



SAS/ASSIST[®] 9.1

Administrator's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS®/ASSIST 9.1: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS®/ASSIST 9.1: Administrator's Guide

Copyright © 2004, SAS Institute Inc., Cary, NC, USA

ISBN-13: 978-1-59047-208-8

ISBN-10: 1-59047-208-X

All rights reserved. Produced in the United States of America.

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

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

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st printing, January 2004

1st electronic book, March 2008

2nd printing, January 2008

2nd electronic book, July 2011

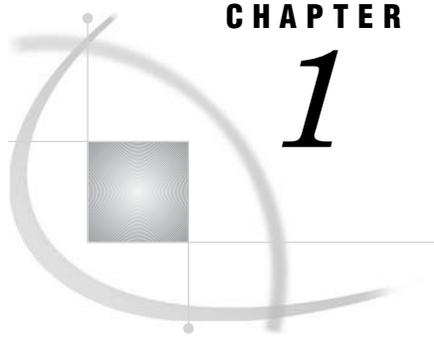
SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/publishing or call 1-800-727-3228.

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

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

Contents

Chapter 1	△ Introduction	1
	Purpose and Audience	1
	Chapter Overview	1
	Available Resources	2
Chapter 2	△ Setting Up SAS/ASSIST Software	3
	Introduction	3
	Using the SASPARM Table	3
	Setting Up SAS/ASSIST Software for Individual Users	4
	Command-line Options	5
Chapter 3	△ Setting Up and Customizing Master and Group Profiles	7
	Introduction	7
	Creating a Master Profile	7
	Opening and Using the Master/Group Profile Window	8
	Creating and Updating Group Profiles	10
	Editing the Options for Existing Master and Group Profiles	13
	Assigning and Changing the Status of Profile Options	15
	Profile Option Values	18
Chapter 4	△ Printing Reports and Graphics	19
	Introduction	19
	Printing Reports	19
	Printing Graphics	20
Chapter 5	△ Setting Up Query Managers	31
	Introduction	32
	Overview	32
	Defining and Using Query Managers	33
	Defining and Using a Public Query Manager	35
	Defining and Using a Personal SAS Query Manager	47
	Deleting Query Managers	54
	Creating SAS Query Manager Tables	55
	Creating DB2 Query Manager Tables	60
Chapter 6	△ Creating Metadata	69
	Overview	69
	Metadata Feature	69
	Generating Metadata in Batch	71
	Sample Metadata-Generating Program	73
Appendix 1	△ Recommended Reading	75
	Recommended Reading	75
	Glossary	77
	Index	85



CHAPTER

1

Introduction

Purpose and Audience 1

Chapter Overview 1

Available Resources 2

Purpose and Audience

This document contains information on setup and maintenance of SAS/ASSIST software.

The *SAS/ASSIST Administrator's Guide* is written for users at customer sites who have front-line responsibility for SAS at their site. This position varies from site to site, but generally includes the SAS Installation Representative and the SAS Support Consultant.

Note: The displays in this document were captured in the z/OS operating environment. Δ

Chapter Overview

Chapter 1, "Introduction," on page 1

provides an introduction of the document and a chapter overview.

Chapter 2, "Setting Up SAS/ASSIST Software," on page 3

describes how to set up SAS/ASSIST software, including using the SASPARM table, the SAS autoexec file, and command-line options. This chapter also includes a checklist for setting up SAS/ASSIST software.

Chapter 3, "Setting Up and Customizing Master and Group Profiles," on page 7

describes the Master/Group Profile window and how you can use it to create and customize profiles.

Chapter 4, "Printing Reports and Graphics," on page 19

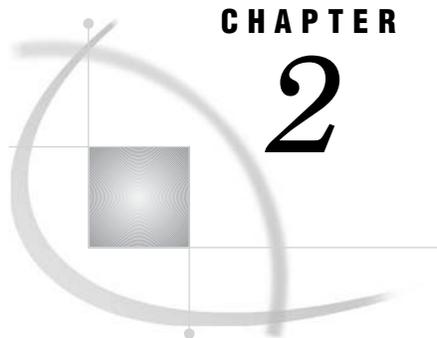
describes how to print graphics while using SAS/ASSIST software. Included is a discussion of the Form window subsystem used to print reports and device drivers used to print graphics.

Chapter 5, "Setting Up Query Managers," on page 31

describes types of Query Managers available in SAS/ASSIST software. Included are examples of different types of Query Managers and a discussion on creating Query Manager Tables.

Available Resources

System Administrators should have available the *Getting Started with SAS/ASSIST* and *Doing More with SAS/ASSIST* documents. They should also have available the companion documentation for their operating environment.



CHAPTER

2

Setting Up SAS/ASSIST Software

<i>Introduction</i>	3
<i>Using the SASPARM Table</i>	3
<i>Setting Up SAS/ASSIST Software for Individual Users</i>	4
<i>Command-line Options</i>	5

Introduction

This chapter describes how to set up SAS/ASSIST software, including using the SASPARM table, the SAS autoexec file, and command-line options. This chapter also includes a checklist for setting up SAS/ASSIST software.

Using the SASPARM Table

The SASPARM table is used by SAS/ASSIST software to store information about your SAS/ASSIST session, such as the active table, graphics options values, the active graphics device name and description, and the name of the catalog where programs are saved. You can create your own SASPARM table and use it with SAS/ASSIST software. You can also create a default SASPARM table that is used each time a new user at your site invokes SAS/ASSIST software.

When SAS/ASSIST software is invoked for the first time, a SASPARM table is created and is stored in the SASUSER library. The table is a permanent table, so each time you exit and then access SAS/ASSIST software again, the same SASPARM table is used.

Each time a SAS/ASSIST session is started, SAS/ASSIST software looks for the SASUSER.SASPARM table. If the SASUSER.SASPARM table does not exist, SAS/ASSIST software looks for a SASPARM table in the SASHELP library. If it finds a SASHELP.SASPARM table, it copies the table into the user's SASUSER library to create the SASUSER.SASPARM table. If it does not find a SASHELP.SASPARM table, it creates a new SASUSER.SASPARM table.

Therefore, if you want to set up system defaults for all your SAS/ASSIST software users, you can create a SASHELP.SASPARM table and modify it to store the information for your site's setup. Whenever a new user invokes SAS/ASSIST software, the SASHELP.SASPARM table is used to create the SASUSER.SASPARM table.

To create a SASHELP.SASPARM table to be used as a template, rename or delete your existing SASUSER.SASPARM table. Invoke SAS/ASSIST software and use the options under the **Setup** menu to customize your SAS/ASSIST session. See "Doing More with Setup" in *Doing More with SAS/ASSIST* and "Setting Up SAS/ASSIST Software for Individual Users" on page 4 for more information. Copy your customized SASUSER.SASPARM table to SASHELP.SASPARM.

Another way to use an alternate SASPARM table is to use the PARMSDS= command-line option. Refer to “Command-line Options” on page 5 in this chapter for more information.

Setting Up SAS/ASSIST Software for Individual Users

Use the following checklist to set up SAS/ASSIST software for individual users.

- 1 Create a SAS autoexec file to customize the users’ access. To invoke SAS/ASSIST software from a SAS autoexec file, include the command **DM 'ASSIST' ASSIST;** in the SAS autoexec file. Refer to the SAS companion documentation for your operating environment for more information.
- 2 Use a customized SASPARM table to set default parameters for your site’s users. See “Using the SASPARM Table” on page 3 for more information.
- 3 Set up forms for users to access when printing reports, if necessary. See “Printing Reports” on page 19 for more information.
- 4 Copy and modify device drivers for users to access when printing graphics, if necessary. See Chapter 4, “Printing Reports and Graphics,” on page 19 for more information.
- 5 Customize other options for the user by typing commands into the start program in the Master profile. Access the Master/Group Profile window by selecting **Tasks ► Setup ► Profiles ► Master/group**

In the Master/Group Profile window, you can

- Specify any logon and logoff exits or an alternate main menu. See Chapter 12, “Doing More with Setup,” in *Doing More with SAS/ASSIST* for more information.
- Set up a default private menu for the user. See Chapter 5, “Saving and Using Task Window Selections,” in *Getting Started with SAS/ASSIST* for more information.

You can also customize the start program in the Master/Group Profile window. Specify the **Type of start program** as either **Catalog** or **File**, depending on whether you want to reference a catalog entry or an external file. Type the physical name of the location of the start program in the **Value** field for **SAS/ASSIST start program**.

You can customize the following options in the start program:

- Use the LIBNAME and FILENAME statements in the start program to specify the libraries and files that users should see when a list of libraries or files is displayed in SAS/ASSIST software. By using libnames and filenames that are in the start program, users select their active table. Only those tables specified in the start program and those that are found in the default libraries that the SAS System assigns, such as the SASUSER library, are listed.
- Set the graphics defaults. See Chapter 4, “Printing Reports and Graphics,” on page 19 for more information.
- Set any prefix and suffix commands. See Chapter 6, “Doing More with Graphics,” in *Doing More with SAS/ASSIST* for more information.
- Define any custom formats or informats. See Chapter 3, “Doing More with Data Management,” in *Doing More with SAS/ASSIST* for more information.
- Set up any remote connections for users so that a list of connections appears for them to choose from when they select **Tasks ► Remote ► Establish**

- connection...** See the “Doing More with Remote Connect” chapter of *Doing More with SAS/ASSIST* for more information.
- Make sure the MAPS libref is assigned to give users access to the MAPS tables. See *SAS/GRAPH Software: Reference, Volumes 1 and 2* for more information.

Command-line Options

Listed below are the various command-line options you can use with the ASSIST command. Refer to Chapter 12, “Doing More with Setup,” in *Doing More with SAS/ASSIST* for more information on the LOGON=, LOGOFF=, and PRIMARY= command-line options.

PARMSDS=

specifies an alternate SASPARM table. The syntax for this command is

```
ASSIST PARMSDS='libref.member-name'
```

The *libref.member-name* argument specifies a two-level name for the alternate SASPARM table. SAS/ASSIST software uses the specified alternate SASPARM table instead of the SASUSER.SASPARM table.

LOGON=

specifies an application that initiates when you invoke SAS/ASSIST software. The syntax for this command is

```
ASSIST LOGON='libref.applds.applname.appltype'
```

The *libref.applds.applname.appltype* argument specifies a four-level name for the application. The default application database is SASUSER.SASAPPL. This option accomplishes the same function as **Logon exit...** in **Setup**; LOGON=NO overrides **Logon exit...**

LOGOFF=

specifies an application that initiates when you exit SAS/ASSIST software. The syntax for this command is

```
ASSIST LOGOFF='libref.applds.applname.appltype'
```

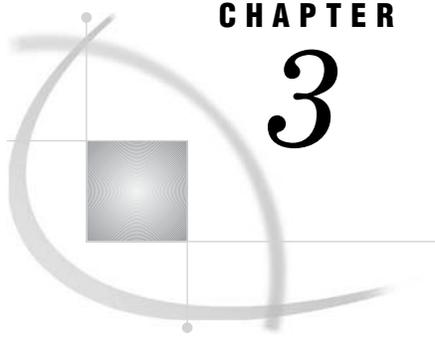
The *libref.applds.applname.appltype* argument specifies a four-level name for the application. The default application database is SASUSER.SASAPPL. This option accomplishes the same function as **Logoff exit...** in **Setup**; LOGOFF=NO overrides **Logoff exit...**

PRIMARY=

specifies an alternate primary menu application that appears when you access SAS/ASSIST software with the ASSIST command or when you select the **Main menu** button from within SAS/ASSIST software. The syntax for this command is

```
ASSIST PRIMARY='libref.applds.applname.appltype'
```

The *libref.applds.applname.appltype* argument specifies a four-level name for the application. The default application database is SASUSER.SASAPPL. This option accomplishes the same task as **Alternate main menu...** in **Logon/logoff exits...**; PRIMARY= NO overrides **Alternate main menu...**



CHAPTER

3

Setting Up and Customizing Master and Group Profiles

<i>Introduction</i>	7
<i>Creating a Master Profile</i>	7
<i>Opening and Using the Master/Group Profile Window</i>	8
<i>Creating and Updating Group Profiles</i>	10
<i>Modifying Group Profiles</i>	12
<i>Deleting Group Profiles</i>	12
<i>Editing the Options for Existing Master and Group Profiles</i>	13
<i>Assigning and Changing the Status of Profile Options</i>	15
<i>Status of D - Default</i>	15
<i>Status of U - Update</i>	15
<i>Status of R - Restrict</i>	15
<i>Defining Exceptions</i>	16
<i>Profile Option Values</i>	18

Introduction

The User Profile and Master/Group Profile windows enable a user, a database administrator (DBA), or a SAS administrator to set up SAS/ASSIST User profiles at three levels: Master, Group, and User. It is not necessary for an installation to have all three levels. The Master/Group profile can be used to set correct standard values for lower profile levels, restrict options to certain values, and overwrite existing options at lower profile levels. When you select **Tasks ► Setup ► Profiles ► Master/group**, the Master profile selection appears automatically in the Setup window, but it can be edited only by users who have write access to the SAS library that contains the Master profile. This chapter describes the Master/Group Profile window and how you use it to create and customize profiles.

Creating a Master Profile

The Master profile controls User profiles and Group profiles. Creating a Master profile requires write access to the SASHELP library. Typically, the SAS administrator or DBA creates a Master profile by completing the following steps. For some operating environments, these steps may have been completed during installation. Refer to the SAS installation instructions for your operating environment for additional information.

- 1 Type the physical pathname of the SAS library that will store the Master profile on line 1 of the Program Editor window of the SAS windowing environment. You must have WRITE access to this library.

- 2 Save this text to a file named SASHELP.QASSIST.PARMS.SOURCE by selecting **Save As Object...** from the **File** menu of the Program Editor window. In the Save As Object window, select the **Sashelp** library, then the **Qassist** catalog. Type **parms** in the **Entry Name** field, leave the **Entry Type** as the default value **SOURCE**, and select **Save**.
- 3 Invoke SAS/ASSIST software. The Master profile is created and stored in the QASSIST catalog in the SAS library you specified in the SASHELP.QASSIST.PARMS.SOURCE entry.

To make changes to the Master profile, use the Master/Group Profile window.

Opening and Using the Master/Group Profile Window

The Master/Group Profile window enables you to specify option values for all users or a group of users at a site. The Master profile controls the Group and User profiles and the Group profiles control the User profiles. This means that changes that are made in the Master profile are reflected in the Group and User profiles, depending on the status value of the profile options. Each option value is assigned a status: **Default**, **Update**, or **Restrict**. For more information on status values, see “Assigning and Changing the Status of Profile Options” on page 15.

When you start SAS/ASSIST software, Group and User profiles are compared with the Master profile and option values are updated according to their assigned status. If a user is not a member of a user group, the user’s profile is affected only by the Master profile. In this case, the User profile is at a lower profile level than the Master profile. If a user is a member of a user group that is associated with a Group profile, the user’s option values are controlled directly by that Group’s profile and indirectly by the Master profile. In this case, the Group profile is at a lower profile level than the Master profile, and the User profile is at a lower profile level than the Group profile. For more information on profile options, see “Assigning and Changing the Status of Profile Options” on page 15.

The Master/Group Profile window is used to

- set site-specific standard values for lower profile levels
- restrict the options to certain values
- overwrite existing option values at lower levels.

To open the Master/Group Profile window, follow this path:

Tasks ► Setup ► Profiles ► Master/group...

The following display shows the Master/Group Profile window.

Display 3.1 Master/Group Profile Window

SAS/ASSIST: Master/Group Profile			
File View Tools Help			
Name:	MASTER	Updated:	16JUL1999:11:18:23
Description:	Master profile		
Type:	ALL		
Cmd	Option	Value	Status
—	Message window	ALL	D
—	Permanent task catalog	SASUSER.PRIVATE	D
—	Update data automaticly	Yes	D
—	Save selections on end	Yes	D
—	Confirm changes	No	D
—	Menu style	New	D
—	Alternate menu bar		D
—	SAS/ASSIST style	Workplace	D
—	Show startup dialog	Yes	D
—	SAS/ASSIST start program		D
—	Type of start program	Catalog	D
—	SAS/ASSIST stop program		D
—	Type of stop program	Catalog	D
—	End SAS on exit	No	D
—	Logon - application name		D
—	Primary Menu after Logon	Yes	D
—	Catalog for private menu	SASUSER.PRIVATE	D
—	Use private menu	No	D
—	Private menu type	Block	D
—	Private menu title	PRIVATE MENU	D

The Master profile appears automatically in the Master/Group Profile window. The Master profile can be edited only by users who have write access to the SAS data library that contains the Master profile. Users without write access can open the Master/Group profile in browse mode. The window is empty when there is no Master profile. Like the User Profile window, the Master/Group Profile window contains all of the possible profile options. However, the Master/Group Profile window displays only a subset of the available options when the **Type** field has any value other than **ALL**. To see other subsets of options, type a ? in the **Type** field and press ENTER. Select a type from the list or enter a different type in the **Type** field. The default value for the **Type** field is **ALL**, which shows all the possible options. The types of options include the following:

- All
- Access to DB2 data (z/OS only)
- Access to SAS data
- Batch (z/OS only)
- General
- Private menu
- Query
- Query Manager
- Report Engine
- System Administration.

Scroll the window down as needed to see additional options. For more information on individual profile options, type an **I** in the **Cmd** field for the option and press ENTER. Most options have a default value. To change the value, select a new value from the list

or type over the current value. Table 3.1 on page 10 lists the commands that you can use in the **Cmd** field next to an option:

Table 3.1 Commands for the Master/Group Profile Window

Cmd	Action	Description
?	Lists commands	Displays a list of commands for the Cmd field.
I	Information	Opens the Option Information window, which provides information about the current option, including a description, a value, and the date the value was updated. You can edit the value in this window by typing a new value in the Value field. Alternatively, you can type a ? in the Value field to list all possible values for the option. Select a value from the list.
S	Information	Is identical to the I command.
X	Defines exceptions	Defines exceptions to any restricted values. Restricted values are specified by an R in the Status field. Exceptions to restricted values are specified by an asterisk (*) next to the status.

Note: Unlike the User Profile window, the Master/Group Profile window does not accept the **R** (Reset Value) command. To reset a value, type over the current value. Δ

After you have made changes to your Master profile, choose **Save** from the **File** menu to save your Master profile to the default location.

Creating and Updating Group Profiles

- 1 Select **Create Group Profile...** from the **Tools** menu of the Master/Group Profile window to create a new Group profile. The Create Group Profile window appears as shown in the following display.

Display 3.2 Create Group Profile Window

```

SAS/ASSIST: Create Group Profile
-----
Name:                GROUP1
Description:         Group profile 1
Catalog:            ASSIST
Physical name of SAS library:

[OK] [Cancel] [Help]

```

- 2 Type the name of the Group profile, a description, and a catalog for storage in the Create Group Profile window. The default catalog name is ASSIST.

- 3 Type the location of the SAS library to contain the catalog in the **Physical name of SAS library** field. The library must already exist or the profile is not saved.
- 4 Select **OK**. The Group profile is created, and you return to the Master/Group Profile window with the Group profile displayed. The **Name** and **Description** fields of the Master/Group Profile window show the name and description of the Group profile.
- 5 Select **Update User Group...** from the **Tools** menu to add users to a Group profile. The Update User Group window is shown in the following display.

Display 3.3 Update User Group Window

SAS/ASSIST: Update User Group

Name: ACCOUNT

Userids:

USERID1
USERID2
USERID3

16JUL1999:12:12:43

Status

D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D

OK Cancel Help

— Primary Menu after Logon Yes
— Catalog for private menu SASUSER.PRIVATE
— Use private menu No
— Private menu type Block
— Private menu title PRIVATE MENU

You can add users to a group or delete users from a group in this window. To add a user, type in a userid on the blank line and press ENTER. To delete a user, space over the userid to be deleted and press ENTER. The example in the previous display shows the ACCOUNT group with three users.

- 6 Select **OK** to save your results and return to the Master/Group Profile window when you are finished updating your user group.

Note: A user can belong to only one group at a time. For example, if a user already belongs to a group and is added to a new group, that user is automatically removed from the old group. Δ

Userids are stored in a SAS table called USER that is located in the same library as the Master profile. The table contains the columns USERID and GROUPID. The USER table can be updated within SAS/ASSIST software using the Update User Group window. You can also use a SAS program to populate the USER table with userids for various groups, for example:

```
libname libref 'physical filename';
data libref.user(keep=userid groupid);
  set data_set_name;
  select(DEPT);
  when('FINANCE')
    groupid=DEPT;
  when('SALES','MARKETING')
    groupid='Sales';
  otherwise
    groupid='Common';
```

```

end;
userid=empid;
run;

```

In the previous code, *libref* is the libref you chose to represent the library that contains the Master profile. *physical filename* is the physical filename of the location of the library containing the Master profile. *data_set_name* is the name of the table containing the IDs that you want to add to various user groups. In the example program, this table has a column called **DEPT** that contains the name of each employee's department, such as FINANCE, SALES, or MARKETING. This table also contains a column called EMPID that contains the userids for each employee. This program puts any employee userids who are in the FINANCE department into the Finance group. It also puts any employee userids who are in the SALES or MARKETING department into the Sales group.

Modifying Group Profiles

After you create a Group profile, you can modify its description. You can also change its storage location by altering the catalog and the physical name of the SAS library. To modify a Group profile, select **Modify Group Profile...** from the **Tools** menu of the Master/Group Profile window. The following display shows the Group profile ACCOUNT in the Modify Group window. Note that the profile name cannot be modified.

Display 3.4 Modify Group Profile Window

SAS/ASSIST: Master/Group Profile
File View Tools Help

Name: MASTER Updated: 16JUL1999:11:18:23
Description: Master profile
Type: ALL

Cmd Option	Value	Status
— Message window	All	D
— Permanent task catalog	SASUSER.PRIVATE	D
— Update data automaticly	Yes	D
— Save selections on end	Yes	D
— Confirm changes	No	D
— Menu style	New	D
— Alternate menu bar		D
— SAS/ASSIST style	Workplace	D
— Show startup dialog	Yes	D
— SAS/ASSIST start program		D
— Type of start program	Catalog	D
— SAS/ASSIST stop program		D
— Type of stop program	Catalog	D
— End SAS on exit	No	D
— Logon - application name		D
— Primary Menu after Logon	Yes	D
— Catalog for private menu	SASUSER.PRIVATE	D
— Use private menu	No	D
— Private menu type	Block	D
— Private menu title	PRIVATE MENU	D

After you make your changes to the Group profile, select **OK** to save your changes and return to the Master/Group Profile window.

Deleting Group Profiles

You can delete a Group profile by selecting **Delete Group Profile...** from the **Tools** menu. In order to delete a Group profile, the Master profile must be the open profile. For details on opening the Master profile, see “Editing the Options for Existing Master and Group Profiles” on page 13. To delete a Group profile, select **Delete Group**

Profile... from the **Tools** menu of the Master/Group Profile window. A list of profiles appears as shown in the following display. Notice that the Master profile is the open profile.

Display 3.5 Delete Group Profile Window

SAS/ASSIST: Master/Group Profile			
File View Tools Help			
ACCOUNT	MASTER	Updated:	16JUL1999:11:18:23
QC	tion: Master profile		
RD			
SALES	ALL		
Cnd Option	Value	Status	
— Message window	All	D	
— Permanent task catalog	SASUSER.PRIVATE	D	
— Update data automaticly	Yes	D	
— Save selections on end	Yes	D	
— Confirm changes	No	D	
— Menu style	New	D	
— Alternate menu bar		D	
— SAS/ASSIST style	Workplace	D	
— Show startup dialog	Yes	D	
— SAS/ASSIST start program		D	
— Type of start program	Catalog	D	
— SAS/ASSIST stop program		D	
— Type of stop program	Catalog	D	
— End SAS on exit	No	D	
— Logon - application name		D	
— Primary Menu after Logon	Yes	D	
— Catalog for private menu	SASUSER.PRIVATE	D	
— Use private menu	No	D	
— Private menu type	Block	D	
— Private menu title	PRIVATE MENU	D	

When you select a profile for deletion, a window prompts you to confirm or cancel the DELETE operation. Select **Yes** to proceed with the DELETE operation or **No** to cancel the operation. If users are still connected to the Group profile that is to be deleted, then a message appears, indicating that users are still controlled by the profile. If you continue and delete the group, the group is deleted, but the users are not deleted from the table that contains the userids for each group.

You can also use a SAS program to delete all users of a given group from the USER table, for example:

```
libname libref 'physical filename';
data libref.user;
  set libref.user;
  if groupid='name-of-groupid'
  then delete;
run;
```

In the previous code, *libref* is the libref you chose to represent the library that contains the Master profile. *physical filename* is the physical filename of the location of the library that contains the Master profile. *name_of_groupid* is the name of the group whose users you want to delete from the group.

Editing the Options for Existing Master and Group Profiles

To edit the options for an existing Master or Group profile, choose **Open...** from the **File** menu in the Master/Group Profile window. The Open Profile window, shown in the following display, lists all saved Group profiles as well as the Master profile.

Assigning and Changing the Status of Profile Options

Each option value in the Master/Group Profile window is assigned a status. Type a ? in the **Status** field to list the possible status values. To change a value, make a new selection from the list or type over the current value. The possible values for the **Status** field are as follows:

- D** for default
- U** for update
- R** for restrict.

For most option values, the initially assigned value of **Status** is **D** (default). Only the **Standard catalog** and **User - library** options are set up as restricted options.

The value that you use for the option status depends on how your User, Group, and Master profiles are set up at your site. See Chapter 3, “Setting Up and Customizing Master and Group Profiles,” on page 7 and “Creating and Updating Group Profiles” on page 10 for information about creating Master and Group profiles. A site may choose to have a Master profile and one or more Group profiles.

When the status for an option is changed, the change is reflected at lower profile levels only after a user exits and reinvokes SAS/ASSIST software. When a user starts SAS/ASSIST software, Group and User profiles are checked with the Master profile. The results of the comparison vary depending on the status of each option.

Status of **D** - Default

Option values with **D** (default) status transfer to lower profile levels automatically when the lower level profile is created. For example, all option values with **D** status in the Master profile are automatically transferred to new Group profiles as the Group profiles are created. In the absence of Group profiles, and in the case where a user is not defined to a Group, the values are transferred to the User profiles of all new users. When a new user is added to a group, the option values with a status of **D** in the Group profile are transferred to the user’s User profile. If a user is not defined in any User group, the user acquires the value’s options with a status of **D** from the Master profile. Users can change values with a status of **D**. If values with a status of **D** are later changed in the Master or Group profiles after a Group or User profile has already been created, the values defined at lower profile levels are not affected.

Status of **U** - Update

Option values with **U** (update) status are not transferred to lower profile levels automatically. After values have been changed at the Master or Group level and a user starts SAS/ASSIST software, a window appears and prompts the user to choose between retaining the current option value or allowing the updated value to overwrite the current values. A message appears for each updated option. Once a user chooses not to update a certain value, the user is not prompted about the same update the next time SAS/ASSIST is invoked. When values are changed at the Master level for **R** or **U** status options, or the status has been changed to **R** or **U**, opening a Group profile for editing causes a prompt for each value. The Group profile is updated accordingly at that time. The affected users do not acquire the changes until the next time they start SAS/ASSIST software.

Status of **R** - Restrict

Option values with **R** (restrict) status automatically update lower profile levels. When a user starts SAS/ASSIST software after a higher level profile change is made, a window

appears and notifies the user that an option value was overwritten and that the user's profile was updated. Users at lower profile levels cannot change restricted option values.

Defining Exceptions

Although a restricted value applies to all users at lower profile levels, you can use the Define Exceptions window to specify exceptions to the restrictions. In the Master/Group Profile window, type **x** in the **Cmd** field next to the restricted option for which you want to define exceptions and press ENTER. The Define Exceptions window appears for that option for the open profile. The following display shows the Define Exceptions window for the Master profile.

Display 3.8 Define Exceptions Window for Master Profile

```

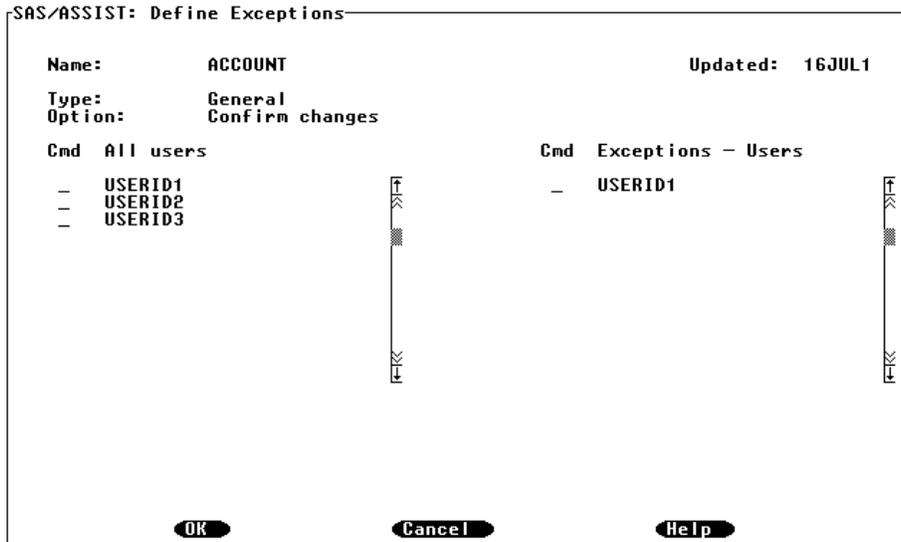
SAS/ASSIST: Define Exceptions
-----
Name:          MASTER                      Updated:  16JUL1
Type:          General
Option:        Confirm changes

Cmd  All groups                               Cmd  Exceptions - Groups
-----
-   ACCOUNT
-   QC
-   RD
-   SALES
-----

```

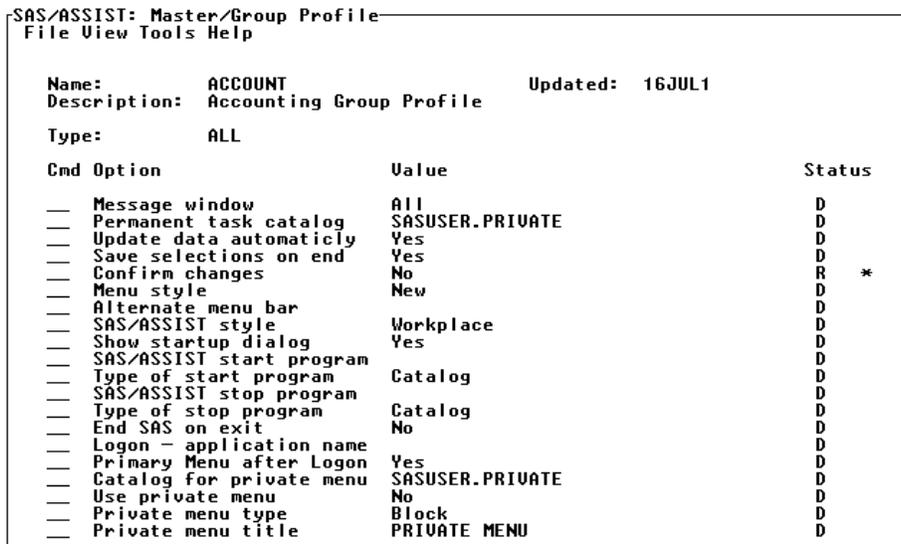
The window shows the name of the profile, the type of option, the option name, and the date the option value was last changed. The left column, **All groups**, lists all the groups. The right column, **Exceptions - Groups**, lists the groups to which the current restricted value does not apply. Type an **S** in the left column **Cmd** field to select groups that you want to define as exceptions for the current restriction. The selected groups appear in the **Exceptions - Groups** (unrestricted) column. To remove a group from the **Exceptions - Groups** (unrestricted) column, type a **D** in the **Cmd** field for **Exceptions - Groups** and press ENTER. When you are defining Group exceptions, the left column in the Define Exceptions window is named **All users** and the right column is named **Exceptions - Users**. The following display shows the Define Exceptions window for the ACCOUNT Group profile.

Display 3.9 Define Exceptions Window



After you define your exceptions, choose **OK** to return to the Master/Group Profile window. As shown in the following display, an asterisk (*) appears next to the **Status** field to indicate that exceptions apply to the restricted value.

Display 3.10 Exceptions Shown in the Master/Group Profile Window



Profile Option Values

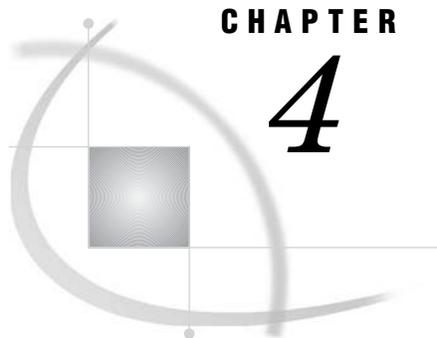
The User Profile window and the Master/Group Profile window have the same options, excluding System Administration options, which are Master profile options only. The two System Administration options available in the Master profile are

□ **Userid**

The **Userid** option is a variable name that contains the userid. If the variable starts with an ampersand (&), it is a macro variable. Otherwise, it is an environment variable. The default value for the **Userid** option is **&SYSJOBID**.

□ **Query exit**

The **Query exit** option is the name of a program or SAS Component Language entry to be called before the query is submitted. For further information, see the sample source code in **SASHELP.QASSIST.SAMPEXIT.SOURCE**.



CHAPTER

4

Printing Reports and Graphics

<i>Introduction</i>	19
<i>Printing Reports</i>	19
<i>Printing Graphics</i>	20
<i>Additional Information</i>	20
<i>Overview</i>	20
<i>Modifying Device Drivers</i>	20
<i>Modifying Graphics Device Defaults</i>	27
<i>Printing with Modified Drivers</i>	28
<i>Printing Directly to the Printer</i>	29

Introduction

This chapter provides instructions for printing reports and graphics while using SAS/ASSIST software. Included is a discussion of the Form window subsystem that is used to print reports and device drivers used to print graphics.

Printing Reports

When SAS prints text, it may need information about the type of printer to be used and the way the text will be formatted. This information is stored in a *form*, a SAS catalog entry of the type FORM. You can create custom forms to define the exact formatting and printing features you want to use. When you do not explicitly select a particular form, a default form is used. Forms are typically used with the mainframe operating environment and are turned off by default on the PC and UNIX operating environments. You can access the Form window subsystem by selecting **Tasks ► Setup ► Environment ► Forms management...**

For additional information on using the Form window subsystem, refer to *SAS Language Reference: Dictionary* and the companion documentation for your operating environment.

For more information on printing, see the companion documentation for your operating environment.

Note: To ensure that row and column separators and boxed tabular reports will be printed legibly when using the standard forms characters, you must use these resources:

- either the SAS Monospace or the SAS Monospace Bold font
- a printer that supports TrueType fonts

Characters that are used in standard forms are specified by using the FORMCHAR system option. Printer fonts are specified by using the SYSPRINTFONT system option.

Character and font specifications vary according to your operating environment and your print methods. If you run under the Windows operating environment, see the FORMCHAR and SYSPRINTFONT system options in the companion documentation. For all other operating environments, see these system options in *SAS Language Reference: Dictionary*. △

Printing Graphics

When printing graphics, you can specify various options that enable you to control your printout. Using the **Catalog management** item, you can create a personal catalog of graphics device drivers customized for your own use. You must have SAS/GRAPH software licensed to use this feature.

Additional Information

For additional information on printing graphics, refer to *SAS/GRAPH Software: Reference, Volumes 1 and 2* and *SAS/GRAPH Software: Using Graphics Devices* for your operating environment.

Overview

This section explains how to copy and modify a SAS/GRAPH device driver. For example purposes, the driver for an HP Series IIIsi laser printer in PCI mode is copied. You will select the driver for printing, create and display a bar chart, change the graphic defaults for the bar chart, and print the bar chart from the Graph window. You also print the bar chart directly to the printer without first displaying the chart in the Graph window.

Note: You do not need to set up your device drivers every time you print a graphic. △

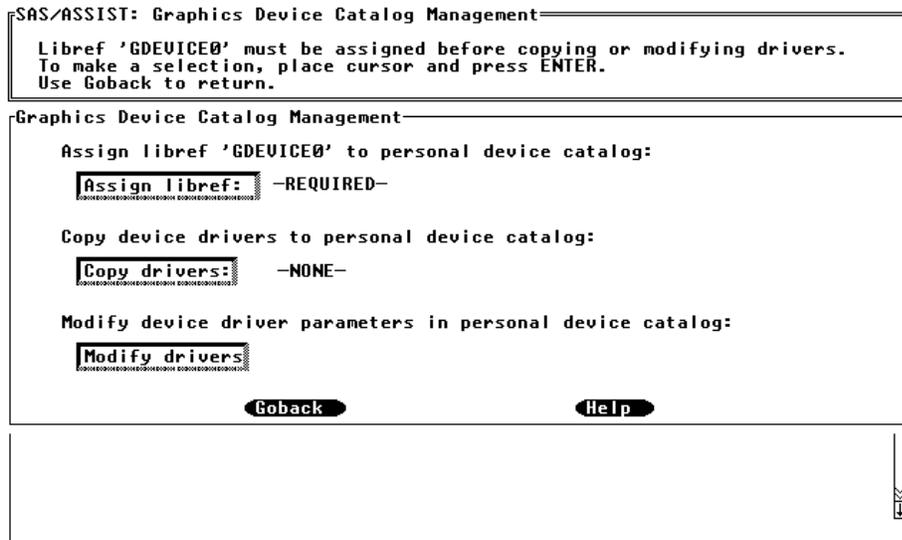
In this section, the HP LaserJet Series IIIsi laser printer is modified for use under the z/OS operating environment. For information on other printers and operating environments, refer to the SAS companion documentation for your operating environment and the documentation for your printer.

Modifying Device Drivers

- 1 Select **Tasks ► Setup ► Environment ► Graphics options... ► Catalog management...**

The Graphics Device Catalog Management window appears as shown in the following display.

Display 4.1 Graphics Device Catalog Management Window

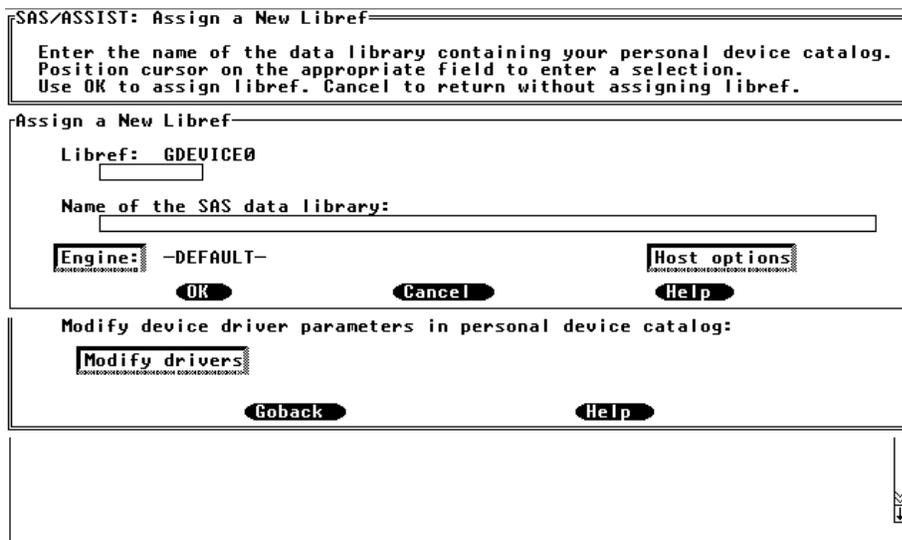


The Catalog Management window enables you to create a personal catalog of graphics device drivers. You can use this feature to select the graphics devices you want from the device catalog that SAS supplies and store them in a personal device catalog. You can then modify these drivers to customize them for your use. If you set up a personal graphics device driver catalog, when you select an item that lists graphics devices, only the device drivers in your personal catalog appear.

2 Select **Assign libref**.

The Assign a New Libref window appears as shown in the following display.

Display 4.2 Assign a New Libref Window



3 Type the name of the SAS data library in which you want to store your graphics devices in the **Name of the SAS data library** field. The format for typing in the library name is determined by your operating environment.

This example uses **USERID.SAS.GRPHDEVS** as the SAS data library name under the z/OS operating environment. Examples of SAS data libraries for various operating environments are listed in the following table.

Table 4.1 Examples of SAS Data Libraries

Operating Environment	What You Type	Example
z/OS	an operating system data set name	USERID.SAS.GRPHDEVS
OpenVMS	the name of a device and directory.	DEVICE:[GRPHDEVS]
UNIX	the name of a directory or subdirectory	/u/userid/grphdevs
Windows	the name of a drive and directory or subdirectory	C:\GRPHDEVS

4 Select **OK**. A message appears indicating the libref was assigned successfully.

If the SAS data library does not exist, one of the following occurred, depending on your operating environment. See the SAS companion documentation for your operating environment for more information.

- Under some operating environments, such as UNIX, the directory must exist or you must first create it before you copy device drivers to it. If the directory does not exist, an error message appears indicating that the SAS data library does not exist.
- Under some operating environments, such as z/OS, a prompt appears asking if you want to create the SAS data library you typed in the **Name of the SAS data library** field. Type **Y** to create the library. The operating environment may ask you if you want to delete or catalog the library once it has been created. Type **C** to catalog the library.

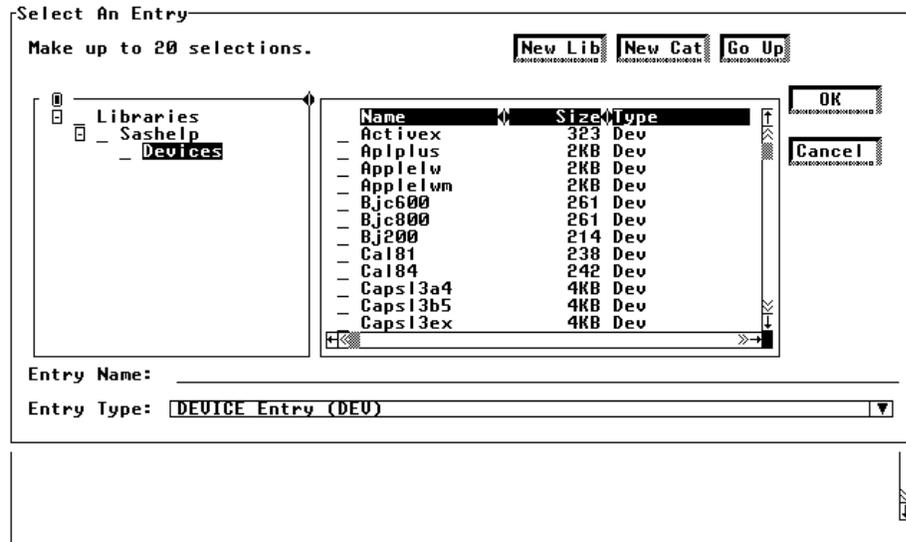
You assign the libref GDEVICE0 to your personal graphics device driver catalog so that the devices displayed for you to choose from are those in your personal device driver catalog. Since the default libref is GDEVICE0, you cannot type a libref in the **Libref** field.

After you have assigned the GDEVICE0 libref, you can copy device drivers from the catalog supplied by SAS into your personal graphics device driver catalog.

5 Select **Goback** to return to the Graphics Device Catalog Management window.

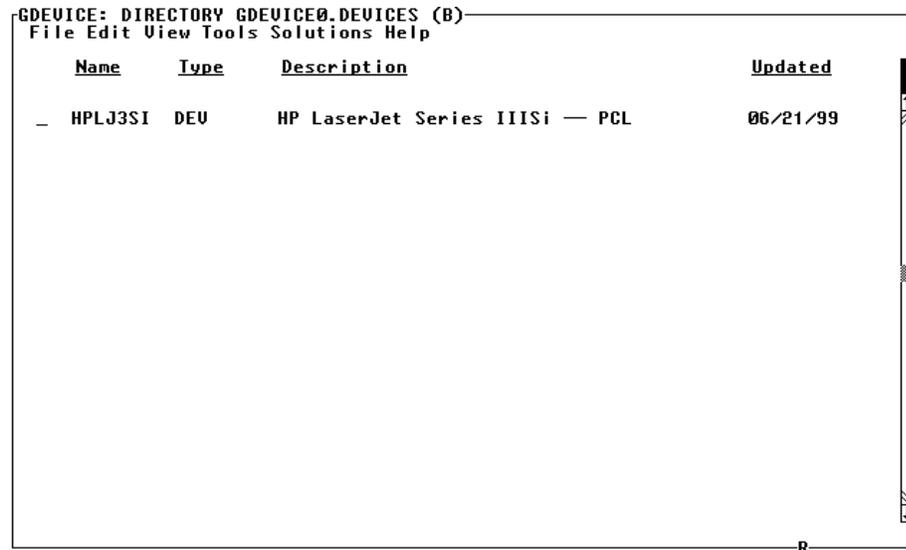
6 Select **Copy drivers**. The Select an Entry window appears, as shown in the following display, with a list of device drivers that you can choose from for your personal graphics devices catalog.

Display 4.3 Catalog Directory Window



- 7 Scroll to the right to display the desired driver and select it.
- 8 Select **OK** to return to the Graphics Device Catalog Management window.
- 9 Select **Modify drivers**; the graphics devices Directory window appears, as shown in the following display. In this example, the driver for an HP Laserjet Series IIIsi was copied.

Display 4.4 Graphics Devices Directory Window



- 10 Type **s** in the **CMD** selection field to the left of the desired driver to select it and press ENTER.

The Detail window appears as shown in the following display.

Display 4.5 Detail Window

```

GDEVICE: Detail
File Edit View Tools Solutions Help
Catalog: GDEVICE0.DEVICES                      Entry: HPLJ3SI

Orig Driver: HPLJ3SI      Module: SASGDHPJ      Model: 3018
Description: HP LaserJet Series IIISI — PCL      Type: PRINTER

Lrows: 0   Xmax: 8.500 IN   Hsize: 8.000 IN   Xpixels: 2550
Lcols: 0   Ymax: 11.000 IN  Usize: 10.000 IN  Ypixels: 3300
Prows: 66  Horigin: 0.250 IN
Pcols: 85  Uorigin: 0.500 IN

Aspect: 0.000      Rotate: _____
Driver query: Y    Queued messages: N
                  Paperfeed: 0.000 IN

```

```

S HPLJ3SI DEV HP LaserJet Series IIISI — PCL 06/21/99
- HPLJ3SIM DEV HP LaserJet Series IIISI—A4 size paper 06/21/99
- HPLJ300 DEV HP LaserJet Series II (1 meg)—300 DPI 06/21/99
- HPLJ300S DEV HP LaserJet Series II—300 DPI, 4 x 5 in 06/21/99
- HPLJ4SI DEV HP LaserJet IV — 600 dpi 06/21/99
- HPLJ5P2 DEV HP LaserJet—8 x 10, 150 DPI 06/21/99
- HPLJ5P3 DEV HP LaserJet—4 x 5, 300 DPI 06/21/99
- HPLJ75 DEV HP LaserJet Series II—75 DPI 06/21/99
- HPPJXL DEV HP PaintJet XL (180 dpi) 06/21/99

```

Note: To display a list of commands that can be typed in the **CMD** selection field next to each graphics device, type ? in the selection field and press ENTER. △

11 Select **Host File Options Window** from the **Tools** menu. The Host File Options window appears as shown in the following display.

Display 4.6 Host File Options Window

```

GDEVICE: Host File Options
File Edit View Tools Solutions Help
Catalog: GDEVICE0.DEVICES                      Entry: HPLJ3SI

Gaccess: GSASFILE

Gsfname: _____      Gsfmode: PORT          Gsflen: 4096
Trantab: _____      Devmap: _____      Devtype: DISK

Gprotocol: SASGPASC

Host file options:
_____
_____
 Close file at end of driver or procedure termination
 Close file at end of each graph

```

```

- HPLJ300S DEV HP LaserJet Series II—300 DPI, 4 x 5 in 06/21/99
- HPLJ4SI DEV HP LaserJet IV — 600 dpi 06/21/99
- HPLJ5P2 DEV HP LaserJet—8 x 10, 150 DPI 06/21/99
- HPLJ5P3 DEV HP LaserJet—4 x 5, 300 DPI 06/21/99
- HPLJ75 DEV HP LaserJet Series II—75 DPI 06/21/99
- HPPJXL DEV HP PaintJet XL (180 dpi) 06/21/99

```

Note: You can change the values in the Host File Options window on all operating environments. However, most of the changes listed in this section are specific to the z/OS operating environment. △

12 Modify the options in the Host File Options window as needed.

- The **Gaccess** field specifies the format and destination of graphics data written to a device or graphics stream file (GSF).

- The **Gsfmode** field specifies the disposition of records written to a graphics stream file or to a device or communications port by the device driver.
- The **Gsflen** field controls the length of the records written to the GSF.
- The **Devtype** field specifies the information required by SAS/GRAPH routines to determine the nature of the output device.
- The **Gprotocol** field specifies the protocol module to use when routing output directly to a printer or creating the GSF to send to a device attached to your host by a protocol converter.

13 Select **Close file at end of each graph** to send the graphic to the printer as soon as you choose to print it, not when the graph procedure ends.

For more information on other printers and operating environments, refer to the SAS companion documentation for your operating environment and the documentation for your printer.

14 Select **File** from the menu bar, then select **Close**. A message appears asking if you want to modify the driver supplied by SAS.

15 Select **Yes** since the driver you are modifying is the device driver that you copied into your personal driver catalog. Press ENTER.

The Detail window appears again.

Under some operating environments, you may need to enter additional information using the Host Commands window. Select **Tools** from the menu bar of the Detail window to access the Host Commands window. Select **Host Commands Window**; the Host Commands window appears as shown in the following display.

Display 4.7 Host Commands Window

```

GDEVICE: Host Commands
File Edit View Tools Solutions Help
          Catalog: GDEVICE0.DEVICES      Entry: HPLJ3SI

Driver Initialization:
 1 _____
 2 _____

Pre-Graph commands:
 1 _____
 2 _____

Post-Graph commands:
 1 _____
 2 _____

Driver Termination:
 1 _____
 2 _____

- HPLJS3  DEU    HP LaserJet Series III (1 meg)—300 DPI  06/21/99
- HPLJS3A4 DEU   HP LaserJet Series III—A4 size paper  06/21/99
- HPLJ0   DEU    HP LaserJet—8 x 10, 75 DPI             06/21/99
- HPLJ100 DEU    HP LaserJet Series II—100 DPI           06/21/99
- HPLJ150 DEU    HP LaserJet Series II—150 DPI           06/21/99
S HPLJ3SI DEU    HP LaserJet Series IIISi — PCL         06/21/99
    
```

The Host Commands window stores the host commands issued at driver initialization, before and after each display is produced, and at driver termination. These commands are typically used to send graphics output to a hardcopy device, such as a printer. You can specify multiple commands for each field. The available commands and their delimiters are host dependent. There are four fields in the Host Commands window:

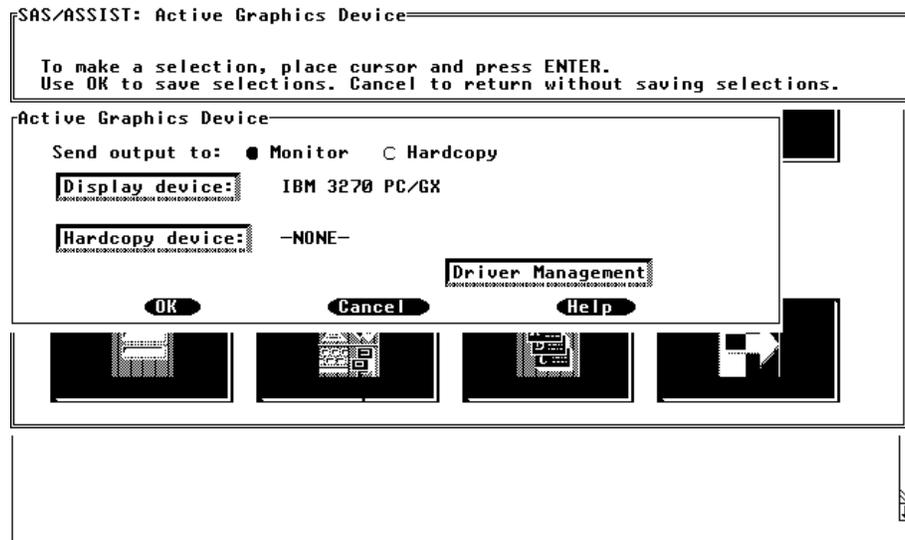
- The **Driver Initialization** field specifies the commands to issue each time the device driver is initialized.

- The **Pre-Graph commands** field specifies the commands to issue before each display is produced.
- The **Post-Graph commands** field specifies the commands to issue after each display is produced.
- The **Driver Termination** field specifies the commands to issue each time the device driver terminates.

For more information on the Host Commands window and commands in specific operating environments, refer to the SAS companion documentation for your operating environment and the SAS companion documentation for your printer.

- 16 Select **Close** from the **File** menu to return to the Detail window from the Host Commands window.
- 17 Select **File** and **Close** twice from the Detail window, then select **Goback** to return to the WorkPlace menu.
- 18 Select **Tasks ► Setup ► Environment ► Graphics options... ► Active graphics device...**The Active Graphics Device window appears as shown in the following display.

Display 4.8 Active Graphics Device Window



- 19 Select **Hardcopy device**; the Device Types window appears.
- 20 Select **Printers...**

The Devices window appears with a list of the devices in your personal device catalog, as shown in the following display.

Display 4.9 Devices Window



21 Select the desired driver; the Active Graphics Device window reappears.

22 Select **OK** to return to the WorkPlace menu.

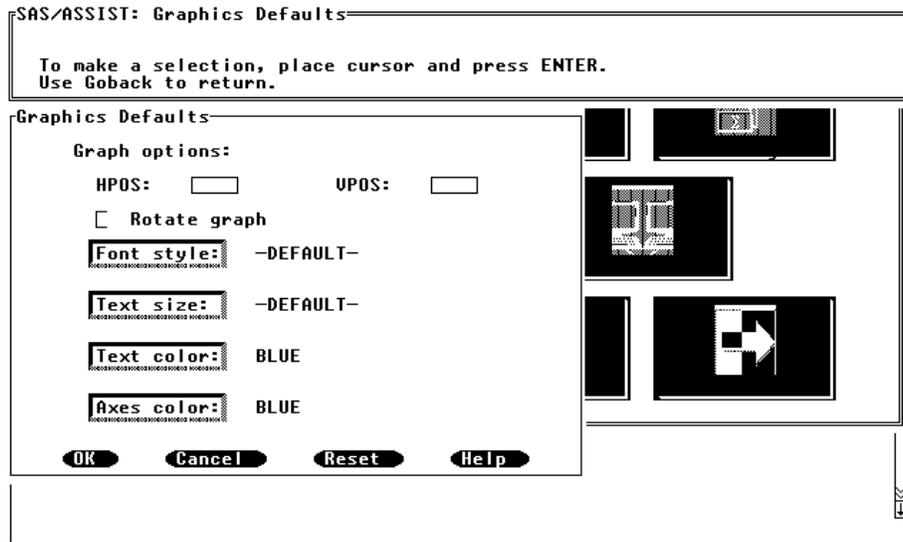
You do not need to set up your device drivers every time you print a graphic. The graphics device that you select remains active until you change it, even when you exit your SAS session. However, to make sure the list of drivers in your personal device catalog is always displayed when you select an item that requests a driver, you need to have the GDEVICE0 libref assigned to your personal device catalog. You can assign the GDEVICE0 libref each time you use SAS/ASSIST software by selecting **Assign libref** from the Graphics Device Catalog Management window or you can assign the libref in your SAS autoexec file so the GDEVICE0 libref is assigned automatically each time you invoke SAS. For more information on assigning librefs in your SAS autoexec file, refer to the SAS companion documentation for your operating environment.

Modifying Graphics Device Defaults

Depending on your display device (monitor) and hardcopy device (printer), you may need to select colors other than the default of blue to view and print your graphic. Follow these instructions to complete that function:

- 1 Follow this path to change the text and axes colors for a bar chart from the default of blue: **Tasks** \blacktriangleright **Setup** \blacktriangleright **Environment** \blacktriangleright **Graphics options...** \blacktriangleright **Graphic defaults...** The Graphics Defaults window appears as shown in the following display.

Display 4.10 Graphics Defaults Window



- 2 Select **Text color** to display a list of valid colors for the text of your graph and select **Axes color** to display a list of valid colors for the axes of your graph. Select the color you prefer to change the text and axes color of your graph.
- 3 Select **Font style** to display a list of valid fonts for the text of your graph. Select the font you prefer to change the font style of your graph. If no font is specified, the hardware font of the graphics device is used.
- 4 Select **Text size** to display a list of valid heights for the text of your graph. Select the height you prefer to change the text size of your graph.
- 5 Type a value for **HPOS** and **VPOS**. **HPOS** specifies the number of columns in your graphics display area. **VPOS** specifies the number of rows in your graphics display area.
- 6 Select **Rotate graph** if you want to rotate the graph by 90 degrees.

These changes are reflected the next time you run a report.

Printing with Modified Drivers

- 1 Select the modified driver as the active graphics device.
- 2 Select **Print...** from the **File** menu in the Graph window.
- 3 Select **OK**. The graphic prints using the modified driver that you copied, modified, and selected as the active printer device.
- 4 Select **Close** from the **File** menu to return to the previous window.

If a hardcopy device was selected along with a monitor, the graphic in the Graph window uses the hardcopy device (HP LaserJet Series IIIsi) parameters, not the display device parameters.

If you do not select a graphics device for **Hardcopy device** in the Active Graphics Device window, and you select **Print** from the **File** menu after the graphic appears in the Graph window, a window appears asking you to type your hardcopy device name.

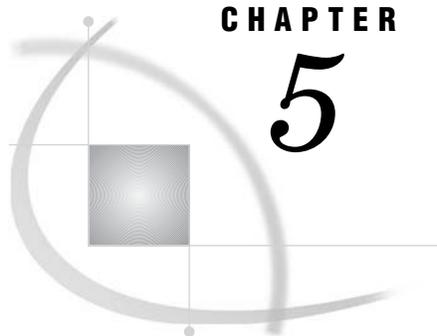
Type the name of your device and select **OK** to print your graph.

If a hardcopy device was not selected, the graph in the Graph window uses the display device parameters, not the hardcopy device parameters.

If you only select a hardcopy device along with the **Hardcopy** item in the Active Graphics Device window, the graphic prints directly to the hardcopy device. The graphic does not appear in the Graph window, and your current window remains displayed. Printing directly to the printer is illustrated in the next section.

Printing Directly to the Printer

- 1 Select **Active graphics device...** from the **Graphics options...** menu to print the graphic directly to the hardcopy device without displaying the graphic in the Graph window. The Active Graphics Device window appears.
- 2 Select **Hardcopy** for the **Send output to** field.
- 3 Select **Hardcopy device**; the Device Types window appears.
- 4 Select **Printers....**
The Devices window appears with a list of the devices in your personal device catalog.
- 5 Select the desired driver; the Active Graphics Device window reappears.
- 6 Select **OK**, then **Run** to print the graphic directly to the printer and remain at your current window.



CHAPTER

5

Setting Up Query Managers

<i>Introduction</i>	32
<i>Overview</i>	32
<i>Defining and Using Query Managers</i>	33
<i>Defining and Using a Public Query Manager</i>	35
<i>Defining and Using a SAS Query Manager</i>	35
<i>Defining and Using a DB2 Query Manager</i>	35
<i>Creating Master DB2 Query Manager Tables</i>	37
<i>Creating Public SAS Query Manager Tables</i>	38
<i>Creating a Query Manager Definition</i>	40
<i>Create a Storage Location for the Query Manager Definition</i>	40
<i>Point to the Query Manager Definition with Profiles</i>	40
<i>Create the Public Query Manager Definition</i>	42
<i>Using Initiation and Termination Programs</i>	44
<i>Using the Query Manager Profile</i>	46
<i>Save the Query Manager Definition</i>	47
<i>Selecting a Query Manager</i>	47
<i>Updating a Query Manager</i>	47
<i>Defining and Using a Personal SAS Query Manager</i>	47
<i>Editing the Default Personal SAS Query Manager</i>	48
<i>Creating Dynamic SAS Query Manager Views</i>	50
<i>Creating a New Dynamic, Personal SAS Query Manager</i>	51
<i>Selecting a Personal SAS Query Manager</i>	52
<i>Updating a Personal SAS Query Manager</i>	53
<i>Deleting Query Managers</i>	54
<i>Creating SAS Query Manager Tables</i>	55
<i>_SACOLS Table</i>	55
<i>_SAINDX Table</i>	57
<i>_SARELS Table</i>	58
<i>_SATAB Table</i>	59
<i>Creating DB2 Query Manager Tables</i>	60
<i>TABAUTH Table</i>	61
<i>_DB2COLS Table</i>	62
<i>_DB2INDX Table</i>	64
<i>_DB2RELS Table</i>	65
<i>_DB2TAB Table</i>	66

Introduction

This chapter describes types of Query Managers available in SAS/ASSIST. Included are examples of different types of Query Managers and a discussion on creating Query Manager tables.

Overview

The Query Manager is the link between a query and the data to be queried. It specifies the location of the data and the tables that describe the structure of the data. It also specifies the options and SAS statements that are activated when the Query Manager is used. A Query Manager must be set up to query data. By default, if no Query Manager is defined, a personal SAS Query Manager is created the first time you access the Query window. A user, SAS administrator, or database administrator (DBA) can set up Query Managers.

If Master and Group profiles are set up at your site, you can share Query Managers with a group of users or with the entire site. You can specify the location of the Query Managers in Master, Group, and User profiles. System administrators can restrict users from overwriting the location of the Query Managers that are specified in their User profile. For additional information on profiles, see Chapter 3, “Setting Up and Customizing Master and Group Profiles,” on page 7.

A Query Manager consists of definitions and tables. Query Manager definitions specify

- the type of data (for SAS data or, on z/OS, DB2 data)
- the name of a program that creates or updates the Query Manager tables
- the location of the Query Manager tables
- whether the Query Manager tables are dynamic (views) or static (tables)
- whether the user is allowed to save the relations being used when tables are joined
- any SAS statements that are submitted when the Query Manager is activated and deactivated
- user profile options that are set when the Query Manager is activated.

Query Managers are based on SAS tables that contain information about the structure of the data tables to be accessed. These SAS tables are generated or updated by SAS programs that retrieve the information either from SAS dictionary tables or, for DB2 on z/OS, the DB2 systems catalog. The SAS programs can be modified to retrieve information from external files or external data dictionaries. Storing the data table structure information in SAS tables has the following advantages:

- You can access the table structure information from SAS quickly.
- Only limited interaction with the SAS dictionary tables or DB2 systems catalog is needed.
- Only relevant information is displayed.
- You can customize and extend the information that is available.
- You can restrict READ access to DB2 catalogs.

One program is provided to create SAS Query Manager tables. Five programs are provided to create five different types of DB2 Query Manager tables. These programs are run from Query and Reporting. They also can be run from outside Query and Reporting, either from the Program Editor window or through batch processing.

The four SAS data files that are created for the SAS Query Manager are:

_SACOLS

This table contains information about each column in the available SAS tables.

_SAINDX

This table contains information about the indexes that are defined on columns in tables in the `_SACOLS` table.

_SARELS table

This table contains relations between tables.

_SATAB

This table contains information about each SAS table or SAS view that can be queried with the Query Manager.

The four SAS data files that are created for the DB2 Query Manager are:

_DB2COLS

This table contains information about every DB2 column in the `_DB2TAB` table.

_DB2INDX

This table contains information about indexes that are defined on columns in tables in the `_DB2COLS` table.

_DB2RELS

This table contains primary key/foreign key relations that are defined between DB2 tables.

_DB2TAB

This table contains information about every DB2 table or view that the user can query.

See “Creating SAS Query Manager Tables” on page 55 for a detailed description of the Query Manager tables and the programs that create them. In that section, you will also find information on modifying the programs that create the Query Manager tables so that information can be retrieved from other sources.

Defining and Using Query Managers

You can define the following types of Query Managers:

personal Query Manager

Use a personal Query Manager to define the SAS or DB2 data that you want to query. If no Query Manager has been defined, then a personal SAS Query Manager is created the first time you enter the Query and Reporting Query window. Personal Query Managers also are called user Query Managers.

public Query Manager

Use a public Query Manager if you want to enable a group of users to have the same type of access to the same SAS data libraries or DB2 tables. A public Query Manager can be available either to all users (Master Query Manager) or to a group of users (Group Query Manager).

dynamic Query Manager

Use a dynamic Query Manager if you want your queries to always reflect the most current structure of the SAS or DB2 tables to which you have access in your SAS session. For example, if you add new tables (SAS tables) to your SAS data libraries or add new columns to your tables, these changes are automatically

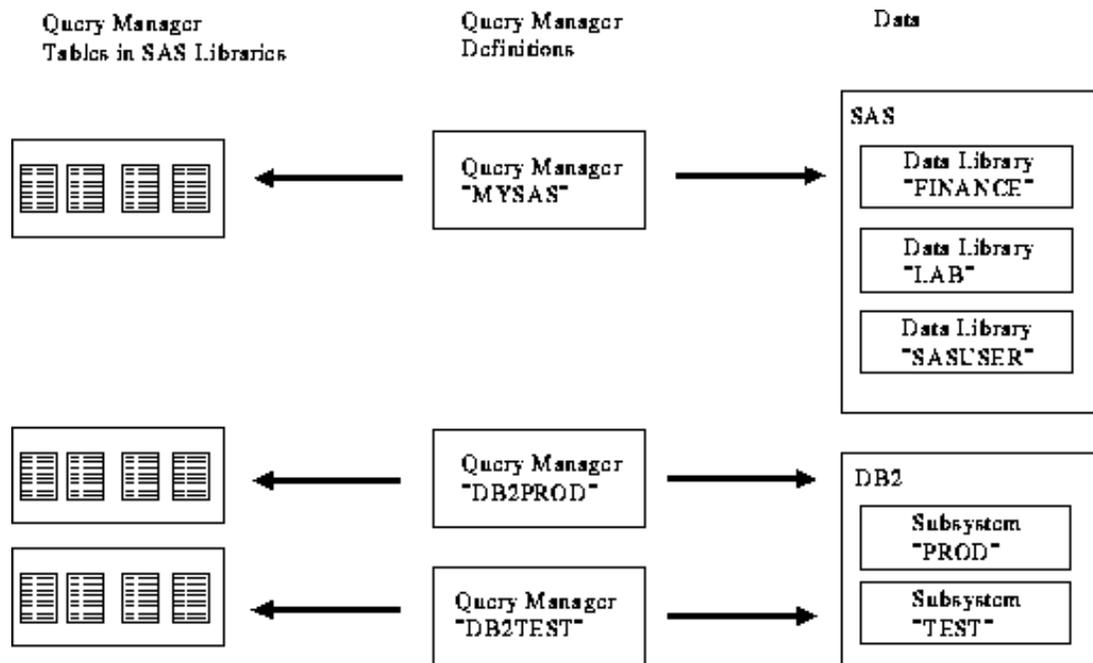
reflected by the dynamic Query Manager. Information about the structure of the tables is displayed more slowly when you use a dynamic Query Manager.

static Query Manager

Use a static Query Manager when you want to access information quickly about the structure of your SAS or DB2 tables. A static Query Manager provides fast access to the Query Manager tables. However, a static Query Manager requires that you manually update the Query Manager tables when the structure of the tables changes, for example, when new tables (SAS tables) are added to your SAS data libraries or when new columns are added to the tables. A static Query Manager is the default Query Manager.

You can combine the various types of Query Managers. For example, you can create a static, public SAS Query Manager or a dynamic, personal DB2 Query Manager. The following display shows an example of a setup for a user who has three Query Managers that give access to three SAS data libraries and two different DB2 subsystems.

Figure 5.1 Query Manager Examples



This chapter describes how to set up two Query Managers. Not all possible types of Query Managers are described. The following table shows which Query Manager features are used in which examples. Each example is a complete description of the setup for a Query Manager.

Table 5.1 Query Manager Examples

	Personal SAS Query Manager Example	Public SAS Query Manager Example
Master/Group Profile		X
Database SAS	X	X
Database DB2		X

	Personal SAS Query Manager Example	Public SAS Query Manager Example
Update tables using Update Query Manager	X	X
Update tables from Program Editor window or in batch		X
Tables' location specified by libref	X	X
Tables' location specified by physical file name		X
Static Query Manager	X	X
Dynamic Query Manager	X	
Allow update of relations from Join window	X	
Do not allow update of relations from Join window		X
Use Initiation Program		X
Use Termination Program		X
Use Generation Program		X
Use Query Manager profile		X

Defining and Using a Public Query Manager

If you are a SAS administrator or a database administrator (DBA), you can create a Group or Master Query Manager. Group and Master Query Managers give a group of users or all users the same type of access to the same data libraries through Query and Reporting. This section shows you how to create a static, Group or Master SAS Query Manager.

Defining and Using a SAS Query Manager

The Query Manager tables for this Query Manager can be updated by submitting the table creation program from the Program Editor window or through batch processing.

To create a public Query Manager, complete the following steps:

- 1 create Query Manager tables in a public SAS data library that users can access.
- 2 create Query Manager definitions and save them in a SAS data library. Users must have access to this library. Make sure that the Group or Master profile references the SAS data library that contains the Query Manager definitions. Master and Group profiles are optional. For additional information about creating profiles, see Chapter 3, "Setting Up and Customizing Master and Group Profiles," on page 7.

Defining and Using a DB2 Query Manager

A DBA must create the DB2 Query Manager because READ access to the DB2 systems catalog is required to run the table creation programs. The subsets of the DB2 Query Manager tables can be updated by using **Update Query Manager** from the **Tools** menu of the Select window. The master DB2 Query Manager tables for this

Query Manager can be updated by submitting the table creation program again from the Program Editor window or through batch processing.

Depending on the number of DB2 tables and your site's policy for accessing data, choose one of the following strategies.

- Let all users run their queries using the master Query Manager tables.
- Set up group Query Manager tables for groups of users.
- Let users have their own personal Query Manager tables.
- Use a combination of the above strategies.

The strategy you choose may depend on whether users should see only the tables that they actually can query. It also depends on whether or not users create their own tables and how often. You can easily change data access by making changes to the Master or Group profiles. This section shows how to generate a set of Query Manager tables that contain only the tables to which a user has READ access.

A two-step process is used to create the DB2 Query Manager tables:

- 1 Access the DB2 systems catalog and create a master set of the Query Manager tables in SAS. By default, these master tables contain information about all the tables in a DB2 subsystem and information about table authorizations. Optionally, create a subset of the master tables.
- 2 Create Query Manager definitions and make them available to all users or to groups of users. The Query Manager definitions supply all or a subset of the master tables to the users.

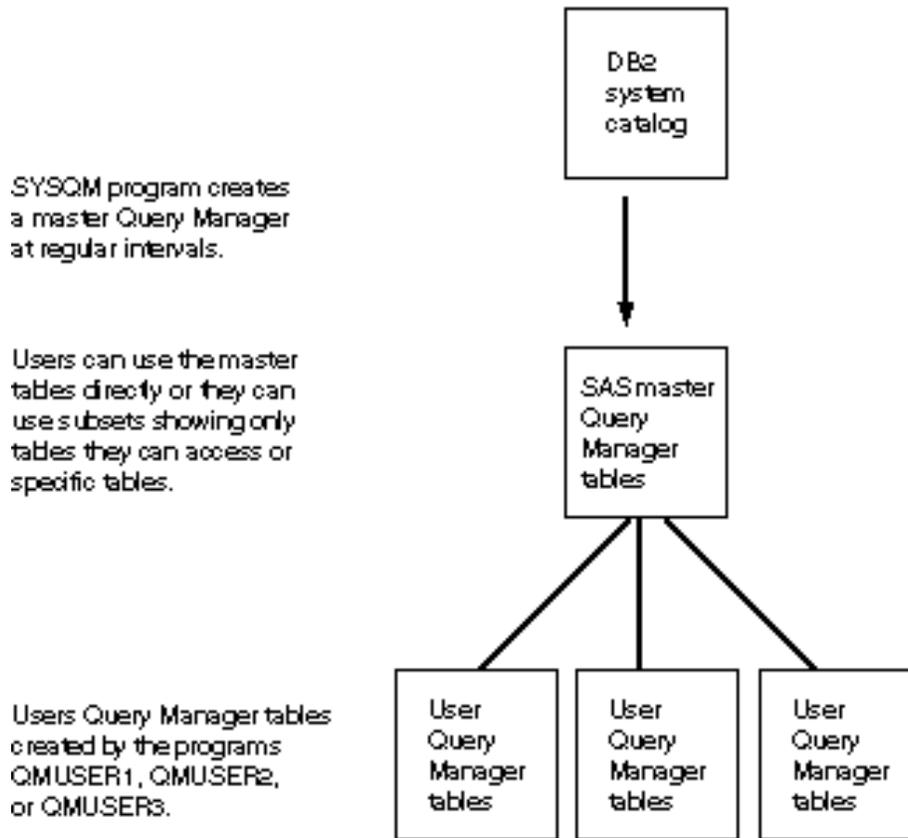
The program that generates the master set of Query Manager tables is located in SASHELP.QASSIST.SYSQM.SOURCE. This program is intended to be run as a batch job by the DBA. However, it also can be run from the Program Editor window. Running this program requires READ access to the tables in the DB2 systems catalog.

All users can use the master Query Manager tables. However, it is more efficient for users or groups of users to use subsets of these tables for their own Query Managers. The following three programs are supplied with Query and Reporting to create subsets of the master DB2 Query Manager tables.

- SASHELP.QASSIST.QMUSER1.SOURCE creates a subset for a specified set of creators.
- SASHELP.QASSIST.QMUSER2.SOURCE creates a subset based on the tables to which the user is granted READ access.
- SASHELP.QASSIST.QMUSER3.SOURCE creates a subset based on a creator table that contains all creators whose tables are contained in the subset.

The SASHELP.QASSIST.DB2DYNA.SOURCE program creates a set of dynamic DB2 Query Manager tables that read from the DB2 systems catalog directly. A dynamic DB2 Query Manager is useful if the structure of the DB2 tables changes frequently or if DB2 tables are added and deleted frequently. Using a dynamic Query Manager on a large DB2 system is considerably slower than using a static one because the DB2 catalog must be read at the time a user chooses the Select window. Individual users who use a dynamic DB2 Query Manager must be granted SELECT privileges on the DB2 catalog tables discussed in Chapter 5, "Setting Up Query Managers," on page 31.

The following figure shows an example of a table setup for a DB2 Query Manager. You may want to create multiple DB2 Query Manager definitions. For example, you may have one Query Manager definition that points to your production DB2 subsystem and another Query Manager definition that points to your test subsystem. Both Query Manager definitions can refer to the same set of Query Manager tables if the two subsystems' data definitions are the same.

Figure 5.2 Example DB2 Query Manager Setup

Creating Master DB2 Query Manager Tables

To create the master DB2 Query Manager tables, you must be granted READ access to the DB2 systems catalog. Allocate a SAS data library to store the master Query Manager tables. Give the library a libref of SYSQM, for example:

```
libname sysqm 'userid.dbquery.tables';
```

Use the program SASHELP.QASSIST.SYSQM.SOURCE to create the tables. You can run this program in batch or from the Program Editor window as follows:

- In batch, modify the program by using the JCL statements that are necessary to run the program in batch at your site. Include a DDNAME called SYSQM that references the SAS data library you created to store the Query Manager tables. Submit the job.
- Copy the program SASHELP.QASSIST.SYSQM.SOURCE into the Program Editor window, as shown in the following display.

Display 5.1 Editing Source Statements to Create Master DB2 Query Manager Tables

```

Program Editor
File Edit View Tools Run Solutions Help
00002 /*+-----+
00003 | Name : SASHELP.QASSIST.SYSQM.SOURCE
00004 | Purpose: Create site System Query Manager.
00005 | Freq. : Should run daily at off-peak hours.
00006 | Product: SAS/ASSIST: Query & Reporting From DB2.
00007
00008 | Note : All though this program can be run in foreground
00009 | it is highly recommended to run it as a batchjob.
00010 | This program should be run at off-peak hours
00011 | only, to prevent deadlocks on the SYSIBM-tables.
00012 | This program requires write access to the
00013 | system datasets, and should therefore be run
00014 | by the DBA/SAS-administrator.
00015
00016 | Usage : A SYSQM libref is required.
00017
00018 | In batch:
00019 | //SYSQM DD DSN=xxx.yyy.zzz,DISP=OLD
00020
00021 | In foreground:
00022 | libname SYSQM 'xxx.yyy.zzz';
00023 +-----+*/
00024
00025 /* START site-specific changes */
00026
00027 %let ssid = xxxx; /* DB2 Subsystem ID */
00028

```

At the beginning of the program, add a LIBNAME statement that references the library you created to store the Query Manager tables. Use the required libref SYSQM, for example,

```
libname sysqm 'userid.dbquery.tables';
```

In the first %LET statement, edit the DB2 Subsystem ID to the appropriate SSID. Submit the job for processing.

For DB2 Query Managers, three programs are supplied with Query and Reporting that create subsets of the master Query Manager tables:

- SASHELP.QASSIST.QMUSER1.SOURCE
- SASHELP.QASSIST.QMUSER2.SOURCE
- SASHELP.QASSIST.QMUSER3.SOURCE

See “Creating DB2 Query Manager Tables” on page 60 for more information about these programs.

To create a subset based on the tables to which the user is granted READ access, copy SASHELP.QASSIST.QMUSER2.SOURCE into the Program Editor window. Save the program to a library to which users have READ access. You can edit the program using **Generation Program** from the **Tools** menu of the Query Manager Administration window as described in “Create the Public Query Manager Definition” on page 42.

Creating Public SAS Query Manager Tables

To create Query Manager tables for a public SAS Query Manager, submit a LIBNAME statement from the Program Editor window that references the SAS data library to contain the Query Manager tables. This library must not contain Query Manager tables for any other Query Manager definition.

Submit a LIBNAME statement for each SAS data library that you want to make available to other users, as in the following examples. When you create the Query Manager tables, these SAS libraries are included in the tables. Make sure that these libraries are allocated each time the Query Manager is used. You can allocate the libraries in an AUTOEXEC.SAS file, by using the Initiation and Termination programs, or by using a SAS/ASSIST start program. For additional information, see “Using

Initiation and Termination Programs” on page 44 of this document and “Using Logon/Logoff Exits and an Alternate Main Menu” in Chapter 12, “Doing More with Setup,” in *Doing More with SAS/ASSIST*.

Copy the SASHELP.QASSIST.SASQM.SOURCE program into the Program Editor window. This program creates the public Query Manager tables. Edit selected program statements for your site’s use or for a group’s use as follows:

- Uncomment the first %LET statement. Change the assignment of the macro variable QMLIB (currently set to **WORK**) to the libref that points to the SAS data library to contain the public Query Manager tables. For example, change the statement to `%let qmlib=mgr;`
- Uncomment and change the assignment of the macro variable EXCLLIB so that all the SAS data libraries you want to exclude from the public Query Manager, if any, are listed.

The following display shows the edited SASQM.SOURCE program. The libref MGR is assigned to the macro variable QMLIB. The macro variable EXCLLIB points to the libraries SASHELP and MAPS by default. This example adds SASUSER, WORK, MGR, and LIBRARY to the libraries to be excluded from the Query Manager.

Display 5.2 Editing Source Statements to Create Public Query Manager Tables

```

Program Editor
File Edit View Tools Run Solutions Help
00001  /*+-----+
00002  | Name   : SASHELP.QASSIST.SASQM.SOURCE
00003  | Purpose: Create Query Manager tables on SAS tables
00004  | Product: SAS/ASSIST: Query & Reporting
00005
00006  | Note   : This program creates Query manager tables
00007  |         on SAS tables.
00008  |         To define the library for Query manager tables
00009  |         set the macro variable: qmlib
00010  |         To exclude libraries in the query manager tables
00011  |         set the macro variable: excllib
00012  |-----+
00013
00014  /*+-----+
00015
00016
00017
00018  %let qmlib=mgr;
00019  /* Output library for query manager tables          */
00020
00021  %let excllib=('SASHELP','MAPS', 'SASUSER', 'WORK', 'MGR','LIBRARY');
00022  /* Libraries to be excluded in query manager tables */
00023  /* To select all:                                   */
00024  /* %let excllib=('');                               */
00025
00026
00027  %macro upc(macvar,value);

```

Submit the program to generate the Query Manager tables.

Note: Save this program and re-submit it periodically to update the Query Manager tables. You can use the Program Editor window or batch processing to re-submit this program. For example, you may want to batch submit this job every night to make sure that users have updated table information in the morning. Δ

To help users join SAS tables, a SAS administrator can do one of the following:

- Using SAS/FSP software, edit the _SARELS table in the library that contains the Query Manager tables. Add all the relations so that joins can be made automatically. For further information, see “Creating SAS Query Manager Tables” on page 55 of this document and “Using Logon/Logoff Exits and an Alternate Main Menu” in Chapter 12, “Doing More with Setup,” in *Doing More with SAS/ASSIST*.
- Create a public Query Manager that references the Query Manager tables you have just created. From Query and Reporting, save any joins you want done

automatically. See Chapter 10, “Query and Reporting,” in *Doing More with SAS/ASSIST* for details on saving joins.

Creating a Query Manager Definition

To create a Query Manager definition, follow these steps:

- 1 create a storage location for the Query Manager definition
- 2 point to the Query Manager definition with profiles
- 3 create the Query Manager definition
- 4 save the Query Manager definition.

Create a Storage Location for the Query Manager Definition

Create a SAS data library to store the Master and Group Query Manager definition. Submit a LIBNAME statement that references the SAS data library, as shown in the following example:

```
libname public 'userid.system.public';
```

All users must have READ access to this library.

Point to the Query Manager Definition with Profiles

After you create a place to store the public Query Manager definition, point to it using the Master and Group profiles. Follow this path to open the Master or Group Profile window shown in the following display:

Tasks \blacktriangleright **Setup** \blacktriangleright **Profiles** \blacktriangleright **Master/group...**

Display 5.3 Master/Group Profile Window

SAS/ASSIST: Master/Group Profile			
File View Tools Help			
Name:	MASTER	Updated:	16JUL1999:11:18:23
Description:	Master profile		
Type:	ALL		
Cnd	Option	Value	Status
—	Message window	All	D
—	Permanent task catalog	SASUSER.PRIVATE	D
—	Update data automaticly	Yes	D
—	Save selections on end	Yes	D
—	Confirm changes	No	D
—	Menu style	New	D
—	Alternate menu bar		D
—	SAS/ASSIST style	Workplace	D
—	Show startup dialog	Yes	D
—	SAS/ASSIST start program		D
—	Type of start program	Catalog	D
—	SAS/ASSIST stop program		D
—	Type of stop program	Catalog	D
—	End SAS on exit	No	D
—	Logon - application name		D
—	Primary Menu after Logon	Yes	D
—	Catalog for private menu	SASUSER.PRIVATE	D
—	Use private menu	No	D
—	Private menu type	Block	D
—	Private menu title	PRIVATE MENU	D

By default, all the options for the Master profile are shown. You can edit the fields in this window only if you have WRITE access to the SAS data library that contains the Master profile. If your site has no Master profile, this window is empty.

In the **Type** field, type **Query Manager** and press ENTER. You can also type a ? and press ENTER to display a list of **Type** options. Select **Query Manager** from the list. The

Query Manager options appear. The following display shows the Query Manager profile options.

Display 5.4 Query Manager Profile Options Window

SAS/ASSIST: Master/Group Profile			
File View Tools Help			
Name:	MASTER	Updated:	16JUL1999:11:18:23
Description:	Master profile		
Type:	QUERY MANAGER		
Cmd Option	Value	Status	
— Standard name	SASQM	D	
— Standard catalog	MANAGER	R	
— Standard type	User	D	
— User - library	SASUSER	R	
— Group - path		D	
— Master - path		D	
— Additional information	SASHELP.QASSIST.QSADDINF.PROGRAM	D	

You can create public Query Managers for all users (Master Query Manager) or for a group of users (Group Query Manager).

- If you are creating a public Query Manager for all users, in the **value** field for **Master - path** type the external file name of the SAS data library for the master Query Manager definition. In the previous display, the path for the master Query Manager definition is **userid.system.public**. A libname to the specified path is assigned automatically.

To ensure that the profile change is available to all users, type **R** or **U** in the **Status** field, to set the value to RESTRICT or UPDATE, respectively. Then select **Save** from the **File** menu. See “Assigning and Changing the Status of Profile Options” on page 15 for more information about profile option status.

- If you are creating a public Query Manager for a group of users, open the appropriate Group profile and point the Group profile to the location of the public Query Manager definitions.

To open a Group profile, select **Open** from the **File** menu. The Open Profile window appears.

To select a profile, type **S** in the **Cmd** field and press ENTER. The Master/Group Profile window appears again with the profile information for the chosen group as shown in the following display. The Group profile reflects the changes made to options with a status of **U** in the Master profile. The following display shows the Group profile for the Accounting Group.

Display 5.5 Accounting Group Profile Window

SAS/ASSIST: Master/Group Profile			
File View Tools Help			
Name:	ACCOUNT	Updated:	16JUL1
Description:	Accounting Group Profile		
Type:	QUERY MANAGER		
Cnd Option	Value	Status	
— Standard name	SASQM	D	
— Standard catalog	MANAGER		
— Standard type	User	D	
— User - library	SASUSER		
— Group - path		D	
— Master - path		D	
— Additional information	SASHELP.QASSIST.QSADDINF.PROGRAM	D	

In the **Value** field for **Group - path**, type the external file name of the SAS data library for the public Query Manager definitions. In the previous display, the path for the group Query Manager definitions is **userid.system.public**. A libname to the specified path is assigned automatically.

To ensure that the profile change is available to all users, type **R** or **U** in the **Status** field of the window, to set the value to RESTRICT or UPDATE, respectively. See “Assigning and Changing the Status of Profile Options” on page 15 for more information about profile option status.

Select **Save** from the **File** menu to save your changes. Select **Close** from the **File** menu to return to the WorkPlace menu.

Create the Public Query Manager Definition

After you have set up your Master and Group profiles, you can create the public Query Manager definition. From the Query and Reporting Setup window, choose **Administration**. The Query Manager Administration window appears.

You can change the default settings by typing new values where typing is permitted or by choosing a different selection where two alternatives are listed.

Database

specifies the type of database for which you want to set up a Query Manager. Type the name of the database you want to use or type a **?** in the field and press ENTER to select a database from a list of the database systems that you can access.

Program to generate Query Manager tables

specifies the name of the program that generates and updates the Query Manager tables.

If you select **Yes**, the fields **SAS entry** and **External file** appear. If the program is stored in a catalog entry, select **SAS entry** and type the appropriate four-level SAS name. If the program is stored in an external file, select **External file** and type the physical file name for the program. The external file must already exist.

For DB2 Query Managers, three programs are supplied with Query and Reporting that create subsets of the master Query Manager tables:
 SASHELP.QASSIST.QMUSER1.SOURCE,
 SASHELP.QASSIST.QMUSER2.SOURCE,
 SASHELP.QASSIST.QMUSER3.SOURCE.

You can copy and edit these programs to create subsets of the master DB2 Query Manager tables.

For this example, use the program in PUBPROGS.QMPROGS.QMUSER2.SOURCE. This is the program that you copied and saved in “Defining and Using a Public Query Manager” on page 35. This program generates the subset of the DB2 Query Manager tables.

Select **Generation Program** from the **Tools** menu. The QMUSER2.SOURCE program appears as shown in the following display.

Display 5.6 QMUSER2.SOURCE

```

Program Editor
File Edit View Tools Run Solutions Help
00001  /* Create USER Query Manager, program 2.
00002  Check for table authorization.
00003  NOTE: Only one User Query Manager per LIBREF
00004  LIBREF is defined by macrovariable QMLIB
00005  */
00006  libname sysqm 'licham.db2quwey.table' disp=shr;
00007  %let sysuser = &SYSJOBID;          * Change to user-id if run in bat
00008
00009  /* The following two %LET-stmts are intentionally deactivated, since
00010  parameters by default are set, when running Query & Reporting.
00011  Two activate them, simply move line 14 and 15 outside the comment
00012  and modify them if needed.
00013
00014  %let fysname = xxxxx.yyyyyy.zzzzz; * <— Physical dataset of system
00015  %let qmlib = SASUSER;             * <— User QM library;
00016  */
00017  %let fysname = licham.db2query.tables;
00018  %let qmlib = SASUSER;
00019  libname sysqm "&fysname" disp=shr;
00020
00021  /* Create dataset to hold any error messages */
00022  proc sql;
00023  create table work.__log
00024  (db2rc char(5) label='Return Code',
00025  db2msg1 char(70) label='Message-1',
00026  db2msg2 char(70) label='Message-2',
00027  db2msg3 char(70) label='Message-3',

```

Edit the program following the commented instructions in the program. Choose **Save** from the **File** menu to save the program. Choose **Close** from the **File** menu to return to the Query Manager administration window.

If you supply a program name, you can update the Query Manager tables using **Update Query Manager** from the **Tools** menu of the Select window. See “Defining and Using a Public Query Manager” on page 35 for further information.

If you select **No**, the Query Manager tables cannot be updated using **Update Query Manager** from the **Tools** menu of the Select window. The tables must be updated by batch submitting the program to update the tables or submitting the program from the Program Editor window.

Location of Query Manager tables

specifies the location of the Query Manager tables. You store the Query Manager tables in a SAS data library.

Select **SAS library name** and type the libref or select **Physical file name** and type the external file name of the SAS library. In the previous display, the location of the subsetted DB2 Query Manager tables is SASUSER because this DB2 Query Manager uses the QMUSER2.SOURCE program. This program

generates a set of Query Manager tables containing only the tables to which the user has READ access.

If you use a SAS library name, use **Initiation Program** and **Termination Program** to assign a libref to the SAS library when the Query Manager is used. See “Using Initiation and Termination Programs” on page 44 for more information. Because SASUSER is automatically assigned to your SAS session, you do not need to use the initiation and termination programs to assign the libref.

Query Manager type

For this example, select **Static**. A static Query Manager is based on SAS data files and provides faster access to the Query Manager tables. However you must update the static Query Manager tables when the structure of the tables changes, for example when new tables (SAS data files) are added to your SAS data libraries or when new columns are added to the tables. See “Defining and Using a Public Query Manager” on page 35 for more information.

A dynamic Query Manager is based on PROC SQL views to SAS dictionary tables or, for DB2 on z/OS, system tables. The PROC SQL views supply current information to a dynamic Query Manager each time the Query Manager is referenced. The SASHELP.QASSIST.DB2DYNA.SOURCE program creates a set of dynamic DB2 Query Manager tables that read from the DB2 systems catalog directly. See “Creating Dynamic SAS Query Manager Views” on page 50 and “Creating DB2 Query Manager Tables” on page 60 for more information.

Allow Update of relations

For this example, select **No** so that users can only browse the Query Manager tables from the Join window.

Select **Yes** to allow users to add and delete relations in the Query Manager tables by following this path from the Join Manager window:

View ► Relations ► Current Tables | All Tables

Note: Relations can be updated only when **SAS library name** is selected for **Location of Query Manager tables**. When **Physical file name** is selected for **Location of Query Manager tables**, the library is open for READ access only. Δ

If you select **Yes**, you can save any joins you make in the Join Manager window for future automatic joining of tables. See Chapter 10, “Query and Reporting,” in *Doing More with SAS/ASSIST* for more information.

Using Initiation and Termination Programs

Use **Initiation Program** and **Termination Program** to assign librefs to the SAS libraries that you want users to access when they use the public Query Manager. The SAS libraries are then automatically available to users. You also can use these selections to assign a libref to the location of the Query Manager tables when it is specified with a SAS library name.

Select **Initiation Program** from the **Tools** menu of the Query Manager Administration window. The Initiation Program window appears. The Initiation Program window is an editor in which you can type SAS programming statements that are executed when the Query Manager is activated. For this example, type the LIBNAME statements for each SAS data library you want to make available to other users. Remember to use the same LIBNAME statements that you used when you generated the Query Manager tables. The following display shows the libname statements for the FINANCE and PRODUCTS data libraries.

Display 5.7 Initiation Program Window

```

Edit: Initiation Program
File Edit View Tools Solutions Help
00001 libname finance 'userid.system.finance' disp=shr;
00002 libname products 'userid.system.products' disp=shr;
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
00016
00017
00018
00019
00020
00021
00022
00023
00024
00025

```

Select **Save** from the **File** menu to save the initiation program. Select **Close** from the **File** menu to return to the Query Manager Administration window.

Select **Termination program** from the **Tools** menu. The Termination Program window appears. The Termination Program window is an editor in which you can type SAS programming statements that are executed when the Query Manager is deactivated. For this example, use the Termination Program window to clear the librefs you allocated in the Initiation Program window. The following display shows the LIBNAME statements used to clear the FINANCE and PRODUCTS librefs.

Display 5.8 Termination Program Window

```

Edit: Termination Program
File Edit View Tools Solutions Help
00001 libname finance clear;
00002 libname products clear;
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
00016
00017
00018
00019
00020
00021
00022
00023
00024
00025

```

Select **Save** from the **File** menu to save the termination program. Select **Close** from the **File** menu to return to the Query Manager Administration window.

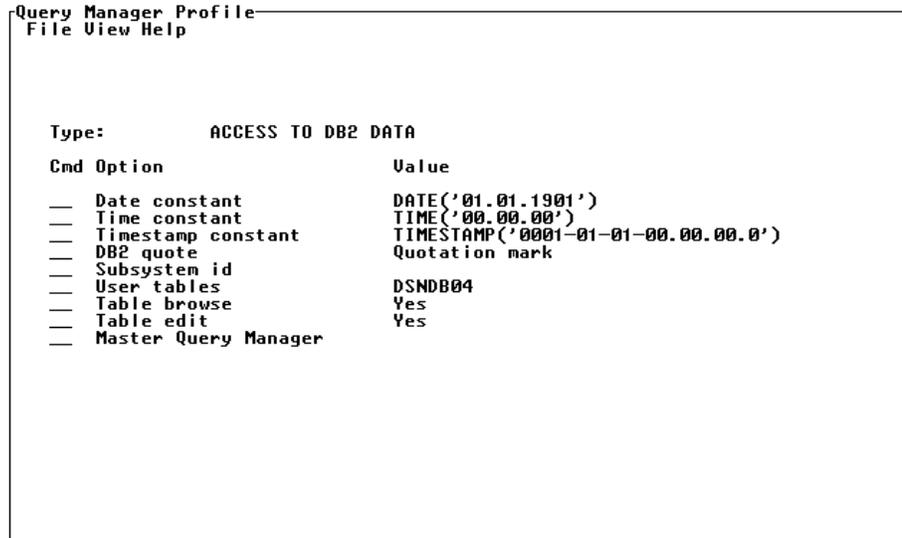
Using the Query Manager Profile

This section is applicable to DB2 Query Managers only.

Use the Query Manager Profile to specify the location of the master DB2 Query Manager tables.

To open the Query Manager Profile window, select **Query Manager Profile** from the **Tools** menu in the Query Manager Administration window. The following display shows the Query Manager Profile window. Make sure that the value for **Type** is **ACCESS TO DB2 DATA**.

Display 5.9 Edited Query Manager Profile Window



In the **Master Query Manager** field, type the physical name of the SAS data library that contains the master DB2 Query Manager tables, for example, `USERID.DB2QUERY.TABLES` as shown in the previous display.

In the **Value** field for **Subsystem id**, type the SSID, for example **PROD** as shown in the previous display. In Query and Reporting, the DB2 subsystem value can be defined using one of the following methods:

- in the DB2 Query Manager profile as described in this section
- using the `DB2SSID=` option statement outside SAS/ASSIST software.
`DB2SSID=`*subsystem-name* specifies the DB2 subsystem name. `DB2SSID=` is valid in the `OPTIONS` statement, as part of the configuration file, and when you invoke SAS
- in the Master, Group, or User profile.

When you save the DB2 Query Manager definition from the Administration window, all profile options are defined specifically for this Query Manager, including the DB2 subsystem. You can set up a DB2 Query Manager for each subsystem. Then, when users need to use a different subsystem, they can select another DB2 Query Manager.

Verify that the User tables and date and time constants have the appropriate values in the Query Manager profile. Change these values and any other values as necessary.

To return to the Query Manager Administration window, select **Close** from the **File** menu.

Save the Query Manager Definition

To save the Query Manager definition, select **Save As...** from the **File** menu. The Save As window appears.

Type the name, description, and the catalog where the Query Manager definitions are stored. The above display shows the ACCOUNT group Query Manager stored in the PUBLIC.MANAGER catalog.

Select **OK** to save the Query Manager definition. To return to the WorkPlace menu, select **Close** from the **File** menu, then **Goback**.

Exit SAS/ASSIST software and clear the libref to the SAS data library where you stored the Master and Group Query Manager definitions. If you have a libref allocated to the physical file name of the Query Manager definitions, Query and Reporting cannot allocate them. For example, submit:

```
libname public clear;
```

Selecting a Query Manager

To select a Query Manager, choose **Select Query Manager** from the **Tools** menu in the SQL Editor window.

You may get a message that the list of Query Managers is being updated. A list of Query Managers is displayed, including any Query Managers that you have created.

Select your newly created Query Manager from the list. The list of Query Managers shows the Query Manager name, description, and type. The type of Query Manager can be **M** for Master, **G** for Group, or **U** for User. See Chapter 3, “Setting Up and Customizing Master and Group Profiles,” on page 7 for more information on Master, Group, and User profiles.

Updating a Query Manager

You need to update static Query Managers periodically. For example, to update your master DB2 Query Manager tables, batch submit the SYSQM.SOURCE program that you edited in “Creating Master DB2 Query Manager Tables” on page 37. You can submit this program again from the Program Editor window or you can batch submit this program to update the master Query Manager tables.

To update the tables for your Query Manager, click on **Select** in the Query window. Next, select **Update Query Manager** from the **Tools** menu.

You can also update the tables for your Query Manager in the SQL Editor window. Select **Query Manager** from the **Tools** menu in the SQL Editor window. The Select DBMS window appears. Select **Update Query Manager** from the **Tools** menu.

The Create/Update Query Manager Tables window appears. The window indicates which program is used to update the Query Manager tables.

Note: SAS Query Managers are updated in a similar way. Follow the same steps whether you are updating a DB2 or a SAS Query Manager. △

Select **Create** to update the tables. A message indicates that the tables have been created and that processing has ended successfully. Select **Goback** to return to the previous window.

Defining and Using a Personal SAS Query Manager

Use a personal SAS Query Manager to query the SAS tables in your SAS session. By default, a personal SAS Query Manager is created the first time you enter the Query

window. The default personal Query Manager uses the program in SASHELP.QASSIST.SASQM.SOURCE to generate the Query Manager tables for all available SAS data libraries in the current SAS session except SASHELP and MAPS.

Use a dynamic Query Manager instead of a static Query Manager if you want your queries to always reflect the most current structure of the SAS or DB2 tables to which you have access in your SAS session. If, for example, you add new tables (SAS tables) to your SAS data libraries or add new columns to your tables, these changes are automatically reflected by the dynamic Query Manager. However, information about the structure of the tables is displayed more slowly when you use a dynamic Query Manager.

This section shows you how to edit the default static, personal SAS Query Manager. It also shows you how to create a new dynamic, personal Query Manager by using the default settings. The Query Manager tables for this Query Manager can be updated using **Update Query Manager** from the **Tools** menu of the Select window.

Editing the Default Personal SAS Query Manager

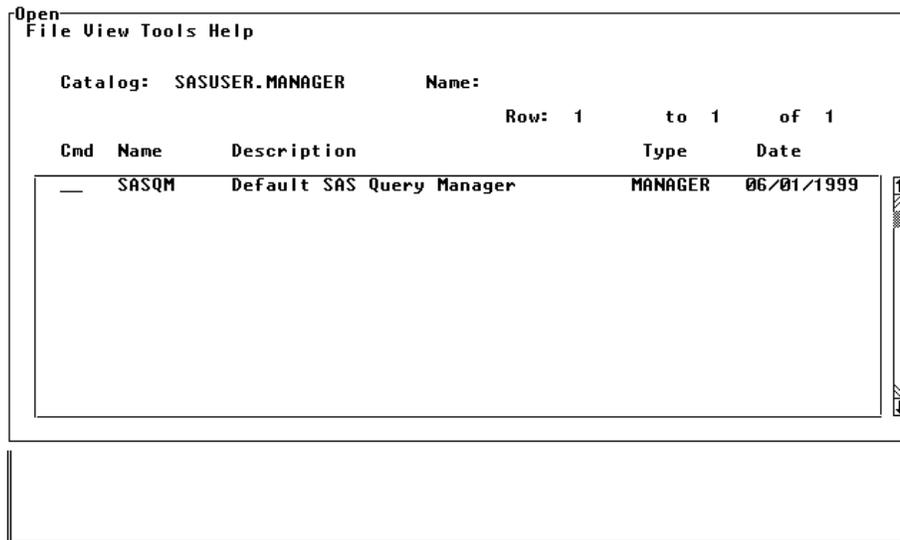
To edit the default personal SAS Query Manager, follow this path:

Tasks \blacktriangleright **Data Management** \blacktriangleright **Query** \blacktriangleright **Query and Reporting** \blacktriangleright **Setup...**

Click on **Administration**, then select **Open...** from the **File** menu.

The Open window appears as shown in the following display.

Display 5.10 Open Window



The Open window shows the default personal Query Manager that was created for you the first time you entered Query and Reporting. To select **SASQM** from the list, type **S** in the **Cmd** field next to it and press ENTER. The Query Manager Administration window appears, as shown in the following display.

Display 5.11 Query Manager Administration Window

```

SAS/ASSIST: Query Manager Administration <SASQM>
File Tools Help
NOTE: SASQM has been loaded.

                Query Manager definitions.
Database:      [ SAS          ]
Program to generate Query Manager tables:
     Yes                 SAS entry
     No                  External file
    [ SASHELP.QASSIST.SASQM.SOURCE ]
Location of Query Manager tables:
     SAS library name
     Physical file name
    [ SASUSER ]
Query Manager type:      Allow update of relations:
     Static                 Yes
     Dynamic                  No

```

The previous display shows the values for the default personal Query Manager Administration window fields. You can edit the default settings by typing new values where typing is permitted or by choosing a different selection where two alternatives are listed.

Database

specifies the type of database for which you want to set up a Query Manager. In the previous display, the database is **SAS**. Type the name of the database you want to use or type a ? in the field and press ENTER to select a database from a list of the database systems that you can access.

Program to generate Query Manager tables

specifies the name of the program that generates and updates the Query Manager tables.

If you select **Yes**, the options **SAS entry** and **External file** appear. If the program is stored in a catalog entry, select **SAS entry** and type the appropriate four-level SAS name. In the previous display, the program in SASHELP.QASSIST.SASQM.SOURCE generates the Query Manager tables. This is the default program. It generates Query Manager tables for all available SAS data libraries in the current SAS session, except SASHELP and MAPS.

If the program is stored in an external file, select **External file** and type the physical file name for the program. The external file must already exist.

After you supply a program name, you can update the Query Manager tables from Query and Reporting. See "Updating a Personal SAS Query Manager" on page 53 for further information.

If you select **No**, the Query Manager tables cannot be updated using **Update Query Manager** from the **Tools** menu of the Select window. The tables must be updated by submitting the program to update the tables from the Program Editor window, or by batch submitting the program. You can also select **No** when you create a dynamic Query Manager because there is no need to update the Query Manager tables. Dynamic Query Manager tables always reflect the most current structure of the SAS or DB2 tables.

Location of Query Manager tables

You store the Query Manager tables in a SAS data library. Select **SAS library name** and type the libref or select **Physical file name** and type the external file

name of the SAS library. By default, the personal Query Manager tables are located in the SASUSER library. Only one set of SAS Query Manager tables can reside in the same SAS data library.

Query Manager type

Select **Static** or **Dynamic**. A static Query Manager is based on SAS data files and provides faster access to the Query Manager tables. However, you must manually update the static Query Manager tables when the structure of the tables changes, for example when new tables (SAS data files) are added to your SAS data libraries or when new columns are added to the tables. See “Updating a Personal SAS Query Manager” on page 53 for more information.

A dynamic Query Manager is based on PROC SQL views to SAS dictionary tables or, for DB2 on z/OS, system tables. The PROC SQL views supply current information to the dynamic Query Manager each time the Query Manager is referenced. See “Creating Dynamic SAS Query Manager Views” on page 50 for further information.

Allow Update of relations

Select **Yes** to allow users to add and delete relations in the Query Manager tables by following this path from the Join Manager window:

View ► Relations ► Current Tables | All Tables

Note: Relations can be updated only when **SAS library name** is selected for **Location of Query Manager tables**. When **Physical file name** is selected for **Location of Query Manager tables**, the library is open for READ access only. Δ

You can save any joins you make in the Join Manager window for future automatic joining of tables. See Chapter 10, “Query and Reporting,” in *Doing More with SAS/ASSIST* for more information.

Select **No** if you want the user to only browse the Query Manager tables from the Join window.

To save any changes you made to the Query Manager definition, select **Save** from the **File** menu. By default, Query and Reporting looks for user-defined Query Managers in the SASUSER.MANAGER catalog. The default name of the personal SAS Query Manager is SASQM.

To return to the WorkPlace menu, select **File**, **Close**, and then **Goback**.

Creating Dynamic SAS Query Manager Views

To create a dynamic Query Manager, you create PROC SQL views instead of SAS data files for all of the Query Manager tables except the _SARELS table. The following steps create the views that describe the data to be queried:

- 1 Submit a LIBNAME statement from the Program Editor window that references the library to contain the dynamic views.
- 2 Copy the SASHELP.QASSIST.SASDYNA.SOURCE program into the Program Editor window. This program creates the dynamic views and the static _SARELS table.
- 3 Change the assignment of the macro variable QMLIB to the libref that references the SAS data library to contain the dynamic Query Manager views.

The following display shows the edited SASDYNA.SOURCE program. The libref DYNAMIC is assigned to the macro variable QMLIB.

Display 5.12 Editing Source Statements to Create Dynamic Query Manager Views

```

Program Editor
File Edit View Tools Run Solutions Help
00001 /*+-----+
00002 | Name : SASHELP.QASSIST.SASDYNA.SOURCE
00003 | Purpose: Create Query Manager tables on SAS tables
00004 | Product: SAS/ASSIST: Query & Reporting
00005 |
00006 | Note : This program Dynamic creates Query manager
00007 | Tables on SAS tables.
00008 | To define the library for Query manager tables
00009 | set the macro variable: qmlib
00010 +-----+*/
00011
00012
00013
00014 %let qmlib=dynamic;
00015 /* Output library for query manager tables */
00016
00017
00018 proc sql;
00019
00020 create view &qmlib._sacols as
00021 select *
00022 from dictionary.columns;
00023 create view &qmlib._satlab as
00024 select *
00025 from dictionary.tables;
00026 ;
00027 create view &qmlib._saindx as

```

- 4 Submit the program to generate the Query Manager views. The views for this Query Manager do not need to be updated because they always reflect the most current structure of the SAS or DB2 tables.

Creating a New Dynamic, Personal SAS Query Manager

To create a new personal SAS Query Manager using the default personal Query Manager, click on **Administration** in the Query and Reporting Setup window. Next, select **Open...** from the **File** menu.

In the Open window, change the catalog by typing **SASHELP.QASSIST** in the **Catalog** field and pressing ENTER. Type **S** in the **Cmd** field for **SASQM** and press ENTER to select it from the list. The Query Manager Administration window appears.

You can change the default values for the default personal Query Manager Administration window fields by typing new values where typing is permitted or by choosing a different selection where two alternatives are listed. See “Defining and Using a Personal SAS Query Manager” on page 47 for descriptions of the fields in this window. For this example, select **No** in the **Program to generate Query Manager tables** field. Select **Dynamic** as the **Query Manager type**.

To save the Query Manager definition in a catalog for user-defined Query Managers, select **Save As...** from the **File** menu. The Save As window appears as shown in the following display.

Display 5.13 Save As Window

SAS/ASSIST: Query Manager Administration <SASQM>
File Tools Help

Query Manager definitions.

Database: [SAS]

Save As

Type: MANAGER
Name: SASQM
Description: Default SAS Query Manager
Catalog: SASUSER.MANAGER

OK Cancel

[SASUSER]

Query Manager type: Allow update of relations:

Static Yes
 Dynamic No

Type a name, description, and catalog of your choice. Make sure you type a new catalog name over SASHELP.QASSIST. By default, Query and Reporting looks for user-defined Query Managers in the SASUSER.MANAGER catalog. Select **OK** to save the Query Manager definition.

To return to the WorkPlace main menu, select **File, Close**, and then **Goback**.

Selecting a Personal SAS Query Manager

To use your personal Query Manager, follow this path:

Tasks ► Data Management ► Query ► Query and Reporting ► SQL Editor...

Then select **Select Query Manager...** from the **Tools** menu.

You may get a message that the list of Query Managers is being updated. A list of Query Managers appears, including any Query Managers that you have created, as shown in the following display.

Display 5.14 Query Manager Selection List

Name	Description (Type)
SASQM	Default SAS Query Manager (U)

Row: 1 to 21 of 22

Select your Query Manager from the list. The list of Query Managers shows the Query Manager name, description, and type. The type of Query Manager can be **M** for Master, **G** for Group, or **U** for User depending on how the User, Master, and Group profiles are set up. See Chapter 3, “Setting Up and Customizing Master and Group Profiles,” on page 7 for more information on Master, Group, and User profiles.

Updating a Personal SAS Query Manager

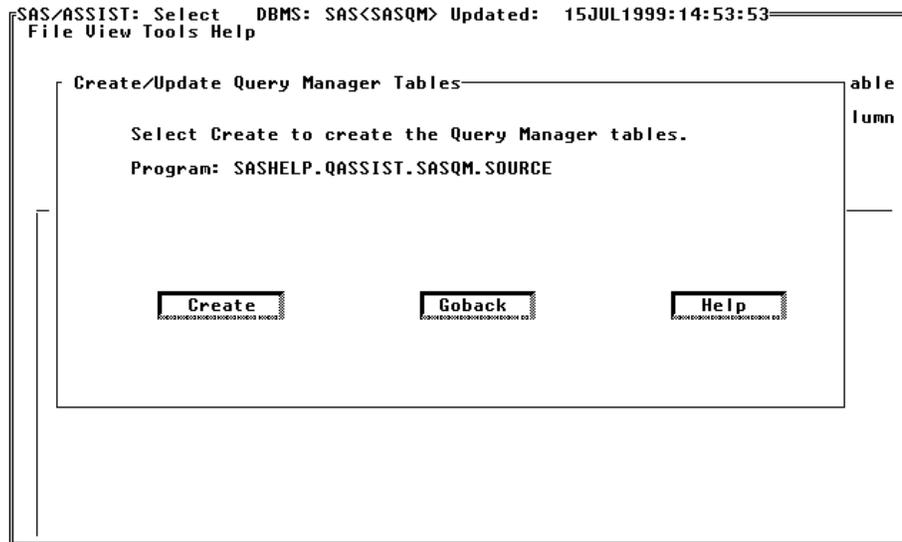
When you create a dynamic Query Manager, there is no need to update the dynamic Query Manager tables because they always reflect the most current structure of the SAS or DB2 tables.

When you create a static Query Manager, you must update the Query Manager periodically to show any changes made to the structure of your data tables. To update your personal SAS Query Manager, follow this path: **Tasks ► Data Management ► Query ► Query and Reporting ► Query...**

Click on **Select**. Then select **Update Query Manager** from the **Tools** menu.

Note: You can also select **Query Manager** from the **Tools** menu in the SQL Editor window. The Select DBMS window appears. Select **Update Query Manager** from the **Tools** menu. △

The Create/Update Query Manager Tables window appears, as shown in the following display.

Display 5.15 Create/Update Query Manager Tables Window

The window indicates which program is used to update the Query Manager tables. In the previous display, the SASHELP.QASSIST.SASQM.SOURCE program is used.

Select **Create** to update the tables. A message indicates that the tables have been created and that processing has ended successfully.

Select **Goback** to return to the previous window.

Deleting Query Managers

To delete a Query Manager, submit a LIBNAME statement that references the SAS data library that contains the Query Manager definitions.

Open the Result Manager window by following this path:

Tasks ► Results ► Result Manager...

The Result Manager window appears. In the **catalog** field, type the name of the catalog where the Query Manager that you want to delete is stored and press ENTER. The list of Query Managers appears in the Result Manager window as shown in the following display.

Display 5.16 Query Managers in the Result Manager Window

SAS/ASSIST: Result Manager
File View Tools Help

Catalog: SASUSER.MANAGER Name: Type: ALL
Row: 1 to 5 of 5

Cmd	Name	Description	Type	Date
—	DYNAMIC	Dynamic Query Manager	MANAGER	07/19/1999
—	SASQM	Default SAS Query Manager	MANAGER	07/15/1999
—	ACCOUNT	Accounting Group Query Manager	MANAGER	07/19/1999
—	DB2	DB2 Query Manager	MANAGER	07/19/1999
—	PUBLIC	Public SAS Query Manager	MANAGER	07/19/1999

Type **D** in the **cmd** field next to the Query Manager you want to delete and press **ENTER**. Type **V** to verify and delete the entry and press **ENTER**. Select **Close** from the **File** menu to exit the Result Manager window.

Creating SAS Query Manager Tables

One sample program is provided to create SAS Query Manager tables. This program can be run from Query and Reporting or from outside Query and Reporting, either from the Program Editor window or in batch. The program is located in SASHELP.QASSIST.SASQM.SOURCE. By default, the program generates Query Manager tables for all available SAS data libraries except SASHELP and MAPS. The Query Manager tables are stored in a library defined by the Query Manager definition. The program can be edited to exclude more libraries and to store the Query Manager tables at a different location. See “Creating Public SAS Query Manager Tables” on page 38 for an example of customizing and using this program.

The Query Manager tables are described in the next sections.

_SACOLS Table

The **_SACOLS** table contains one row for each column in the available SAS tables.

Table 5.2 **_SACOLS Table**

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	LIBNAME	Library Name	CHAR	8	SIMPLE
2	MEMNAME	Member Name	CHAR	8	SIMPLE
3	MEMTYPE	Member Type	CHAR	8	
4	NAME	Column Name	CHAR	8	SIMPLE

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
5	TYPE	Column Type	CHAR	4	
6	LENGTH	Column Length	NUM	8	
7	NPOS	Column Position	NUM	8	
8	VARNUM	Column Number in Table	NUM	8	
9	LABEL	Column Label	CHAR	40	
10	FORMAT	Column Format	CHAR	16	
11	INFORMAT	Column Informat	CHAR	16	
12	IDXUSAGE	Column Index Type	CHAR	9	
13	INDEX	Index	CHAR	1	
14	F_LIB	First library	CHAR	1	SIMPLE
15	F_MEM	First member	CHAR	1	SIMPLE

The following list describes the columns in the `_SACOLS` table:

LIBNAME

contains the name of the SAS data library for the SAS table containing the column.

MEMNAME

contains the name of the SAS table containing the column.

MEMTYPE

contains the type of table (DATA/VIEW).

NAME

contains the name of the column.

TYPE

contains the type of column (CHAR/NUM).

LENGTH

contains the length of the column.

NPOS

contains the column starting position in the table.

VARNUM

contains the column number.

LABEL

contains the column label.

FORMAT

contains the format associated with the column.

INFORMAT

contains the informat associated with the column.

IDXUSAGE

indicates whether the column is part of an index (simple/composite/both).

INDEX

indicates whether the column is part of an index or is blank.

F_LIB

contains a quoted flag **1** for the first column for a given library; otherwise, it contains **0**.

F_MEM

contains a quoted flag **1** for the first column for a given table; otherwise, it contains **0**.

_SAINDX Table

The `_SAINDX` table contains information about SAS indexes that are defined on columns in tables in the `_SACOLS` table. The `_SAINDX` table contains one row for every column that is part of an index. The `_SAINDX` table is not required by the Query Manager, but users often find this information helpful.

Table 5.3 `_SAINDX` Table

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	LIBNAME	Library Name	CHAR	8	SIMPLE
2	MEMNAME	Member Name	CHAR	8	SIMPLE
3	MEMTYPE	Member Type	CHAR	8	
4	NAME	Column Name	CHAR	8	
5	IDXUSAGE	Column Index Type	CHAR	9	
6	INDXNAME	Index Name	CHAR	8	
7	INDXPOS	Position of Column in Concatenated Key	NUM	8	
8	NOMISS	Nomiss Option	CHAR	3	
9	UNIQUE	Unique Option	CHAR	3	
10	IXFIELDS	Column name(s) in index	CHAR	200	

The following list describes the columns in the `_SAINDX` table:

LIBNAME

contains the name of the SAS data library for the SAS table containing the column.

MEMNAME

contains the name of the SAS table containing the column.

MEMTYPE

contains the type of table (DATA/VIEW).

NAME

contains the name of the column.

IDXUSAGE

contains the type of index (SIMPLE/COMPOSITE).

INDXNAME

contains the name of the index.

INDXPOS

contains the column position in the index.

NOMISS

indicates whether no missing values are allowed (yes/blank).

UNIQUE

indicates only unique values (yes/blank).

IXFIELDS

contains the names of all columns that are part of the index. They are separated by commas and listed in column order.

`_SARELS` Table

The `_SARELS` table contains relations between SAS tables. This information is used to join tables automatically. The relation information either must be entered manually or saved from the Join Manager window within Query and Reporting. The `_SARELS` table has one row for every column that is part of a relation.

Table 5.4 `_SARELS` table

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	RELNAME	Relation name	CHAR	8	
2	RELDISC	Relation description	CHAR	30	
3	TABLE	Table name	CHAR	27	
4	RTABLE	Referenced table	CHAR	27	
5	NAME	Foreign key column	CHAR	18	
6	RNAME	Primary key column	CHAR	18	
7	COLSEQ	Position within foreign key	NUM	8	
8	COLCOUNT	Number of columns in foreign key	NUM	8	
9	ORIGIN	Keep track of relation-originator	CHAR	3	

The following list describes the columns in the `_SARELS` table:

RELNAME

Contains the name of the relation.

RELDISC

Contains the description of the relation.

TABLE

Contains *libname.name* of one of the tables in the relation.

RTABLE

Contains *libname.name* of the other table in the relation.

NAME

Contains the name of the column in the table from the **TABLE** column in the relation.

RNAME

Contains the name of the column from the table in the **RTABLE** column in the relation.

COLSEQ

Contains the number of the current column in the relation.

COLCOUNT

Contains the total number of columns in the relation.

ORIGIN

Describes where the relation was created (from the Join Manager window in Query and Reporting or manually).

_SATAB Table

The **_SATAB** table contains one row for each SAS table that can be queried with the Query Manager.

Table 5.5 **_SATAB Table**

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	LIBNAME	Library Name	CHAR	8	SIMPLE
2	MEMNAME	Member Name	CHAR	8	SIMPLE
3	MEMTYPE	Member Type	CHAR	8	
4	MEMLABEL	Dataset Label	CHAR	40	
5	TYPEMEM	Dataset Type	CHAR	8	
6	CRDATE	Date Created	NUM	8	
7	MODATE	Date Modified	NUM	8	
8	NOBS	Number of Observations	NUM	8	
9	OBSLEN	Observation Length	NUM	8	
10	NVAR	Number of Variables	NUM	8	
11	PROTECT	Type of Password Protection	CHAR	3	
12	COMPRESS	Compression Routine	CHAR	8	
13	REUSE	Reuse Space	CHAR	3	
14	BUFSIZE	Bufsize	NUM	8	
15	DELOBS	Number of Deleted Observations	NUM	8	
16	INDXTYPE	Type of Indexes	CHAR	9	
17*	SCREEN	FSEdit screen name	CHAR	35	

* Optional.

The following list describes the columns in the _SATAB table:

LIBNAME

contains the name of the SAS data library containing the table.

MEMNAME

contains the name of the SAS table.

MEMTYPE

contains the type of table (DATA/VIEW).

MEMLABEL

contains the label of the table.

TYPEMEM

contains the table type.

CRDATE

contains the date the table was created.

MODATE

contains the date the table was last modified.

NOBS

contains the number of rows in the table.

OBSLEN

contains the length of one row.

NVAR

contains the number of columns in the table.

PROTECT

if password protected, contains the type of protection.

COMPRESS

YES if table is compressed.

REUSE

YES if deleted space should be reused.

BUFSIZE

contains the buffer size.

DELOBS

contains the number of deleted rows.

INDXTYPE

contains the types of indexes being used.

SCREEN

contains the name of a PROC FSEDIT screen for browsing and editing SAS tables (optional column).

Creating DB2 Query Manager Tables

The following five sample programs are provided to create the DB2 Query Manager tables. These programs can be edited to meet your needs. These programs can be run from Query and Reporting or from outside Query and Reporting, either from the Program Editor window or in batch. All the sample programs are located in source entries in the SASHELP.QASSIST catalog. See Chapter 5, "Setting Up Query Managers," on page 31 for an example of using some of these programs.

Note: Users need READ access to the SYSIBM.SYSTABLES and SYSIBM.SYSCOLUMNS tables when they use a dynamic DB2 Query Manager. △

SYSQM.SOURCE

Use this program to generate a set of master DB2 Query Manager tables from which all other DB2 Query Manager tables are generated. This program should be run as a batch job, when the activity on the DB2 systems catalog is low. READ access to the following DB2 tables is required to run this program:

SYSIBM.SYSTABLES

SYSIBM.SYSCOLUMNS

SYSIBM.SYSRELS

SYSIBM.SYSKEYS

SYSIBM.SYSFOREIGNKEYS

SYSIBM.SYSINDEXES

SYSIBM.SYSTABAUTH

QMUSER1.SOURCE

QMUSER3.SOURCE

Use these programs to generate subsets of the master Query Manager tables based on a specified set of creators. You can modify the programs to reflect the names of the creators that you want in the Query Manager tables. The programs can be run from the Program Editor window, in batch, or from Query and Reporting when specified in a Query Manager definition. Use QMUSER1 to generate smaller subsets of creators and QMUSER3 to generate larger subsets, because the creator names are kept in a separate table.

QMUSER2.SOURCE:

Use this program to generate a set of Query Manager tables containing only the tables to which the user has READ access. This program uses the TABAUTH table generated by the SYSQM.SOURCE program.

DB2DYNA.SOURCE:

Use this program to create a set of dynamic Query Manager tables that read from the DB2 systems catalog directly. A dynamic Query Manager is useful if the structure of DB2 tables changes frequently or if DB2 tables are added and deleted frequently. Be aware that using a dynamic Query Manager on a large DB2 system is considerably slower than using a static one because the DB2 catalog must be read at the time a user chooses the Select window.

TABAUTH Table

The TABAUTH table resides in the library that contains the master DB2 Query Manager tables. It is created by the program in SASHELP.QASSIST.SYSQM.SOURCE. It is used to create personal DB2 Query Manager tables and contains only the tables to which the user has READ access. This table is used by the SASHELP.QASSIST.QMUSER2.SOURCE program.

Note: If privileges are granted to RACF groups instead of to single DB2 users, you can modify QMUSER2.SOURCE and the DB2 RACF exit to use these grants. See SASHELP.QASSIST.SAMP_SCL.SOURCE for an example of the DB2 RACF exit. △

Table 5.6 TABAUTH Table

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	CREATOR	Table creator	CHAR	8	SIMPLE
2	NAME	Table name	CHAR	18	SIMPLE
3	GRANTEE	Table access granted to	CHAR	8	SIMPLE

`_DB2COLS` Table

The `_DB2COLS` table contains one row for every DB2 column in the `_DB2TAB` table. The column names, column types, and column lengths must be specified as they are described in the `_DB2COLS` table. The rest of the table structure can be customized. This table is shown here because of the way that the sample Query Manager source programs create the table. It is recommended that you index the `_DB2COLS` table for the fastest possible access.

Table 5.7 `_DB2COLS` Table

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	CREATOR	Table creator	CHAR	8	SIMPLE
2	NAME	Table name	CHAR	18	SIMPLE
3	COLNAME	Column name	CHAR	18	SIMPLE
4	INDEX	Index	CHAR	1	
5	LABEL	Label/Column name	CHAR	40	
6	COLTYPE	Column type	CHAR	8	
7	LENGTH	Length	NUM	8	
8	DECIMALS	Decimals	NUM	8	
9	NULLS	Nulls allowed?	CHAR	1	
10	COLNO	Column number	NUM	8	
11	TEXT1	Text1	CHAR	70	
12	TEXT2	Text2	CHAR	70	
13	TEXT3	Text3	CHAR	70	
14	TEXT4	Text4	CHAR	70	
15	F_CREA	First.Creator	CHAR	1	
16	L_CREA	Last.Creator	CHAR	1	
17	F_NAME	First.Name	CHAR	1	
18	L_NAME	Last.Name	CHAR	1	
19	FORMAT		CHAR	12	

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
20*	SASNAME	Name that is used when creating a SAS table or view	CHAR	8	
21*	SASLEN	Length in SAS	NUM	8	

* Optional: If the name exists, it is used as a column name in SAS tables or views.

The following list describes the columns in the `_DB2COLS` table:

CREATOR

contains the authorization ID for the table that contains the column.

NAME

contains the name of the table that contains the column.

COLNAME

contains the name of the column.

INDEX

contains an indication of whether the column is part of an index. By default, the indicator is an asterisk (*).

LABEL

contains the DB2 label for the column if one is defined; otherwise, it contains the column name.

COLTYPE

contains the DB2 data type for the column.

LENGTH

contains the length of the DB2 column.

DECIMALS

contains the number of decimals for the DB2 data type decimal.

NULLS

contains an indication of whether or not null values are allowed in the DB2 column: **N** No (the default) or **Y** Yes.

COLNO

contains the number of the column in the DB2 table.

TEXT1 -TEXT4

holds descriptions of the column. The default is the contents of the `REMARKS` column of the `SYSIBM.SYSCOLUMNS` table, divided into 70-byte strings.

F_CREA

is a quoted flag. **1** is the first column for a given creator; otherwise, `F_CREA` contains **0**. `F_CREA` must have the contents as specified.

L_CREA

is a quoted flag. **1** is the last column for a given creator; otherwise, `L_CREA` contains **0**. `L_CREA` must have the contents as specified. All rows in the `_DB2COLS` table between `F_CREA='1'` and `L_CREA='1'` must belong to the same CREATOR.

F_NAME

is a quoted flag. **1** is the first column in the table; otherwise, `F_NAME` contains **0**. `F_NAME` must have the contents as specified.

L_NAME

is a quoted flag. **1** is the last column in the table; otherwise, `L_NAME` contains **0**. `L_NAME` must have the contents as specified. All rows in the `_DB2COLS` table between `F_NAME='1'` and `L_NAME='1'` must belong to the same table NAME.

FORMAT

contains the SAS format being used when the field is displayed. This field is defined in the `SASHELP.QASSIST.SYSQM.SOURCE` program (optional column).

SASNAME

assigns a SAS name to a column in the SAS table. DB2 column names can be 18 characters long. If they truncate in SAS tables, the names might not be meaningful. Use `SASNAME` to assign a more meaningful eight-character name to a SAS column (optional column).

SASLEN

Defines the length of the column when loaded into SAS from DB2 (optional column).

`_DB2INDX` Table

The `_DB2INDX` table contains information about indexes that are defined on columns in tables in the `_DB2COLS` table. The `_DB2INDX` table contains one row for every column that is part of an index. `_DB2INDX` is not required by the DB2 Query Manager, but users often find the information helpful.

Table 5.8 `_DB2INDX` Table

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	CREATOR	Table creator	CHAR	8	SIMPLE
2	NAME	Table name	CHAR	18	SIMPLE
3	COLNAME	Column name	CHAR	18	SIMPLE
4	IXNAME	Index name	CHAR	18	
5	CLUSTER	Cluster index	CHAR	1	
6	UNIQUE	Unique index?	CHAR	1	
7	ICREATOR	Index creator	CHAR	8	
8	ORDERING	Order of col. in key	CHAR	1	
9	IXCOLS	Columns in index	CHAR	200	

The following list describes the columns in the `_DB2INDX` table:

CREATOR

contains the name of the creator of the table.

NAME

contains the name of the table.

COLNAME

contains the name of the column that contributes to the index.

IXNAME

contains the name of the index.

CLUSTER

indicates whether the index is a cluster index. **Y** (yes) indicates a cluster index and is the default value; **N** (no) indicates it is not a cluster index.

UNIQUE

indicates whether the index is unique.

D duplicate values are allowed

U unique and duplicate values are not allowed

P it is a primary key and is unique.

ICREATOR

contains the creator of the index

ORDERING

contains information about index ordering.

IXCOLS

contains the names of all the columns that are part of the index, separated by commas and listed in the order in which the columns appear.

_DB2RELS Table

The `_DB2RELS` table contains primary key/foreign key relations that are defined between DB2 tables. `_DB2RELS` is not required by the Query Manager, but if relations are defined between DB2 tables, the tables can be joined automatically. If relations are not used in your DB2 system, you can provide the information directly or from other sources (for example, from an external data dictionary). The `_DB2RELS` table has one row for every column that is part of a relation. This table must be sorted by `RELNAME`, `TABLE`, `RTABLE`, and `COLSEQ`.

Table 5.9 `_DB2RELS` Table

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	TABLE	Table	CHAR	27	SIMPLE
2	RTABLE	Referenced table	CHAR	27	SIMPLE
3	COLCOUNT	No. of columns in foreign key	NUM	8	
4	RELNAME	Relation name	CHAR	8	SIMPLE
5	ORIGIN	Keep track of relation-originator	CHAR	3	
6	COLNAME	Foreign key column	CHAR	18	SIMPLE
7	COLSEQ	Position within foreign key	NUM	8	
8	PKEYCOL	Primary key column	CHAR	18	
9	CREATOR	Creator of table	CHAR	8	SIMPLE
10*	RELDESC	Relation description	CHAR	30	

* Optional: If this column exists, it is used as a relation description in the menu when the tables are automatically joined.

The following list describes the columns in the `_DB2RELS` table:

TABLE

contains the *creator.name* of the table that contains the foreign key(s).

RTABLE

contains the *creator.name* of the referenced table, that is, the table that contains the primary key(s).

COLCOUNT

contains the number of columns in the foreign key (of the table specified in the `TABLE` column).

RELNAME

contains the name of the primary key/foreign key relation.

ORIGIN

describes where the relation was created: in DB2, in the Master Query Manager, or by the user. This information is used when updating the table so user-defined relations are not affected.

COLNAME

contains the name of the column (from the table in the `TABLE` column) that contributes to the primary key/foreign key relation.

COLSEQ

contains the position in the foreign key that the column in the `COLNAME` column has.

PKEYCOL

contains the name of the column in the primary key with the number that corresponds to the value of the column `COLSEQ`.

CREATOR

contains the name of the creator of the table.

RELDESC

describes the primary key/foreign key relation(s) for the Joins/relations menu in the Join Table window (optional column).

_DB2TAB Table

The `_DB2TAB` table contains one row for every DB2 table or view that the user can query.

Table 5.10 `_DB2TAB` Table

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
1	CREATOR	Table creator	CHAR	8	SIMPLE
2	NAME	Table name	CHAR	18	SIMPLE
3	LABEL	Table label	CHAR	30	
4	TYPE	Alias/Table/View	CHAR	1	
5	DBNAME	Database	CHAR	8	

Column Number	Column Name	Column Label	Column Type	Column Length	Index Type
6	TSNAME	Table space	CHAR	8	
7	NPAGES	Number of pages in table	NUM	8	
8	ROWS	Number of rows in table	NUM	8	
9	TEXT1	Text1	CHAR	70	
10	TEXT2	Text2	CHAR	70	
11	TEXT3	Text3	CHAR	70	
12	TEXT4	Text4	CHAR	70	
13	P_KEY	Primary Key	CHAR	200	
14*	SCREEN	FSEdit screen name	CHAR	35	

* Optional.

The following list describes the columns in the `_DB2TAB` table.

CREATOR

contains the authorization ID for the table.

NAME

contains the name of the table.

LABEL

contains the label of the table.

TYPE

contains information about the type of DB2 object in question, **T** for a table, **V** for a view, **A** for an alias. **T** is the default.

DBNAME

contains the name of the DB2 database where the table or view is stored.

TSNAME

contains the name of the DB2 table space where the table resides.

NPAGES

contains the number of pages that the table uses. Default: **-1** if it is a view or if a `RUNSTATS` utility has not been run for the table space.

ROWS

contains the number of rows in the table, unless it is a view or a `RUNSTATS` utility has not been run for the table space.

TEXT1 –TEXT4

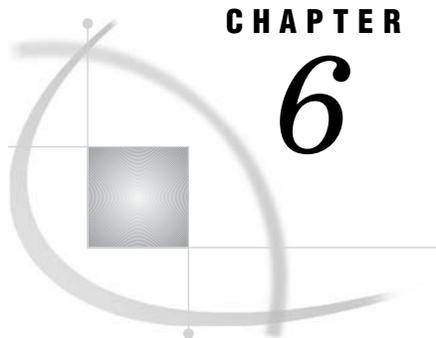
contains a description of the table. The `REMARKS` column in `SYSIBM.SYSTABLES` is divided into 70-byte strings.

P_KEY

contains the names of the columns in the primary key (separated with commas).

SCREEN

contains the name of a `PROC FSEdit` screen to be used for browsing and editing DB2 tables (optional column).



CHAPTER

6

Creating Metadata

Overview	69
Metadata Feature	69
Generating Metadata in Batch	71
Sample Metadata-Generating Program	73

Overview

SAS/ASSIST software is *metadata driven*. This means that, instead of examining each data source and extracting information each time you invoke SAS/ASSIST software, the software examines permanently saved, locally stored files that contain this information. This information, which includes table names, number of observations, column names and attributes, and more, is called *metadata*.

As an administrator, you can set up batch programs to create metadata and make it available to users from within their SAS/ASSIST session. Metadata creation can be customized globally for an entire site, or locally for a particular division or user. The existence of locally stored metadata makes it possible for SAS/ASSIST software to instantly display comprehensive selection lists, enabling users to search across all tables or columns.

Metadata Feature

In SAS/ASSIST software, metadata is either generated at startup or generated prior to startup in batch mode. When users invoke the software, the batch-generated metadata is immediately available and can be appended to the metadata that is generated at startup by the user. To use the metadata library, the profile option **User Metadata Library** must be set to **Yes** either in the User Profile or the Master Profile.

Metadata is created in batch mode by using a SAS program. A sample program is provided with SAS/ASSIST software. The sample program is located in the SASHELP library at SASHELP.QASSIST.METADICT.SOURCE. The batch program must also contain a list of LIBNAME statements. When a library is both defined in the batch program and assigned in a user session, metadata for data sources that are located in this library is made available to the user. Conversely, if a library is defined in the batch program but not assigned in an individual user session, data sources that are located in this library are not visible to this particular user.

The batch-generated metadata and the user-generated metadata are stored together in two SAS tables, `_ASTDCOL` and `_ASTDTAB`. If the metadata is created in batch mode prior to startup, both `_ASTDCOL` and `_ASTDTAB` are automatically created in the SASUSER library the first time a user invokes the software. If the metadata is

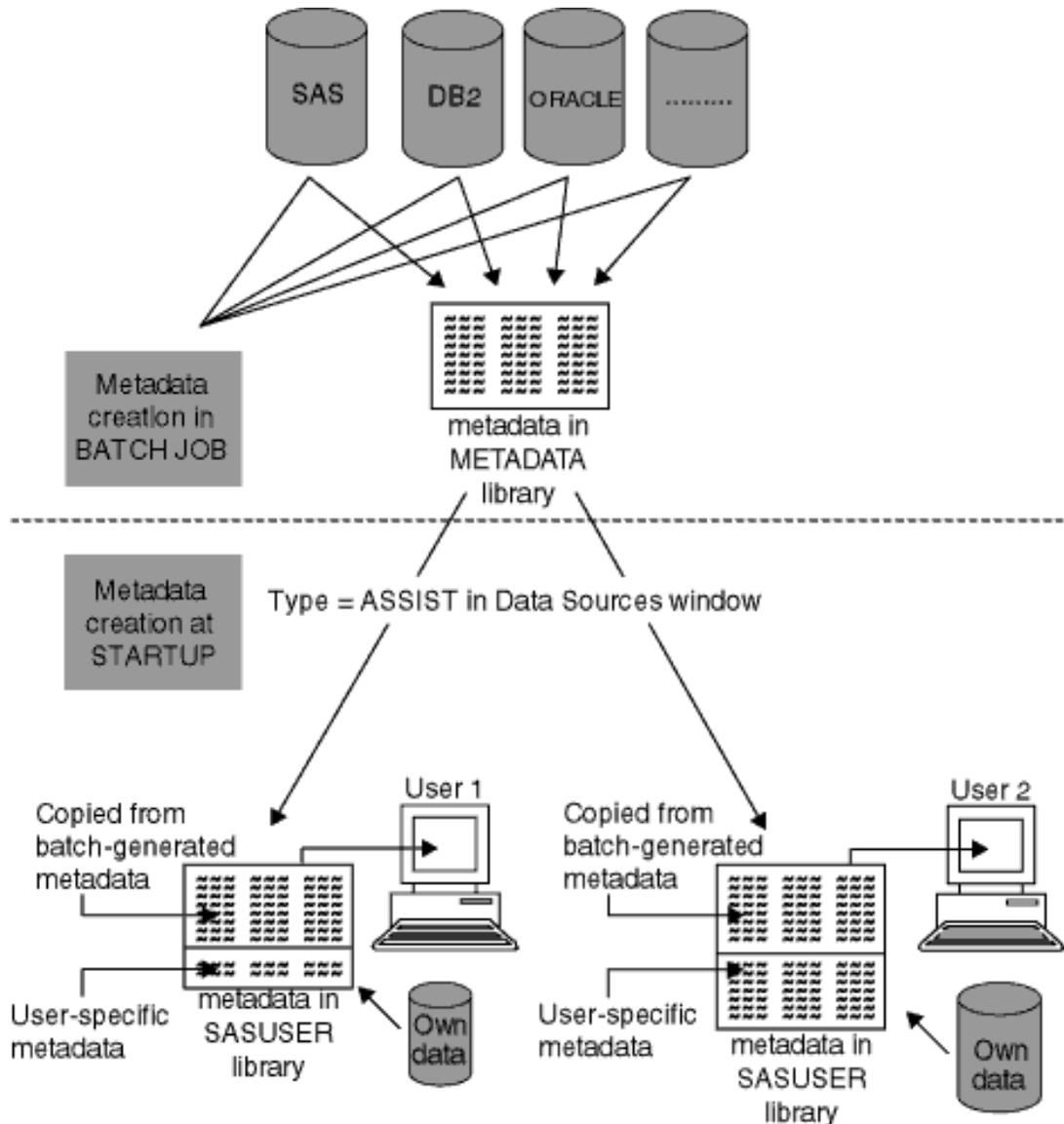
generated at startup, only the `_ASTDCOL` table is generated, and the user is unable to view the `_ASTDTAB` table.

The advantages of creating metadata in batch mode are as follows:

- The use of resources is reduced when users invoke SAS/ASSIST software. If metadata is created at startup by the individual user, then the software accesses all available data in order to gather information. Creating metadata in batch mode avoids the need for each user to access remote hosts in order to examine the data.
- The metadata can be customized to suit the needs of individual users and groups. You can subset metadata generation in the sample program so that users see only the metadata for the information they normally work with. This simplifies the process of searching for tables or columns.
- The processing time is significantly reduced when users update metadata during a session. Creating metadata at startup can be time consuming, especially when a large number of tables are accessed or when data resides on remote hosts. Batch-generated metadata, however, is stored in two tables in a shared location. These tables are appended to the user-generated metadata.

The following figure illustrates and compares metadata creation in batch and metadata creation at startup.

Figure 6.1 Metadata Creation in SAS/ASSIST



Generating Metadata in Batch

The sample metadata-generating program that is supplied with SAS/ASSIST software can be modified to suit the needs of individual organizations. This section explains how to modify the original program to suit your needs. The original program is included in Chapter 6, “Creating Metadata,” on page 69 of this document.

To create metadata:

- 1 Invoke SAS. In the text editor, create a program that defines the library in which metadata will be stored and the libraries that point to the underlying data sources. The METADATA library must be assigned the name METADATA.

```
libname metadata <shared location>;
libname library1 <data source 1>;
```

```
libname library2 <data source 2>;
...
```

Depending on which DBMS you are assigning libraries to, you might want to define additional LIBNAME options that are specific to that system.

- 2 Save and run the program.
- 3 Copy the sample metadata-generating program to your text editor in SAS. Issue the following command in the command line:

```
copy sashelp.qassist.metadict.source
```

- 4 Define the libraries that are to be included in the metadata. In the program code, you will find this line:

```
%let sellib=('INFORMIX', 'SASDATA');
```

Change the libraries in the parentheses to the libraries that you defined in the LIBNAME statements in step 1. For example, in the following code, the program is modified to include four libnames: DEPOTS, RETURNED, STOCKS, and SUPPLIER.

```
%let sellib=('DEPOTS', 'RETURNED', 'STOCKS', 'SUPPLIER');
```

- 5 Further modify the code as desired to customize the task of metadata creation, as shown in the following code:

```
create table _astdcol as
  select *
    from dictionary.columns
   where (libname=DEPOTS and memname in (ROSSLAIGHRE,SHANNON,COVENTRY,LUTON))
        or (libname=SUPPLIER and memname in(MATERIALS,UTILITIES,PACKAGING)) or
        libname in (RETURNED,STOCKS)
   order by libname, memname, name;

create table _astdtab as
  select *
    from dictionary.tables
   where (libname=DEPOTS and memname in (ROSSLAIGHRE,SHANNON,COVENTRY,LUTON))
        or (libname=SUPPLIER and memname in(MATERIALS,UTILITIES,PACKAGING)) or
        libname in (RETURNED,STOCKS)
   order by libname, memname;
```

In the above code, the program is modified so that only a subset of tables in the DEPOTS and SUPPLIER libraries are included in metadata creation. The rest of the program, which does not require any further modifications, is shown in “Sample Metadata-Generating Program” on page 73.

- 6 Save the file using the following command:

```
save metadata.create.metadict.source
```

- 7 Create the metadata in batch:

- If this is the first time you are creating metadata in batch, then open the program you created in step one, and modify it to include the following code:

```
libname metadata <shared location>;
libname library1 <data source 1>;
libname library2 <data source 2>;
dm af c=sashelp.qassist.metadict.scl;
```

Save the program and execute it in a batch job.

- If you have previously created metadata in batch, simply run the following command in SAS:

```
af cat=sashelp.qassist.metadict.scl
```

When the SAS code you edited above is executed, two SAS tables, `_ASTDCOL` and `_ASTDTAB`, are created in the `METADATA` library. Remember to check the log when this is finished.

Note: In subsequent batch jobs, the metadata is automatically updated if the metadata tables exist. If they do not exist, the metadata is automatically recreated by SAS/ASSIST software. △

Run the batch job each time that the data structure changes. You might want to run it as a scheduled job.

Sample Metadata-Generating Program

The following program is included with SAS/ASSIST software and can be modified as explained in “Generating Metadata in Batch” on page 71.

```

/*-----+
| Name   : SASHELP.QASSIST.METADICT.SOURCE
| Purpose: Create ASSIST dictionary tables
| Product: SAS/ASSIST:
|
| Note   : This program creates dictionary tables with
|           long names and their alias.
|           To define the library for Dictionary tables
|           set the macro variable: dictlib
|           To select libraries in the query manager tables
|           set the macro variable: sellib
|-----+*/

%let dictlib=Metadata;
/* Output library for query manager tables */

%let sellib=('INFORMIX','SASDATA'); /* <--- THIS MUST BE
CHANGED TO THE ACTUAL LIBNAMES */
/* Libraries to be excluded in Dictionary tables */
/* To select all: */
/* %let sellib=('*'); */

%macro upc(macvar,value);
  %global &macvar
  %let &macvar=%upcase(&value);
%mend;

/* upcase the value of sellib */
%upc(sellib,&sellib);
proc sql;
  create table work.__log

```

```

        (db2rc   char(5)  label='Return Code',
         db2msg1 char(70) label='Message-1',
         db2msg2 char(70) label='Message-2',
         db2msg3 char(70) label='Message-3',
         db2msg4 char(70) label='Message-4');
quit;

proc sql;
  drop table _astdcol;
  drop table _astdtab;

  create table _astdcol as
    select *
      from dictionary.columns
     where libname in &sellib
     order by libname, memname, name;

  create table _astdtab as
    select *
      from dictionary.tables
     where libname in &sellib
     order by libname, memname;

  alter table _astdtab
    modify crdate format=DATETIME19.,
    moddate format=DATETIME19.
;

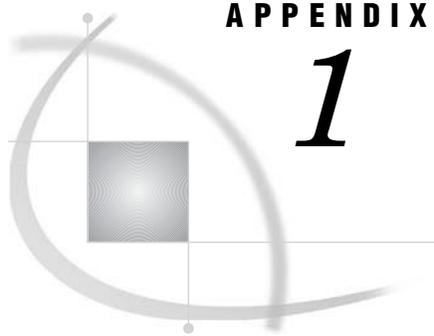
/* data &dictlib._astdtab(index=(libname memname) */
data &dictlib._astdtab(label="ASSIST Dictionary tables: Table info");

  set _astdtab;
run;

/*data &dictlib._astdcol(index=(libname memname name f_lib f_mem) compress=NO */
data &dictlib._astdcol(compress=NO label="ASSIST Dictionary tables: Column info");

  set _astdcol;
  attrib
  index length= $1 format=$1. label="Index"
  f_lib length= $1 format=$1. label="First library"
  f_mem length= $1 format=$1. label="First member"
  ;
  by libname memname;
  f_lib=put(first.libname,1.);
  f_mem=put(first.memname,1.);
  if idxusage eq ' ' then index=' ';
  else index='*';
  if label=' ' then label=name;
run;

```



APPENDIX

1

Recommended Reading

Recommended Reading 75

Recommended Reading

Here is the recommended reading list for this title:

- *Getting Started with SAS/ASSIST*
- *Doing More with SAS/ASSIST*
- *SAS/CONNECT User's Guide*
- *SAS/GRAPH Software: Reference, Volumes 1 and 2*
- *SAS Language Reference: Concepts*
- *SAS Language Reference: Dictionary*
- *Base SAS Procedures Guide*

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

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513
Telephone: 1-800-727-3228
Fax: 1-919-531-9439
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Customers outside the United States and Canada, please contact your local SAS office for assistance.

Glossary

active table

in SAS/ASSIST software, a SAS table that has been specified for a particular task.

analysis variable

a numeric variable that is used to calculate statistics. An analysis variable usually contains quantitative or continuous values, but this is not required.

arithmetic operator

in SAS software, any of the symbols (+, -, /, *, and **) that are used to perform addition, subtraction, division, multiplication, and exponentiation in SAS expressions.

base SAS software

software that includes a programming language that manages your data, procedures for data analysis and reporting, procedures for managing SAS files, a macro facility, help menus, and a windowing environment for text editing and file management.

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.

button

a component of a graphical user interface. A button is usually in the form of a rectangle or square that contains a label. The button is programmed to execute a command, to open a window, or to perform some other function when a user selects it. For example, many graphical user interfaces include buttons that have labels such as OK, Cancel, and Help.

BY variable

a variable named in a BY statement whose values define groups of observations to process.

character variable

a variable whose values can consist of alphabetic and special characters as well as numeric characters.

chart variable

the variable whose values you are charting.

check box

an item in a window that you can select without affecting any other items. You can deactivate a check box by selecting it again.

class variable

a variable that is used to group, or classify, data. 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.

column

in relational databases, a vertical component of a table. Each column has a unique name, contains data of a specific type, and has certain attributes. A column is analogous to a variable in SAS terminology.

crosstabulation table

a frequency table that shows combined frequency distributions or other descriptive statistics for two or more variables. These tables are also referred to as two-, three-, or n - way tables; contingency tables; and stub-and-banner tables. See also frequency table.

cumulative frequency

the total number of observations in all ranges up to and including a given range.

cumulative percent

the percentage of observations in all ranges up to and including a given range.

DATA step

a group of statements in a SAS program that begins with a DATA statement and ends with either a RUN statement, another DATA statement, a PROC statement, or the end of the job. The DATA step enables you to read raw data or other SAS data sets and to create SAS data sets.

data value

a unit of character or numeric information in a SAS data set. A data value represents one variable in an observation.

dependent column

See response variable.

dialog box

a type of window that opens to prompt you for additional information or to ask you to confirm a request.

engine

a component of SAS software that reads from or writes to a file. Each engine enables SAS to access files that are in a particular format.

explanatory variable

a variable that is included in a regression model in order to explain the variation in the dependent (or response) variable. Explanatory variables, which are also called independent, input, predictor, or regressor variables, can be deterministic or probabilistic.

external file

a file that is created and maintained by a host operating system or by another vendor's software application. SAS can read data from and route output to external files. External files can contain raw data, SAS programming statements, procedure output, or output that was created by the PUT statement. An external file is not a SAS data set. See also fileref.

fileref

a name that is temporarily assigned to an external file or to an aggregate storage location such as a directory or folder. The fileref identifies the file or the storage location to SAS. In SAS/ASSIST software, you assign a fileref with the Setup task.

format

a pattern or set of instructions that SAS uses to determine how the values of a variable (or column) should be written or displayed. SAS provides a set of standard formats and also enables you to define your own formats.

frequency chart

a graphic illustration of the number of times a value or range of values for a given variable occurs.

frequency count

the number of times a value or range of values for a given variable occurs.

frequency table

a table that lists the values of a variable and the number of observations with each value. See also crosstabulation table, frequency count.

function key

a keyboard key that can be defined to have a specific action in a specific software environment.

independent column

See explanatory variable.

index

in SAS software, a component of a SAS data set that enables SAS to access observations in the SAS data set quickly and efficiently. The purpose of SAS indexes is to optimize WHERE-clause processing and to facilitate BY-group processing.

informat

a pattern or set of instructions that SAS uses to determine how data values in an input file should be interpreted. SAS provides a set of standard informats and also enables you to define your own informats.

interactive line mode

a method of running SAS programs in which you enter one line of a SAS program at a time at the SAS session prompt. SAS processes each line immediately after you press the ENTER or RETURN key. Procedure output and informative messages are returned directly to your display device.

label, variable

a descriptive label of up to 40 characters that can be printed in the output by certain procedures instead of, or in addition to, the variable name.

length, variable

the number of bytes used to store each of a variable's values in a SAS data set.

libref

a name that is temporarily associated with a SAS data library. The complete name of a SAS file consists of two words, separated by a period. The libref, which is the first word, indicates the library. The second word is the name of the specific SAS file. For example, in VLIB.NEWBDAY, the libref VLIB tells SAS which library contains the file NEWBDAY. You assign a libref with a LIBNAME statement or with an operating system command.

local session

a SAS session running on the local host. The local session accepts SAS statements and passes those that are remote-submitted to the remote host for processing. The

local session manages the output and messages from both the local session and the remote session.

logical operator

an operator used in expressions to link sequences of comparisons. The logical operators are AND, OR, and NOT.

menu bar

the primary list of items at the top of a window which represent the actions or classes of actions that can be executed. Selecting an item executes an action, opens a pull-down menu, or opens a dialog box that requests additional information. See also pop-up menu, pull-down menu.

message area

the area immediately beneath a window's command line or menu bar which displays messages from SAS.

metadata

a description or definition of data or information.

midpoint

a value that represents one data value or the middle of a range of data values. When a midpoint represents a range of values, the algorithm used to calculate it depends on the procedure.

missing value

in SAS, a term that describes the contents of a variable that contains no data for a particular row or observation. By default, SAS prints or displays a missing numeric value as a single period, and it prints or displays a missing character value as a blank space.

numeric variable

a variable that contains only numeric values and related symbols, such as decimal points, plus signs, and minus signs.

observation

a row in a SAS data set. All of the data values in an observation are associated with a single entity such as a customer or a state. Each observation contains one data value for each variable.

permanent SAS data set

a data set that remains after the current program or interactive SAS session terminates. Permanent SAS data sets are available for future SAS sessions.

pop-up menu

a menu that appears when it is requested. These menus are context-specific, depending on which window is active and on the cursor location. See also pull-down menu.

PROC SQL view

a SAS data set (of type VIEW) that is created by the SQL procedure. A PROC SQL view contains no data. Instead, it stores information that enables it to read data values from other files, which can include SAS data files, SAS/ACCESS views, DATA step views, or other PROC SQL views. A PROC SQL view's output can be either a subset or a superset of one or more files. See also view.

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. See also pop-up menu.

raw data

data that has not been read into a SAS data set.

remote session

a SAS session running in a special mode on the remote host. No output or log messages are displayed on the remote host; instead, the results of a remote SAS session are transmitted back to the log and output files on the local host.

response variable

in regression analysis, the variable that you are interested in modeling for the purpose of seeing how it changes as a function of the factors. The response variable is also referred to as the dependent variable.

row

in relational database management systems, the horizontal component of a table. It is analogous to a SAS observation.

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. 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 file

a SAS data set that contains data values as well as descriptor information that is associated with the data. The descriptor information includes information such as the data types and lengths of the variables, as well as which engine was used to create the data. SAS data files are of member type DATA. See also SAS data set, SAS data view.

SAS data library

a collection of one or more SAS files that are recognized by SAS and which are referenced and stored as a unit. Each file is a member of the library.

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. SAS data views can be created by the SAS DATA step and by the SAS SQL procedure.

SAS log

a file that contains a record of the SAS statements that you enter as well as messages about the execution of your program.

SAS name

a name whose construction follows certain rules and that can appear in a SAS statement (for example, names of variables and SAS data sets).

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.

SAS program

a group of SAS statements that guide SAS through a process or series of processes.

SAS statement

a string of SAS keywords, SAS names, and special characters and operators that instructs SAS to perform an operation or that gives information to SAS. Each SAS statement ends with a semicolon.

SAS Support Consultant

an individual at your computing installation who is designated as a support person for SAS software users at the installation. The consultant can help you with questions about using SAS software.

SAS table

another term for SAS data set. See SAS data set.

SAS windowing environment

an interactive windowing interface to SAS software. In this environment you can issue commands by typing them on the command line, by pressing function keys, or by selecting items from menus or menu bars. Within one session, you can perform many different tasks, including preparing and submitting programs, viewing and printing results, and debugging and resubmitting programs.

Sasuser library

a default, permanent SAS data library that is created at the beginning of your first SAS session. The Sasuser library contains a Profile catalog that stores the customized features or settings that you specify for SAS. You can also store other SAS files in this library.

scroll bar

a graphical window element that is used to bring the undisplayed contents of an associated object (such as a text box or a list box) into view. A scroll bar is shaped like a horizontal or vertical bar and has a scroll arrow at each end. If the associated object contains text that is not in view, the scroll bar contains a box that you can drag in order to move up or down in the viewing area. Alternatively, you can click on the arrows to move up or down in smaller increments.

selection field

the portion of a window (shown on the display as an underscore) where you can enter a short command to perform an action, such as B for browse.

selection-field command

a command that enables you to perform actions from a selection field in a SAS windowing environment. For example, entering D in the selection field beside the name of a SAS data set in the DIRECTORY window enables you to delete that SAS data set.

SQL (Structured Query Language)

a standardized, high-level query language that is used in relational database management systems to create and manipulate database management system objects. SAS implements SQL through the SQL procedure.

table

in general, a two-dimensional system of representing data in rows and columns.

temporary SAS data set

a data set that exists only for the duration of the current program or interactive SAS session. Temporary SAS data sets are not available for future SAS sessions.

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.

view

a generic term (used by many software vendors) for a definition of a virtual data set (or table). The definition is named and stored for later use. A view contains no data; it merely describes or defines data that is stored elsewhere.

WHERE expression

a type of SAS expression that specifies a condition for selecting observations for processing by a DATA step or a PROC step. WHERE expressions can contain special operators that are not available in other SAS expressions. WHERE expressions can appear in a WHERE statement, a WHERE= data set option, a WHERE clause, or a WHERE command.

WHERE processing

a method of conditionally selecting observations for processing in a DATA or PROC step. WHERE processing involves using a WHERE expression in a WHERE statement, a WHERE= data set option, a WHERE clause, or a WHERE command. See also WHERE expression.

Work data library

a SAS data library that is automatically defined by SAS at the beginning of each SAS session or SAS job. The Work library contains SAS files that are temporary by default. When the libref User is not defined, SAS uses Work as the default library for SAS files that are created with one-level names.

Index

A

Account Profile window 14
 Active Graphics Device window 26
 applications
 launching at logoff 5
 launching at logon 5
 Assign a New Libref window 21
 ASSIST command, options 5

C

Catalog Directory window 23
 Create Group Profile window 10

D

D (default) status value 15
 _DB2COLS table
 creating 62
 definition 33
 DB2DYNA.SOURCE program 61
 _DB2INDX table
 creating 64
 definition 33
 _DB2RELS table
 creating 65
 definition 33
 _DB2TAB table
 creating 66
 definition 33
 default (D) status value 15
 defaults, setting for the system 3
 Define Exceptions window 16
 Delete Group Profile window 13
 Detail window 23
 device drivers
 displaying 27
 modifying 20
 printing with 28
 dynamic Query Manager
 creating views (SAS) 50
 definition 33

F

files, specifying 4
 formats, custom 4

G

Graphics Device Catalog window 20
 graphics device defaults
 modifying 27
 setting 4
 Graphics Devices Directory window 23
 Group profiles 7
 creating 10
 deleting 12
 editing 13
 modifying 12
 pointing to Query Manager definitions 40
 propagating to lower level profiles 15
 propagation, exceptions to 15
 setting options for 8
 status values 15
 updating 10

H

Host Commands window 25
 Host File Option window 24

I

informats, custom 4
 Initiation Program 44

J

joining tables
 automatically 39
 _DB2RELS table 33, 65
 _SARELS table 33, 58

L

launching applications
 at logoff 5
 at logon 5
 libraries, specifying 4
 LOGOFF= option 5
 LOGON= option 5

M

MAPS tables 5

- Master/Group Profile window
 - opening and using 8
 - Query exit option 18
 - System Administrator options 18
 - Userid option 18
- Master profiles 7
 - creating 7
 - editing 8, 13
 - pointing to Query Manager definitions 40
 - propagating to lower level profiles 15
 - propagation, exceptions to 15
 - setting options for 8
 - status values 15
- master tables (DB2)
 - creating 37
 - specifying location of 46
 - subsetting 36, 38, 43

O

- Open Profile window 14

P

- PARMSDS= option 5
- personal Query Manager 33
- personal Query Manager (SAS)
 - creating 51
 - defining and using 47
 - editing the default 48
 - selecting 52
 - updating 53
- prefix commands, setting 4
- primary menu application, alternative to 5
- PRIMARY= option 5
- printers, printing directly to 29
- printing graphics 20
 - device drivers, displaying 27
 - device drivers, modifying 20
 - directly to the printer 29
 - graphics device defaults, modifying 27
 - with modified drivers 28
- printing reports 19
- public Query Manager
 - defining and using (DB2) 35
 - definition 33
- public Query Manager definitions
 - creating 40, 42
 - pointing to 40
 - storage locations for 40
- public Query Manager (SAS), defining and using 35, 38

Q

- Query exit option 18
- Query Manager 32
 - combining types of 34
 - deleting 54
 - dynamic 33
 - examples 34
 - personal 33
 - public 33, 42
 - selecting 47
 - static 34
 - updating 47

- Query Manager (DB2)
 - data files 33
 - master tables, creating 37
 - master tables, specifying location of 46
 - master tables, subsetting 36, 38, 43
 - public 35
 - Query Manager Profile 46
- Query Manager (DB2) tables
 - creating, sample programs for 60
 - _DB2COLS 33, 62
 - _DB2INDX 33, 64
 - _DB2RELS 33, 65
 - _DB2TAB 33, 66
 - READ access, listing 61
 - TABAUTH 61
- Query Manager definitions
 - public, creating 40
 - public, pointing to 40
 - public, storage locations for 40
 - saving 47
- Query Manager Profile Options window 41
- Query Manager (SAS)
 - data files 33
 - dynamic views, creating 50
 - personal, creating 51
 - personal, defining and using 47
 - personal, editing the default 48
 - personal, selecting 52
 - personal, updating 53
 - public, defining and using 35, 38
- Query Manager (SAS) tables
 - _SACOLS 33, 55
 - _SAINDX 33, 57
 - _SARELS 33, 58
 - _SATAB 33, 59

R

- remote connections 4
- restrict (R) status value 15

S

- _SACOLS table
 - creating 55
 - definition 33
- _SAINDX table
 - creating 57
 - definition 33
- _SARELS table
 - creating 58
 - definition 33
- SAS data libraries
 - librefs for Query Manager 44
 - names under various operating environments 22
- SAS tables, joining
 - automatically 39
 - _DB2RELS table 33, 65
 - _SARELS table 33, 58
- SAS tables, master (DB2)
 - creating 37
 - specifying location of 46
 - subsetting 36, 38, 43
- SASHELP.QASSIST.QMUSERx.SOURCE program 36, 38, 43, 61

- SASHELP.QASSIST.SYSQM.SOURCE program 37, 39, 60
 - SASHELP.SASPARM table 3
 - SASPARM table 3
 - alternatives to 3, 5
 - initial creation 3
 - SASHELP.SASPARM table 3
 - SASUSER.SASPARM table 3
 - setting system defaults 3
 - template for 3
 - SASUSER.SASPARM table 3
 - _SATAB table
 - creating 59
 - definition 33
 - setting up SAS/ASSIST software 3
 - ASSIST command, options 5
 - files, specifying 4
 - for individual users 4
 - formats, custom 4
 - graphics defaults 4
 - informats, custom 4
 - launching applications at logoff 5
 - launching applications at logon 5
 - libraries, specifying 4
 - MAPS tables 5
 - prefix commands, setting 4
 - primary menu application, alternative to 5
 - remote connections 4
 - SASPARM table 3
 - suffix commands, setting 4
 - static Query Manager 34
 - status values
 - D (default) 15
 - Group profiles 15
 - Master profiles 15
 - R (restrict) 15
 - U (update) 15
 - User profiles 15
 - suffix commands, setting 4
 - system defaults, setting 3
- ## T
- tables, joining
 - automatically 39
 - _DB2RELS table 33, 65
 - _SARELS table 33, 58
 - tables, master (DB2)
 - creating 37
 - specifying location of 46
 - subsetting 36, 38, 43
 - template for SASPARM table 3
 - Termination Program 44
- ## U
- U (update) status value 15
 - Update User Group window 11
 - User profiles 7
 - setting options for 8
 - status values 15
 - USER table 11
 - Userid option 18
 - userid, and USER table 11

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.

SAS® Publishing delivers!

Whether you are new to the workforce or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart.

SAS® Press Series

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from the SAS Press Series. Written by experienced SAS professionals from around the world, these books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information—SAS documentation. We currently produce the following types of reference documentation: online help that is built into the software, tutorials that are integrated into the product, reference documentation delivered in HTML and PDF—free on the Web, and hard-copy books.

support.sas.com/publishing

SAS® Learning Edition 4.1

Get a workplace advantage, perform analytics in less time, and prepare for the SAS Base Programming exam and SAS Advanced Programming exam with SAS® Learning Edition 4.1. This inexpensive, intuitive personal learning version of SAS includes Base SAS® 9.1.3, SAS/STAT®, SAS/GRAPH®, SAS/QC®, SAS/ETS®, and SAS® Enterprise Guide® 4.1. Whether you are a professor, student, or business professional, this is a great way to learn SAS.

support.sas.com/LE



THE
POWER
TO KNOW®

