceptional condition is detected before an exception is flagged.

Note: The **Threshold** field is present only if multiple exceptions are detected. △

Using Expression: a clause that determines an exceptional condition. It contains a logical or arithmetic operator.

You can take the following actions:

- Click **Recommendation** to view the suggested actions that can help you handle or minimize the effects of this exception. This can include running follow-up reports and can be delivered as output to the Web.
- Click **Graph** to view a graph of observations for the group responsible for the exception.
- Click **View Obs** to view the observations in the BY group that triggered the exception. To return to the View Results window, close the inner Observations satisfying exception criteria window.
- Click **Goback** to return to the prior window, and ultimately to the Exception Reporting window.

CHAPTER

29

Reporting: Working with Galleries

Working with Galleries: Main Topics

Overview of Galleries

You can direct reports to a Web gallery interactively or in batch mode.

- For more information about directing reports interactively, see
  - "Creating/Editing/Viewing a Report Definition" on page 481
  - "Directing a Report to the Web" on page 491
  - the GUI information in the section "To Direct a Report to the Web" in the topic "How the OUT*= Parameters Work Together" in Chapter 3: Report Macros in the SAS IT Resource Management macro reference documentation.

- For more information about directing reports in batch mode, see
  - the report macros that are listed in the topic "Macros Used for Analyzing Data" in Chapter 3: Report Macros in the SAS IT Resource Management macro reference documentation
  - the batch mode information in the section "To Direct a Report to the Web" in the topic "How the OUT*= Parameters Work Together" in Chapter 3: Report Macros in the SAS IT Resource Management macro reference documentation
  - the topic "Examples of the OUT*= Parameters" in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.

For additional information about galleries, see the following list of topics, which are in alphabetical order:

Adding a Report Definition to the QuickStart Report Job

To add a new report to the QuickStart batch report job, perform the following steps.

1. Start SAS and the SAS IT Resource Management GUI, and activate the QuickStart PDB with write access.
2. From the **Reporting** tab in the SAS IT Resource Management main window, select the **Manage Report Definitions** task.

   In the **Folders** list box, select the report folder *ADMIN.ITSVRPT* if this is the folder in which to store the new report definition. You do not have to add your report definition to the ADMIN.ITSVRPT folder, but all of the other QuickStart report definitions are stored in that folder, so it might be helpful to put all of the report definitions for the QuickStart job in one location.

   The names of user-created report definitions should not begin with the letter Q, to avoid conflicts with the names of the report definitions that are supplied by running the QuickStart Wizard.
3. Create a report definition for the new report that you want to add to the QuickStart job. For assistance with completing the Manage Report Definitions window, select **Help** in that window.
4. When you have completed the modifications, you can save the report definition by selecting **File ▶ Save Report Definition**
5. Determine where you want to add the new report by finding out the name of an existing report and then adding your new report adjacent to that existing report. To determine the name of the existing report, open your Web browser and display the welcome.htm page that was created by the QuickStart job. On the left side of the welcome.htm page, there is a directory or tree structure of the reports that are

displayed on this welcome.htm page. On the right side of the welcome.htm page, there are small graphs or "thumbnails" of each report. Select the report that is adjacent to the location where you want to add the new report. The report name is displayed in the bottom-right corner of the browser window.

If the new report is an exception report, and you want information about how to add or remove exception reports, see "Modifying Exception Reports in the QuickStart Report Job" on page 566.

6 Edit your QuickStart batch report job and find the reference to the existing report that will be adjacent to your new report. The report name will appear as the value of a parameter in a call to the %CPRUNRPT macro. If there are multiple references to the report in the batch job, read the comments and reports adjacent to each reference to determine which comments apply to the report name that you are trying to locate.

7 You can add the new report to an existing report group (gallery), to an existing report directory but in a separate group, or to a new directory. Copy the existing %CPRUNRPT macro call that refers to the existing report that will be adjacent to your new report, and insert this code in one of the following locations.

- To add a report to an existing report group:
  - **a** Insert the copy of the %CPRUNRPT macro call adjacent to the original %CPRUNRPT macro call and update the copy of the %CPRUNRPT macro call to reference only the new report name.
  - **b** Within the new %CPRUNRPT call, do not change the values of OUTLOC=, OUTMODE=, HTMLDIR=, WEBSTYLE=, and OUTDESC=, so the report output will be displayed in the same gallery as the existing report in the %CPRUNRPT call that you copied.
  - **c** Review the values of all other parameters in the new %CPRUNRPT macro call in order to ensure that they apply to your new report. (For more details about specific parameters, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation.)
- To add a report to an existing report directory but in a separate group:
  - **a** Insert the copy of the %CPRUNRPT macro call adjacent to the original %CPRUNRPT macro call, and update the copy of the %CPRUNRPT macro call to reference only the new report name.
  - **b** Within the new %CPRUNRPT call, do not change the values of OUTLOC=, OUTMODE=, HTMLDIR=, and WEBSTYLE=, so the report output will be displayed in the same directory as the existing report that you copied.
  - **c** Update the value of the OUTDESC= parameter with a value that is different from the value that is specified in the copied %CPRUNRPT macro call. This forces the new report into a group (gallery) of its own because one gallery is created for each value of the OUTDESC= parameter.
  - **d** Review the values of all other parameters in the new %CPRUNRPT macro call to make sure they apply to your new report. (For more information about the specific parameters of %CPRUNRPT, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation.)
- To add a report to a new report directory:
  - **a** Insert a copy of the call to the %CPRUNRPT macro adjacent to the original call to the %CPRUNRPT macro and update the copy to reference only the new report name.

b Within the copy, update the values of the OUTLOC= and HTMLDIR= parameters, specifying values that are different from any others that are used in the batch job. This forces the report output to be stored in its own separate directory.
c Add initialization statements as necessary before the copy. For example, to initialize the directory, include a call to the %CPWEBINI macro by using the values of HTMLDIR= and OUTLOC= (that you specified in the call to the %CPRUNRPT macro) for the values of DIR= and CAT= (in the call to %CPWEBINI).
d Within the copy, do not modify the values of the OUTMODE= and WEBSTYLE= parameters, so the new report style will match the existing reports.
e Review the values of all other parameters in the copy to make sure that they apply to your new report. (For more details about the specific parameters, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation.)
f Create the new directory that is referenced in HTMLDIR= above or, if you want to access the new directory via the report explorer tree, follow the instructions in "Updating the Report Explorer Tree" on page 570, to create the directory and point to the directory from the report tree.

Combining Report Explorer Trees

To combine one or more "explorer trees" produced by multiple QuickStart report jobs, copy the Web-based output from each QuickStart report job into one common location. You can then run the %CPHTREE macro with an empty input file to "link" the report trees together. You will need to copy the report output to a common location after each run of the report jobs. When you copy the report output, replace the old files with newer ones and rerun %CPHTREE to link the report trees together.

Here is an example:

```
* Link QuickStart report trees together after copying GIF and HTML files;
* This %CPCAT creates an empty file for input to %CPHTREE;

%CPCAT;  cards4;
;;;;
%CPCAT( cat=work.temp.temp.source );
%CPHTREE( SUPERLOC=/common/SUPERLOC,
          cat=work.temp.temp.source,
          ptitle2=Combined Reports );
```

Note: The four semicolons above (;;;;) MUST start in column one. △

Changing the Style in an Existing Gallery

To change the style in an existing gallery:

1 Convert the existing gallery.
2 Convert the daily batch job for reporting, and then run the job.
3 When you direct new reports to the gallery, use the new style.

Converting the Existing Gallery

To convert a gallery from one style to another, find or re-create the call to %CPHTREE that originally created your gallery and specify the new gallery style as

the value of the WEBSTYLE= parameter. Then rerun the call. (Rerunning the call will add the necessary files to your gallery and will not overwrite or initialize existing reports in the gallery.)

For example, to convert a gallery from the style named GALLERY2 to the style named DYNAMIC, submit a call to the %CPHTREE macro with the WEBSTYLE= parameter set to *DYNAMIC*. More specifically, convert the gallery by using one of the following methods:

- If you wrote your own catalog entry and call to the %CPHTREE macro to create the gallery:
  1. Locate the catalog entry and call.
  2. If your call did not specify the WEBSTYLE= parameter, add *WEBSTYLE=DYNAMIC* to the call. If your call did specify the WEBSTYLE= parameter, set the WEBSTYLE= parameter to *DYNAMIC*. For examples, see below.
  3. Resubmit the call.
- If you recently used the QuickStart Wizard to create your gallery:
  1. On UNIX or Windows, locate the *x*rptstr.sas file that the wizard created in the qs/cntl or qs\cntl directory. On z/OS, locate the *X*RPTSTR member that the wizard created in the QS.CNTL PDS. (*X* is a one-letter code that identifies the collector.)
  2. If your call did not specify the WEBSTYLE= parameter, add *WEBSTYLE=DYNAMIC* to the call. If your call did specify the WEBSTYLE= parameter, set the WEBSTYLE= parameter to *DYNAMIC*. For examples, see below.
  3. Resubmit the program in the file or member.
- If you do not have a copy of the %CPHTREE macro call or the xrptstr.sas file or XRPTSTR member that was used to create your gallery:
  1. Create a catalog entry and call to the %CPHTREE macro that correspond to your gallery. For more information about the catalog entry and the call to %CPHTREE, see "Grouping Reports in a Web Gallery" on page 46 and the topic "%CPHTREE" in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation .
  2. In your call to %CPHTREE, set the WEBSTYLE= parameter to *DYNAMIC*. For examples, see below.
  3. Resubmit the call to %CPHTREE.

Here are examples of calls to the %CPHTREE macro:

- On z/OS, the call to the %CPHTREE macro might look like this:

```
%CPHTREE(
   superloc=TMP.MYWEB,
   subloc=MYTREE,
   bigspace=%QUOTE((CYL,(1,1,10))),
   smlspace=%QUOTE((TRK,(1,1,10))),
   dsparms=%QUOTE(recfm=vb,lrecl=6156,blksize=6160,dsorg=po),
   cat=work.mytree.source,
   title1=Server Reports
   title2=z/OS SMF Data,
   webstyle=dynamic
   );
```

- On UNIX or Windows, the call to the %CPHTREE macro might look like this:

```
%CPHTREE(
   superloc=/tmp/myweb,
   subloc=mytree,
   cat=work.mytree.source,
   title1=Server Reports,
   title2=z/OS SMF Data,
   webstyle=dynamic
   );
```

Note: On Windows, use back slashes (\) instead of forward slashes (/). △

Note: For more information about submitting SAS code or a SAS program, see "Working with the Interface for Batch Mode" on page 18. △

Note: You can omit the WEBSTYLE= parameter and instead specify the global macro variable CPWSTYLE, as described below. However, in the call to the %CPHTREE macro, it is recommended that you explicitly specify the WEBSTYLE= parameter. △

Converting the Daily Batch Job for Reporting

1 Locate the daily batch job for reporting.
   - If you created your own daily batch job for reporting, the job is probably one of the jobs that is scheduled to run every night and might have a filename like *xreport.sas* or a member name like *xREPORT*, where *x* identifies the collector whose data is read to produce the reports.
   - If you used the QuickStart Wizard to create your PDB and gallery, you will find the batch job for reporting in the same directory as the *xrptstr.sas* file or *xRPTSTR* member that is described above. The batch job for reporting is named *xreport.sas* (on UNIX and Windows) or *xREPORT* (on z/OS), where *x* is a one-character abbreviation that identifies the collector.

   You might want to print the job for your convenience in carrying out the following steps.

2 In the daily batch job for reporting, locate the calls to the following reporting macros:
   - %CPCCHRT
   - %CPCHART
   - %CPG3D
   - %CPHTREE
   - %CPMANRPT
   - %CPPLOT1
   - %CPPLOT2
   - %CPPRINT
   - %CPRUNRPT
   - %CPSPEC
   - %CPSRCRPT
   - %CPTABRPT
   - %CPWEBINI
   - %CPXHTML.

3 Immediately after the call to %CPSTART (or at least before the program encounters calls to macros on the above list), add the following line of code:

```
%let CPWSTYLE = new_gallery_style ;
```

where *new_gallery_style* is the style that you want to convert to. The global macro variable named CPWSTYLE sets the default value for the WEBSTYLE= parameter. The setting is temporary; it lasts until the end of the job. The setting affects any call to a macro in the above list that does not explicitly specify the WEBSTYLE= parameter. For example, in the following code, the value of CPWSTYLE *does not affect* the first macro call to the %CPRUNRPT macro but *does affect* the second macro call to %CPRUNRPT.

```
%let CPWSTYLE=DYNAMIC ;
     .
     .
     .
%CPRUNRPT(myreport_1,
   folder=myfolder_1,
   htmldir=/tmp/path/and/directory/name/of/my/gallery,
   outmode=web,
   webstyle=gallery2
   );
     .
     .
     .
%CPRUNRPT(myreport_2,
   folder=myfolder_2,
   htmldir=/tmp/path/and/directory/name/of/my/gallery,
   outmode=web
   );
```

For more information about the global macro variable CPWSTYLE, see the topic "Global and Local Macro Variables" in "Chapter 1: Using SAS IT Resource Management Macros" in the SAS IT Resource Management macro reference documentation.

4 By adding the specification of CPWSTYLE in the previous step, you converted (to the new gallery style) all the macro calls that do not specify the WEBSTYLE= parameter. What remains is to convert all the macro calls that *do* specify the WEBSTYLE= parameter.

Locate all the macro calls that do specify the WEBSTYLE= parameter. For these macro calls you have two choices:

- remove the WEBSTYLE= specification, so that the specification of a value for the CPWSTYLE global macro variable *does* affect the macro call.

  This is the recommended choice. When all macro calls are affected by the value of CPWSTYLE, then a single change (to the value of CPWSTYLE) changes the reports to another gallery style.

- change the value in the WEBSTYLE= specification to the new gallery style.

  Here is an example of a macro call that directs a report to a *directory-based report* structure:

```
%CPRUNRPT(myreport,
   folder=myfolder,
   htmldir=/tmp/path/and/directory/name/of/my/gallery,
   outmode=web,
   webstyle=GALLERY2
   );
```

To convert this macro call from the gallery style named GALLERY2 to the Web style named DYNAMIC, change the call as follows:

```
%CPRUNRPT(myreport,
   folder=myfolder,
   htmldir=/tmp/path/and/directory/name/of/my/gallery,
   outmode=web,
   webstyle=DYNAMIC
   );
```

Here is an example of a macro call that directs a report to a *PDS-based report* structure:

```
%CPRUNRPT(MYREPORT,
   folder=MYFOLDER,
   htmldir=HLQ.AND.PDS.NAME.OF.MY.GALLERY,
   outmode=web,
   webstyle=gallery2
   );
```

To convert this macro call from the gallery style named GALLERY2 to the gallery style named DYNAMIC, change the call as follows:

```
%CPRUNRPT(MYREPORT,
   folder=MYFOLDER,
   htmldir=HLQ.AND.PDS.NAME.OF.MY.GALLERY,
   outmode=web,
   webstyle=dynamic
   );
```

5 Run the batch job for reporting.

Generating New Reports in the New Style

If you use batch mode to produce new reports in the gallery, use the WEBSTYLE= parameter as described above. That is, either explicitly specify the value of the WEBSTYLE= parameter to be the new gallery style or omit the specification of the WEBSTYLE= parameter so the macro call will be affected by the value of CPWSTYLE. Omitting the WEBSTYLE= parameter is recommended, so that you can easily make changes between gallery styles by changing the value of CPWSTYLE.

The style that is specified in the batch job overrides the style that is specified in the GUI if all three of these criteria are true:

- the batch job contains a call to the %CPRUNRPT macro
- the call refers to a report definition that you generated by using the GUI, and the report definition explicitly specifies a gallery style (as described in the previous paragraph)
- the call to the %CPRUNRPT macro explicitly specifies a gallery style, or the call to %CPRUNRPT does not specify a gallery style but CPWSTYLE specifies a default gallery style.

If you use the GUI to generate new reports to the gallery, specify the new style as the value of the **Web Style** field. For more information about the **Web Style** field, see “Directing a Report to the Web” on page 491.

Note: If you accidentally use a different gallery style when you direct a report to the gallery, the gallery’s style will temporarily change. However, the gallery will change back to the new style as soon as a report in the new style is directed to the gallery. △

Creating and Displaying Reports on the Web

Overview

SAS IT Resource Management enables you to create a report definition and specify an output format of "Web" in order to display the report output by using a Web browser such as Microsoft Internet Explorer or Netscape Navigator. You can create reports for the Web by

- selecting Web Output Mode from the Report Output Options window in the SAS IT Resource Management GUI for UNIX and Windows. This is further described in "Creating Web-Based Reports by Using the SAS IT Resource Management GUI" on page 560.
- selecting OUTMODE=WEB in your report macro. For more information about this parameter, see the OUTMODE= parameter (which is on most of the reporting macros) in "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.

When you run your report definition (either interactively or in batch), your Web reports are saved as images. The Web output format also generates text output files that utilize a markup language called HTML. The HTML files reference your report images, and when you display the HTML file by using your Web browser, your report images are displayed within that file.

When you select an output type of Web, you must set additional report options (either in batch or in the SAS IT Resource Management GUI for UNIX and Windows) that determine things such as where your HTML output and images are stored, the size of your images, and the style of your Web report. The options that you set are saved with the report definition.

To create reports that you can display on the Web, follow these steps:

1. Decide whether you will you create your HTML reports from the GUI or you will use batch report macros. You can also create the reports by using the GUI and then save the macro source to run in batch. (See "Creating Web-Based Reports by Using the SAS IT Resource Management GUI" on page 560 and "Creating Web-Based Reports by Using Report Macros" on page 560.)
2. Decide how you want to store and access your reports. Do you want to store all Web reports in one location? This decision determines how you will access the reports (see "Accessing Your Web Reports and Managing the Report Files" on page 561).
3. Either by using the SAS IT Resource Management GUI for UNIX and Windows or by using the report macros, create a report definition, select an output format of Web, and specify the appropriate output options for your Web report by completing the Output Options window (see "Directing a Report to the Web" on page 491). To access the window, follow this menu path: **Locals ▶ Output Options ▶ Set Attributes ▶** ***select the "Web" option***
4. Indicate whether or not to delete the contents of the currently specified SAS catalog and the Web output directory or add new reports to an existing Web page each time you run your report definitions (see "Accessing Your Web Reports and Managing the Report Files" on page 561).
5. If you are creating the Web reports on a z/OS server, you can transfer your reports to a UNIX or Windows host to display them through your Web browser (see "Transferring Web Files from z/OS to a UNIX or Windows Host" on page 561).

Note: When you create reports to be displayed through your Web browser you can use a palette that is specifically designed for Web reports (see "Overview of Palette Definition Folders" on page 517 and "Overview of Palette Definitions" on page 525). △

Note: In some cases, you can successfully view SAS IT Resource Management galleries by using either *file:* or a network address (that is, in the format *machinename**directory**welcome.htm* instead of *http:*). However, there are situations where this will not work. If you are viewing galleries that were created with SAS IT Resource Management 2.4 and earlier versions, and if you are viewing these galleries with Internet Explorer release 5.5 (and later releases),

- for non-interactive graphics, http: is recommended; using *file:* or \\ protocols will not work.
- for interactive graphics (either the Java applet or the Windows ActiveX drivers), you must access your SAS IT Resource Management Web reports by using a Web server.

△

Creating Web-Based Reports by Using the SAS IT Resource Management GUI

To create a new Web report from the SAS IT Resource Management GUI for UNIX and Windows, follow these steps:

1. Create a report definition for your report, as you would for any type of report, by providing the required information in the Manage Report Definitions window. For information about this topic, see "Using the Manage Report Definitions Tool" on page 468.
2. When you have created the report definition in that window, select the Web output option by following this menu path from the Manage Report Definitions window: **Locals ▶ Output Options**

   This opens the Report Output Options window.
3. Select **`WEB`** from the Report Output Options window, and then select **`Set Attributes`** to open the Generate Web Output window.
4. The options in the Generate Web Output window enable you to specify additional report options, such as where your report images are stored, where your HTML files are stored, the Web report style, and more. Provide the required information in the Generate Web Output window as described in "Directing a Report to the Web" on page 491.

Creating Web-Based Reports by Using Report Macros

You can specify OUTMODE=WEB on most batch report macros, to create reports that you can display by using a Web browser. The WEB option is valid on all platforms where the report macros are supported (UNIX, Windows, z/OS).

To create a Web report by using batch report macros, follow these steps:

1. Determine which type of report you want to create and then use the appropriate report macro to create that report.

   For a complete list of report macros, see "Chapter 3: Report Macros" in the SAS IT Resource Management macro reference documentation.
2. Create a report definition by including the macro in your program and then submitting or running the program in batch. When you include the macro, you must specify OUTMODE=WEB, and the HTMLDIR= parameter. You should also review the other Web parameters on your report macro, in order to set other Web report options, such as where your report images are stored, the Web report style, and more.
3. If you create the reports on a z/OS server and the report structure is not located in the z/OS UNIX File System area, you might want to transfer the files to UNIX or

Windows to view them. For assistance see "Transferring Web Files from z/OS to a UNIX or Windows Host" on page 561.

Accessing Your Web Reports and Managing the Report Files

When you create reports that you can display by using your Web browser, many HTML files are created for you. All of these files can be accessed through one main file, or "Web page." Your HTML files are created in the location that you specify for your HTML directory, and they contain links to your report images. To view your reports, use your Web browser to open the *welcome.htm* page in your HTML output directory.

As you add new reports to an existing HTML directory, the welcome.htm page is updated and the new reports are appended to that page. If you do not occasionally clean out this directory, the HTML directory and the associated SAS catalog can become quite large.

You can clean out the HTML and IMAGE directories and the SAS catalog by selecting the **`Clear Gallery`** option in the Web Output window in the GUI. Or, in batch, you can use the %CPWEBINI macro or use the WEBCLR= parameter on the %CPRUNRPT macro or %CPXHTML macro to clear all reports, or you can selectively delete reports by using the %CPMANRPT macro.

(For more information about the %CPWEBINI macro, see the topic %CPWEBINI in the SAS IT Resource Management macro reference documentation. For more information about the %CPRUNRPT macro, see the topic %CPRUNRPT in the SAS IT Resource Management macro reference documentation. For more information about the %CPXHTML macro, see the topic %CPXHTML in the SAS IT Resource Management macro reference documentation. And for more information about the %CPMANRPT macro, see the topic %CPMANRPT in the SAS IT Resource Management macro reference documentation.)

You can clear or selectively delete reports as often as you like, depending on your site's needs. If a report is in the catalog, the macros and parameter assume that the report is also in the directory, so it is helpful to ensure that the reports in the catalog and the reports in the directory correspond. Notice that when you delete reports from the directory, some static files remain. For more information about static files, see "Customizing a Report Gallery's Static Files" on page 562.

One advantage of storing all reports (or groups of related reports) in the same location or catalog is that you can access everything from one main Web page. You can group specific types of reports together by using a common report output description, and you can select different descriptions on the welcome.htm page to display different groups of reports.

However, if you want to delete one type of report more often than another type of report, or if you like the option of accessing different types of reports from different "home" or welcome.htm Web pages, then you might want to store them in separate HTML directories.

Transferring Web Files from z/OS to a UNIX or Windows Host

If you are creating Web-based report files (specifying OUTMODE=WEB) on z/OS and not using the z/OS UNIX File System area, and then you want to move the files to a directory-based file system (UNIX, Windows, or the z/OS UNIX File System) in order to display them in a Web browser, you can

- use the %CMFTPSND macro
- use the sample FTP scripts that are provided with SAS IT Resource Management.

The %CMFTPSND macro sets up control statements and invokes FTP, enabling you to transfer files from a PDS to a specific host and location. This method can be used for any SAS IT Resource Management reports, including those created with the QuickStart

Wizard and those created with SAS IT Resource Management report macros. For information about %CMFTPSND, see the topic %CMFTPSND in the SAS IT Resource Management macro reference doc.

The sample jobs that are provided with SAS IT Resource Management can also assist you in transferring your Web output files from z/OS PDSs (if you are not using the z/OS UNIX File System area for the report structure) to a directory-based file system (UNIX, Windows, or the z/OS UNIX File System). The CMFCPFTP job can be used with any SAS IT Resource Management report macros. However, if you use these scripts to transfer report files that are created by the QuickStart Wizard programs, the report files are transferred but the "tree" structure that is created by %CPHTREE is not.

The CMTCPFTP job is available in CPMISC on z/OS. This job refers to another member named TCPSEND. The TCPSEND member is automatically created when you create reports with OUTMODE=WEB on z/OS and do not use the z/OS UNIX File System area for the report structure.

The TCPSEND member contains commands to transfer your HTML output files to another location. The name of the target directory is created based on the last qualifier of the PDS names that are used in the HTMLDIR= and IMAGEDIR= parameters. For example, if *HTMLDIR=myuser.html.itrmweb* and *IMAGEDIR=myuser.gifs.itrmweb*, then the TCPSEND file will contain commands to send the files to the *itrmweb* directory. If the values were *myuser.itrmweb.html* and *myuser.itrmweb.gifs*, the files would be sent to the HTML and GIFS directories, respectively.

To use this sample CMTCPFTP job to transfer your own files, follow the instructions at the beginning of that member and modify the file appropriately for your site. When you run the CMFCPFTP job, your Web output files will be transferred to the appropriate directory on the target host. Before you transfer the files, be sure that those subdirectories exist on your target host.

Customizing a Report Gallery's Static Files

A report gallery's static files are the files that are not cleared when the reports in the gallery are cleared. You can customize the *help.htm*, *info.htm*, *instruct.htm*, and *notes.htm* files. And you can make and customize an alternate *welcome.htm* file.

Customizing Gallery Help Files

Using the QuickStart Wizard or SAS IT Resource Management Report macros, you can create reports that you can display by using a Web browser. Multiple reports are displayed in one "Web page," and this is referred to as a report gallery. If you have created a gallery of reports and want to provide help or information about the entire set of reports or for an individual gallery, you can do this by attaching a help file to the Web page.

To provide a link to a help file from your report gallery, create a file called *help.htm* in the directory that contains the gallery. In the *help.htm* file, use HTML tags to define a name anchor that matches the value of the OUTDESC= parameter that you specified when you create this gallery of reports. For example, if you create a gallery of reports, you might specify parameters like the following ones to indicate the location of your HTML output and the description of the reports in that location:

```
htmldir=/tmp/myreports
outdesc=Useful CPU Reports
```

In this example, the file */tmp/myreports/help.htm* might contain help on the gallery of reports in the */tmp/myreports* location. The name anchor for these reports could be included in *help.htm* as follows:

```
<A NAME="Useful CPE Reports"><HR></A>
<H2>Help on "Useful CPU Reports"</H2>
This gallery contains reports on CPU activity. These
reports are generated by running the job in file...
....
<A HREF="Javascript:history.go(-1)">Back to Report Gallery</A>
```

Other reports in the same directory might have used a different value for the OUTDESC= parameter. These reports will appear in different galleries. Help for these other galleries can be included in the same *help.htm* file. If you use name anchors to identify the help for each gallery, each gallery page can use a link to the same *help.htm* file, but each link would specify the name anchor for the help on that gallery. For example, you might decide to create another gallery of reports in the same directory with the following parameters:

```
HTMLDIR=/tmp/myreports
OUTDESC=Useful Disk Reports
```

The file */tmp/myreports/help.htm* might also contain explanatory notes on that gallery of reports in a different section of the html file. The name anchor in *help.htm* might look like this for that section:

```
<A NAME="Useful Disk Reports"><HR></A>
<H2>Help on "Useful Disk Reports"</H2>
This gallery contains reports on disk activity. These reports
are generated one time each night by job...
....
<A HREF="Javascript:history.go(-1)">Back to Report Gallery</A>
```

If you specify a name anchor in the *help.htm* file and link to it from your gallery Web page, the **Help** button in each of these galleries will automatically link to the *help.htm* file at the appropriate "name" anchor, which would be "Useful Disk Reports" in this example.

The *Javascript:history.go(-1)* HREF line in the file returns to the previous page, mimicking the browser's **Back** button.

You might have a set of reports that have the same value of OUTDESC but are very different reports. For example, an "Overview" set of reports might include summaries of CPU, Disk, and Network Activity. You can also have sections of your *help.htm* file that refer to a specific report. In this case, the name anchor in the notes file should match the name of the report as saved in your SAS IT Resource Management Report folder.

For example, if your report is saved in the folder ADMIN.ITSVRPT and called CPUOVER, the corresponding section in the help file might look like this:

```
<A NAME="CPUOVER"><HR></A>
<H2>Help on CPU Overview Report</H2>
This gallery contains reports on disk activity. These reports
are generated one time each night by job...
....
<A HREF="Javascript:history.go(-1)">Back to Report Gallery</A>
```

This section will be the target of the *Help* link when the large GIF file is displayed alone.

You can create the *help.htm* file by using your favorite file editor or utility or by using the SAS IT Resource Management %CPENTCPY macro. Examples of using %CPENTCPY appear in the SAS catalog entry PGMLIB.JSWIZCAT.JPSERVI.SOURCE.

Customizing Gallery Note Files

If you have created a gallery of reports by using SAS IT Resource Management report macros and would like to provide explanatory notes for an individual gallery, modify or create the *notes.htm* file in the directory containing your report gallery. Within that file, define a name anchor that matches the value of the OUTDESC= parameter for the gallery. For example, suppose that you use the following parameters to create a gallery of reports:

```
HTMLDIR=/tmp/myreports
OUTDESC=Useful CPU Reports
```

The file */tmp/myreports/notes.htm* might contain explanatory notes on that gallery of reports in a section of the file that looks like this:

```
<A NAME="Useful CPU Reports"><HR></A>
<H2>Help on "Useful CPU Reports"</H2>
This gallery contains reports on CPU activity. These reports
are useful to you because...
....
<A HREF="Javascript:history.go(-1)">Back to Report Gallery</A>
```

In this example, the name anchor defines the target location for the link on your report gallery Web page.

Other reports in the same directory, but with different values of the OUTDESC= parameter, will appear in different galleries. Notes for these other galleries are included in the same *notes.htm* file but following a different name anchor. For example, suppose that another gallery of reports is created in the same directory with the following parameters:

```
HTMLDIR=/tmp/myreports
OUTDESC=Useful Disk Reports
```

The file */tmp/myreports/notes.htm* might also contain explanatory notes on that gallery of reports in a section that is identified by a name anchor as shown in this example:

```
<A NAME="Useful Disk Reports"><HR></A>
<H2>Help on "Useful Disk Reports"</H2>
This gallery contains reports on disk activity. Disk reports
are useful to you because...
....
<A HREF="Javascript:history.go(-1)">Back to Report Gallery</A>
```

The **Notes** button in each of these galleries will automatically link to the *notes.htm* file at the appropriate name anchor or label. The code *Javascript:history.go(-1) HREF* acts like the browser's **Back** button.

You might have a set of reports that have the same value of OUTDESC but are very different reports. For example, an "Overview" set of reports might include summaries of CPU, Disk, and Network Activity. You can also have sections of your *notes.htm* file that refer to a specific report. In this case, the name anchor in the notes file should match the name of the report as saved in your SAS IT Resource Management Report Folder.

For example, if your report is saved in the folder ADMIN.ITSVRPT and called CPUOVER, the corresponding section in the notes file might look like this:

```
<A NAME="CPUOVER"><HR></A>
<H2>Help on CPU Overview Report</H2>
This report gives you an overview of CPU activity for
previous day...
....
```

```
<A HREF="Javascript:history.go(-1)">Back to Report Gallery</A>
```

This section will be the target of the Notes link when the large GIF file is displayed alone.

You can create the *notes.htm* file by using your favorite file creation utility or by using the SAS IT Resource Management %CPENTCPY macro. Examples of using %CPENTCPY are available in the SAS catalog entry PGMLIB.JSWIZCAT.JPSERVI.SOURCE.

Other Static Files

You can also customize report gallery *instruct.htm* files and *info.htm* files.

And you can make alternative *welcome.htm* files. For more information about alternative *welcome.htm* files, see "Making an Alternate Welcome" on page 565.

Making an Alternate Welcome

Each time that you run SAS IT Resource Management report definitions to generate Web reports to a specified directory, SAS IT Resource Management creates a new welcome.htm file in that directory. You cannot influence how that file is created, and you should not modify that file after it is created (because your modifications will be lost as soon as the file is re-created).

You can, however, create and use a similar file that displays a different initial page. (For example, you might want to change the frame layout to conform to your organization's standards.) As long as the frames that are defined in your file include frames called group, bymenu, valmenu, and gallery, they can use the navigational files that are created automatically in the specified directory by SAS IT Resource Management. Notice that as you add more frames on the page (for example, for a header or footer banner), there is less display space for the report gallery.

If you generate Web reports to more than one directory, remember to copy the alternate file to each of the directories. You need to copy to each directory only one time; SAS IT Resource Management does not delete the alternate file when it generates a new welcome file.

Then tell your users to point their browsers to your file in the specified directory, instead of to welcome.htm.

For example, suppose welcome.htm contains the following:

```
<HTML><HEAD><TITLE> SAS IT Resource Management</TITLE></HEAD>
<!-- Generated by  SAS IT Resource Management V2.1 9904  -->
<!-- On Tuesday, May 11, 1999 at  8:42  -->
<FRAMESET cols="225,*" frameborder="1">
  <FRAMESET rows="60%,*,*,*" frameborder="1">
    <FRAME SRC="info.htm" NAME="info" scrolling="no">
    <FRAME SRC="group.htm" NAME="group" scrolling="no">
    <FRAME SRC="empty.htm" NAME="bymenu" scrolling="no">
    <FRAME SRC="empty.htm" NAME="valmenu" scrolling="no">
  </FRAMESET>
  <FRAME SRC="empty.htm" NAME="gallery">
</FRAMESET>
<NOFRAMES>
  You must use a frame-capable browser.
</NOFRAMES>
</HTML>
```

In that case, you could make an alternate file like this one:

```
<HTML><HEAD><TITLE> SAS IT Resource Management</TITLE></HEAD>
<!-- Generated by  SAS IT Resource Management V2.2 9810  -->
<FRAMESET rows="10%,*,10%" frameborder="0">
   <FRAME SRC="samphead.htm" NAME=header scrolling="no">
   <FRAMESET cols="225,*" frameborder="0">
      <FRAMESET rows="60%,*,*,*" frameborder="0">
         <FRAME SRC="info.htm" NAME="info" scrolling="no">
         <FRAME SRC="group.htm" NAME="group" scrolling="no">
         <FRAME SRC="empty.htm" NAME="bymenu" scrolling="no">
         <FRAME SRC="empty.htm" NAME="valmenu" scrolling="no">
       </FRAMESET>
       <FRAME SRC="noreps.htm" NAME="gallery">
   </FRAMESET>
   <FRAME SRC=sampfoot.htm" NAME=footer scrolling="no">
</FRAMESET>
<NOFRAMES>
You must use a frame-capable browser.
</NOFRAMES>
</HTML>
```

Or you could make an alternate file like this one:

```
<HTML><HEAD><TITLE> SAS IT Resource Management</TITLE></HEAD>
<!-- Generated by  SAS IT Resource Management V2.2 9810  -->
<FRAMESET rows="10%,*,10%" frameborder="0">
   <FRAME SRC="samphead.htm" NAME=header scrolling="no">
   <FRAMESET cols="225,*" frameborder="0">
      <FRAME SRC="info.htm" NAME="info" scrolling="no">
      <FRAMESET rows="15%,*" frameborder="0">
          <FRAMESET cols="33%,33%,*" frameborder="0">
             <FRAME SRC="group.htm" NAME="group" scrolling="no">
             <FRAME SRC="empty.htm" NAME="bymenu" scrolling="no">
             <FRAME SRC="empty.htm" NAME="valmenu" scrolling="no">
          </FRAMESET>
          <FRAME SRC="noreps.htm" NAME="gallery">
      </FRAMESET>
   </FRAMESET>
   <FRAME SRC=sampfoot.htm" NAME=footer scrolling="no">
</FRAMESET
<NOFRAMES
You must use a frame-capable browser.
</NOFRAMES
</HTML
```

Modifying Exception Reports in the QuickStart Report Job

After you run the QuickStart Wizard, you might want to perform the following tasks to update the exception reports in the QuickStart report job:

- “Removing a Follow-up Report from a Rule in the QuickStart Report Job” on page 567
- “Removing a Rule from QuickStart Report Job” on page 567
- “Adding a Follow-up Report to a Rule in the QuickStart Report Job” on page 568.

For more information about the QuickStart Wizard, see “Using the QuickStart Wizard” on page 50.

Removing a Follow-up Report from a Rule in the QuickStart Report Job

After you run the QuickStart Wizard and create the batch report job, you might want to remove a follow-up report from the rule but leave the rule itself in the Exception Summary table. To do so, follow these steps:

Note: You must already have run the process job and the report batch job to create the gallery before you can complete the following steps. △

1. Open your Web browser and display the welcome.htm page that was created by the QuickStart job. On the left side of the welcome.htm page, there is a directory, or tree structure, of reports. Select a folder that contains the category of reports that you want to modify. Then select the **`Exception Summary`** folder under the **`Exceptions`** folder, and select **Exceptions ▶ Exception Summary** On the right side of the page, select the cell that contains the follow-up report that you want to remove. A new page opens. On the page, select **`Report`**. A new window opens. When you select an exception, the Exception Rule detail page for that exception is displayed. The names of selected follow-up reports are listed after the **`Actions`** label on the Exception Rule detail page. The follow-up report name is an eight-character name that follows the RUN_REPORT= string.

   Note: The welcome.htm file is located in the qs subdirectory that you chose when you ran the QuickStart. △

2. From the SAS IT Resource Management main window, select the **`Reporting`** tab, and then select the **`Perform Exception Analysis`** task.
3. From the list of exception folders, select the folder that contains the exception rule and report that you want to remove. The QuickStart default exception folder is ADMIN.CPXRULE.
4. From the list of rules, select the exception rule that you want to modify, and then select **File ▶ Rules ▶ Open**
5. On the **`Recommendations`** tab, find **`Run_Report`**. Within this code, delete the name of the report definition that you want to remove. If you do not want any reports to run with this exception, remove the **`Run_Report`** code.

Removing a Rule from QuickStart Report Job

To remove an exception rule from the list of default exception rules in the QuickStart Wizard, follow these steps:

Note: You must already have run the process job and the report batch job to create the gallery before you can complete the following steps. △

1. Determine the name of the rule that you want to remove from the Exception Rule report. To determine the rule name, open your Web browser and display the welcome.htm page that was created by the QuickStart job and that contains the report that you want to modify. On the left side of the welcome.htm page, there is a directory, or tree structure, of reports. Select the **`Exception Summary`** folder under the **`Exceptions`** folder. On the right side of the page, select the exceptions that you want to remove. When you select an exception, the Exception Rule detail page for that exception is displayed. The name of the rule is listed after the **`Rule Name`** label on the Exception Rule detail page. The exception rule name is an eight-character name that begins with the letter Q (for example, QPCPUUS).

   Note: The welcome.htm file is located in the qs subdirectory that you chose when you ran the QuickStart. △

2. From the SAS IT Resource Management main window, select the **`Reporting`** tab, and then select the **`Perform Exception Analysis`** task.

3 From the list of exception folders, select the folder that contains the exception rule that you want to remove. The default QuickStart exception folder is ADMIN.CPXRULE.

4 Select the exception rule that you want to delete, and then select **File ▶ Rules ▶ Delete** Select **Yes** to confirm that you want to delete the rule.

Adding a Follow-up Report to a Rule in the QuickStart Report Job

To add a report to an exception rule to the Quick Start batch job, follow these steps:

1 From the SAS IT Resource Management main window, select the **Reporting** tab, and then select the **Perform Exception Analysis** task.

2 From the list of exception folders, select the folder that contains the exception rule that you want to modify. The default QuickStart exception folder is ADMIN.CPXRULE.

3 Select the exception rule that is associated with the report that you are adding, and then select **File ▶ Rules ▶ Open**

4 Select the **Recommendations** tab. If the RUN_REPORT= command is not already there, add it on a new line after the rest of the recommendation. Follow the command by the name or names of the report definitions that you want to run to produce follow-up reports for exceptions. If you do not specify the location of the report definitions, the exception system searches for the report definitions in the *admin.itsvrpt* folder. To specify the folder, use the FOLDER= parameter.

   For example, you could type the following command to run the *QNCPUSPC* and *QNCPUQUE* report definitions that are in the default folder:

   ```
   RUN_REPORT=QNCPUSPC QNCPUQUE
   ```

   Or you could type the following command to run the *QNCPUSPC* and *QNCPUQUE* report definitions that are in a folder named *myrptlib.itsvrpt*:

   ```
   RUN_REPORT=QNCPUSPC, folder=MYRPTLIB.ITSVRPT
   ```

   For more information about the RUN_REPORT= command, see “Using the RUN_REPORT= command” in “Creating a Rule Definition” on page 536.

Removing a Report from the QuickStart Report Job

To remove a report name from the source statements of the QuickStart batch report job, perform the following steps.

1 Determine the name of report that you want to remove. To determine the report name, open your Web browser and display the welcome.htm page that was created by the QuickStart job and that contains the report that you want to remove. On the left side of the welcome.htm page, there is a directory or tree structure of the reports that are displayed on this welcome.htm page. On the right side of the welcome.htm page, there are small graphs or “thumbnails” of each report. Select the report that you want to remove. The report name is displayed in the bottom-right corner of the browser window.

   If the report is an exception report, see “Modifying Exception Reports in the QuickStart Report Job” on page 566.

2 Edit the QuickStart batch report job, and locate in the code the name of the report that you want to remove. The report name will appear as the value of a parameter in a call to the %CPRUNRPT macro. If there are multiple references to the report in the batch job, read the comments and reports adjacent to each reference to determine which comments apply to the report that you want to remove.

3 If the call to the %CPRUNRPT macro contains references to multiple reports, remove or comment out only the report that you want to remove. If the report to be removed is the only report that is referenced by the macro, you can remove the entire call to the macro (remove %CPRUNRPT, the open and close parentheses, and all the parameters between them up to and including the next semicolon).

4 If this report is the only report in a directory of reports, you might also want to update the report explorer tree so it will not point to the directory. You might also want to remove from the batch report job any initialization statements specific to that directory of reports. To remove the initialization statements for a specific directory, read the comments in the report batch job that indicate which statements are used for initialization of just the single directory (typically, a %CPWEBINI macro plus, optionally, some subsetting code). Remove the initialization statements as indicated in these comments. To remove a reference to a directory from the report explorer tree, see "Updating the Report Explorer Tree" on page 570.

Updating an Existing QuickStart Report

To modify one of the reports that are produced by the QuickStart batch report job, perform the following steps.

1 Determine the name of report that you want to modify. To determine the report name, open your Web browser and display the welcome.htm page that was created by the QuickStart job and that contains the report that you want to modify. On the left side of the welcome.htm page, there is a directory or tree structure of the reports that are displayed on this welcome.htm page. On the right side of the welcome.htm page, there are small graphs or "thumbnails" of each report. Select the report that you want to modify. The report name is displayed in the bottom-right corner of the browser window.

2 Start SAS and the SAS IT Resource Management GUI for UNIX and Windows, and activate the QuickStart PDB with write access.

3 From the **`Reporting`** tab in the SAS IT Resource Management main window, select the **`Manage Report Definitions`** task.

In the **`Folders`** list box, verify that the report folder *ADMIN.ITSVRPT* is highlighted.

4 From the **`Current PDB Reports`** list box, select the report you want to modify. For assistance with completing that window and modifying the report definition, select **`Help`** in the Manage Report Definitions window. You can test the report changes interactively by selecting **`Run`**; however, within the GUI you might receive a larger quantity of output than you receive when you run the batch report job. This is because the QuickStart batch job subsets the data prior to running many of its reports. If you run the report within the GUI, you might want to subset the data (select **`Subset Data`** from the **`Locals`** menu) to limit the scope of the report to just a small subset of the data while you test. Be sure to remove this limitation before you save the report.

5 When you have completed the modifications, you can save the report definition by selecting **File ▶ Save Report Definition**

If you save the report in the ADMIN.ITSVRPT folder and you use the existing report name, the batch job will automatically use the modified report the next time the report batch job runs. If you save the updated report definition into a different folder or name, you will need to modify the report name in the source statements within the batch report job. The original report is also available in PGMLIB.ITSVRPT if you want to refer to it later.

Updating the Report Explorer Tree

Your Web gallery is based on a hierarchical list of all report directories. This hierarchical list is called the “report structure,” or the “report explorer tree," or just "the tree.” The tree was created when SAS IT Resource Management and your collector were set up to work together at your site.

In the instructions that follow, it is assumed that you have the necessary permissions. The SAS IT Resource Management administrator at your site has the permissions that are necessary to update the tree. If you are not the administrator, you can ask the administrator to update the tree for you.

Here are the steps to update the tree:

1. Find the setup case that applies to your collector. For the mapping of collectors to setup cases, see “Setting Up the Server, with Setup Cases” on page 38.
2. After you know which setup case applies to your collector, find that setup case: “Overview of Setup Case 1” on page 53, “Overview of Setup Case 2” on page 65, “Overview of Setup Case 3” on page 81, “Overview of Setup Case 4” on page 99, or “Overview of Setup Case 5” on page 119.
3. In that setup case, find the topic named “The Report Structure.”
4. In that topic, find the name of the program that calls the %CPHTREE macro. (The report explorer tree is created by a program that calls the %CPHTREE macro.)

   For more information about the %CPHTREE macro, see the topic %CPHTREE in the SAS IT Resource Management macro reference documentation.
5. Find that program on your system. The program is likely to be in the same location as the programs in the SAS IT Resource Management daily jobs. If the program is not there, look for a file (on UNIX or Windows) or a file or PDS member (on z/OS) that contains the word *%CPHTREE*.
6. In the setup case that applies to your collector, find the topic “Optional Customization.”
7. In that topic, find the instructions for customizing the report structure. Any steps that are required are explained either in the topic “Optional Customizations” or in topics, such as “The Report Structure,” that appear earlier in that setup case.
8. Save a copy of the program that calls the %CPHTREE macro and/or any file or catalog entry that the program refers to, in case you want to refer to the original versions later.
9. As described in the instructions for customizing the report structure, edit the report structure program and/or edit the catalog entry that is specified by the CAT= parameter in the call to the %CPHTREE macro in the report structure program. Then remove the original tree or not (after consideration of the issues that are mentioned in those instructions), and run the edited version of the report structure program. That run creates the revised tree.
10. In the daily job that produces reports, find the name and location of the program that produces the reports.
11. Save a copy of that program under a different name, in case you want to refer to the original version later.
12. Edit the program that produces the reports so that it directs reports to locations in the revised tree. For more information about the program, see the topic “The Report Program(s)” in the setup case that applies to your collector.

CHAPTER

30

Reporting: Working with Remote Server Profiles

Working with Remote Server Profiles: Main Topics

Overview of Remote Server Profiles

A remote server profile provides the specifications for connecting to a remote host and for accessing SAS IT Resource Management components on that host.

Your need for a connection to a remote server depends on the SAS IT Resource Management software on your host and on the location of the PDB that you want to activate.

- If the SAS IT Resource Management software on your host has *only a client license*, the PDBs that you want to activate are typically on one or more servers (hosts on which the SAS IT Resource Management software has a server license) that are remote (see the notes below). If a PDB that you want to activate is not local (see the notes below), you must connect to the host on which a remote PDB is located in order to use that remote PDB.
- If the SAS IT Resource Management software on your host has *only a server license*, the PDBs that you can activate are local (see the notes below), and you have no need (or ability) to connect to a remote host.
- If the SAS IT Resource Management software on your host has *both a client license and a server license*, typically the PDBs that you want to activate are local (see the notes below). If a PDB that you want to activate is not local, you must connect to the host on which the remote PDB is located in order to use that remote PDB.
-

  Note: A *local PDB* is a PDB that you access from one or more disk drives that are attached to the local host or from one or more disk drives whose contents are accessible through a service such as Network Neighborhood, NFS, AFS, or DFS.

Thus, if a PDB satisfies the definition of a local PDB, you can access the PDB without connecting to a remote server. △

Note: A *remote PDB* is a PDB that you access by connecting to a remote host that has the PDB available as a local PDB. Thus, if a PDB does not satisfy the definition of a local PDB, you must connect to a remote server if you want to access the PDB. △

The typical way to manage your connection to a remote server is to

- connect at the same time that you activate the remote PDB.

  When you activate a remote PDB, SAS IT Resource Management automatically asks for your user ID and password on the remote server and uses them (along with the remote server profile that is associated with that PDB) to connect you to the remote server.

  For more information about activating a remote PDB while automatically connecting to the remote server, see "Activating a PDB" on page 299.

  Note: An active PDB on a remote server is in read-only mode. While the PDB is activated in read-only mode, other work that requires write mode (such as the process task and reduce task) cannot take place. Thus, it is important to de-activate the remote PDB as soon as you finish using it. (If you de-activate a PDB by making a local *demonstration* PDB your active PDB, you are less likely to affect anyone else's work than if you activate a *production* PDB.) △

  Note: If a site library is specified in the profile that is associated with the remote PDB, that site library becomes the active site library. By default, an active site library on a remote server is in read-only mode. For information about switching it to write mode, see "Viewing/Editing Your Active Site Library's Access Mode" on page 281. △

  Note: If a site library is not specified in the profile that is associated with the remote PDB, the active site library and its access mode are unchanged. △

- work with the remote PDB.

  For example, run report definitions to generate reports on the data in that PDB.

- disconnect at the same time that you de-activate the remote PDB.

  De-activate the remote PDB by activating a local PDB. (If you do not have an immediate use for a local production PDB, you can activate a local demonstration PDB in order for your access mode to the local PDB to have the least effect on others' work.) At that time, SAS IT Resource Management automatically asks if you want to disconnect; *agree* to disconnect.

  For more information about activating a local PDB while agreeing to disconnect from the remote server, see "Activating a PDB" on page 299.

The specifications for a connection to a remote host (and for accessing SAS IT Resource Management components on that remote host) are provided by a remote server profile on the SAS IT Resource Management client software.

Note: On a host that has both the SAS IT Resource Management client license and server license, the functionality is additive. Thus, you can also use remote-server profiles on a machine with both licenses. △

Three sample profiles are shipped with SAS IT Resource Management: one for UNIX, one for z/OS (MVS), and one for SAS/SHARE. The sample profiles (and any other that you make) are stored in your SASUSER library. You cannot use a sample profile as is; you must customize it. In order to keep an unchanged, original copy of the sample profile, you might want to copy the sample profile and edit the copy, rather than edit the original.

Typically, you use one profile for each remote server, regardless of the number of PDBs that are local to a server.

For more information about remote server profiles, see the following topics:

- "Copying a Remote Server Profile" on page 573
- "Creating a Remote Server Profile" on page 574
- "Deleting a Remote Server Profile" on page 576
- "Viewing/Editing a Remote Server Profile" on page 577.

Note: To see more information about the fields in a remote profile, follow this path *for UNIX:*

From the main window, select **Administration ▶ Manage PDBs ▶ File ▶ Add PDB Entry**

Then click on the right arrow that is associated with the Remote Server Profile field. Click to highlight UNIX or MVS. Right-click to open the drop-down menu. Click **`Properties`** to display the Remote Server Profile Definition window. Click **`Help`** to view detailed information about the fields on the remote server profile. Click **`Close`** to return to the Remote Server Profile Definition window.

For Windows:

Because SAS IT Resource Management clients usually have network access to the server if the server is on Windows, the clients rarely need to use a remote profile to access a PDB on a Windows server.

△

Copying a Remote Server Profile

You can copy a remote server profile

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client license. △

Copying a Remote Server Profile (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can copy a remote server profile by following these steps:

1. From the main window's **`Administration`** tab, select **`Manage PDBs`**. The Manage PDBs window opens.
2. Select **`Locals`**. A menu opens. From the menu, select **`Remote Profile`**. The Remote Server Connection Profiles window opens.
3. Select the remote server profile that you want to copy.
4. Select **`File`**. A menu opens. From the menu, select **`Copy`**. The Remote Server Profile Definition window opens.

   All of the fields have the same values as the profile that you copied, except for the **`Profile Name`** field.
5. In the "to-profile," make any changes that you need for your purposes. For more information, select **`Help`** for that window or see "Creating a Remote Server Profile" on page 574.

   Note: SAS IT Resource Management generated a profile name that is unique within your set of remote server profiles. You can change the name of the profile, as long as the new name is also unique within your set of remote server profiles. The profile name can be as many as eight characters in length and it is required. △

For TCP/IP, the profile works faster if the profile name is the same name as in the **Server Name** field. For APPC, the profile name must be the same name as in the **Server Name** field. For NetBIOS that uses SAS/CONNECT, the profile name (and the server name) should be the same as the -NETNAME setting that was used in creating the spawner on the server.

Note: If you want to use a port number in the **Server Name** field (in order to make a remote connection by using the z/OS spawner), use two underscores preceding of the port number. Thus, if the port number is *nnnn*, the value in the **Server Name** field should be *__nnnn*. △

Note: The remote server profiles are in your SASUSER library. Thus, you can make changes to the remote server profiles without affecting other users. △

Note: The logon and logoff scripts are physical files and can be located anywhere. Thus, changing the script files can affect others, if others use the same scripts. △

6 Select **OK**. SAS IT Resource Management returns you to the Remote Server Connection Profiles window. The "to-profile" is now in the list of profiles.

7 Select **OK** to return to the Manage Report Definitions window.

8 Select **Close** to return to the main window.

Creating a Remote Server Profile

You can create a remote-server profile

□ by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client license. △

Note: It is faster to copy a profile that is similar to what you want to use and then edit it. But you can create an entirely new profile. This topic covers creating an entirely new profile. △

Creating a Remote Server Profile (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can create a remote-server profile by following these steps:

1 On the main window's **Administration** tab, select Manage PDBs. The Manage PDBs window opens.

2 Select **Locals**. A menu opens. From the menu, select **Remote Profile**. The Remote Server Connection Profiles window opens.

3 Select **File**. A menu opens. From the menu, select **New**. The Remote Server Profile Definition window opens.

4 On the **General** tab,

   □ in the **Profile Name** field, type a profile name. The profile name can be up to eight characters in length and it is required. The profile name must be unique within your set of remote server profiles.

   Note: The profile works faster if the profile name is the same name as in the **Server Name** field. △

   □ in the **Description** field, type a longer description of the profile. This field is optional.

5 Select the **Remote Server** tab.

   □ In the **Server Name** field, type the name of the server. This field is required. The value is an Internet address.

- In the **Server OS** field, select the operating system that is running on the server host: **MVS** (z/OS), **UNIX**, or **Windows**.
- In the **Server Type** field, select whether you want to make the connection by means of **Connect** (SAS/CONNECT) or **Share** (SAS/SHARE). SAS/CONNECT is the single-user server, and SAS/SHARE is the multi-user server. (To connect with SAS/SHARE, you must contact your SAS IT Resource Management administrator to request a SAS/SHARE server on the host where the remote PDB resides.)

  Also, if the remote server is using a spawner, select the spawner box. (This box is a toggle. To reverse the selection, select the box again.)

  Note: Both SAS/CONNECT and SAS/SHARE servers can be used with TCP/IP.
  Use of a spawner for TCP/IP is optional. △
- If you select "Remote server is using a spawner," the **Service Name** field opens. Type the service name.

  Or if you want to use a port number (instead of a service name) in the **Service Name** field, use two underscores preceding the port number. Thus, if the port number is *nnnn*, the value in the **Service Name** field would be *__nnnn*.

6 Select the **Communication** tab.

- In the **Communication Method** field, select **TCP** (TCP/IP).

  Note: TCP/IP is available on all client/server combinations that SAS IT Resource Management supports.
  TCP/IP must be installed on both the client and server. TCP/IP is almost always installed by default with all operating systems. △
- TCP/IP requires a script that is specific for your site.

  Select the **Logon Script** box. (This box is a toggle. To reverse the selection, select the box again.) Also, below the box, provide the full path and name of the script. Sample scripts (named *TCP*.scr*) are provided in the *sasmisc* directory in the location where SAS IT Resource Management is installed at your site.

  Depending on the needs of your site, you might need to edit the script file. Select the **Edit** button to browse the logon script, determine whether changes are required, and make changes if necessary.
- In the **SAS Command on server** field, type the name of the command (on the server host) that invokes SAS.

7 Select the **Libraries** tab.

- If you want to use the site library on the remote host, in the **Location of SITELIB on server** field, type the full path and name (or fully qualified name) of that site library. If you do not know the location, ask your SAS IT Resource Management administrator.

  If you want to use the site library on the local host, leave this field blank.
- In the **Location of PGMLIB on server** field, type the full path and name (or fully qualified name) of the program library. If you do not know the location, ask your SAS IT Resource Management administrator.
- In the **Location of ITRM Macros on server** field, type the full path and name (or fully qualified name) of the directory or PDS that contains the SAS IT Resource Management macros. If you do not know the location, ask your SAS IT Resource Management administrator.

- In the **Location of MXG Format Library on server** field, type the full path and name (or fully qualified name) of the MXG Format Library. If you do not know the location, ask your SAS IT Resource Management administrator.

8 Select the **Advanced** tab.

- If you want to submit any SAS code on the client host before signing on to the server host, type the code in the **SAS Code to submit locally** field, after the two comment lines. Typically, this field is not used.
- If you want to submit any SAS code on the server host after signing on to the server host, type the code in the **SAS Code to submit to server** field. Typically, this field is not used.

9 Select **OK**. If some required field is not filled in, SAS IT Resource Management displays a message in the message area at the bottom of the window.

SAS IT Resource Management returns you to the Remote Server Connection Profiles window. The profile that you created is now in the list of profiles.

10 Select **OK** to return to the Manage PDBs window.

11 Select **Close** to return to the main window.

For more information about the fields in a remote profile, follow these steps:

1 On the main window's **Administration** tab, select Manage PDBs. The Manage PDBs window opens.

2 Select **Locals**. A menu opens. From the menu, select **Remote Profile**. The Remote Server Connection Profiles window opens.

3 Select a remote server profile and click the right mouse button. A menu opens. From the menu, select **Properties**. The Remote Server Profile Definition window opens.

4 Click **Help** to display additional information about the fields in the remote server profile.

Note: Because SAS IT Resource Management clients usually have network access to the server if the server is on Windows, the clients rarely need to use a remote profile to access a PDB on a Windows server. △

Deleting a Remote Server Profile

You can delete a remote server profile

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client license. △

Deleting a Remote Server Profile (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can delete a remote-server profile by following these steps:

- On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
- Select the **Locals** menu. A menu opens. From the menu, select **Remote Profile**. The Remote Server Connection Profiles window opens.
- Select the name of the profile that you want to delete.
- Select **File**. A menu opens. From the menu, select **Delete**. No confirmation is requested.

SAS IT Resource Management redisplays the Remote Server Connection Profiles window. The remote server profile that you deleted is no longer in the list of profiles.

- To return to the Manage PDBs window, select **OK**.
- To return to the main window, select **Close**.

Viewing/Editing a Remote Server Profile

You can view or edit the values in a remote-server profile

- by using the SAS IT Resource Management GUI for UNIX and Windows.

Note: Your local host must have a SAS IT Resource Management client license. △

Viewing/Editing a Remote Server Profile (UNIX, Windows)

In the SAS IT Resource Management GUI for UNIX and Windows, you can display and/or edit the values in a remote-server profile by following these steps:

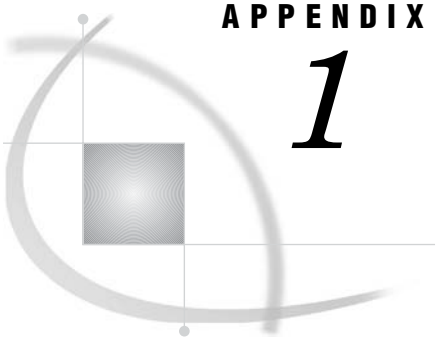
1. On the main window's **Administration** tab, select **Manage PDBs**. The Manage PDBs window opens.
2. Select **Locals**. A menu opens. From the menu, select **Remote Profile**. The Remote Server Connection Profiles window opens. In it is a list of the sample profiles and the additional profiles, if any, that have been created.
3. Select the profile that you want to view.
4. Select **File**. A menu opens. From the menu, select **Open**. The Remote Server Profile Definition window opens.
5. View and/or edit the values of the fields. For information about each field, select **Help** in the Remote Server Profile Definition window or see "Creating a Remote Server Profile" on page 574.

   Note: You can change the name of the profile, as long as the new name is unique within your set of remote server profiles. The profile name can be up to eight characters in length and it is required. For TCP/IP, the profile works faster if the profile name is the same as the server name. For NetBIOS that uses SAS/CONNECT, the server name (and the profile name) should be the same as the -NETNAME setting that was used in creating the spawner on the server. △

   Note: If you want to use a port number in the **Server Name** field (in order to make a remote connection by using the z/OS spawner), use two underscores preceding the port number. Thus, if the port number is *nnnn*, the value in the **Server Name** field should be *__nnnn*. △

   The remote server profiles are in your SASUSER library. Thus, you can make changes to the remote server profiles without affecting other users.

   Note: The logon and logoff scripts are physical files and can be located anywhere. Thus, changing the script files might affect others, if others use the same scripts. △
6. When you finish viewing or changing the profile, select **OK**. SAS IT Resource Management returns you to the Remote Server Connection Profiles window.
7. Select **OK** to return to the Manage Report Definitions window.
8. Select **Close** to return to the main window.



APPENDIX

1

Recommended Reading

Here is the recommended reading list for this title:

- *Getting Started with SAS IT Resource Management*
- *SAS IT Resource Management 2.7: Macro Reference*
- *SAS IT Resource Management 2.7: Server Setup Guide*
- *SAS IT Resource Management: Administration and Batch Reporting Course Notes*
- *SAS IT Resource Management: Interactive Reporting Course Notes*

For a complete list of SAS publications, see the current *SAS Publishing Catalog*. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513
Telephone: (800) 727-3228*
Fax: (919) 677-8166
E-mail: **sasbook@sas.com**
Web address: **support.sas.com/pubs**
* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.

Glossary

activate
to specify that a particular performance data warehouse, site library, or table is the one that you want to use now.

active PDB
the performance data warehouse that you are currently using.

active site library
the library of site-related information that you are currently using.

active table
the table to which the control statements in a stream of %CPDDUTL control statements currently refer.

administration tasks
tasks that manage the SAS IT Resource Management server. Administration tasks typically require write access to a performance data warehouse (PDB), to a site library, or to a program library. See also reporting tasks.

age limit
in a table, a unit of time that describes how long existing data is to be kept in a performance data warehouse (PDB) when new data is processed and reduced. An age limit specification pertains to the data that is already in a particular level of a particular table, and does not pertain to any data that is being read in. In a report gallery, the age limit specification describes how long the reports are to be kept.

aged data
data that is deleted from a level of a table in the performance data warehouse (PDB) by the process task or the reduce task because the data exceeds the age limit for keeping data in that level of that table.

aggregate
another term for summarize.

analysis variable
a numeric variable that is used to calculate statistics or to display values. Usually an analysis variable contains quantitative or continuous values, but this is not required.

API (application programming interface)
a set of software functions that facilitate communication between applications and other kinds of programs, services, or devices.

archive
a set of archive libraries for a specific performance data warehouse (PDB).

archive library
a SAS library that contains the data that was incoming to the detail level of a performance data warehouse (PDB) during a particular execution of the process task. An archive library is one of the libraries in the archive. See also detail-level data.

archive location
for a performance data warehouse (PDB), specifies where the archive is stored, if at least one table specifies archiving. The archive location (root) on z/OS is a partitioned data set (PDS). The archive location (root) on UNIX and Windows is a directory.

archive status
for a table, a value that specifies whether data that was incoming to the detail level of the table is to be copied to an archive.

archived data
data that was incoming to the detail level and which was then copied to an archive library during an execution of the process task.

autoexec file
a file that contains SAS statements that are executed automatically when SAS is invoked. The autoexec file can be used to specify SAS system options, as well as to specify librefs and filerefs for data sources that are used frequently.

availability
the percentage of time that a system is available to respond to the requests of its users.

axis
a one-dimensional line that represents the scale that is used for plotting the values of x, y, or z coordinates.

backload
to process and reduce data that was logged before the most recently logged data in the performance data warehouse (PDB).

backup copy
a complete copy of one or more libraries.

batch job
a unit of work that is submitted to an operating system for batch processing. On z/OS, a batch job is a set of JCL statements; on Windows, a batch job is a task; and on UNIX, a batch job is a background process.

batch mode
a method of executing SAS programs in which a file that contains SAS statements plus any necessary operating environment commands is submitted to the computer's batch queue. After you submit the program, control returns to your terminal or workstation, where you can perform other tasks. Batch mode is sometimes referred to as running in the background. The program output can be written to files or printed on an output device.

browser
See Web browser.

BY variables list
1) in the detail level of a table definition, a list of variables that specifies the sort order and grouping of the observations in the detail level of that table. 2) in a report definition, a list of variables that specifies the sort order and grouping of the data in

the report. 3) in a rule definition, a list of variables that specifies the sort order and grouping of the data in the results. See also exception, CLASS variables list.

calculation status
1) for a formula variable at any level, an indicator that specifies whether the value of the formula variable is to be calculated at that level. 2) for a summary statistic at a summary level, an indicator that specifies whether the value of that summary statistic is to be calculated at that summary level.

capacity planning
the process of measuring performance, availability, and workload volume in a particular computing environment and comparing these measurements over time to baseline values. The object of capacity planning is to predict when resources will need to be upgraded.

catalog entry
See entry type, SAS catalog.

chart
a graph in which graphical elements such as bars or pie slices show the magnitude of data values. The graphical elements can represent one data value or a range of data values.

CLASS variable
a variable that is used to group, or classify, data. The data type of CLASS variables can be either character or numeric. Class variables can have continuous values, but they typically have a few discrete values that define the classifications of the variable.

CLASS variables list
1) in the day, week, month, or year level of a table definition, a list of variables that specifies the sort order and grouping of the data that is summarized into observations in that level of that table. There is one CLASS variables list for each summary level of each table. 2) in a report definition, zero, one, or more variables that may interact with other variables (such as the Group, SubGroup, or Stack variables) to specify the grouping of the data in the report.

client
See SAS IT Resource Management client.

collector
another term for data collector.

collector name
one of the two identifiers that specifies a particular set of collector-support software. The other identifier is tool name. For example, the pair of identifiers specifies which collector-specific staging code to run during the process task.

collector support
a set of software components that provide a variety of functions that enable SAS IT Resource Management to work with data from a particular collector. Each collector-support component is identified by both a collector name and a tool name.

combined view
a view that provides access to variables from two or more other views.

continuous values
values of a variable that theoretically could indicate an uncountable number of possible values. For example, a variable that measures the temperature of water in degrees Celsius under normal conditions is continuous because it can have any value from 0 to 100, even though the thermometer that is used is not able to measure temperatures beyond a particular level of decimal precision.

control data set
one of the data sets that the duplicate-data-checking macros maintain and use to determine whether data from a log has already been processed.

control statement
an instruction or a string of keywords that you provide to the data dictionary utility in order to direct the utility's actions in batch mode. For example, control statements can direct the utility to copy a table definition from the supplied table definitions in the master data dictionary to the active performance data warehouse (PDB), to create and print table definitions, and to perform a variety of other functions.

cron
a UNIX utility for executing batch jobs on a periodic basis.

current site library
another term for active site library.

current table
another term for active table.

daemon
a process that starts and waits either for a request to perform work or for an occurrence of a particular event. After the daemon receives the request or detects the occurrence, it performs the appropriate action. If nothing else is in its queue, the daemon then returns to its wait state.

data collector
software that collects logged data that is related to IT services. The data collector could be a facility of the operating system, a product that is provided by a third-party vendor, or software that is written by consultants or by SAS IT Resource Management customers.

data dictionary
See master data dictionary, PDB's data dictionary.

data dictionary utility
a macro (%CPDDUTL) that is used for adding, changing, deleting, or checking the information in the data dictionary of the performance data warehouse (PDB). The utility can also generate a preliminary version of a table definition (and its associated variable definitions) from a SAS data set or view, from a character-delimited external file, or from other types of files.

data reduction
the summarization of data for a day, week, month, or year. The observations for the current week, month, and year contain week-to-date, month-to-date, and year-to-date data. Data reduction decreases the amount of memory that is required for storing data, but it still preserves information that can be used for analyzing trends.

data source software
software that logs data that is related to IT services.

data type
an attribute of a variable. In SAS IT Resource Management, a variable can have a data type of either character or numeric. See also interpretation type.

data visualization
the act of representing data visually or graphically. Charts, plots, and graphs are some results of data visualization.

date format
in SAS software, the instructions that tell SAS how to write numeric values as date values.

datetime format
in SAS software, the instructions that tell SAS how to write numeric values as datetime values.

day-level data
performance data that is summarized for an entire day. Day-level data is summarized from detail-level data.

derived variable
a variable whose values are calculated from regular variables, constants, or both. See also regular variable, formula variable.

detail-level data
in general, performance data that has not been summarized. (For data collectors such as SAP that log an extremely large volume of data, detail-level data can be partially summarized.) Some of the data might have been changed by the staging code, by user-written exit routines for the process task, by partial summarization, or by some other means. See also day-level data, week-level data, month-level data, year-level data.

duplicate data
1) observations that have identical values in all of the BY or CLASS variables. 2) observations that come from the same log. The meaning that is intended is explained when the term is used.

duplicate-data-checking macros
a set of macros (%CPDUPINT, %CPDUPDSN, %CPDUPCHK, and %CPDUPUPD) that control whether data from a log can be reprocessed into a performance data warehouse (PDB). See also control data set.

duration
1) in records that represent interval data but which do not contain a value for the length of the interval, a value in seconds that represents the length of time between records. 2) in records that represent interval data and which do contain a value for the length of the interval, that value in seconds. 3) in observations for interval-type tables, the variable that contains the length of the interval in seconds.

entry type
a characteristic of a SAS catalog entry that identifies the catalog entry's attributes and structure to SAS software. When a catalog entry is created, SAS automatically assigns the entry type as part of the name.

ETL (extract, transform, load)
another term for process.

event-type table
a table in which each observation represents an event. The timestamp on the observation represents the date and time of the event. See also interval-type table.

exception
an observation that results from evaluating a rule. Each set of unique values of the rule's BY variables represents a group of observations against which the rule ran. The total number of exceptions indicates the number of groups for which the rule is true, not the number of individual observations for which the rule is true. See also results.

exit code
additional code that is to be executed at a specified point in the supplied software.

exit point

a point in the supplied software where additional code runs (if additional code is provided).

external detail
a view that enables data that is formatted like detail-level data to be reduced directly into the summary levels of the performance data warehouse (PDB) and therefore to bypass the detail level in the PDB. The data to be reduced might or might not be in a PDB.

external file
a file that is created and maintained by a host operating system or by another vendor's software application. Either data or stored SAS statements can be read from an external file. Data can be written as output to an external file. A SAS data set is not an external file.

external name
1) when referring to a SAS IT Resource Management table, the name of the object that contains the data that is to be merged into the table. The object can be a staged data set, a staged view, or a character-delimited file. 2) when referring to a SAS IT Resource Management variable, the name of the corresponding variable in the staged data set, in the staged view, or in the character-delimited file.

fileref (file reference)
a short name (or alias) for the full physical name of an external file. A SAS FILENAME statement maps the fileref to the full physical name.

folder
a repository for objects that are of the same type. For example, report definitions are stored in one or more report definition folders, rule definitions are stored in one or more rule definition folders, and palette definitions are stored in one or more palette definition folders.

formula variable
a variable whose values are calculated from regular variables, derived variables, other formula variables, and/or constants. The values are calculated when they are needed; they are not stored. The formula's source statements are stored. See also regular variable, derived variable.

fully populated report
a report that contains as much data as the report definition allows.

fully populated table
a table that contains as much data as the age limit specifications allow.

gallery
all the reports (and the associated left and right frames) that are visible in a Web browser when a particular welcome.htm file is selected.

generic collector
one of the data collectors for which no table definitions are supplied.

Generic Collector Facility
a component of SAS IT Resource Management that enables users to process data into a table for which the user provides the table definition. The table is typically defined by using the GENERATE SOURCE control statement.

global macro variable
a macro variable that can be referenced in any referencing environment in a SAS program, except when there is a local macro variable that has the same name. A global macro variable exists until the end of the session or program. See also macro variable.

graph report
a report that uses high-resolution graphics. See also text report.

graphics option
in a SAS GOPTIONS statement, an option that controls some attribute of the graphics output. The specified value remains in effect only for the duration of the SAS session. Some graphics options override parameters that have been specified for a graphics output device.

graphics output device
any terminal, printer, or other output device that is capable of displaying or producing graphical output.

graphics stream file (GSF)
a file that contains device-dependent graphics commands from a SAS/GRAPH device driver. This file can be sent to a graphics device or to other software applications.

GSF
See graphics stream file (GSF).

GUI
graphical user interface.

host
an operating environment that is identified by a name. For example, a domain name or an IP address can be used to identify a computer or a router, and an LPAR name can be used to identify an LPAR.

ID variable
a variable that contains an alternate identifier for the data in a CLASS variable. For example, the value of a CLASS variable could be a device address, and the value of the corresponding ID variable could be the name of the device. ID variables are useful for identification because their values rarely change.

ID variables list
a list of ID variables. There is one ID variables list for each summary level of each table.

index variables list
a list of variables for which a simple (non-compound) index is to be built. There is one index variables list for each level of each table. If a variable is indexed, a WHERE expression that includes that variable, such as a WHERE expression that subsets data for a report definition or a rule definition, typically runs more quickly.

install a table
to store a table definition and its associated variable definitions (as well as override control statements, if any) in the master data dictionary, so that the table definition can be used for other performance data warehouses (PDBs) as if it were a supplied table definition.

install collector support
to integrate user-written collector-support software into SAS IT Resource Management so that it looks and works like the collector-support software that is supplied with SAS IT Resource Management.

Internet
the proper name of the super-network that connects many smaller networks around the world and which enables all the computers on those networks to exchange information. All of the computers on the Internet use the same set of communication protocols, called TCP/IP. The World Wide Web uses the Internet as an electronic highway. See also internet, intranet.

internet
any group of interconnected networks that use Internet Protocol (IP) for exchanging information. The individual networks are not necessarily owned by the same company or organization. The Internet is the biggest example of an internet. See also intranet.

interpolate
to estimate values that are between two or more known values.

interpretation type
an attribute of a variable that further defines the data type. For example, the data type Numeric has interpretation types such as Count, Time, Datetime, and Gauge. The interpretation type determines the default settings of other attributes of the variable, such as maximum length and format.

interval-type table
a table in which each observation represents an interval of time. The timestamp in the observation typically represents the beginning of the interval, and the duration in the observation represents the length of the interval. See also event-type table.

intranet
a private network of interconnected networks within a single company or organization. These networks use the same TCP/IP communication protocols that are used on the public, worldwide Internet, but they are either closed to the Internet or they are connected to the Internet via a firewall. A firewall enables internal users to access the Internet, but prevents unauthorized outsiders from accessing the intranet. In other words, intranets enable companies or organizations to share information without making that information available to everyone who has access to the Internet. See also Internet, internet.

IT
information technology.

IT services
services that an IT organization provides to customers. IT services could include support for computer systems, networks, applications, telephones, fax machines, and Web pages, as well as other types of software and hardware services.

keep status
1) for a table, a value that indicates whether the table is to be used or ignored. 2) for a variable in a table, a value that indicates whether the variable is to be used or ignored.

keyword parameter
a parameter that is identified by name. For example, in the macro call %CPDUPDSN (SOURCE=WWW);, 'SOURCE' is a keyword parameter that specifies the input for the macro %CPDUPDSN. Keyword parameters can be used in any order on the macro call, but they must follow any positional parameters.

LAN
local area network.

level
a logical grouping of all data that is consolidated for a specific unit of time. Within a table, there are five levels in which data is stored: detail, day, week, month, and year.

libref (library reference)
a short name (or alias) for the full physical name of a SAS library. A SAS LIBNAME statement maps the libref to the full physical name. A libref is the first part of a multi-level SAS filename and indicates the SAS data library in which a SAS file is

stored. For example, in the name SASUSER.ACCTS, SASUSER is the libref, and ACCTS is a file in the library that the SASUSER libref refers to. See also SAS library.

local PDB
a performance data warehouse (PDB) that is located on a disk that is attached to the local computer or which is on a disk drive that is accessible to the local computer through a service such as Network Neighborhood, NFS, AFS, or DFS.

LPAR
logical partition.

machine
any type of data processing hardware that can be recognized by SAS IT Resource Management as a single unit. A machine can be a single physical machine such as a computer, a router, or a telephone switch. It can also be a logical entity such as a logical partition of an S/390 or z900 operating environment or a domain on the Solaris operating system. A computer in which multiple CPUs share the same memory could also be regarded as a single machine.

macro call
the programming code that invokes a macro.

macro source
another term for macro call.

macro variable
a variable that is part of the SAS macro programming language. The value of a macro variable is a string that remains constant until you change it. Macro variables are sometimes referred to as symbolic variables.

maintain a table
to apply changes from a table definition in a master data dictionary to a table definition in a performance data warehouse (PDB).

management information base
See MIB (management information base).

master data dictionary
in the SAS IT Resource Management program library, the master data definitions of tables and their associated variables, as well as override control statements, if any. Master copies of user-written definitions can be installed in the data dictionary, too.

MIB (management information base)
a virtual database for data that is gathered by one or more SNMP agents.

migrate a table
to add a table definition and the definitions of its associated variables from the master data dictionary to a performance data warehouse's (PDB's) data dictionary. When a table is migrated, any override %CPDDUTL control statements that are associated with the table definition or with the definitions of its associated variables in the master data dictionary are also applied to the PDB's data dictionary.

month-level data
performance data that is summarized for an entire month or for the current month-to-date. Month-level data is summarized from detail- level data.

MXG software
a collection of SAS programs (from Merrill Consultants) that operates on raw performance data that is created by z/OS, VM, and AS/400 operating systems and subsystems, and by other programs that log data in the same format.

network

a configuration of data processing devices and software that are connected for the purpose of information exchange.

operating environment
a computer, or a logical partition of a computer, and the resources (such as an operating system and other software and hardware) that are available to the computer or partition.

override control statements
one or more %CPDDUTL control statements that are automatically applied to a table definition in the active performance data warehouse (PDB) as the last stage of adding (migrating) a table definition and the definitions of its associated variables from the master data dictionary to the PDB's data dictionary. In the master data dictionary, the override statements, if any, are stored separately from the definitions of the table and its associated variables.

package
collector support that can be distributed to and incorporated into existing SAS IT Resource Management installations in order to provide additional, updated, or new functionality. See also collector support.

palette
a stored set of SAS and SAS/GRAPH options, such as colors and patterns, that can be associated with one or more report definitions.

parameter
a data item that is passed to a routine. In the SAS macro facility, a parameter is a constant or variable that is passed to the macro routine when the routine is invoked. See also keyword parameter, positional parameter.

PDB
See performance data warehouse.

PDB level
another term for level.

PDB options
characteristics of a specific performance data warehouse (other than data, table definitions, and variable definitions). PDB options include archive device, archive path, archive parameters, archive engine, active collector, Daylight Saving Time definition, Greenwich Mean Time deviation, start of week, work shift schedule, and definitions of work shift codes. On z/OS, you can also specify whether to build MXG views as a PDB option.

PDB's data dictionary
in a SAS IT Resource Management performance data warehouse (PDB), a SAS library whose name and libref are DICTLIB. The library contains metadata such as definitions of tables and variables and status information for the data that the tables contain.

performance data
information about how your IT system is doing its work. You use this information to determine whether your system is doing work efficiently and to determine how to change its load or its tunable parameters so that it will perform more efficiently.

performance data warehouse (PDB)
logically, a data warehouse that contains detailed and summarized performance data, as well as information (metadata) that is needed for managing the data. Physically, each PDB consists of a coordinated set of nine SAS libraries that contain performance data plus other information that is related to one or more IT services.

permanent SAS library
a SAS library that is not deleted when a SAS session ends, and which is therefore available to subsequent SAS sessions.

PGMLIB
the libref (library reference name) and library name for a SAS IT Resource Management program library that is on the local server host. See also RPGMLIB.

positional parameter
in a SAS macro, a parameter that is identified only by its position in the definition and invocation of the macro. Positional parameters must be separated by commas. If a positional parameter is not used, a comma must be inserted to indicate that the parameter is not being used. For example, if filename is the second positional parameter in a macro, you would either type the name of your data file in the second position (parm1, filename, parm3), or use a comma as a placeholder (parm1, , parm3).

process
to extract raw data from a data collector or data source, to optionally transform the data, and to load the data into a detail-level SAS library in a performance data warehouse (PDB).

process step
another term for process task.

process task
the task (or step) that processes the raw data into the detail level of a performance data warehouse (PDB).

program library
the SAS library in which most of the SAS IT Resource Management software resides. For example, the master data dictionary resides in the program library. See also PGMLIB, RPGMLIB.

protocol
a set of rules that govern data communications between computers, between computers and peripheral devices, and between software applications. TCP/IP, FTP, and HTTP are examples of protocols.

pull-down menu
the list of menu items or choices that appears when you choose an item from a menu bar or from another menu.

QuickStart Wizard
a wizard in the GUIs for the SAS IT Resource Management server. This wizard enables you to select a data collector and the type of log that you want to work with. The wizard creates a new performance data warehouse (PDB). The wizard also creates several batch jobs, including jobs that you can use to process data into the PDB, to reduce the data in the PDB, and to generate reports about the data.

reduce
another term for summarize.

reduce step
another term for reduce task.

reduce task
the task (or step) that reduces the data in a performance data warehouse (PDB) from the detail level to the summarized levels.

reduction level
another term for summary level.

regular variable
a variable whose values are acquired from the raw data and stored in the performance data warehouse (PDB). See also derived variable, formula variable.

remote PDB
a performance data warehouse (PDB) that SAS IT Resource Management can access by connecting to a remote host on which the PDB is available as a local PDB. See also local PDB.

remote server profile
a stored set of values that are used for connecting to a remote server host.

report
a graph or text that is generated by running a report definition.

report definition
a specification that is used for generating a report. A report definition includes information such as the table and level, the names of the variables, the report style, and other attributes.

report gallery
another term for gallery.

reporting tasks
tasks that generate or manage reports. Reporting tasks typically require 'read' access to a performance data warehouse (PDB), to a site library, or to a program library. See also administration tasks.

response time
the amount of time between the submission of a request for service and the user's receipt of the system's reply to that request.

restore
to recover the contents of a backup copy.

results
1) exceptions that are generated by running (evaluating) one or more rule definitions. The exceptions are written to the results data set. 2) reports (text, graphs, or both) that are generated by running one or more report definitions. The reports are written to a SAS catalog and then to a directory or PDS. See also exception.

retrieve
to recover some or all of the contents of an archive.

RPGMLIB
the libref (library reference name) for the program library on a remote server host. See also PGMLIB.

rule definition
a specification for detecting exceptions. A rule definition includes information such as the table and level, the expression, the message, and other attributes. See also exception.

SAS catalog
a SAS file that stores many different kinds of information in smaller units called catalog entries. A single SAS catalog can contain several different types of catalog entries, such as graphs (.GRSEGs) and source code (.SOURCEs). See also SAS catalog entry.

SAS catalog entry
a separate storage unit within a SAS catalog. Each entry has an entry type that identifies its purpose to SAS. Some catalog entries contain system information such

as key definitions. Other catalog entries contain application information such as window definitions, Help windows, formats, informats, macros, or graphics output.

SAS data set
a file whose contents are in one of the native SAS file formats. There are two types of SAS data sets: SAS data files and SAS data views. SAS data files contain data values in addition to descriptor information that is associated with the data. SAS data views contain only the descriptor information plus other information that is required for retrieving data values from other SAS data sets or from files whose contents are in other software vendors' file formats.

SAS data view
a type of SAS data set that retrieves data values from other files. A SAS data view contains only descriptor information such as the data types and lengths of the variables (columns), plus other information that is required for retrieving data values from other SAS data sets or from files that are stored in other software vendors' file formats. Both PROC SQL views and SAS/ACCESS views are considered to be SAS data views. SAS data views are of member type VIEW.

SAS datetime value
an integer that represents a date and a time in SAS software. The integer represents the number of seconds between midnight, January 1, 1960, and another specified date and time. For example, the SAS datetime value for 9:30 a.m., June 5, 2000, is 1275816600.

SAS IT Resource Management client
a computer on which SAS IT Resource Management software is installed with a client license. The client accesses PDBs on servers through a telecommunications protocol such as TCP/IP. Most client access is limited to "readonly" and includes support for reporting, data visualization, and querying PDB definitions and status.

SAS IT Resource Management server
a computer on which SAS IT Resource Management software is installed with a server license. The server has 'write' access to the performance data warehouses (PDBs) and is used for processing and reducing data into one or more PDBs that are accessible through the host operating environment's file system. In addition, the server is used for administering PDBs and can also be used to schedule batch work, to create report and rule definitions, to generate reports, and to manage report and rule definitions and reports.

SAS library
a collection of one or more SAS files that are recognized by SAS and that are referenced and stored as a unit. Each file is a member of the library.

SAS procedure
a program that produces reports, manages files, or analyzes data and which is accessed with a PROC statement. Many procedures are included in SAS software.

scheduled job
in SAS IT Resource Management, one of a set of batch jobs that typically run at night and that process data into one or more performance data warehouses (PDBs), reduce data in the PDB(s), generate reports, and back up the PDB(s).

schema
the physical layout of logged or collected data.

server
See SAS IT Resource Management server.

service-level agreement (SLA)

an agreement that specifies the performance characteristics that a system is to meet. For example, availability, throughput, and response time are characteristics that are often included in SLAs.

setup
the task of customizing the collector and SAS IT Resource Management software in order to enable data from the collector to be processed into a performance data warehouse (PDB) and reduced in the PDB.

simple index variable
a variable for which an index is built. The index contains information about the values of the variable and about the observations that contain those values.

site
a group of performance data warehouses (PDBs) that are located on the same server host and which share the same default values, the same list of holidays, and the same work shift code for holidays. The shared information, known as site options or site properties, is stored in a site library. See also site library, site options.

site library
a SAS library in which site-specific information is stored. See also SITELIB, site.

site options
options whose values are intended as default values for all performance data warehouses (PDBs) for a specific site. These site options include archive device, archive path, archive parameters, archive engine, active collector, Daylight Saving Time definition, Greenwich Mean Time deviation, start of week, work shift schedule, and definitions of work shift codes. Site options can also include some site-level specifications that do not act as PDB defaults. These site options include the list of holidays and the work shift code for holidays. On z/OS, you can also specify whether to build MXG views as a site option.

SITELIB
the libref (library reference name) for the active site library. See also site library, active site library.

SNMP (Simple Network Management Protocol)
a protocol or standard that is used for exchanging network management information, which is information that enables the network to be managed.

SNMP agent
a software module that performs network management functions. See also SNMP (Simple Network Management Protocol).

staged data
raw data that has been read, optionally transformed, and either written to a SAS data set or made available through a SAS data view.

staging code
SAS statements that read raw data, transform it, and either write it to a SAS data set or make it available through a SAS data view.

suffix
in SAS IT Resource Management, the eighth character in the name of a variable at the day, week, month, or year level. The suffix indicates the type of statistic (for example, N means count, S means sum, and no suffix means average). Earlier characters in the name contain the name of the detail-level variable on which the statistic is based.

summarize

to calculate summary statistics in order to represent the values of variables in the detail level of a performance data warehouse (PDB). The summary statistics are stored in the day, week, month, and/or year levels of the PDB.

summary level
a performance data warehouse (PDB) level in which the performance data is summarized for a specific unit of time. The four summary levels are day, week, month, and year. The data at a particular summary level is stored in a corresponding library. For example, data that is summarized by day is stored in the DAY library of the PDB.

supplied collector support
collector support that is supplied with SAS IT Resource Management.

supplied format
a format that is shipped with SAS IT Resource Management.

supplied report definition
a report definition that is shipped with SAS IT Resource Management.

supplied rule definition
a rule definition that is shipped with SAS IT Resource Management.

supplied software
software that is shipped with SAS IT Resource Management.

supplied table definition
a table definition that is shipped with SAS IT Resource Management.

supplied variable definition
a variable definition that is shipped with SAS IT Resource Management.

table
the table definition, the status information about the data that is associated with the table definition, and the data itself. A table's data is contained in SAS data sets from each of the performance data warehouse's five data libraries: DETAIL, DAY, WEEK, MONTH, and YEAR. A table's metadata is contained in the performance data warehouse's DICTLIB library.

table definition
1) the attributes of a particular table. 2) the attributes of a table and the attributes of its associated variables. 3) the attributes of a table and of its associated variables, plus the status information about the data that is associated with the table and its variables. The meaning that is intended is explained where the term is used. See also table.

text report
a report that tabulates or prints data values or which produces low- resolution graphics. See also graph report.

throughput
the rate at which requests for work are serviced by a computer system.

time format
in SAS software, the instructions that tell SAS how to write numeric values as time values.

tool name
one of a pair of identifiers that specifies a particular set of collector support. The other identifier is collector name. For example, the pair of identifiers specifies which collector-specific staging code to run during the process task.

tuning

the process of tracking the performance of an IT service, comparing the performance to service objectives in order to identify bottlenecks and hardware problems, and then adjusting the service to eliminate the problems. Tuning also involves distributing work equitably to the available computer hardware in order to optimize the use of resources.

utilization
1) for resources that can be partially occupied (for example, memory), the fractional usage of a resource. 2) for resources that cannot be partially busy (for example, a CPU), the ratio of time that a resource is being used to the total elapsed time.

variable
a column in a SAS data set or in a SAS data view. The data values for each variable describe a single characteristic for all observations. See also derived variable, formula variable, regular variable.

view
logically, a set of observations that contains values for regular, derived, and formula variables. Physically, a view provides information about regular and derived variables and their locations, as well as source statements for calculating the values of formula variables. There is one view for each level of each table in a performance data warehouse (PDB).

Web browser
a software application such as Microsoft Internet Explorer or Netscape Navigator that provides access to information that is on the Internet or on your company's intranet.

Web gallery
another term for gallery.

Web-enabled report
a report that has attributes that enable it to be displayed by a Web browser.

week-level data
performance data that is summarized for an entire week or for the current week-to-date. Week-level data is summarized from detail-level data.

weight
a numerical coefficient that is assigned to an item and which indicates the relative importance of the item in a frequency distribution or population.

weighting variable
a variable whose values represent the weights for each observation.

workload
the amount of work a system is experiencing. Workload can be measured in terms such as characters per second, processes per second, or transactions per second.

year-level data
performance data that is summarized for an entire year or for the current year-to-date. Year-level data is summarized from detail-level data.

Index

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z

Your Turn

If you have comments or suggestions about *SAS IT Resource Management 2.7: User's Guide*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
E-mail: **yourturn@sas.com**

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
E-mail: **suggest@sas.com**