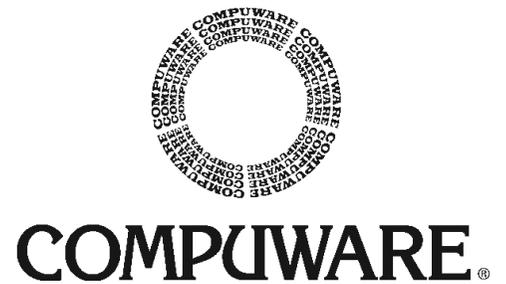


STROBE MVS

System Programmer's Guide

Release 3.0



Please direct questions about STROBE MVS
or comments on this document to:

STROBE MVS Technical Support

Compuware Corporation
124 Mount Auburn Street
Cambridge MA 02138-5758
1-800-585-2802 or
1-617-661-3020
1-617-498-4010 (fax)
strobe-sup@compuware.com

Outside the USA and Canada, please contact
your local Compuware office or agent.

This document and the product referenced in it are subject to the following legends:

Copyright 2002 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS-Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

STROBE, iSTROBE and APMpower are trademarks or registered trademarks of Compuware Corporation.

DFSMS, MVS/ESA, MVS/XA, OS/390, Parallel Sysplex, RACF, Resource Measurement Facility. The following are registered trademarks of International Business Machines Corporation: AD/Cycle, BatchPipes, BookManager, CICS, DB2, IBM, IMS/ESA, OS2, Language Environment, Software Mall, MQSeries, and VisualGen are trademarks of International Business Machines Corporation.

AD/Cycle, BookManager, CICS, DB2, IBM, IMS/ESA, Language Environment, MQSeries, OS/2, Software Mall, VisualGen, and VTAM are trademarks of International Business Machines Corporation. Microsoft is a registered trademark of Microsoft Corporation. Windows, Windows NT, and Windows 98 are trademarks of Microsoft Corporation. Attachmate and EXTRA! are registered trademarks of Attachmate Corporation. RUMBA is a registered trademark of Wall Data Inc. WRQ and Reflection are registered trademarks of WRQ, Inc.

Adobe ® Acrobat ® Reader copyright © 1987-2001 Adobe Systems Incorporated. All rights reserved. Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

All other company and product names are trademarks or registered trademarks of their respective owners.

Contents

Figures	xi
Tables	xiii
Summary of Changes	xv
Changes to STROBE	xv
Changes to this Manual	xvi
Introduction	xvii
How This Manual Is Organized	xvii
How to Use This Manual	xvii
The STROBE Library	xvii
STROBE Feature Manuals	xviii
Online Documentation	xviii
Online Help	xviii
Other Compuware Application Performance Management Products	xix
iSTROBE	xix
SQL Analysis Feature	xix
APMpower	xix
Compuware APM Technical Support	xix
Compuware APM Training	xx
Compuware APM Service Offerings	xx
APM Installation Assurance	xx
Application Performance Management Consulting	xx
Application Performance Assessment	xx
Chapter 1. Overview	1-1
The Role of the STROBE System Programmer	1-1
The Basic STROBE MVS System	1-2
The Session Management Facility	1-2
The Session Requester	1-2
The Session Manager	1-2
The STROBE Environment	1-3
The Measurement Facility	1-4
How the Measurement Task Gets Control	1-5
The Reporting Facility	1-5
Multisystem Functions	1-6
STROBE-XCF Groups	1-6
Batch-Submission Requests	1-6
Access to STROBE	1-6
STROBE/ISPF	1-6
STROBE Command Language	1-7
Batch Execution	1-7
APMpower	1-7
iSTROBE	1-7
Optional STROBE Features	1-8
Subsystem Support Features	1-8
Language Features	1-8
Development Environment Features	1-9
The Data Collector Interface	1-9
The STROBE Advanced Session Management Feature	1-9

The History Manager	1-9
AutoSTROBE	1-9
CA-Panexec Feature	1-10
Chapter 2. Planning the Installation	2-1
An Overview	2-1
Parameter Data Set	2-2
Planning for Multisystem Environments	2-2
Operating System Requirements	2-2
Setting up the STROBE-XCF Group	2-2
Requirements	2-2
Recommendations	2-2
Planning for Libraries and Data Sets	2-3
Data Set Requirements	2-3
Direct Access Storage Requirements	2-4
Data Set Naming Conventions	2-4
Planning for Enqueues	2-5
Planning for Archiving	2-6
Planning for Security	2-7
Sample Data Sets	2-7
Access Rules	2-7
READ Access to Load Libraries	2-7
Automatic Profile Creation	2-8
Security for Multisystem Environments	2-8
Deciding on a User-Exit Program	2-8
Planning for Products and Features	2-8
Planning for Measurement Session History	2-11
License Management System	2-12
Maintaining Test and Production Versions	2-12
Planning for Migration	2-12
Other Factors to Review	2-13
System Support Requirements	2-13
Virtual Storage Requirements	2-13
In the Measured Program's Address Space	2-13
In the STROBE Address Space	2-14
In the Measurement Services Address Space	2-14
In the User's TSO Address Space	2-14
In Batch Jobs Submitted to Run Reports	2-14
In the Common Storage Area	2-14
DB2 Multi-Thread Environments	2-14
Compatibility Issues	2-15
Checkpoint/Restart During Measurement	2-16
Limited Function Address Spaces	2-16
Address Spaces Using VIO, TCAM, and TSO	2-16
Programs That Attach Tasks as Job Step Tasks	2-16
BatchPipes Support	2-16
SmartBatch	2-17
BPXBATCH	2-17
CA-ACF2	2-17
IPCS for CICS	2-17
CICSplex Sysplex Manager (CP/SM)	2-17
CA-1/External Data Manager	2-17
CA-INTERTEST	2-17
HSM	2-17
Hiperbatch	2-18
FOCUS	2-18
CA-Panexec	2-18

Transrelate Utility Manager	2-18
Dynamic Linklist Support	2-18
PDSMAN	2-18
MIM	2-18
SAS	2-18
BMC STOP X-37	2-18
BMC Application Restart Control	2-18
BMC Data Accelerator	2-18
BMC UNLOAD	2-19
BMC ARC	2-19
Control-D and Control-M	2-19
C Programs with Compiler Option OPTIMIZE(2)	
Inter-Procedural Analysis (IPA)	2-19
VSAM Data Set Management	2-19
Installation Checklists	2-19
Checklist for the STROBE MVS Install	2-19
Checklist for the STROBE ADABAS/NATURAL Feature	2-21
Checklist for the SQL Analysis Feature	2-22
General DB2 Values	2-22
BIND Values for the Target DB2	2-22
GRANT Values for the DB2 Catalog	2-22
Checklist for the iSTROBE Product	2-22
Chapter 3. Installing STROBE	3-1
Before You Install	3-1
Installation Procedures	3-1
Step 1. Install License Management System Certificates	3-2
Step 2. Convert Reserves to Enqueues	3-2
Step 3. Delete or Archive Earlier Versions	3-3
Step 4. Establish Security	3-3
Step 5. Create the Control Data Set	3-3
Step 6. Install the Basic System	3-4
Step 7. Install the ADABAS/NATURAL Feature	3-5
Step 8. Copy Modules to a Link List Data Set	3-8
Maintaining Two Releases	3-9
Running from a Non-Link List Authorized Library	3-9
Copy the STROBE Modules	3-10
Step 9. Refresh Link List	3-11
Step 10. Edit Member ALLOCATE (or Member ALLOCTST for Test)	3-12
Customize the Parameter Data Set	3-12
Customize the Log Data Set	3-19
Customize the Queue Data Set	3-20
Customize the System Message Data Set	3-20
Define the ALLOCATE Member Start Procedures	3-21
Submit the Allocation Job	3-22
Step 11. Edit Member ALLOCASM (or Member ALLOTSTA for Test)	3-22
Customize the Request Group Data Set	3-22
Customize the History Data Set	3-23
Initializing the History Data Set	3-24
Customize the AutoSTROBE Data Set	3-25
Define the ALLOCASM Member Start Procedures	3-26
Submit the Allocation Job	3-27
Step 12. Start the Address Space	3-27
Step 13. Test the Installation	3-28
Measure a Sample Program	3-28
Testing in a Sysplex Environment	3-29
Create a Performance Profile	3-29

Create an Indexed Performance Profile (Optional)	3-31
Step 14. Install the ISPF Libraries.	3-33
STROBE/ISPF Components.	3-34
Create the Libraries	3-34
Modify Option Menu Panel	3-36
iSTROBE FTP Communications	3-37
Accessing the STROBE Packaging Utility.	3-38
Output Data Set Formats	3-38
Installing the Java Targeting Selector (Java Feature Users Only).	3-39
Modify CLISTS	3-40
Modify the ISPF/PDF Panels.	3-44
Test CLISTS	3-44
Step 15. Install the SQL Analysis Feature	3-44
MVS Component Parts.	3-45
Installation Procedures.	3-45
Other Considerations.	3-46
Bind the DB2 Version-Specific DBRMs	3-47
Using Shadow Catalogs (Optional)	3-48
SQL Analysis Feature Performance (Optional).	3-49
Grant DB2 Authority	3-50
Establish Dynamic EXPLAIN Authority.	3-50
Step 16. Verify the SQL Analysis Feature Installation	3-50
Verifying the Installation	3-50
Verifying Product Use.	3-51
Step 17. Install a TSO CLIST to Execute the Session Requester (Optional).	3-52
Step 18. Put into Production (Optional, if you had a test version).	3-52
Migration Utility	3-53
Upgrading to STROBE MVS for Sysplex Release 3.0	3-53
Migrating Queue Data Sets.	3-53
Migrating Group Data Sets (Advanced Session Management Only)	3-54
Migrating the AutoSTROBE Data Set (Advanced Session Measurement Only)	3-54
Migrating Data Sets Across Different Environments	3-57
Migration Requirements	3-58
Migration Utility Parameters	3-58
Setting Migration Utility Parameters	3-59
Local to XCF Environment STROBE Migration.	3-59
Migrating to STROBE Release 3.0 with Advanced Session Management	3-60
Copying Measurement Requests	3-60
Listing Migration Utility Information.	3-60
Chapter 4. Customizing STROBE	4-1
Setting Up Security	4-1
How Access Checking Works	4-1
Enabling and Disabling the Access Filter.	4-2
Defining Access Authority	4-2
Define User Profiles to Control What Can be Measured.	4-2
Define Privileged Access.	4-3
Defining RACF Access Rules	4-3
Initializing the Examples	4-4
Example 1: Access to a Specific System	4-4
Example 2: Access to a Set of Jobs	4-4
Example 3: Privileged Access to a Session Manager.	4-4
Defining CA-Top Secret Access Rules.	4-5
Initializing the Examples	4-5

Example 1: Access to a Specific System	4-6
Example 2: Access to a Set of Jobs	4-6
Example 3: Privileged Access to Session Manager	4-6
Defining CA-ACF2 Access Rules	4-6
Defining a Resource Type for Release 6.2 and 6.3	4-6
Defining Resource Rules	4-7
Initializing the Examples	4-7
Example 1: Access to a Specific System	4-7
Example 2: Access to a Set of Jobs	4-8
Example 3: Privileged Access to Session Manager	4-8
Implementing Security in a Multisystem Environment	4-8
Security Database Recommendation	4-8
Restricting Access on Particular Systems	4-8
Implementing Security for Automatic Profile Creation	4-9
Providing User-Exit Programs	4-9
STROBE Processing	4-9
Processing ADD and ADDGRP Commands	4-9
Processing CHANGE and CHGGRP Commands	4-10
Processing DELETE or DELGRP Commands	4-10
Processing LIST and LISTGRP Commands	4-10
Processing SEND Commands	4-10
User Exits in a Multisystem Environment	4-11
Implementing a User-Exit Program	4-11
Writing a User-Exit Program	4-11
Parameters That Cannot Change	4-12
Parameters That Can Change	4-14
Return Codes	4-15
Linking a User-Exit Program	4-16
Testing a User-Exit Program	4-16
Modifying STROBE Procedures	4-17
Functions of the Procedures	4-17
Changing the Procedures	4-18
Procedures Invoking Language Processors	4-18
Report Parameters	4-18
Customizing the Module Descriptor Program	4-18
Adding and Changing Module Descriptors	4-18
Link Editing a New Copy of STREMODS	4-19
Customizing the Loading of Data Collectors	4-21
CICS System Task Table	4-22

Chapter 5. STROBE Administration and Maintenance	5-1
Administering STROBE	5-1
Submitting Commands to the Session Manager from an Operator's Console 5-1	
Submitting Commands to the Session Requester with a Batch Job	5-1
Managing the STROBE Environment	5-2
Starting STROBE	5-2
Stopping STROBE	5-2
Stopping STROBE with the STOP Command	5-3
Stopping STROBE with the CANCEL Command	5-3
Stopping STROBE with the TERMINATE Command	5-3
Stopping a STROBE XCF Group	5-3
Measurement Services Address Space (MSAS)	5-4
Stopping and Starting MSAS	5-4
Starting and Stopping AutoSTROBE	5-4
Starting and Stopping AutoSTROBE Autoloader Processing	5-4
Establishing and Modifying the STROBE Environment	5-5

.....	Examples5-5
Determining Session Manager Status	5-5
Managing STROBE Data Sets	5-6
Managing the Log Data Set	5-6
Managing a GDG Log Data Set	5-6
Managing a Non-GDG Log Data Set	5-7
Managing the Queue Data Set	5-8
Checking the Status	5-8
Expanding the Queue Data Set	5-9
Synchronizing the Queue Data Set	5-9
Managing the System Message Data Set	5-9
Expanding the System Message Data Set	5-10
Managing the History Data Set	5-10
Reorganizing the History Data Set	5-10
Expanding the History Data Set	5-11
Printing the History Log Data Set	5-11
Managing the AutoSTROBE Data Set	5-12
Managing the Request Group Data Set	5-12
Expanding the Request Group Data Set	5-13
Resequencing Request Element Numbers	5-13
Transferring Ownership of a Request Group	5-13
Controlling Request Group Display	5-13
Maintaining STROBE	5-14
Creating the Maintenance Data Set	5-14
PTF Update Tapes and FTP Files	5-14
Individual PTFs	5-15
Maintaining the SQL Analysis Feature	5-15
Replace the SQL Analysis Feature MVS Component Object	5-15
Relink the SQL Analysis Feature MVS Component	5-16
Bind the DB2 Version-Specific DBRM	5-16
Syntax for Administrative Commands	5-17
The ADDGRP Command	5-18
JOB	5-18
GROUP	5-18
The CLEAN Command	5-18
QUEUE	5-18
COMPLETE	5-18
The COPYGRP Command	5-18
GROUP	5-18
NEWGROUP	5-18
GRPOWNER	5-19
NEWOWNER	5-19
The CHGGRP Command	5-19
ELEMENTNUMBER	5-19
GROUP	5-19
GRPOWNER	5-19
PARAMETER	5-19
The DELGRP Command	5-19
GROUP	5-19
The DELETE Command	5-19
ELEMENTNUMBER	5-20
GROUP	5-20
GRPOWNER	5-20
The ENV Command	5-20
CHECK	5-20
DISABLE	5-20

ENABLE	5-20
STATUS	5-20
The LIST Command	5-20
Controlling LIST Operands	5-21
Specifying the List Format	5-21
Format 1: List Selected Requests	5-22
Format 2: List Requests with a Given Status	5-22
Format 3: List Requests by Ownerid	5-22
Format 4: Show the Status of All Requests	5-23
The LISTGRP Command	5-23
GROUP	5-23
GRPOWNER	5-23
The LMREFRESH Command	5-23
The LOGCLEAR Command	5-23
The LOGSWITCH Command	5-24
The RESYNC Command	5-24
The SHOW Command	5-24
ABENDS	5-24
CSASIZE	5-24
ENQUEUE	5-24
MEMBERS	5-24
SVCS=(nnn,nnn,...)	5-24
QUEUE	5-24
LOG	5-24
DEFAULTS	5-25
The SHOWPTF Command	5-25
TYPE=ISPF	5-25
{rr jobname}	5-25
The STATUS Command	5-25
The SUBMIT Command	5-25
GROUP	5-25
GRPOWNER	5-25
The TERMINATE Command	5-26
The TEST Command	5-26
EXITON	5-26
EXITOFF	5-26
Chapter 6. Error Diagnosis and Trouble Reporting	6-1
Error Messages	6-1
Return Codes	6-1
ABENDs	6-1
Diagnostic Program	6-1
Reporting Problems to Technical Support	6-2
Appendix A. Sample User-Exit Programs	A-1
STROBE Sample User Exit Program	A-1
MVS System IEFU85 Exit	A-12
Appendix B. STROBE Data Sets	B-1
The Control Data Set	B-1
Object Modules	B-3
L1OBJ	B-3
L2OBJ	B-4
Other Members	B-6
STROBE/ISPF Common Libraries	B-6
The Parameter Data Set	B-6
The Log Data Set	B-6

Data Set Characteristics	B-7
Processing Considerations	B-7
As a Generation Data Set	B-7
As a Non-Generation Data Set	B-7
The Queue Data Set	B-7
Data Set Characteristics	B-8
Multisystem Considerations	B-8
Processing Considerations	B-8
Size Considerations	B-9
The System Message Data Set	B-9
Data Set Characteristics	B-9
Processing Considerations	B-9
Size Considerations	B-9
The Request Group Data Set	B-9
Data Set Characteristics	B-10
Processing Considerations	B-10
Serialization Requirements	B-10
Size Considerations	B-10
The History Data Set	B-10
History Data Set Characteristics	B-11
Serialization Requirements	B-11
Size Considerations	B-11
The History Log Data Set	B-11
The AutoSTROBE Data Set	B-11
AutoSTROBE Data Set Characteristics	B-11
Serialization Requirements	B-12
Size Considerations	B-12
The Sample Data Set	B-12
Sample Data Set Characteristics	B-13
Disposition of the Sample Data Set	B-13
The Map Data Set	B-13
The STROBE Performance Profile	B-13
Wide	B-13
Narrow	B-14
Temporary Data Sets	B-14
Appendix C. Symbolic Parameters for STROBE Job Control Procedures	C-1
Appendix D. SQL Analysis Feature Threshold Set	D-1
Index	IX-1

Figures

1-1.	Address Spaces Used by STROBE	1-4
2-1.	DB2 Multi-Thread Above the Line Storage Requirements	2-15
3-1.	Creating the STROBE Control Data Set	3-4
3-2.	Install the Basic STROBE MVS System (Member INSTALL)	3-5
3-3.	STROBE ADABAS/NATURAL Configuration Module (Member STRBANCM).....	3-7
3-4.	(Member STRBANCM Cont.).....	3-8
3-5.	Assemble and Link the Configuration Module (Member STRBANCJ).....	3-8
3-6.	Copy STROBE to a Link List Data Set (Member COPY)	3-11
3-7.	Customize the Parameter Data Set (Member ALLOCATE)	3-14
3-8.	Customize the Log Data Set (Member ALLOCATE)	3-19
3-9.	Customize the Queue Data Set (Member ALLOCATE).....	3-20
3-10.	Customize the System Message Data Set (Member ALLOCATE)	3-21
3-11.	Create the Procedure to Start STROBE (Member ALLOCATE)	3-22
3-12.	Customize the Request Group Data Set (Member ALLOCASM)	3-23
3-13.	Customize the History Data Set (Member ALLOCASM)	3-25
3-14.	Customize the AutoSTROBE Data Set (Member ALLOCASM)	3-26
3-15.	Create the Procedure to Start STROBE (Member ALLOCASM)	3-27
3-16.	Measure a Sample Program (Member TESTSTR).....	3-28
3-17.	Create a Performance Profile (Member REPORT).....	3-30
3-18.	Member INDEXCOB.....	3-32
3-19.	Member INDEXCOB (Concluded).....	3-33
3-20.	Install the STROBE/ISPF Common Libraries (Member SPFINSTL)	3-35
3-21.	Modify the STROBE/ISPF Options Menu (Member STRBDFLT).....	3-36
3-22.	Editing FTP Commands (Member STRBPJCL).....	3-38
3-23.	Install the Java Targeting Selector (Member JAVAINST).....	3-40
3-24.	TSO CLIST for Testing STROBE/ISPF (Member STROBE)	3-41
3-25.	Modify the CLIST for Executing APMpower (Member APMPOWER).....	3-43
3-26.	Install APMpower and iSTROBE SQL Analysis Feature (Member SQLINSTL)	3-46
3-27.	(Member STRBPLOD).....	3-46
3-28.	Identifying a DB2 Load Library (Member STRBPLOD)	3-46
3-29.	Bind for the DB2 Version-Specific DBRM (Member STRANBND).....	3-47
3-30.	SQL Analysis Feature Indexes	3-49
3-31.	Grant Statement for SPUFI (Member STRANGNT)	3-50
3-32.	Execute the Verification Utility (Member SQLAFVER)	3-51
3-33.	Create a Sample with SQLAF (Member STRANSMP)	3-52
3-34.	TSO CLIST to Execute the Session Requester (Member SRCLIST).....	3-52
3-35.	Migration Utility (Member MIGRATEQ).....	3-55
3-36.	Migration Utility (Member MIGRATEG).....	3-56
3-37.	Migration Utility (Member MIGRATEA)	3-57
4-1.	STROBE User-exit Program Parameters	4-16
4-2.	Module Descriptor Program (Member MODCSECT)	4-19
4-3.	Link Edit the STREMODS Program (Member MODLINK).....	4-20
4-4.	CICS System Task Table (Member 9STRBCTSK).....	4-22
4-5.	CICS System Task Table (Member STRBCTSJ).....	4-23
5-1.	JCL for Submitting Commands Through the Session Requester	5-2
5-2.	JCL for Starting STROBE	5-2
5-3.	Copying the Log to a Backup Data Set (Member (STRBCLOG)	5-8
5-4.	History Data Set (Member HSTREORG)	5-11
5-5.	History Data Set (Member HSTPDLOG)	5-12
5-6.	Controlling Request Group Processing	5-14
5-7.	Loading Maintenance Data Set (Member MAINTNCE)	5-14
5-8.	Replace SQL Analysis Feature objects (Member SQLAFRPL)	5-17
B-1.	Log Data Set Characteristics	B-7
B-2.	Queue Data Set Characteristics.....	B-8
B-3.	System Message Data Set Characteristics	B-9
B-4.	Request Group Data Set Characteristics	B-10

B-5.	History Data set Characteristics.....	B-11
B-6.	History Log Data Set Characteristics	B-11
B-7.	Data Set Characteristics	B-12
B-8.	Sample Data Set Characteristics	B-13
D-1.	Assemble and Link Rule Threshold Set (Member STRANRDJ)	D-2
D-2.	Defining Rule Sets (Member STRANRDM)	D-3

Tables

2-1.	Planning Your STROBE Installation	2-1
2-2.	Planning For Data Sets.....	2-3
2-3.	Direct Access Storage Requirements	2-4
2-4.	Planning for Enqueues	2-5
2-5.	Planning for STROBE Features.....	2-9
2-6.	Storage Requirements for STROBE Modules.....	2-13
3-1.	An Overview of the Allocation of Data Sets.....	3-12
3-2.	STROBE Parameter Data Set Options	3-15
3-3.	An Overview of the STROBE/ISPF Installation.....	3-33
3-4.	SQL Analysis Feature Optimizing Index Locations	3-50
3-5.	Parameters for the Migration Utility	3-58
4-1.	RDEFINE and PERMIT Keywords.....	4-4
4-2.	TSS ADD and TSS PERMIT Keywords	4-5
4-3.	\$KEY Keywords.....	4-7
4-4.	Parameters the User-exit Program Cannot Modify	4-12
4-5.	Parameters the User-exit Program Can Modify.....	4-14
D-1.	SQL Analysis Threshold Rules	D-1

Summary of Changes

STROBE and this manual have changed from STROBE MVS from Sysplex Release 2.5. This section discusses the functional changes to STROBE, as well as the changes to this manual, from STROBE MVS for Sysplex Release 2.5 to Release 3.0.

Changes to STROBE

STROBE MVS for Sysplex Release 3.0 introduces:

- AutoSTROBE enhancements
 - AutoSTROBE SMF Candidate Utility-This tool enables you to scan System Management Facility (SMF) data sets to determine which job steps may require AutoSTROBE monitoring and measuring.
 - AutoSTROBE Autoloader- AutoSTROBE now will look at each batch job and its resource consumption as it ends. If the resource consumption is significant enough, the Autoloader will add the job step to the AutoSTROBE monitor candidate list.
 - STROBE - CHANGE AUTOSTROBE REQUEST panel is a new panel that allows you make changes to an AutoSTROBE request.
 - STROBE - AutoSTROBE STATUS panel provides an option to generate Performance Profiles.
 - Installation parameter to set minimum program execution time required to trigger AutoSTROBE basis data collection.
 - STROBE - ADD AUTOSTROBE REQUEST panel contains:
 - fields to disable AutoSTROBE action regardless of whether basis data thresholds are exceeded.
 - a field allowing you specify an ADD ACTIVE request, an ADD QUEUED request or both.
 - a field that allows you to set by individual AutoSTROBE request whether a measurement or warning is generated when thresholds are exceeded.
 - Installation parameter to set input values for AutoSTROBE algorithms that calculate thresholds
 - STROBE - STATUS panel identifies AutoSTROBE-initiated requests
- z/OS Memory Object Storage report detailing information about z/OS memory objects
- Java Virtual Machine (JVM) Support
 - New Java code measurement targeting capabilities through ISPF panels.
 - New Java reports showing new types of JVM Java (Just-In-Time JITted and interpreted) method measurement data at a summary and more detailed levels in batch, CICC and DB2 environments
- DDIO indexing-STROBE can use Compuware DDIO files in addition to map data sets as input to Performance Profile Report indexing.
- zOS V1R3, V1R4 support

- IMS 8.1 support
- CICS Transaction Server Version 2.2 support
- ADABAS through and including V7.1.3 support
- Natural through and including Version 3.1.5 support
- MQSeries Release 5.3 support
- IBM C/C++ Releases for OS/390 V2R9, V2R10 and IBM C/C++ for z/OS V1.1 through V1.4 support

Changes to this Manual

This section describes the changes made to this manual since Strobe MVS for Sysplex Release 2.5.0.

- Updates to the descriptions of the STROBE parameter library to include all new AutoSTROBE parameters.
- Updates to the chapter 2 sections on Planning for STROBE and Compatibility Issues providing more detail.
- Updates to the chapter 3 section describing the STROBE Migration Utility explaining how to migrate an AutoSTROBE data set.
- Updates to all material explaining how the Measurement Services Address Space functions and how to start and stop it and the STROBE Autoloader.
- Updates to chapter 4 providing more detail on using STROBE and various system security packages.
- Updates to the chapter 5 section describing STROBE administrative commands including new material on the LIST and RESYNC commands.
- Updates to all JCL examples to reflect the JCL that is currently shipped.
- Updates to tables describing new modules that are part of STROBE MVS for Sysplex Release 3.0.

Introduction

This manual is a system programmer's guide to the STROBE MVS Application Performance Measurement System, a product designed for IBM MVS/ESA, IBM OS/390, and IBM z/OS systems. This manual describes how STROBE works and provides instructions for its installation, customization, administration, and maintenance.

How This Manual Is Organized

"Summary of Changes" describes the changes made to STROBE and to this manual since the last release.

Chapter 1, "Overview" provides an overview of the STROBE system, with an explanation of the structure and function of its components.

Chapter 2, "Planning the Installation" summarizes the steps to prepare for STROBE installation.

Chapter 3, "Installing STROBE" describes how to install STROBE.

Chapter 4, "Customizing STROBE" explains how to customize the STROBE system.

Chapter 5, "STROBE Administration and Maintenance" discusses STROBE system administration and maintenance.

Chapter 6, "Error Diagnosis and Trouble Reporting" explains STROBE error diagnosis and trouble-reporting procedures.

Appendix A, "Sample User-Exit Programs" provides a sample of a STROBE user-exit program and a sample of an MVS System IEFU85 exit.

Appendix B, "STROBE Data Sets" describes the STROBE data sets.

Appendix C, "Symbolic Parameters for STROBE Job Control Procedures" lists the symbolic parameters for STROBE job control procedures.

Appendix D, "SQL Analysis Feature Threshold Set" describes how to define rule threshold sets for the SQL Analysis Feature.

How to Use This Manual

Use this manual to install, customize, and maintain STROBE.

The STROBE Library

The STROBE base product manuals include:

- *STROBE MVS Concepts and Facilities*, document number CWSTGX3A
STROBE MVS Concepts and Facilities explains how to decide which programs and online regions to measure, when to measure them, and how to interpret the reports in the STROBE Performance Profile.
- *STROBE MVS Messages*, document number CWSTXM3A

STROBE MVS Messages lists all messages and abnormal termination (ABEND) codes, describes how to interpret them, and in many cases suggests a corrective action.

- *STROBE MVS System Programmer's Guide*, document number CWSTXI3A

The *STROBE MVS System Programmer's Guide* explains how to install and maintain STROBE.

- *STROBE MVS User's Guide*, document number CWSTUX3A and the *STROBE MVS User's Guide with Advanced Session Management*, document number CWSTUA3A

The *STROBE MVS User's Guide* explains how to use STROBE to measure application performance. The *STROBE MVS User's Guide with Advanced Session Management* explains how to use STROBE with the STROBE Advanced Session Management Feature to measure application performance. Users who have the STROBE Advanced Session Management Feature will use this manual rather than the *STROBE MVS User's Guide*.

- *STROBE MVS Application Performance Measurement System Quick Reference*

The *STROBE MVS Application Performance Measurement System Quick Reference* is a convenient reference for how to use STROBE and for interpreting the STROBE Performance Profile.

STROBE Feature Manuals

These manuals describe the optional features of the STROBE MVS Application Performance Measurement System. Each manual describes measurement concepts applicable to and specific data made available by the feature.

- *STROBE MVS User's Guide with Advanced Session Management*, document number CWSTUA3A
- *STROBE ADABAS/NATURAL Feature*, document number CWSTUN3A
- *STROBE CA-IDMS Feature*, document number CWSTUR3A
- *STROBE CICS Feature*, document number CWSTUC3A
- *STROBE COOL:Gen Feature*, document number CWSTUG3A
- *STROBE CSP Feature*, document number CWSTUP3A
- *STROBE DB2 Feature*, document number CWSTUD3A
- *STROBE IMS Feature*, document number CWSTUI3A
- *STROBE Interface Feature*, document number CWSTUF3A
- *STROBE Java Feature*, document number CWSTUJ3A
- *STROBE MQSeries Feature*, document number CWSTUM3A
- *STROBE UNIX System Services Feature*, document number CWSTUU3A

Online Documentation

STROBE manuals are available in HTML, Adobe Acrobat PDF format, and IBM BookManager format, on CD-ROM and at Compuware's technical support Web site at <http://frontline.compuware.com>.

Online Help

STROBE products provide the following online information:

- STROBE/ISPF Online Tutorials, Option T from the STROBE/ISPF STROBE OPTIONS menu

- STROBE/ISPF Online Message Facility, Option M from the STROBE/ISPF STROBE OPTIONS menu

Other Compuware Application Performance Management Products

The following products and features work in conjunction with the STROBE MVS Application Performance Measurement System. These tools extend the benefits of application performance management (APM).

iSTROBE

iSTROBE enables you to view and analyze STROBE Performance Profile data on a workstation using a standard Web browser. Easy to install and easy to use, iSTROBE guides you through the performance analysis process and offers recommendations for improving performance. iSTROBE simplifies the performance analysis of applications that you measure with STROBE. For more information on iSTROBE, see the *iSTROBE Getting Started Guide*.

SQL Analysis Feature

The SQL Analysis Feature works in conjunction with STROBE and iSTROBE or APMpower to supply access path analyses and database and SQL coding recommendations for DB2 applications measured by STROBE. The SQL Analysis Feature pinpoints the most resource-consumptive static or dynamic SQL statements, explains why these statements might be inefficient, and provides recommendations to improve the performance of the DB2 application. For more information on the SQL Analysis Feature, see the *STROBE MVS User's Guide* or the *STROBE MVS User's Guide with Advanced Session Management*.

APMpower

The APMpower Application Performance Analysis System extends the benefits of STROBE to application developers who use workstations to develop, test, and maintain MVS applications. Developers employ the APMpower graphical user interface and advanced analytical aids to navigate the Performance Profile, analyze and improve application performance, and share performance knowledge across the IS organization. For more information about APMpower, see the APMpower documentation.

Compuware APM Technical Support

For North American customers, for technical support, please contact the Technical Support department by telephone at (800) 585-2802 or (617) 661-3020, by fax at (617) 498-4010, or by e-mail at strobe-sup@compuware.com.

To access online technical support, visit Compuware's FrontLine page on the World Wide Web at <http://frontline.compuware.com> and select the product "STROBE and APMpower."

For other international customers, please contact your local Compuware office or STROBE supplier.

Compuware APM Training

Compuware's Education Resources Group offers a range of training options for organizations that use STROBE, iSTROBE, and APMpower. To arrange Application Performance Management training, please contact Compuware at 1-800-835-3190 or visit Compuware's Education Resources Group at <http://www.compuware.com/training>

For other international customers, please contact your local Compuware office or STROBE supplier for a complete list of APM Training offerings.

Compuware APM Service Offerings

For North American customers, for information about current service offerings, please contact your local Compuware sales office or call Compuware Corporate Headquarters at 1-800-COMPUWARE (266-7892) or visit Compuware's APM Product page on the World Wide Web at <http://www.compuware.com/products/strobe>.

For other international customers, please contact your local Compuware office or STROBE supplier for a complete list of Services offerings.

APM Installation Assurance

The APM Installation Assurance service assists you in planning for, installing, customizing and using APM products. The service will help you maximize the value and benefits derived from the APM product family.

Consulting engineers work closely with your IT personnel to understand your operating environment and your organization's APM goals. The engineer will assist you in developing a customization and installation plan for STROBE, iSTROBE, and APMpower. The engineer will oversee the installation process and verify product readiness. The engineer will also help set up measurement request schedules, request groups, history records, AutoSTROBE measurement requests, and will verify the installation of the SQL Analysis Feature.

With APM Installation Assurance services, your organization can immediately maximize the value received from your investment in the APM product family. You will also benefit from a fully customized installation that will enhance the product functionality and increase the automation aspects of your APM initiatives.

Application Performance Management Consulting

The Application Performance Management (APM) Consulting services assist you in identifying and resolving specific performance problems in your OS/390 business-critical applications.

Using STROBE, iSTROBE, and APMpower, consulting engineers work closely with your IT personnel to measure an application's performance, identify performance improvement opportunities and make recommendations for implementing solutions.

With APM Consulting services, your organization cannot only resolve problems quickly and effectively, but also gain the skills necessary to prevent application performance degradation in the future.

Application Performance Assessment

The Application Performance Assessment (APA) service assists you in achieving a higher level of performance for your OS/390 business-critical applications.

Using STROBE, iSTROBE, and APMpower, consulting engineers work closely with your IT personnel to evaluate the efficiency of business-critical applications, identify opportunities for improving performance and document the potential savings that can result from implementing recommended solutions.

With APA services, you cannot only improve application performance quickly and effectively, but also gain the knowledge and skills necessary to implement and sustain a process-oriented application performance management (APM) program.

Chapter 1.

Overview

Read this chapter for information about

- the role of the STROBE system programmer
- the basic STROBE MVS system
- how users access STROBE
- optional STROBE features

The Role of the STROBE System Programmer

As the STROBE system programmer, your tasks include

- learning about STROBE, assisting STROBE users when necessary, and answering their questions
- planning for and installing new releases of STROBE
- customizing STROBE to meet the special requirements of your site
- administering STROBE
- applying program temporary fixes (PTFs) between releases
- diagnosing user errors and reporting system malfunctions

You should be familiar with the structure and function of the STROBE system, the organization of the STROBE load module library, and the procedures for installing, changing, administering, diagnosing errors, and reporting problems with the STROBE system. This manual addresses each of these topics.

If you need to	You should read
Understand how STROBE works	This chapter and <i>STROBE MVS Concepts and Facilities</i>
Plan the installation of new releases of STROBE	Chapter 2, "Planning the Installation"
Install new releases of STROBE	Chapter 3, "Installing STROBE"
Customize STROBE to establish access rules for security, add a user-exit routine, add descriptions of modules in the STROBE Performance Profile, modify the parameter data set, or customize how STROBE loads data collectors	Chapter 4, "Customizing STROBE"
Maintain and manage the STROBE system	Chapter 5, "STROBE Administration and Maintenance"
Diagnose errors and report trouble	Chapter 6, "Error Diagnosis and Trouble Reporting"

The Basic STROBE MVS System

The basic STROBE MVS system is made up of the following functions

- session management
- measurement
- reporting

Optional STROBE features augment the basic functions of these components providing support for specific operating environments such as CICS or DB2.

The Session Management Facility

The session management facility enables submission and tracking of measurement requests. The facility includes

- the session requester, which interprets the STROBE command language or STROBE/ISPF selections and communicates with the session manager
- the session manager, which adds and manages measurement requests and measurement sessions
- the STROBE environment, which contains the STROBE SVC, attach filter, common control block interfaces, and subsystem interface

The session management facility uses additional user-provided parameters to control the collection of data by STROBE measurement features, such as attribution data for compiler run-time library routines or performance statistics for CICS.

The Session Requester

The session requester operates in the user's address space, where it receives, edits, and directs measurement requests to the session manager through commands. These commands can originate from STROBE/ISPF, TSO terminals, batch jobs, and system consoles.

The session requester checks each measurement request it receives for proper syntax. It verifies that the submitter has access to the target job and optionally invokes a user-written exit program, which can further validate or modify the request. It directs

- correctly coded requests to the session manager
- messages to the session manager for placement in the log data set and the SYSPRINT data set

The session requester uses the STROBE SVC to initiate specific session manager functions, such as modifying a request.

The Session Manager

The session manager is initialized with the STROBE address space. Once started, the session manager executes as a started task within that address space until it terminates or an IPL takes place.

The session manager receives measurement requests from the session requester. It maintains a STROBE Request Element (SRE) for each measurement request in a local work queue, and records its actions in the log data set. If the target application is already executing, the session manager initiates measurement immediately. Otherwise, measurement begins when the application begins executing. The session manager also communicates with the Measurement Services Address Space (MSAS) to handle the AutoSTROBE function.

The session manager uses the following STROBE data sets:

Log data set	The log data set contains historical information such as messages, dumps, and trace data for use during problem determination.
Parameter data set	The parameter data set contains parameters that control the initialization and operation of STROBE.
Queue data set	The queue data set keeps track of measurement requests that STROBE is currently processing.
System message data set	The system message data set is a repository for messages generated for the measurement sessions.
Request group data set	The request group data set contains all the request groups associated with the STROBE Advanced Session Management Feature.
AutoSTROBE data set	The AutoSTROBE data set contains status and basis data that AutoSTROBE collects for programs that it monitors. The basis data is used to calculate the normal behavior of a monitored program as represented by the first ten monitored executions of the program.

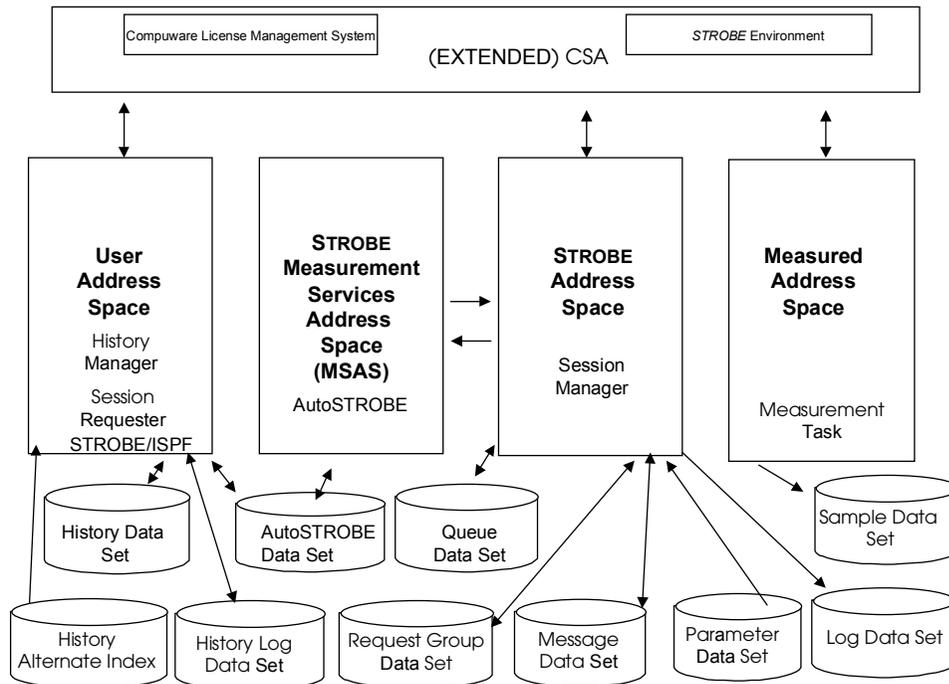
The STROBE Environment

The STROBE environment resides in the extended common system area (ECSA). The STROBE environment consists of

- run-time support modules
- two entries in the SVC table—one for ATTACH (SVC 42) and one for the STROBE SVC (by default, SVC 67)

STROBE executes in four address spaces. Each of these address spaces communicates with the STROBE environment as shown in Figure 1-1.

Figure 1-1. Address Spaces Used by STROBE



The Measurement Facility

STROBE collects performance data by deploying a measurement task within the measured address space. Once activated, the measurement task dynamically allocates a sample data set and begins sampling. The session manager sends sampling control commands to the measurement task. The measurement task responds to these commands and records its actions with messages sent through the STROBE SVC.

STROBE calculates a sampling rate based on the session duration and target sample size specified in the measurement request. Using the rate to determine when to take samples, it sets the timer interval, and waits for the interval to expire. When the timer interval expires, the measurement task regains control and takes a sample. It examines the system control blocks associated with the target program or subsystem and records the following information:

- status of the task activity (active or waiting)
- location of the next instruction that will execute
- load module responsible for the execution or wait
- current online transaction being processed
- devices accessed by the programs executing within the address space
- cylinder accessed (in the case of direct access devices)

STROBE uses this sampling technique repeatedly during the measurement session to collect data about the system resource usage within the address space being measured.

When appropriate, the measurement task calls data collector programs to provide information about the subsystems in which the target program executes, such as:

- transaction identification
- additional module identification
- attributor information

- subsystem-specific information, such as DL/I request data for IMS or SQL data for DB2

The measurement task writes the measurement data to the specified sample data set.

Once STROBE begins measuring a program, it accepts sampling control commands for the measurement session. With these commands, users can suspend, resume, or terminate sampling. Users can also switch from the current sample data set to a new sample data set, freeing the current data set for use in producing a Performance Profile without interrupting the measurement task.

When users stop or terminate measurement sessions, or when target programs terminate, control returns to the measurement task. The measurement task closes the sample data set and ends the measurement session.

How the Measurement Task Gets Control

The STROBE attach filter is an interface that allows STROBE to determine, as tasks and job steps are activated, whether they are targets for STROBE measurement. The measurement task gets control in the following ways:

- If the program being attached is a measurement target, the attach filter includes the STROBE measurement task as part of the job step.
- If the requested target application is already executing, STROBE invokes the STROBE SVC to implant the measurement task.
- When the job to be measured is not active at the time of the request, STROBE attaches the target program and the measurement task as job step tasks.
- If the job is already active, STROBE attaches the measurement task in the same manner as other target program tasks.
- When an active program has no other tasks, STROBE attaches the measurement task as a subtask.

The Reporting Facility

Following a measurement session, STROBE organizes and analyzes the measurement data and produces the Performance Profile. The Performance Profile is a set of reports that shows how user programs and system service routines are using system resources. The reporting facility includes

- The Reporter—a set of procedures that analyze, reduce, and report measurement data to produce Performance Profiles. The Reporter produces the Performance Profile by processing measurement data in the sample data set, together with any indexing data supplied. Users can produce a Performance Profile through STROBE/ISPF or as a batch processing job at any time following the measurement session.

If you are using the STROBE Advanced Session Management Feature, the Reporter also creates and stores measurement session history records.

- The Indexers—a set of language-specific programs that relate hexadecimal addresses and offsets in the object module to procedures or statements in the source program. The Indexers collect information from the output listings of a source language compiler and produce map data sets. When STROBE is provided with a map data set, it correlates execution times with the names and/or statement numbers of the program procedures within the object module. Without indexing, detailed program activity is given by the hexadecimal address relative to the base of each control section. A different indexing program is used for each compiler and code generator. The compilers and code generators supported by the Indexers include
 - ADS/O
 - Assembler
 - C

- C++
- CA-Optimizer
- COBOL
- COOL:Gen
- CSP
- FORTRAN
- Java
- NATURAL
- PL/I
- VisualAge PL/I

Multisystem Functions

When STROBE is initialized on an MVS system with the MVS cross-system coupling facility (XCF), it can optionally become a member of a STROBE-XCF group. STROBE uses XCF for communication and configuration management.

STROBE-XCF Groups

The STROBE-XCF group provides a single point of control for users to manage measurement requests in a multisystem environment. Users can submit measurement requests from one system targeted for other systems in the STROBE-XCF group and can view the status of requests queued to or running on other systems.

When sample and map data sets are shared, a user on one system can produce STROBE Performance Profiles from data gathered by other STROBE-XCF group members operating on other systems in the sysplex.

Batch-Submission Requests

In earlier releases, STROBE had a cross-system measurement function that allowed a measurement to be submitted on one system for execution on another. These requests are now called batch-submission requests. Batch-submission requests require job control statements to route jobs and job output to the user. Therefore, the user monitors these requests through the system output display facility rather than through STROBE.

It is recommended that you implement STROBE-XCF groups for multisystem request management, because they simplify the management of requests among different systems.

Access to STROBE

Users access STROBE through

- STROBE/ISPF
- STROBE command language
- Batch execution (STROBE-supplied procedures)
- APMpower (a workstation product)
- iSTROBE (a browser product)

The following sections describe each of these access methods.

STROBE/ISPF

STROBE/ISPF is an ISPF/PDF dialog that provides users with full-screen control of STROBE measurement, sampling, and reporting facilities.

STROBE/ISPF provides panels that allow users to

- submit a request for STROBE measurement before or during execution of the job being measured
- modify pending measurement requests
- stop and restart measurement sessions
- create and browse Performance Profiles
- browse the STROBE log data set
- navigate a tutorial that demonstrates how to use STROBE
- access online help/messages
- create measurement session history records
- use the AutoSTROBE function

STROBE Command Language

The STROBE command language provides users with an alternate way to perform many of the same tasks as STROBE/ISPF, as well as the additional tasks required for administering STROBE. You can enter a subset of commands from the operator console to facilitate administrative tasks. Also supplied are a set of procedures to help perform standard STROBE tasks.

Batch Execution

STROBE users can submit batch jobs to

- add, monitor, change, and delete STROBE measurement requests
- index the measurement data
- create Performance Profiles

APMpower

APMpower is a workstation product that enables users to submit measurement requests and analyze results interactively from an OS/2 or Windows workstation. Users can

- access application performance-improvement tips and a database of module descriptions
- learn methods for detecting, diagnosing, and resolving application performance issues
- share performance knowledge across the IS organization

iSTROBE

iSTROBE is a web browser interface product that enables you to view and analyze STROBE Performance Profile data on a workstation. Application environments including CICS, DB2, IMS, MQSeries, OS/390 UNIX System Services or VisualAge for Java are supported by iSTROBE. Users can:

- view consolidated reports to quickly identify performance issues
- analyze performance by proceeding through various series of selected reports
- obtain STROBE recommendations for improving performance
- obtain assistance for navigating through profile reports.

Optional STROBE Features

STROBE provides additional features for your data center configuration, such as:

- subsystem features that provide additional performance information for subsystems
- language features that provide additional information for languages
- development environment features that provide performance information for applications developed and deployed in environments such as COOL:Gen
- the STROBE Interface Feature, which supports user-written data collector programs
- the STROBE Advanced Session Management Feature, which significantly improves the process of creating, submitting, and managing measurement requests and provides the capability to analyze performance trends

Subsystem Support Features

STROBE features supplement the STROBE Performance Profile with subsystem-specific information for

- ADABAS
- CA-IDMS
- CICS
- DB2
- IMS
- MQSeries
- UNIX System Services

Each feature includes a data collector program and an attribution program.

- STROBE invokes the data collector program to acquire subsystem-specific data, such as transaction and module identification, in addition to the usual data acquired when the STROBE measurement task takes a sample.
- STROBE invokes the attribution module to identify the user-coded statements that are responsible for the execution of service modules within the subsystem.

Language Features

STROBE language features provide indexing and descriptions of language library functions. The language features are available for:

- CA-Optimizer
- C
- C++
- COBOL
- FORTRAN
- NATURAL
- PL/I

STROBE also furnishes language library routine attribution for:

- ADABAS/NATURAL
- CA-IDMS
- CICS
- COBOL
- CSP
- DB2
- COOL:Gen
- IMS
- C

- C++
- Java
- PL/I
- VisualAge PL/I

Development Environment Features

STROBE provides features that support the following development environments:

- COOL:Gen
- CSP
- Java

The Data Collector Interface

The STROBE Interface Feature supports user-written data collector programs that supplement performance data collected by STROBE and its features. The *STROBE Interface Feature* describes the interface and explains how to code a data collector program.

The STROBE Advanced Session Management Feature

The STROBE Advanced Session Management Feature extends the capabilities of STROBE by facilitating the submission and management of measurement requests. With this feature, users can

- measure multiple, related address spaces concurrently
- measure multiple steps of a job with a single measurement request
- schedule a measurement request to be automatically submitted on specific dates and times
- create, save, and submit groups of measurement requests
- identify a job for STROBE to measure by including a global search character in the jobname
- analyze trends in performance by collecting and displaying measurement history information
- cause STROBE to activate the AutoSTROBE function that automatically measures applications that show poor or unusual performance.

The History Manager

The history manager is available if you have the STROBE Advanced Session Management Feature. The history manager handles all requests for storing, retrieving, updating, and sorting of measurement session history records. You create measurement session history records as you create the Performance Profile. Reviewing measurement session history records enables you to analyze trends in performance of job steps, DBRMs, or transactions.

The history manager also maintains an audit trail by tracking any changes to the history data set. These changes are written to the history log data set.

AutoSTROBE

AutoSTROBE is a new STROBE function that enables you to automatically cause measurement of applications when their behavior deviates from a pre-established norm based on three different categories of performance-related data. AutoSTROBE resides in the Measurement Services Address Space (MSAS) shown in Figure 1-1 on page 1-4 and only is available if you have the STROBE Advanced Session Management Feature. See the *STROBE MVS User's Guide with Advanced Session Management* for information about using AutoSTROBE.

CA-Panexec Feature

The CA-Panexec Feature facilitates reporting on libraries that are managed by CA-Panexec.

Chapter 2.

Planning the Installation

This chapter discusses what you need to know and the decisions you need to make before installing STROBE.

An Overview

Before beginning your STROBE installation, it is helpful to do some planning. Review the following table to get an idea of the types of decisions you need to make. You might find it useful to copy the installation checklist for the basic STROBE MVS system on page 2-19 and the parameter data set table Table 3-4 on page 3-50 for planning purposes.

Table 2-1. Planning Your STROBE Installation

To plan for	Refer to
STROBE parameter data set options, establishing values for the operation of the STROBE address space and the STROBE measurement task	"Parameter Data Set" on page 2-2
Naming conventions, the characteristics of STROBE data sets, and the DASD they require	"Planning for Libraries and Data Sets" on page 2-3
Multisystem installation	"Planning for Multisystem Environments" on page 2-2
STROBE enqueues and the effect of resource serialization products	"Planning for Enqueues" on page 2-5
Archiving STROBE data sets	"Planning for Archiving" on page 2-6
Securing STROBE, deciding how to authorize user and system administrator access	"Planning for Security" on page 2-7
User-exit routine, determining if one is needed and how to code it for STROBE	"Deciding on a User-Exit Program" on page 2-8
Compuware License Management System and the tasks that might be required to use a new version of STROBE and new Features	"License Management System" on page 2-12
Maintaining two versions of STROBE, the location of the modules, and the requirements for unique module and data set names	"Maintaining Test and Production Versions" on page 2-12
Installing one or more of the following features: <ul style="list-style-type: none"> - STROBE Advanced Session Management Feature - STROBE ADABAS/NATURAL Feature - SQL Analysis Feature - iSTROBE - CA-Panexec Feature 	"Planning for Products and Features" on page 2-8
Collecting Measurement Session History	"Planning for Measurement Session History" on page 2-11
Orderly migration from an earlier release, enabling use of existing measurement requests with this release	"Planning for Migration" on page 2-12

You may also want to review

- Operating system environments and programs that STROBE requires. (See “System Support Requirements” on page 2-13.)
- Storage allocations for the affected address spaces. (See “Virtual Storage Requirements” on page 2-13.)
- STROBE use with other products. (See “Compatibility Issues” on page 2-15.)

Parameter Data Set

The STROBE parameter data set, a cataloged, partitioned data set (PDS), is the control center for STROBE operations, providing parameters that govern the use of STROBE at your site. The parameter data set is used solely by the session manager and can be write-protected, with read access defined for the STROBE address space only. At startup, STROBE reads the data set pointed to by the //PARMLIB DD statement in the STRBCSM procedure. During initialization, STROBE parses the data set, records the results in the STROBE log, and closes and deallocates the data set. The parameter data set table Table 3-4 on page 3-50 describes each of the parameters.

Note: STROBE requires the use of a type 2, 3, 4, or 5 SVC for communication between the session manager modules and the measurement task. Because STROBE dynamically installs it, the SVC requires no advance definition.

Planning for Multisystem Environments

Operating System Requirements

MVS/ESA 4.1 or higher with XCF services.

Setting up the STROBE-XCF Group

To set up the STROBE-XCF group, specify a group name for the XCFGROUP keyword of the parameter data set. STROBE establishes the group at initialization.

Requirements

The following are requirements if you are using STROBE in a multisystem environment.

- A given MVS system can have only one member of a STROBE-XCF group.
- All members of a STROBE-XCF group must be at the same release level.
- The following data sets must be shared by all members of a STROBE-XCF group:
 - queue data set
 - system message data set

Recommendations

The following are recommendations if you are using STROBE in a multisystem environment.

- While each member of a STROBE-XCF group can have its own unique functions, it is recommended that each member of the STROBE-XCF group has identical user-exit routines, parameter data sets, and system security databases.
- Allocate the following data sets to be shared across the STROBE-XCF group:

- log data set
- request group data set
- history data sets
- AutoSTROBE data set

Planning for Libraries and Data Sets

The session management facility programs, measurement task modules, data collector programs, the STROBE/ISPF program, and the APMpower MVS Component must reside in an authorized link list data set or an APF-authorized library.

Note: It is strongly recommended using an authorized link list data set, except during testing.

Data Set Requirements

Plan for the following data sets when installing STROBE.

Table 2-2. Planning For Data Sets

STROBE Data Set	Created by	Quantity
Parameter data set	Installation procedure	One
Log data set	Installation procedure and if GDG, session manager	One or more, depending on allocation method
Queue data set	Installation procedure	One
Sample data set	Measurement task	One for each measurement session
Map data set	Indexer	Can be several; however, a map data set may include data from mapping one or more modules.
AutoSTROBE data set	Installation procedure	One. It is recommended that members of the STROBE-XCF group share this data set.
System message data set	Installation procedure	One
Performance Profile	Reporter	One for each Performance Profile requested by a user
Request group data set	Installation procedure	One. It is recommended that members of the STROBE-XCF group share this data set.
History data sets	Installation procedure	Three: the history data set, an alternate index, and a history log data set. It is recommended that members of the STROBE-XCF group share this data set.
Temporary data sets	Indexer, Reporter, STROBE/ISPF	As needed for each execution of an indexer, the reporter, or STROBE/ISPF
APMpower PDF (optional)	Reporter	One for each Performance Profile requested by APMpower
iSTROBE XML file (optional)	Reporter	One for each Performance Profile requested by iSTROBE.

Direct Access Storage Requirements

The direct access storage space requirements for STROBE libraries and data sets are shown in the following table.

Table 2-3. Direct Access Storage Requirements

Libraries and Data Sets	Mbytes	Blocks	Block Size*or CI Size	3380 Tracks
STROBE unauthorized library	9.1	1007	8906	738
STROBE link list library	2.4	270	8906	54
STROBE/ISPF	3.5	1110	8880	131
STROBE control data set	29.7	9516	3120	1101
STROBE parameter data set	0.6	20	3120	2
STROBE log data set (non-GDG)	4.2	1400	3024	20-100
STROBE log data set (GDG)	4.2	1400	22528	20-100
STROBE queue data set	2.8**	N/A	22528	60**
STROBE system message data set	2.8**	N/A	22528	60**
STROBE request group data set	2.8**	N/A	22528	60**
STROBE history data set	3.4**	N/A	22528	150**
STROBE AutoSTROBE data set	3.4**	N/A	22528	150**
STROBE alternate index data set	1.7	N/A	22528	75
STROBE history log data set	4.2	1400	4096	100
<p>*Block sizes are defined for compatibility between 3380 and 3390 direct access storage devices.</p> <p>** Storage requirements vary based on the number of requests in the system and are significantly affected by multisystem operations and the STROBE Advanced Session Management Feature. Consider the following guidelines:</p> <ul style="list-style-type: none"> - For system message and queue data sets, plan on 3K for each request plus 1K for growth. For both data sets plan on 5 requests per control interval, 10 requests per track, and 150 requests per cylinder. - For the request group data set, plan on 2K per request plus 1K for growth. Control interval track and cylinder considerations depend on the number of request group elements you use. - For the history data set, plan on 1K per two measurement session history records, plus 1K for growth. Control interval track and cylinder considerations depend on the number of measurement session history records you plan to collect. - For the AutoSTROBE data set, plan on 3.7K being required for every monitored program. <p>For more information, see the description of these data sets in Appendix B, "STROBE Data Sets".</p>				

Data Set Naming Conventions

When planning data set names, be sure that the names are acceptable to your security product and that they accommodate the following STROBE activities:

- the session manager
 - reads the parameter data set
 - reads from and writes to the queue data set
 - creates (when the data set is allocated with a generation data group) and writes to the log data set
 - reads from and writes to the system message data set
 - reads from and writes to the request group data set
- all measured job step(s)

- allocates and writes to a sample data set in the address space of the measured program or subsystem
- all STROBE users (TSO sessions, batch jobs, the reporter)
 - read sample data sets to produce a Performance Profile
 - delete sample data sets
 - read the STROBE log
 - read and write to the history data set
 - read and write to the AutoSTROBE data set
 - write the Performance Profile
 - write a PDF (APMpower) in the user address space.
 - write an XML file (iSTROBE) in the user address space.
 - read the unauthorized STROBE load library
 - read the log data sets

Planning for Enqueues

STROBE serializes access to resources through the enqueue/dequeue facility, as follows:

- In a single-system environment, STROBE uses the system enqueue. The STROBE version is the major name and the STROBE request element (SRE) address is the minor name.
- In a multisystem environment, STROBE uses the system enqueue to serialize requests across the sysplex. STROBE is the major name and the STROBE-XCF group name is the minor name. All systems in the sysplex must use the global resource serialization (GRS) MVS component.

Note: You must change the exclusive reserves on the log, queue, and system message data sets to enqueues. See “Step 2. Convert Reserves to Enqueues” on page 3-2 of the installation instructions for more details. Other enqueue requirements also are listed in Table 2-4.

Table 2-4. Planning for Enqueues

Major Name (QNAME)	Minor Name (RNAME)	Type of Enqueue	Purpose
STROBECV	Log data set name	Exclusive reserve	Serializes access to the log data set across systems
STROBECV	Queue data set name	Shared reserve	Serializes read access to the queue data set across systems
		Exclusive reserve	Serializes write access to the queue data set across systems
STROBECV	System message data set name	Shared reserve	Serializes read access to the system message data set
		Exclusive reserve	Serializes write access to the system message data set
STROBE- <i>n</i> ¹	SREQUEUE	Enqueue	Serializes access to queue of STROBE requests in CSA during deletion or service of requests
	TPFQUEUE	Enqueue	Serializes the temporary queue
STROBECV	SRE (address of STROBE request element)	Exclusive enqueue	Serializes access to an individual SRE request
STROBE- <i>n</i> ¹	Groupname.sre#	Exclusive enqueue	Serializes access to the sysplex-wide list of SRE requests

Table 2-4. Planning for Enqueues

Major Name (QNAME)	Minor Name (RNAME)	Type of Enqueue	Purpose
STROBE- n^1	MSGQUEUE	Exclusive enqueue	Serializes access to the internal queue of messages
STROBE- n^1	SM-INIT	Exclusive enqueue	Reserves STROBE identifier for one session manager
STROBE- n^1	SM-ACTIVE	Exclusive enqueue	Reserves name for validity check of STROBE address space
STROBE- n^1	SR-ACTIVE	Shared enqueue	Reserves name to ensure clean termination
STROBE- n^1	CVUQUEUE	Exclusive enqueue	Serializes access to information about instances of STROBE, users of STROBE, and active measurement requests
STROBEHM	History data set name	Shared reserve	Serializes read access to the history data set across systems
		Exclusive reserve	Serializes write access to the history data set across systems
STROBEHM	A variable based on the view in which you are working	Shared enqueue	Serializes read access to the history data set across systems
		Exclusive enqueue	Serializes write access to the history data set across systems
STROBEHA	AutoSTROBE data set name	Shared reserve	Serializes read access to the AutoSTROBE data set across systems
		Exclusive enqueue	Serializes write access to the AutoSTROBE data set across systems
STROBEHA	A variable based on the view in which you are working	Exclusive enqueue	Serializes write access to the AutoSTROBE data set across systems
AUTOSTRX	REQLIST	Exclusive enqueue	Serializes access to the list of current AutoSTROBE requests
AUTOSTRX	AMPBMONQ	Exclusive enqueue	Serializes access to the the queue of programs currently being monitored.
AUTOSTRX	AMPBMEAQ	Exclusive enqueue	Serializes access to queue of running program for which AutoSTROBE has initiated or queued a measurement
n^1 identifies a unique copy of a module, enabling you to maintain more than one version of STROBE. It is the fourth character of all authorized STROBE modules. For example, STRTSM identifies the test version of STRBSM, the production version of the module.			

Planning for Archiving

Do not archive the STROBE log data set, queue data set, parameter data set, system message data set, or load libraries. Processing and start-up times are affected if STROBE has to retrieve the data sets from archive.

STROBE takes the following action when it needs an archived sample data set that is defined in PARMLIB:

- If your site uses IBM's DFHSM, STROBE HDELETEs the archived data set and allocates a new data set.
- If your site uses DMS with a non-blank volume ID, STROBE uncatalogs the archived data set and allocates a new one. Newer versions of DMS (8.1.7 and greater) use HDELETE.

- If your site uses another archiving package, STROBE forces a recall of the sample data set.

To specify your archiving package, enter a value for the STRMGR option in the STROBE parameter data set. The option is described Table 3-2 on page 3-15.

Planning for Security

Typically, not all users at a given site are authorized to use STROBE. You can restrict who uses STROBE to measure jobs, on what systems those measurement requests can run, and whether users can measure any job, or just their own. Also, you need to assign privileges to those responsible for administering STROBE and those who are authorized to start STROBE as a batch job.

Sample Data Sets

Sample data sets are dynamically allocated and catalogued from the address space of the job to be measured. Your security system must allow the prefix chosen to be allocated, read, written and catalogued from any job, including production batch and online systems. The prefix you choose must be defined to both the security and storage management subsystems.

Access Rules

Your main task in planning for security is to define access rules to your security product. STROBE provides an access filter you can enable to validate user access to a resource through RACF, CA-Top Secret, or CA-ACF2. If you enable the access filter through the parameter data set, STROBE checks during initialization if one of these security products is installed. STROBE disables the filter if a security product is not found.

The security interface controls access to STROBE for the following users:

- issuers of standard STROBE commands
- issuers of privileged STROBE commands that assist in administering the STROBE system
- jobs or TSO user IDs that start the STROBE address space as a batch job

If you want to set up your site to use the access filter, you need to build access rules for each of these types of users. As a starting point, use the installation checklist (on page 2-19) to record user access authority.

If you do not define access rules, the following defaults apply:

- If the security system is RACF or CA-Top Secret, the filter permits access to the session manager.
- If the security system is CA-ACF2, the filter denies access to the session manager.

“Setting Up Security” on page 4-1 describes how to define access rules for the different security products.

READ Access to Load Libraries

As you are setting up the access rules, be sure that all address spaces to be measured have READ access to any load library from which programs are executed. A lesser level of access, such as EXECUTE, is not sufficient.

Automatic Profile Creation

STROBE provides users with the ability to submit batch jobs that automatically generate Performance Profiles. When submitting measurement requests through STROBE/ISPF, users can specify that STROBE automatically produce the Performance Profile when the measurement session completes. This process is described in the *STROBE MVS User's Guide*.

To use the automatic Performance Profile option, the user ID associated with the STROBE session manager must have authority to

- submit jobs on behalf of any STROBE user
- read sample data sets created by any STROBE user
- create a print file on behalf of any STROBE user

For more information, see “Implementing Security for Automatic Profile Creation” on page 4-9.

Security for Multisystem Environments

In a multisystem environment, STROBE assumes one security database is shared by all systems in the STROBE-XCF group, although access rules for each system can vary. If more than one security database is used, it is possible for requests to have different characteristics on different systems. This could lead to unexpected results.

Deciding on a User-Exit Program

The STROBE user-exit program enables you to control functions that the STROBE access filter does not control. For example, you can code an exit routine to

- check user authorization
- validate installation-dependent values
- enforce installation standards
- supply default values

See “Providing User-Exit Programs” on page 4-9 for instructions on coding these types of STROBE programs

Planning for Products and Features

This section provides some general information about the planning considerations for using any of the STROBE Features. Any time you add a STROBE product or Feature at your site, you must replace the previous license certificate in the Compuware License Management System. To enable you to immediately begin using the Feature without having to restart the STROBE session manager, you can use the LMREFRESH command. See “The LMREFRESH Command” on page 5-23 for details about using the command.

Table 2-5. Planning for STROBE Features

Feature	Planning Considerations
STROBE Advanced Session Management Feature	<ul style="list-style-type: none"> - Allocate the request group data set. - Allocate the history data sets. - Set up default and system cost records and history parameters. See "Planning for Measurement Session History" on page 2-11. - Set the MAXREQ parameter at least 400. This parameter represents the maximum number of requests (active, queued, completed, and deferred and AutoSTROBE) that can be present in a system at any given time. <p>AutoSTROBE Requirements</p> <ul style="list-style-type: none"> - Allocate the AutoSTROBE data set. - You must enable SMF 30 recording. - You must define IEFUSI and IEFACTRT in both the PROGxx and SMFPRMxx members in SYS1.PARMLIB. In SMFPRMxx, both IEFUSI and IEFACTRT must be defined in the EXITS option on the SYS statement. See your OS/390 documentation to obtain more information about defining IEFUSI and IEFACTRT. - You must set certain values in the STROBE parameter data set as explained in Chapter 3 in addition to all other parameters required STROBE. - If you plan to use the AutoSTROBE SMF Candidate Utility, determine the names of the SMF data sets you want to use as input. - For CA-ACF2 users. started task STRBMSAS must be defined to ACF-2.
STROBE ADABAS/NATURAL Feature	<p>To measure NATURAL in CICS, TSO, IMS and batch regions, the configurations module for STROBE ADABAS/NATURAL supplies the following information:</p> <ul style="list-style-type: none"> - the NATURAL nucleus name - the region type - the NATURAL release level - for CICS, the system control block name - for IMS, the NATURAL/IMS interface module names <p>STROBE provides a default version of this module containing the names that SOFTWARE AG supplies for each region, the NATURAL release level, the CICS system control block, and the NATURAL/IMS interface module name. If your site values differ from the defaults, record these values on your STROBE ADABAS/NATURAL checklist on page 2-21.</p>
APMpower MVS Support	<p>To enable APMpower MVS support, the standard libraries of an ISPF/PDF application are required. All of the software is installed with and shares the libraries of STROBE/ISPF.</p>

Table 2-5. Planning for STROBE Features

Feature	Planning Considerations
SQL Analysis Feature	<p>You need to know the details of your installation's values for DB2.</p> <ul style="list-style-type: none"> - general DB2 values <ul style="list-style-type: none"> - the version of DB2 - the name of the DB2 load library - the name of the DB2 exit library - BIND values for the target DB2 subsystem <ul style="list-style-type: none"> - the ID of each DB2 subsystem - the authorization ID of the application plan owner (the owner must have DB2 SYSADM authority, a valid secondary ID with BINDADD authority, or BIND authority on plan STRANANA) - authorization ID of the issuer of GRANT authority (if it differs from the application plan owner) - if SELECT authority is not PUBLIC, the IDs of those who can select the plan
iSTROBE product	<p>For the iSTROBE product, you need to supply parameter values such as Internet Protocol (IP) addresses and the name of the File Transfer Protocol (FTP) program used at your site in the STRBDFLT member shown in Figure 3-21 on page 3-36.</p> <p>Use the checklist on page 2-22 to record these values.</p>
STROBE CA-Panexec Feature	<p>The STROBE CA-Panexec Feature supports job libraries and LNKLST libraries managed by the CA-Panexec product. To give STROBE access to control section information from the program libraries, you need to modify one of the Panexec modules. Member PANEPTF of your STROBE control data set contains instructions on how to use the CA-Panexec PTFs and control statements to install the change.</p> <p>Note: Ensure that the Panexec PTFs listed in the instructions have been applied to your CA-Panexec system.</p>
STROBE CICS Feature	<p>When a new release of STROBE is installed, any CICS region which has a measurement request active in its address space will have to be recycled before the new release can occupy that address space. Even if the STROBE upgrade is complete and LLA is refreshed, if the regions is not recycled, any measurement request will fail.</p>
STROBE MQSeries Feature	<p>If you are using the STROBE MQSeries Feature, you can identify a common user module in your site's environment that applications call to access MQSeries. If you specify a common user module, it must be either:</p> <ul style="list-style-type: none"> - a separate load module from the load module(s) that invoke it. - a separate CSECT in the program being measured - A CSECT in one of the load modules that is mapped for measurement <p>The common user module must comply with standard MVS linkage conventions as follows:</p> <ul style="list-style-type: none"> - During its entire operation, register 13 of the common user module must point to its own save area - Offset 4 (i.e. the second word) of that save area must point to a save area of the caller of the user module. - Offset 12 (i.e. the fourth word) of the caller's save area must contain the return address to the calling routine.
STROBE Java Feature	<p>To install the Java target selector, you have to execute member JAVAINST shown in Figure 3-23 on page 3-40. Compuware recommends that whoever installs the target selector have an active Unix Systems Services (USS) account.</p>

Table 2-5. Planning for STROBE Features

Feature	Planning Considerations
STROBE UNIX System Services Feature	<p>The Session Manager Address Space must have a UID of zero specified in the rules of the security package (RACF, Top-Secret, CA-ACF) you are using. This rule gives STROBE the authority to collect process information for all processes.</p> <p>You must enable SMF 92 recording, so STROBE can collect UNIX System Services data.</p> <p>You can use an MVS system IEFU85 exit program to avoid recording excess SMF 92 data. See "MVS System IEFU85 Exit" on page 12 for more information.</p> <p>To measure with the STROBE UNIX System Services Feature, you must apply the PTF for IBM APAR OW35596.</p> <p>To enable the STROBE Session Manager (running as a superuser) to successfully call the UNIX System Service BPX1SEU, you must define user BPXROOT to RACF, ACF2, or TopSecret. This action will prevent STROBE from issuing message STR6429W.</p> <p>For STROBE to correctly identify activity in the program invoking BPXBATCH, you must apply the appropriate PTF for IBM APAR OW38618. Contact IBM to obtain the PTF that is compatible with the specific release of your operating system.</p> <p>To be able to measure a Webserver address space, you must add the STROBE load library to the RACF program control list. You must also ensure that the CSSLIB is APF-authorized.</p>

Planning for Measurement Session History

With the STROBE Advanced Session Management Feature you can also collect and display historical information for measurement sessions. With this measurement session history, you can more easily identify and track changes in the performance of a particular job step, transaction, or DBRM and relate these changes in performance to changes in operating costs. This capability can be useful in all phases of application performance management.

The key planning tasks to effectively use Measurement Session History are as follows:

1. Decide if you want the history option to appear on the STROBE-OPTIONS menu. Set the HISTORY_PLEX={YES|NO} option in the STROBE parameter data set according to your decision, as described in "Customize the Parameter Data Set" on page 3-12. The default for this parameter is YES.
2. Decide if you want the history collection default for the individual STROBE users on the Performance Profile panels to be 'YES' or 'NO'. Set the HISTORY_USER={YES|NO} option in the STROBE parameter data set according to your decision, as described in "Customize the Parameter Data Set" on page 3-12. The default for this parameter is YES (collect measurement session history for all Performance Profiles).
3. Create the default cost record. STROBE uses these cost records to calculate the changes in operating costs between measurement sessions. To create the default cost record:
 - Determine the timeframe for your peak (or prime) and off-peak (or non-prime) processing periods. These timeframes are called cost categories. Assign values to these time frames by specifying the PRIMESTART and PRIMEEND values, as described in "Initializing the History Data Set" on page 3-24.
 - Assign values for the cost of one CPU hour, the cost of 1000 I/Os, and the cost of 1000 service units by specifying the PRIMECPU, NONPRIMECPU, PRIMEIO,

NONPRIMEIO, PRIMESU, and NONPRIMESU values, as described in “Initializing the History Data Set” on page 3-24.

4. Optionally, assign cost values and cost categories for each system/CPU model in your sysplex. Create cost records using this information. For more information on creating cost records, see Chapter 3 in the *STROBE MVS User's Guide with Advanced Session Management*.

License Management System

STROBE MVS for Sysplex Release 3.0 is fully compatible with the Compuware License Management System (LMS) and future STROBE releases will continue to use LMS. To start STROBE MVS for Sysplex Release 3.0, LMS must be installed and running and the certificate authorizing your use of Release 3.0 must have been imported to the system. The STROBE session manager uses LMS to determine if your site has the license certificates required to run STROBE and the Features available for subsystem measurement. For the production version of STROBE and any Features you are using, the session manager will use the default LMS. Refer to the Compuware Enterprise Common Component documentation for details on installing and configuring LMS. Also see “Customize the Parameter Data Set” on page 3-12 for information about the LMSS parameter and using a test LMS version instead of the default LMS for use with a test version of STROBE.

Maintaining Test and Production Versions

You can maintain your current STROBE release while testing a new one by taking one of the following actions:

- rename the new STROBE modules that are link list resident while copying them to an authorized link list data set, *or*
- rename the new STROBE modules and copy them into an authorized library that is not part of the link list concatenation.

In either case, you must identify

- a unique SVC for your test version
- unique names for STROBE link list components
- modules that refer to the renamed components
- an LMS system you want to use if you require your test STROBE version to use a different LMS than the default LMS.

Planning for Migration

A migration utility can be run to upgrade from previous versions of STROBE MVS for Sysplex. The utility allows you to migrate the contents of the queue, group and AutoSTROBE data sets from STROBE 2.1 and 2.2, STROBE MVS 2.2 and 2.2.1, STROBE 2.3.0 and 2.3.1, STROBE 2.4.0, STROBE 2.5.0 to where you are installing STROBE MVS for Sysplex Release 3.0. You can then use measurement requests you want to retain that were created with version of STROBE MVS. See “Migration Utility” on page 3-53 for instructions on using the utility.

Other Factors to Review

System Support Requirements

STROBE supports MVS/ESA, OS/390 and z/OS and requires the sort/merge utility program or an equivalent sort/merge product.

Virtual Storage Requirements

STROBE requires virtual storage for

- Programs and requests (the amount differs with the number of installed STROBE features).
- Four address spaces. The measurement task and data collector modules reside in the measured address space. The session manager resides in its own address space and has the Measurement Services Address Space (MSAS) for AutoSTROBE services. The STROBE/ISPF interface is called from an ISPF/PDF address space in the user's address space. In turn, the STROBE/ISPF interface calls the indexer and reporter programs. These programs can also run as batch jobs.

In the Measured Program's Address Space

STROBE Modules

The following STROBE program components require virtual storage in the measured program's address space.

Table 2-6. Storage Requirements for STROBE Modules

Functions	Extended Private	Private
Measurement task	300K	100K + buffers
STROBE ADABAS/NATURAL Feature	100K	—
STROBE CA-IDMS Feature	100K	—
STROBE CICS Feature	68K (Plus 34K or value of CAPTBUFF parameter)	2K
STROBE CSP Feature	3.5K	—
STROBE DB2 Feature	148K (Plus size of Capture and SQL buffers) per DB2 thread. If you are running DB in a multi-thread environment read the section called "DB2 Multi-Thread Environments" on page 2-14.	8K
STROBE IMS Feature	68.5K (Plus 8K or value of CAPTBUFF parameter)	—
STROBE COOL:Gen Feature	5.5K	—
STROBE MQSeries Feature	21K	—
STROBE language features	102K	—

Note: The virtual storage required by STROBE at your site can be influenced by factors such as the application you are measuring, the system load, other products, and the STROBE Feature(s) you are using. The values shown should be used as estimates.

Sample Data Set Buffers

Storage for the sample data set buffers resides in the private area of the measured address space. Block and buffer sizes are specified during installation in the STROBE parameter file. The STROBE default is two 8K buffers.

In the STROBE Address Space

The session manager requires a region size of 4096K in the STROBE address space.

In the Measurement Services Address Space

MSAS requires a region size of 4096K in the STROBE address space.

In the User's TSO Address Space

To run Performance Profiles in foreground and use STROBE/ISPF, the following amount of virtual storage is required in the user's TSO address space:

STROBE/ISPF or APMpower MVS Component	6MB
STROBE SQL Analysis Feature	7.5MB

In Batch Jobs Submitted to Run Reports

To run Performance Profiles in background, the following amount of virtual storage is required:

Reporter Component	4MB
STROBE SQL Analysis Feature	7.5MB

In the Common Storage Area

STROBE uses the common storage area (CSA) for common control blocks and requests as follows:

Extended common	63K + 3.7K to 6K per request
Common	1K

DB2 Multi-Thread Environments

If you are running the STROBE DB2 Feature with the capture option on in a multi-thread environment, above the line storage increases because multiple buffers may be created for each thread. For all environments, a default amount of above the line storage is available for the buffers. Then depending on the multi-thread environment being measured by the *STROBE DB2 Feature*, additional above the line storage is required. STROBE uses the value of the CAPTBUFF parameter to calculate the amount of this additional storage. The following lists how STROBE determines how much above the line storage to use.

All Environments

STROBE always requires 960 KB of above the line storage for buffers in a multi-thread environments. The *STROBE DB2 Feature* will retain in above the line storage the first 4 kilobytes of each unique SQL statement that it encounters during the sampling process. Either the value of MAXMEM parameter or a maximum of 2,000 kilobytes will be used.

CICS Environment

Beside the storage space required for every environment, additional buffer space is required based on the following calculation:

$(\text{CAPTBUFF value} \times 2) \times \text{Number of Threads}$

If more than 30 threads are running in the environment, STROBE requires an additional 32KB buffer for each thread over 30.

IMS/TSO/ISPF Environments

Beside the storage space required for every environment, additional buffer space is required based on the following calculation:

$\text{CAPTBUFF value} \times 2$

Call Attachment Facility (CAF) and RRSAS Environments

Beside the storage space required for every environment, additional buffer space is required based on the following calculation:

$(\text{CAPTBUFF value} \times (\text{Number of Threads} + 1))$

Figure 2-1 shows how much above the line storage is required by the *STROBE DB2 Feature* if the DB2 system has 50 threads running, the CAPTBUFF parameter value is set at 100, and STROBE samples 20 unique SQL statements:

Figure 2-1. DB2 Multi-Thread Above the Line Storage Requirements

	CICS	IMS TSO/ISPF	CAF RRSAS
	960 KB	960 KB	960 KB
	4KB x 20 SQL stmts = 80KB	4KB x 20 SQL stmts = 80KB	4KB x 20 SQL stmts = 80KB
Additional Above the Line Storage	2 x 100 (CAPTBUFF value) x 50 (Number of Threads) = 10,000 KB	2 x 100 (CAPBUFF value) = 200 KB	100 (CAPTBUFF value) x 51 (Number of Threads+ 1) = 5100 KB
		Total Above the Line Storage 1240 KB (1.24 megabytes)	
	20 Threads x 32KB buffer = 640KB		Total Above the Line Storage 6140 KB (6.14 megabytes)
	Total Above the Line Storage 11,680 KB (11.68 megabytes)		

Compatibility Issues

Under specific conditions, some restrictions and incompatibilities exist between the STROBE product and other proprietary packages. Identify the restrictions and incompatibilities that apply to your system before you install STROBE.

Checkpoint/Restart During Measurement

Because of an MVS restriction, checkpoints cannot be taken while STROBE is measuring a job step. If the measured program does not use the checkpoint return codes, you can measure the program without any additional changes (you will receive warning messages from MVS stating that checkpoints could not be taken). To avoid a nonzero return code from a CHKPT macro, code RD=NC on the EXEC statement of the target job step. The operating system returns a condition code of zero to the CHKPT macro invocation, and execution continues normally.

Limited Function Address Spaces

STROBE will not measure Limited Function Address Spaces (LFASs). These address spaces are normally created by MVS. They are usually accessed by other address spaces via cross memory services and thus have little or no execution on their own behalf. STROBE/ISPF identifies these address spaces on the Display Active panel.

Address Spaces Using VIO, TCAM, and TSO

When a measurement request is made for a job step that uses VIO, TCAM, or TSO but is not yet executing, STROBE will attach its measurement task as a non-job step task rather than a job step task. If the job step subsequently attaches a job step task, the attach will fail.

Programs That Attach Tasks as Job Step Tasks

When STROBE receives a request to measure a job step that is already executing and whose main user task has already attached one or more tasks as job step tasks, MVS requires that the STROBE measurement task be attached as a job step task. Some MVS resource managers, such as TSO and VIO, assume that when any job step task terminates, the job step itself terminates, and so they terminate resources that belong to *all* tasks, not only those belonging to the terminating job step task. The remaining tasks often ABEND. Therefore, when STROBE measures in such an environment, the measurement task must persist throughout the duration of the step. The measurement task can remain inactive, however, with sampling suspended and the sample data set closed.

Whether STROBE permits measurement of active job steps that attach their tasks as job step tasks depends on:

- the JSTCB option in the parameter data set
- whether the NOQUIT keyword is specified in the measurement request.

When you install STROBE, you can set the JSTCB option to permit these job steps to be always measured, never measured, or measured only when the OPTIONS=NOQUIT operand is specified with the ADD command. For more information, see “Customize the Parameter Data Set” on page 3-12.

Once measurement of such a job step begins, you cannot terminate the request with a QUIT command. Although you can end the measurement session with a STOP command, the STROBE request persists until the job step terminates. You cannot remove the STROBE environment until all requests complete.

BatchPipes Support

When STROBE detects that a data set is managed by BatchPipes, reports such as the Resource Demand Distribution and Data Set Characteristics reports will show CPU usage attributed to the BatchPipes processing. To enable STROBE to report on BatchPipes activity, you must either

- copy the BatchPipes Query Service (module ASFPMSQI) to a LINKLIST data set

- add a STEPLIB for the data set containing module ASFPMSQI (typically, "SYS1.SASFPLIB") to the measured job

You also must have applied IBM PTF UN59629.

SmartBatch

If you specify a step number to measure a job step running in an IBM SmartBatch environment, STROBE may not be able to locate the job step if steps for the job are run on different systems.

BPXBATCH

If you are using the STROBE Unix System Services Feature, and you are measuring a program executed by the IBM BPXBATCH product, you must apply the IBM APAR OW38618.

CA-ACF2

To measure address spaces that are defined to the CA-ACF2 Sub-Auth Facility, modify the rules to allow the following modules to submit jobs:

- STRBCRB for all jobs
- IDMBCRB for IDMS jobs
- DFHBCRB for CICS jobs
- DFSBCRB for IMS jobs

Note: You cannot run the session manager STRBCSM as a batch job with CA-ACF2.

IPCS for CICS

You cannot use IPCS to format a CICS dump taken during STROBE measurement. IPCS can be used to format a dump after a measurement has ended. The presence of STRBCRB prevents the dump formatter from recognizing that the dump is from a CICS region.

CICSplex Sysplex Manager (CP/SM)

If you are using CP/SM and you want take a STROBE measurement of CICS region that uses the CP/SM API, you should consider applying IBM APAR PQ51348 to avoid S306-0C abends.

CA-1/External Data Manager

When you are using the CA-1 or CA-Dispatch products with the External Data Manager (EDM), you must use either a job or data set name as the EDM criteria in the TMOEDMxx definition. Do not use the EDM program name as the EDM selection criteria.

CA-INTERTEST

Attribution reports for STROBE measurements of CICS applications running with CA-InterTest may be incomplete. Run the CICS application without CA-InterTest to obtain complete attribution reports.

HSM

STROBE is incompatible with HSM and will not measure programs executing in the HSM address space.

Hiperbatch

When measuring a job that is using hiperbatch, you must specify the NOJSTCB option on the ADD command to prevent early termination of the user task.

FOCUS

STROBE does not measure programs executing in the address space for Focus Versions 7.01, and 7.03 and 7.05. See MIPS#34002 for 7.01 and MIPS#33993 for 7.03.

CA-Panexec

The STROBE CA-Panexec Feature supports job libraries and LNKLST libraries managed by the CA-Panexec product. CA-Panexec prevents STROBE from accessing program libraries to acquire control section information. To acquire this information, you need to use the STROBE CA-Panexec Feature and latest PTFs issued by CA-Panexec. Member PANEPTF in STROBE.CNTRL contains instructions on the CA-Panexec PTFs.

Transrelate Utility Manager

Do not measure address spaces that use the Transrelate Utility Manager.

Dynamic Linklist Support

For OS/390 Version 1, Releases 3 and higher, do not issue the following command during a STROBE measurement session to prevent creating incorrect measurement or attribution data.

```
SETPROG LNKLST,UPDATE,JOB=xxxxxxxx
```

PDSMAN

If you create the STROBE environment after the PDSMAN LPA option is enabled and disabled, the STROBE attach filter may become disabled. Contact Technical Support for more information.

MIM

STROBE is incompatible with and will not allow you to measure a Multiple-Image Manager (MIM) address space.

SAS

A SAS step may hang if it is measured with a STROBE ADD ACTIVE request. If STROBE detects SAS it will force a QUIT as the final session action.

BMC STOP X-37

STROBE log data sets cannot be managed by the BMC/Boole STOP X-37 product.

BMC Application Restart Control

If you are using the BMC Application Restart Control product, you be using version 2.2.02 before you can use STROBE.

BMC Data Accelerator

If you are using the BMC Data Accelerator, it must be running before the STROBE session manager is started.

BMC UNLOAD

When in a JES3 environment, the BMC Unload+ procedure may multitask and try to close files under a task control block other than the one that opened it. For corrective action, see the “JES3 Considerations” section in the *BMC Unload Installation* manual.

BMC ARC

When measuring a job step that uses any of BMC’s AR/CTL products and has any data sets allocated to VIO, you cannot use STROBE to measure the job step with an ADD QUEUED request. Compuware recommends that you use an ADD ACTIVE request, or modify the job step JCL to remove the VIO allocations prior to using STROBE to measure the job step with an ADD QUEUED request.

Control-D and Control-M

Before attempting to measure a Control-D or Control-M system, contact the vendor, BMC Software, for PTF information.

C Programs with Compiler Option OPTIMIZE(2) Inter-Procedural Analysis (IPA)

When C/370 and AD/Cycle C/370 programs use compiler option OPTIMIZE(2) or IPA, the information STROBE needs to produce detailed statement level associations is removed. Therefore, STROBE relates activity and wait time to the C procedure name, rather than to the user-coded C statement.

VSAM Data Set Management

Do not use third-party vendor products to manage any VSAM data sets used by STROBE. These data sets include the queue, system message, group, history and AutoSTROBE data sets.

Installation Checklists

Use these checklists to help you prepare for the basic STROBE installation, the STROBE ADABAS/NATURAL Feature, the SQL Analysis Feature, and the iSTROBE product.

Checklist for the STROBE MVS Install

STROBE SVC #	
STROBE-XCF group name	
Authorized library (preferably link list) name	
STROBE unauthorized load library name	
Data Sets	
STROBE control data set name	
Unit and volume for shared data sets	
STROBE sample data sets default prefix	

STROBE log data set name (GDG)	
STROBE log data set name (non-GDG)	
STROBE parameter data set name	
STROBE queue data set name	
STROBE system message data set name	
STROBE request group data set name	
STROBE history data set name, alternate index data set name, history log data set name	
AutoSTROBE data set name	
STROBE/ISPF Libraries	
Common panel library name	
Common message library name	
Common skeleton library name	
Common message text library name	
Security	
MANAGER IDs	
ADMIN IDs	
Prefix for security profiles	

Checklist for the STROBE ADABAS/NATURAL Feature

You can create as many as 256 entries to identify the characteristics of each of the regions and NATURAL releases that your ADABAS/NATURAL installation supports.

Environment	{CICS TSO BATCH IMS}
Nucleus name	
System control block name	
NATURAL/IMS Interface module name	
NATURAL release number	
Environment	{CICS TSO BATCH IMS}
Nucleus name	
System control block name	
NATURAL/IMS Interface module name	
NATURAL release number	
Environment	{CICS TSO BATCH IMS}
Nucleus name	
System control block name	
NATURAL/IMS Interface module name	
NATURAL release number	
Environment	{CICS TSO BATCH IMS}
Nucleus name	
System control block name	
NATURAL/IMS Interface module name	
NATURAL release number	
Environment	{CICS TSO BATCH IMS}
Nucleus name	
System control block name	
NATURAL/IMS Interface module name	
NATURAL release number	

Checklist for the SQL Analysis Feature

General DB2 Values

The version of DB2	
DB2 load library name	
DB2 exit library name	

BIND Values for the Target DB2

The ID of each DB2 system.	
The authorization ID of the application plan owner. (The owner must have DB2 SYSADM authority, a valid secondary ID with BINDADD authority, or BIND authority on plan STRANANA.)	

GRANT Values for the DB2 Catalog

The authorization ID of the issuer of GRANT authority (if different from the application plan owner).	
If access authority is not PUBLIC, identify the IDs of those who can execute the SQL Analysis Feature.	

Checklist for the iSTROBE Product

Root directory where XML files will be sent by FTP.	
IP address of the root directory.	
Name of FTP program used at your site.	
Name of IBM code page for translation from EBCDIC to ASCII character set	
Name of translation table if you use non-standard character conversions on file downloads.	

Chapter 3. Installing STROBE

An installed STROBE system comprises

- an unauthorized library containing the standard STROBE system components
- link list library copies of the standard STROBE measurement programs, the session management facility programs, and the STROBE feature measurement programs
- ISPF/PDF panel, skeleton, and message libraries, and a data set that contains an explanation of each message produced by the STROBE system
- job control procedures that are included in the system procedure library

Install the system using the components supplied on the STROBE distribution tape. As new STROBE releases become available, you will receive new distribution tapes. Between releases, you may also receive PTFs on tape.

Before You Install

Before you begin installing STROBE

- review the planning activities in Chapter 2, “Planning the Installation”
- fill out the installation checklist on page 2-19
- review the table of data set parameters that begins on page 3-15

Installation Procedures

Follow these procedures to install STROBE. Refer to the appropriate section for detailed information about each step.

For this procedure	Refer to
Optionally, run the migration utility to convert measurement requests for use with STROBE Release 3.0	“Migration Utility” on page 3-53
License Management System Requirements	“Step 1. Install License Management System Certificates” on page 3-2
Convert reserves to enqueues	“Step 2. Convert Reserves to Enqueues” on page 3-2
Remove the current STROBE environment	“Step 3. Delete or Archive Earlier Versions” on page 3-3
If installing STROBE for the first time, construct definitions for your security package	“Step 4. Establish Security” on page 3-3
Execute IEBCOPY to create the STROBE control data set	“Step 5. Create the Control Data Set” on page 3-3
Install the basic STROBE MVS system	“Step 6. Install the Basic System” on page 3-4

For this procedure	Refer to
If using the STROBE ADABAS/NATURAL Feature, install the mainframe component of this feature	"Step 7. Install the ADABAS/NATURAL Feature" on page 3-5
Copy the STROBE modules to a link list data set by editing and executing member COPY or, for test versions, COPYTEST	"Step 8. Copy Modules to a Link List Data Set" on page 3-8
Refresh the link list	"Step 9. Refresh Link List" on page 3-11
Edit and execute member ALLOCATE or, for test version, ALLOCTST	"Step 10. Edit Member ALLOCATE (or Member ALLOCTST for Test)" on page 3-12
Edit and execute member ALLOCASM or, for test version, ALLOTSTA	"Step 11. Edit Member ALLOCASM (or Member ALLOTSTA for Test)" on page 3-22
Start STROBE by executing STRBSM or, for test versions, STRTSM	"Step 12. Start the Address Space" on page 3-27
Test the following components: <ul style="list-style-type: none"> - measurement - reporting - indexing 	"Step 13. Test the Installation" on page 3-28
Install STROBE/ISPF libraries	"Step 14. Install the ISPF Libraries" on page 3-33
(Optional) Install the mainframe component if using the SQL Analysis Feature	"Step 15. Install the SQL Analysis Feature" on page 3-44
(Optional) Test the mainframe component if using the SQL Analysis Feature	"Step 16. Verify the SQL Analysis Feature Installation" on page 3-50.
If providing access to STROBE through a CLIST, install a CLIST for the session requester. Otherwise, proceed to the next step	"Step 17. Install a TSO CLIST to Execute the Session Requester (Optional)" on page 3-52
Put the new release into production	"Step 18. Put into Production (Optional, if you had a test version)" on page 3-52

Step 1. Install License Management System Certificates

To use any new version of STROBE or any new STROBE Feature or product, the appropriate Compuware License Management System (LMS) certificates must be available on your system. STROBE checks LMS to see that the certificates are correct before it starts. Refer to the Compuware Enterprise Common Component and LMS documentation for full details about using license certificates. The iSTROBE product requires its own LMS certificate separate from the STROBE certificate. Do not select an LMS subsystem name ending in STRx. This conflicts with the STROBE session manager that uses a similar convention to identify with the LMS.

Step 2. Convert Reserves to Enqueues

The way in which you perform this task varies with the resource serialization product used by your site. For example, when using the global resource serialization (GRS) MVS component

- Add the major names to the reserve conversion resource name list (RNL).

- Create a generic RNLDEF statement in the RNL in SYS1.PARMLIB. For example, the following statement converts exclusive reserves to enqueues on the STROBE log, queue, and system message, history and AutoSTROBE data sets:

```
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(STROBECV)
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(STROBEHA)
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(STROBEHM)
```

You can create or change RNLs at IPL or use the MVS system command SET GRSRNL to change the RNLs when the system is active.

Note: If you do not convert exclusive reserves to enqueues on STROBE log, queue, and system message data sets, STROBE results are not always predictable. Read “Planning for Enqueues” on page 2-5 for more information.

Step 3. Delete or Archive Earlier Versions

If you are installing a test version, skip this step and return to it when you are ready to put the test version into production.

If you are replacing your STROBE system or test system with a new release, be sure that the first task you perform is to remove the current environment.

Note: Do not delete any modules until you have brought down the STROBE environment. You might inadvertently delete a module that STROBE requires to bring down the environment.

Then replace or remove all modules from the earlier release. Also delete or archive earlier test versions of STROBE modules.

If you want to install this version of STROBE and leave an older version intact, you should assign a different SVC number in the parameter data set for this version and rename all the STROBE modules using the COPYTEST member as described in “Step 8. Copy Modules to a Link List Data Set” on page 3-8.

Step 4. Establish Security

First time installations need to define profiles to control access to certain resources and functions. For details see “Setting Up Security” on page 4-1.

Note: STROBE security checks can be disabled. For more information, see the description of the FILTER parameter on page 3-16.

Step 5. Create the Control Data Set

The STROBE distribution cartridge contains everything you need to install and test the STROBE system. The tape is prepared as input to the MVS IEBCOPY utility program and is tailored for your site.

The tape is a standard-label tape containing 80-byte records. The volume serial number of the tape is an alphanumeric serial number.

The tape contains the STROBE.SYSTEM file, which includes

- job control statements for installing and testing the system
- object modules ready to be processed by the linkage editor to form the executable versions of the STROBE programs
- libraries that support STROBE/ISPF and APMpower
- test programs

To create the STROBE control data set

1. Use IEBCOPY to copy the distribution tape into the STROBE control data set. Figure 3-1 shows sample job control statements you can use for this purpose. Modify the job control statements as follows:
 - Supply a **JOB** statement.
 - Change **STROBE.CNTL** to the name you want to use for your STROBE control data set.
 - Change **VVVVVV** to match the volume serial number of the distribution tape supplied to you.
2. Submit the job.

Figure 3-1. Creating the STROBE Control Data Set

```

//*STROBE   JOB 0,STROBE
//*
//LOAD      EXEC PGM=IEBCOPY
//SYSPRINT  DD SYSOUT=*
//SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,(2,2))

//SYSUT4    DD UNIT=SYSDA,SPACE=(TRK,(2,2))
//IN        DD DSN=STROBE.SYSTEM,
//          DISP=OLD,UNIT=CART,VOL=SER=VVVVVV
//OUT       DD DSN=STROBE.CNTL,
//          UNIT=SYSDA,VOL=SER=,DISP=(NEW,CATLG,DELETE),
//          SPACE=(3120,(9000,500,20))

//SYSIN     DD *
            COPY INDD=IN,OUTDD=OUT
  
```

Once the STROBE control data set is created, it contains all the tools you need for the installation. For more information about its contents, see “The Control Data Set” on page B-1.

Step 6. Install the Basic System

Member **INSTALL** (Figure 3-2 on page 3-5) contains a job that installs the STROBE system. This job performs the following tasks:

- creates an unauthorized STROBE load library
- link-edits the STROBE system
- applies the latest maintenance

To begin the install

1. Edit member **INSTALL** as follows:
 - Supply a **JOB** statement.
 - Change **STROBLIB** to the name of your unauthorized STROBE library.
 - Change **STROBE.CNTL** to the name of your STROBE control data set.
 - Change **PASSWORD** to the 1- to 16-character value to be used by your STROBE installation. The default is **PASSWORD**.

Note: When the access filter is disabled, you must supply this password each time you run a batch job or TSO CLIST to create, modify, or remove the STROBE environment. (If the access filter is enabled and you have **MANAGER** authority granted through a security package such as RACF, you do not need to supply a password.) See “Setting Up Security” on page 4-1 for more information.

- Change UNIT=SYSDA in the STROBE library allocation step if you want to use another unit name for this library.
 - Verify that the specified assembler in the PASSWORD step is correct for your installation.
2. Submit the job. A return code of 4 from the UPDATE step is acceptable. All other steps should complete with a return code of 0.

Figure 3-2. Install the Basic STROBE MVS System (Member INSTALL)

```

//ALLOCATE EXEC PGM=IEFBR14
//STRLIB DD DSN=STROBLIB,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(500,50,60)),DCB=(BLKSIZE=8906,RECFM=U),
// UNIT=SYSDA,VOL=SER=
//*
//LINK PROC LPARM=NCAL
//LKED EXEC PGM=IEWL,REGION=1024K,
// PARM='XREF,LIST,LET,&LPARM'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD DD DISP=SHR,DSN=STROBLIB
//SYSLIN DD DISP=(SHR,PASS),DSN=&DSN
// PEND
//*
//LINK1 EXEC LINK,DSN='STROBE.CNTL(L10BJ)'
//*
//LINK2 EXEC LINK,DSN='STROBE.CNTL(L20BJ)',
// LPARM='NCAL,RENT,REUS,REFR'
//*
//*****
//* CREATE THE STROBE PASSWORD MODULE
//*
//*****
//PASSWORD EXEC PGM=ASMA90,
// PARM='OBJECT,NODECK'
//* PASSWORD EXEC PGM=IEV90,
//* PARM='LOAD,OBJECT,NODECK'
//SYSLIB DD DSN=STROBE.CNTL,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(800,400))
//SYSUT2 DD UNIT=SYSDA,SPACE=(1700,(800,400))
//SYSUT3 DD UNIT=SYSDA,SPACE=(1700,(800,400))
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=&&SYSGO,UNIT=SYSDA,DISP=(MOD,PASS),
// SPACE=(3200,(80,40)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN DD *
STRBPASS PASSWD=PASSWORD
END
//*
//LINK3 EXEC LINK,DSN='&&SYSGO'
//LKED.SYSLIN DD
// DD *
NAME STRBCMJT(R)
//*
//UPDATE EXEC PGM=IMASPZAP,PARM=IGNIDRFULL
//SYSLIB DD DISP=SHR,DSN=STROBLIB
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

```

Step 7. Install the ADABAS/NATURAL Feature

Those sites using the STROBE ADABAS/NATURAL Feature must customize the STROBE ADABAS/NATURAL configurations module. The configurations module defines the names of the CICS, IMS, TSO, batch NATURAL regions, and shared NATURAL modules to the STROBE NATURAL data collector. The module supplies the following information to the STROBE ADABAS/NATURAL Feature:

- The NATURAL nucleus name.
- The region type.
- The NATURAL release level.

- For CICS, the name of the NATURAL/CICS system control block, formerly defined as the thread control block. This name follows the naming convention *ncipCB*, where *ncip* is the value of the parameter PREFIX in the NATURAL/CICS parameter module.
- For IMS, the name of the NATURAL/IMS interface module name. This name follows the the naming convention *NIIVrmxx*, where *vrn* is the NATURAL version, release and modification level (for example 313) and where *xx* is the module name using the following conventions:
 - IF the NATURAL/IMS main interface module
 - BM the NATURAL/IMS batch message program (BMP) front-end
 - OB the NATURAL/IMS message-oriented (NTRD) front-end
 - FR the NATURAL/IMS conversational message processing program (MPP) front-end
 - NC the NATURAL/IMS non-conversational message processing (MPP) front-end

In addition an entry must be coded for the ADABAS/IMS interface, ADALNI.

A default version of the configurations module is provided with the STROBE ADABAS/NATURAL Feature and requires no special installation. This default version contains the NATURAL region definitions shown in member STRBANCM (Figure 3-3 on page 3-7).

To change or add entries to the configurations module

1. Edit member STRBANCM using values from the checklist you completed in Chapter 2, "Planning the Installation".

The definition section starts with

```
STRBANCE TYPE=INITIAL
```

and ends with

```
STRBANCE TYPE=FINAL
```

Each configuration definition starts with

```
STRBANCE TYPE=ENTRY
```

and contains values for the following keywords:

This keyword	Defines
NUCNAME	The 8-character NATURAL shared nucleus name
TCMNAME	The 8-character system control block module name, if the NATURAL nucleus executes in a CICS environment
NIINAME	The 8-character NATURAL/IMS interface module name, if the NATURAL nucleus executes in an IMS environment.
RELEASE	The 3-digit release number of the NATURAL nucleus, without the intervening periods (for example, enter 223 for release 2.2.3)
ENVIRON {CICS TSO BATCH IMS}	The environment in which the NATURAL nucleus executes

2. Edit the assemble-and-link JCL in member STRBANCJ (Figure 3-4 on page 3-8).
 - supply a **JOB** statement
 - change **STROBE.CNTL** to the name of the STROBE control data set
 - change **STROBLIB** to the name chosen for your unauthorized STROBE library

- make any additional changes to customize the job control statements for your installation

3. Execute the job.

Figure 3-3. STROBE ADABAS/NATURAL Configuration Module (Member STRBANCM)

```

STRBANCE TYPE=INITIAL
SPACE
STRBANCE TYPE=ENTRY,
    ENVIRON=BATCH,
    NUCNAME=NAT314SH,
    RELEASE=314,
SPACE
STRBANCE TYPE=ENTRY,
    ENVIRON=BATCH,
    NUCNAME=NAT314BA,
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY,
    ENVIRON=TSO,
    NUCNAME=NAT314SH,
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY,
    ENVIRON=TSO,
    NUCNAME=NTI314RE,
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY
    ENVIRON=CICS
    NUCNAME=NAT314SH
    TCMNAME=NC314CB
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY
    ENVIRON=CICS
    NUCNAME=NCI314RE
    TCMNAME=NC314CB
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY
    ENVIRON=IMS
    NUCNAME=NAT314SH
    NI INAME=NI I314IF
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY
    ENVIRON=IMS
    NUCNAME=NAT314SH
    NI INAME=NC314BM
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY
    ENVIRON=IMS
    NUCNAME=NAT314SH
    NI INAME=NC314OB
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY
    ENVIRON=IMS
    NUCNAME=NAT314SH
    NI INAME=NC314FR
    RELEASE=314
SPACE
STRBANCE TYPE=ENTRY
    ENVIRON=IMS
    NUCNAME=NAT314SH
    NI INAME=NC314NC

```

Figure 3-4. (Member STRBANCM Cont.)

```

SPACE
STRBANCE TYPE=ENTRY
        ENVIRON=IMS
        NUCNAME=NAT314SH
        NIINAME=ADALNI
        RELEASE=314

SPACE
STRBANCE TYPE=ENTRY,
        ENVIRON=BATCH
        NUCNAME=NAT313SH,
        RELEASE=313

SPACE
STRBANCE TYPE=ENTRY,
        ENVIRON=BATCH,
        NUCNAME=NAT313BA,
        RELEASE=313

SPACE
STRBANCE TYPE=ENTRY,
        ENVIRON=TSO,
        NUCNAME=NAT313SH,
        RELEASE=313

SPACE
STRBANCE TYPE=ENTRY,
        ENVIRON=TSO,
        NUCNAME=NTI313RE,
        RELEASE=3

SPACE
STRBANCE TYPE=ENTRY
        ENVIRON=CICS
        NUCNAME=NAT313SH
        TCMNAME=NC313CB
        RELEASE=313

:
:
SPACE
STRBANCE TYPE=ENTRY,
        ENVIRON=CICS,
        NUCNAME=NCI231RE,
        TCMNAME=NCI41CB
        RELEASE=231

SPACE
STRBANCE TYPE=FINAL

```

Figure 3-5. Assemble and Link the Configuration Module (Member STRBANCM)

```

/*
/ASM      EXEC  HLASMCL,
//        PARM='XREF(SHORT),OBJECT'
/**ASM    EXEC  ASMHCL,
//        PARM='XREF(SHORT),LOAD,OBJECT'
//C.SYSLIB DD   DSN=STROBE.CNTL,DISP=SHR
//C.SYSIN  DD   DSN=STROBE.CNTL(STRBANCM),DISP=SHR
/*
//L.SYSLIN DD
//        DD   *
//        NAME STRBANCM(R)
/*
//L.SYSLMOD DD DSN=STROBLIB,UNIT=,DISP=SHR

```

Step 8. Copy Modules to a Link List Data Set

All STROBE measurement and session management modules must reside in an authorized data set, preferably a link list. Member COPY of the STROBE control data set (Figure 3-6 on page 3-11) contains a job to copy selected modules from the STROBE unauthorized library into an authorized link list data set. You can copy the modules with the default names (COPY) or you can copy them with alternate names (COPYTEST) to facilitate testing while your current version of STROBE is in use.

Maintaining Two Releases

You can maintain one STROBE release as you test another one by

- renaming the new STROBE modules that are link list resident while copying them to an authorized link list data set, or
- renaming the new STROBE modules and copy them into an authorized library that is not part of the link list concatenation.

In either case, take the following actions:

- choose an SVC for your test version that differs from the one in your production version.
- rename the STROBE modules as they are copied to link list or to an authorized library.
- in all jobs and CLISTs that call the test version
 - change STRBCSR to STRTCSR
 - change STRBISPF to STRTISPF
 - change STRBSPFM to STRTSPFM
- If you want to use another License Management System other than the default system with your test version, you need to identify the LMS with the LMSS parameter. See “Customize the Parameter Data Set” on page 3-12 for information about this parameter.
- If you are using the data set packaging utility described in the *STROBE MVS User's Guide* with a test version of STROBE, you must use the 3.3 ISPF option to copy STRBCSR to STRTCSR in the unauthorized load library to support the packaging utility and a test STROBE version.

If you are replacing a release of STROBE

- make sure the STROBE environment is not active
- optionally, remove old modules from the link list library

Running from a Non-Link List Authorized Library

Although it is recommended that you execute STROBE from an authorized link list library, you can also use an authorized library that is not a link list data set. If you do not use a link list, you must modify the job control statements for each job step you want to measure.

Make the following changes to run STROBE from an authorized library that is not a link list data set:

- In the STROBE parameter data set, change the CVLIB entry to APFLIB. See page 3-16 for more information about the CVLIB parameter.
- Include the following DD statements in each job step you want to measure:
 - DD statement of STRMTLIB to name the authorized library.
 - DD statement for the authorized library in the STEPLIB concatenation. The other libraries in the STEPLIB concatenation do not have to be authorized.

The following example shows the JCL statements to add to each job step you plan to measure:

```
//STEPLIB DD DSN=user.loadlib
//          DD DSN=strobe.apf.loadlib
//STRMTLIB DD DSN=strobe.apf.loadlib
```

where

- *user.loadlib* is the name of your STROBE user load library
- *strobe.apf.loadlib* is the name of your APF-authorized library

Copy the STROBE Modules

Customize the job control statements in member COPY (Figure 3-6 on page 3-11) or member COPYTEST (if you are renaming and copying the link list components into a link list).

1. Make the following edits:
 - supply a **JOB** statement
 - change **STROBLIB** to the name of your unauthorized STROBE library
 - change **LINKLIB** to the name of an authorized link list data set
 - change **UNIT=SYSDA** to a more appropriate unit if needed
2. Execute the job. All return codes should be zero.

Figure 3-6. Copy STROBE to a Link List Data Set (Member COPY)

```

//COPY EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=STROBLIB
//SYSUT2 DD DISP=SHR,DSN=LINKLIB
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SYSIN DD *
COPYMOD OUTDD=SYSUT2,INDD=((SYSUT1,R))
SELECT MEMBER=STRBVPH1,STRBVPH2,STRBVPH3
SELECT MEMBER=STRBMSAS,STRBUT24,STRBTUSR
SELECT MEMBER=STRBATTR,STRBSAM1,STRBPC1
SELECT MEMBER=STRBCOPT,STRBMAPS,STRBSLMX
SELECT MEMBER=STRBVM11,STRBVM12,STRBGMIM
SELECT MEMBER=STRBAUTO,STRBAEVT,STRBAMON
SELECT MEMBER=STRBAUSI,STRBACTR,STRBASVC
SELECT MEMBER=STRBAE15,STRBAE35,STRBCSR
SELECT MEMBER=STRBALSJ,STRBSNAM,STRBAUHM,STRBAUDT
SELECT MEMBER=STRBCLPA,STRBCLPB,STRBCRB
SELECT MEMBER=STRBCPHO,STRBCINI,STRBCTXR
SELECT MEMBER=STRBCXAC,STRBCMJT,STRBCSM
SELECT MEMBER=OSTAREDC
SELECT MEMBER=STRBALSR FOR LANGUAGE ATTRIBUTION
SELECT MEMBER=STRAVSM FOR LANGUAGE ATTRIBUTION
SELECT MEMBER=STRBALCB FOR COBOL ATTRIBUTION
SELECT MEMBER=STRBACSP FOR CSP ATTRIBUTION
SELECT MEMBER=STRBAIEF FOR COOL:GEN ATTRIBUTION
SELECT MEMBER=STRBALPL FOR PL/I ATTRIBUTION
SELECT MEMBER=STRBALC FOR C ATTRIBUTION
SELECT MEMBER=STRBACPP FOR C ATTRIBUTION
SELECT MEMBER=STRBALJA FOR JAVA ATTRIBUTION
SELECT MEMBER=STRBJDC1 FOR JAVA ATTRIBUTION
SELECT MEMBER=STRBJDC2 FOR JAVA ATTRIBUTION
SELECT MEMBER=STRBJVMA FOR JAVA ATTRIBUTION
SELECT MEMBER=DFHBCRB,STRBDC17 FOR CICS
SELECT MEMBER=STRBDC21,STRBDC31 FOR CICS
SELECT MEMBER=STRBDC32,STRBDC33 FOR CICS
SELECT MEMBER=STRBDC41,STRBDC51 FOR CICS
SELECT MEMBER=STRBDC52,STRBDC53 FOR CICS
SELECT MEMBER=STRBDC61,STRBDC62 FOR CICS
SELECT MEMBER=STRBDFHS FOR CICS
SELECT MEMBER=STRBADFH,STRBDF17 FOR CICS
SELECT MEMBER=STRBDF21,STRBDF31 FOR CICS STATISTICS
SELECT MEMBER=STRBDF32,STRBDF3 FOR CICS STATISTICS
SELECT MEMBER=STRBDF41,STRBDF51 FOR CICS STATISTICS
SELECT MEMBER=STRBDF52,STRBDF53 FOR CICS STATISTICS
SELECT MEMBER=STRBDF61,STRBDF62 FOR CICS STATISTICS
SELECT MEMBER=STRBTXP FOR CICS
SELECT MEMBER=STRBACSQ,STRBDCSQ FOR MQSERIES
SELECT MEMBER=DFSBCRB,STRBADLI FOR IMS
SELECT MEMBER=STRBDFSR,STRBDFSM FOR IMS
SELECT MEMBER=STRBDFSD,STRBASSA FOR IMS
SELECT MEMBER=STRBADBC FOR DBCTL
SELECT MEMBER=STRBDSNA,STRBDSNY,STRBDSNS FOR DB2
SELECT MEMBER=STRBISPF,STRBTDAX FOR ISPF
SELECT MEMBER=STRBSPFM FOR APMPOWER
SELECT MEMBER=IDMBCRB FOR IDMS
SELECT MEMBER=STRBIDMS,STRBADML FOR IDMS
SELECT MEMBER=STRBDM10,STRBDM12 FOR IDMS
SELECT MEMBER=STRBID10,STRBID12 FOR IDMS
SELECT MEMBER=STRBPR10,STRBPR12 FOR IDMS
SELECT MEMBER=STRBPANE FOR PANEXEC
SELECT MEMBER=STRBNATL,STRBADAB,STRBNATB FOR ADABAS/NATURAL
SELECT MEMBER=STRBADAR,STRBANCM FOR ADABAS/NATURAL

```

Step 9. Refresh Link List

After you copy STROBE to a link list data set, refresh LLA. Use the MVS system command

```
MODIFY LLA,REFRESH
```

Step 10. Edit Member ALLOCATE (or Member ALLOCTST for Test)

Member ALLOCATE contains a job that creates the queue, log, and system data sets needed for the execution of the session manager.

Note: If you are using the STROBE Advanced Session Management Feature, you need to edit JCL in another member named ALLOCASM in addition to the ALLOCATE member. Instructions for editing this member are contained in “Submit the Allocation Job” on page 3-22

If you are installing STROBE for the first time, proceed with the instructions in this section to allocate the data sets. If you are upgrading from a previous version of STROBE and want to use measurement requests that were created with the previous version, you need to perform the procedure described in the section “Migration Utility” on page 3-53. The sample JCL provided with the utility allocates queue and if they exist, can also allocate request group data sets. If you plan to use the utility, do not allocate the queue and group data sets before you have completed the migration.

Note: You do not have to migrate history data sets if you are using STROBE to keep a measurement history.

Table 3-1. An Overview of the Allocation of Data Sets

For these customizations	Refer to
Define installation-dependent values that govern the function of STROBE at your site.	“Customize the Parameter Data Set” on page 3-12
Create the STROBE log data set as a unique file or as a data set within a generation data group. It is strongly recommended that you create the log as a generation data set.	“Customize the Log Data Set” on page 3-19
Create a VSAM key-sequenced data set for the STROBE queue data set.	“Customize the Queue Data Set” on page 3-20
Define a VSAM key-sequenced data set for the STROBE system message data set.	“Customize the System Message Data Set” on page 3-20
Define a VSAM key-sequenced data set for the AutoSTROBE data set.	“Customize the AutoSTROBE Data Set” on page 3-25
Define the procedure that allows you to start STROBE from an operator's console or as a batch job.	“Define the ALLOCATE Member Start Procedures” on page 3-21.
Finalize the allocation of data sets.	“Submit the Allocation Job” on page 3-22

If you are creating a test version of STROBE, make the modifications to member ALLOCTST.

Note: When you are editing the ALLOCATE member JCL, you need to supply values for storage unit and volume. If you are using System Managed Storage, determine the storage class and management class instead.

Customize the Parameter Data Set

The STROBE parameter data set provides installation-dependent values that govern the function of STROBE at your site. Use the values you established in your installation checklist (page 2-19) when planning the installation.

- Modify the job control statements for job step PARMLIB (Figure 3-7 on page 3-14)
 - Change **STROBE.PARMLIB** to the name chosen for your STROBE parameter data set
 - Change **UNIT=SYSDA** to the unit for your parameter data set

- If required, supply a value for **VOL=SER=**
- Edit the values in the SYSIN input stream. These values initialize the STROBE parameter data set. Use the data set names from your installation checklist as input.
 - Change **SVC** to the SVC number you are assigning for STROBE
 - Change **DSNAME** to the default prefix for your site.
 - If using a storage manager and you are archiving sample data sets, set **STRMGR** to specify the storage package at your site.
 - If using a STROBE-XCF group in a multisystem environment, modify the following parameters:
 - Supply the group name for **XCFGROUP**
 - Increase the value of **MAXREQ** to 400
 - Retain or change the default values, as needed. A description of the parameters begins Table 3-2 on page 3-15.
 - Depending on the type of log data set you are using, you may have to change additional values. See the parameter data set values table for more information.
 - If you are not using STROBE profile history, change the values of the **HISTORY_PLEX** and **HISTORY_USER** from YES to NO in the parameter data set.

Figure 3-7. Customize the Parameter Data Set (Member ALLOCATE)

```

//PARMLIB EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=STROBE.PARMLIB,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(3120,(20,5,5),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSDA,VOL=SER=
//SYSIN DD DATA,DLM=ZZ
*/ ADD NAME=STROBE
SVC=67 /* SVC NUMBER */
ARCVID= /* ARCHIVE VOLUME ID FOR DASD MGT */
ATTFLTR=ENABLE /* ENABLE ATTACH FILTER AT STARTUP */
AUTOSTROBE=YES /* ENABLE AUTOSTROBE */
AUTOACTION=MEASURE /* ACTION TAKEN WHEN BASELINE EXCEEDED */
AUTOBDMIN=2 /* MIN ELAPSED RUN TIME FOR BASIS DATA */
AUTOCANDIDATE=YES /* AUTOCANDIDATE PROCESSING ENABLE/DISABLE */
AUTOCANTCB=020 /* MIN TCB TIME FOR JSPS CANDIDACY */
AUTOCANELAP=0060 /* MIN ELAPSE TIME FOR JSPS CANDIDACY */
AUTOCANEXCP=1000000 /* MIN EXCPS FOR JSPS CANDIDACY */
AUTOCANFREQ=030 /* MIN RUNS PER WEEK FOR JSPS CANDIDACY */
AUTOELAPPERCNT=50 /* THRESHOLD FOR ELAPSED TIME PERCENTAGE */
AUTOTCBPERCNT=50 /* THRESHOLD FOR TCB TIME PERCENTAGE */
AUTOIOEVENT1=(100,75,3) /* ABNORMAL I/O THRESHOLD, FIRST EVENT */
AUTOIOEVENT2=(0,0,0) /* ABNORMAL I/O THRESHOLD, SECOND EVENT */
AUTOIOEVENT3=(0,0,0) /* ABNORMAL I/O THRESHOLD, THIRD EVENT */
AUTOIOEVENT4=(0,0,0) /* ABNORMAL I/O THRESHOLD, FOURTH EVENT */
AUTOIOEVENT5=(0,0,0) /* ABNORMAL I/O THRESHOLD, FIFTH EVENT */
AUTOMAX=10 /* MAX NUMBER JOBS MONITORED BY AUTOSTROBE */
BASELIN=2 /* MINIMUM PERCENTAGE OF EXECUTION SAMPLES */
BLKSIZE=8192 /* BLKSIZE OF SAMPLE DATA SET */
BUFNO=2 /* NUMBER OF BUFFERS FOR SAMPLE DATA SET */
CVLIB=LINKLIST /* STROBE RUNS FROM THE LINKLIST */
DATACLAS= /* SMS DATACLAS FOR SAMPLE D S ALLOCATION */
DATE=MMDDYYYY /* DATE FORMAT (DEFAULT IS MMDDYYYY) */
DATESEP=/ /* DATE SEPARATOR (DEFAULT IS / ) */
DSNAME=STROBE /* SAMPLE DATA SET NAME PREFIX */
DSNSUFFIX= /* SAMPLE DATA SET NAME SUFFIX */
FILTER=DISABLE /* DISABLE EXTERNAL SECURITY CALLS */
HISTORY_PLEX=YES /* ENABLES HISTORY PROFILE COLLECTION */
HISTORY_USER=YES /* USER DEFAULT FOR PROFILE HISTORY COLLECT */
JSTCB=ALLOW /* ALLOW/ALWAYS/NEVER JOB STEP MEASUREMENTS */
LINEPG=60 /* REPORTER LINES PER PAGE */
LMSS= /* LICENSE MANAGEMENT SUBSYSTEM ID */
LOG_MAX=30 /* MAXIMUM NUMBER OF TRACKS */
LOG_SPACE=(TRK.(10,10)) /* SPACE ALLOCATIONS FOR GDG LOG */
LOG_UNIT=SYSDA /* LOG DATA SET UNIT ALLOCATION */
LOG_VOL= /* LOG DATA SET VOLUME ALLOCATION */
LPALIB=NO /* OBTAIN MAPPING DATA FROM LPALIB */
MAXACT=20 /* MAXIMUM NUMBER OF ACTIVE REQUESTS */
MAXREQ=200 /* MAXIMUM NUMBER OF REQUESTS IN QUEUE */
MGMTCLAS= /* SMS MGMTCLAS SAMPLE D S ALLOCATION */
MSASPROCNAME=STRMSAS /* MSAS PROC NAME */
PROFILE= /* ACCESS FILTER HIGH LEVEL PROFILE ID */
REMV_OLD=NO /* SPECIFY YES IF REMOVING AN OLD STROBE ENV */
RETAINC=7 /* RETENTION PERIOD FOR COMPLETED REQUESTS */
RETAINQ=7 /* RETENTION PERIOD FOR QUEUED REQUESTS */
RETPD= /* RETENTION PERIOD FOR SAMPLE DATA SETS */
STORCLAS= /* SMS STORCLAS FOR SAMPLE D S ALLOCATION */
STRAUTO=STROBE.STRAHIST /* AUTOSTROBE HISTORY DATA SET NAME */
STRMGR= /* DASD MANAGEMENT SYSTEM (HSM/OTHER) */
SVCOVR=NO /* SVC OVERRIDE */
SWAP=NO /* SPECIFY NO TO MAKE SESS MGR NON-SWAP */
TIME=HHMMSS /* TIME FORMAT (DEFAULT IS HHMMSS) */
TIMSEP=: /* TIME SEPARATOR (DEFAULT IS :) */
UNIT=SYSDA /* UNIT FOR SAMPLE DATA SET ALLOCATION */
VOLUME= /* VOLUME FOR SAMPLE DATA SET ALLOCATION */
XCFGROUP= /* XCF GROUP NAME, OR BLANK FOR LOCAL MODE */
*/ ENDUP

```

The following table describes the options that can be included in the STROBE parameter data set. You can modify any of the fields. In the third column, enter the value you wish to include, or retain the listed default value.

Table 3-2. STROBE Parameter Data Set Options

Keyword	Meaning	Default
ARCVID	Volume serial number used by DFHSM, or its equivalent, to indicate that a data set has been archived; for example, DFHSM's MIGRAT. STROBE uses this parameter in conjunction with the STRMGR option (page 3-18) to determine how archived sample data sets are processed. If you use two separate volumes (for example MIGRAT1 indicating DASD, MIGRAT2 indicating tape), you still must specify MIGRAT.	
ATTFLTR	{ENABLE DISABLE} Switch that determines if the attach filter is enabled or disabled. You can override this field at run time by the ENV operation.	ENABLE
AUTOACTION	{MEASURE WARNING} Set to MEASURE when AutoSTROBE is to initiate a measurement when it detects abnormal behavior in an application that has been specified to be monitored. A setting of WARNING causes AutoSTROBE to put the associated request in the held-wn (warning) state and provide data describing the abnormal behavior.	MEASURE
AUTOCANDIDATE	YES NO A switch that controls whether Auto Candidate processing is performed.	YES
AUTOCANELAP	Specifies the minimum elapse time required for a job, job step, program or system to be placed on the AUTO Candidate list for elapsed time.	60
AUTOCANEXCP	Specifies the minimum number of EXCPs required for a job, job step, program or system to be placed on the AUTO Candidate list for EXCP activity.	1000000
AUTOCANFREQ	Specifies the minimum number of runs per week required for a job, job step, program or system to be placed on the AUTO Candidate list for those that are frequently run	30
AUTOCANTCB	Specifies the minimum TCB time in minutes required for job, job step, program or system to be placed on the AUTO Candidate list for TCB time.	20
AUTODBMIN	Minimum time a program must for AutoStrobe to collect basis data.	2
AUTOELAPPERCNT	Percentage AutoSTROBE uses as a threshold to report elapsed time	50
AUTOIOEVENT1	The three values for the first event that defines a threshold for abnormal I/O behavior in a monitored program. The values are a "greater than" percentage, a "less than" percentage for the norm, and a consecutive number of intervals in the program where either percentage was exceeded. None of these values can be negative.	100,75,3
AUTOIOEVENT2	Values you for the second event that defines a threshold for abnormal I/O behavior in a monitored program.	
AUTOIOEVENT3	Values for the third event that defines a threshold for abnormal I/O behavior in a monitored program.	
AUTOIOEVENT4	Values for the fourth event that defines a threshold for abnormal I/O behavior in a monitored program.	
AUTOIOEVENT5	Values for the fifth event that defines a threshold for abnormal I/O behavior in a monitored program.	
AUTOMAX	The maximum number of programs that can be monitored by AutoSTROBE at any time. The maximum value allowed for this parameter is 50.	10
AUTOSTROBE	{YES NO} A switch that controls whether AutoSTROBE is activated and if the AutoSTROBE option is to appear on the STROBE-OPTIONS menu.	YES
AUTOTCBPERCNT	The percentage used in calculating the AutoStrobe threshold for TCB time.	
BASELIN	The minimum percentage of execution samples required for STROBE to gather control section data for load modules. This value must be between 0 and 100. You can override it with the BASELINE operand on the ADD and CHANGE commands.	2
BLKSIZE	The default block size for STROBE sample data sets. This value must be a minimum of 8192 or can be increased to a maximum of 32760. (System administrators should check the available storage on the system(s) where STROBE is running and increase this parameter to as high a value as possible.)	8192

Table 3-2. STROBE Parameter Data Set Options

Keyword	Meaning	Default
BUFNO	The number of buffers assigned to the sample data set. This value must be between 2 and 255. The buffer pool is allocated in the address space of the measured program below the 16-megabyte line. (System administrators should check the available storage on the system(s) where STROBE is running and increase this parameter to as high a value as possible.)	2
CVLIB	{LINKLIST APFLIB} Switch that determines whether STROBE executes from link list or from an authorized library.	LINKLIST
DATACLAS	The default SMS DATACLAS parameter used during sample data set allocation.	
DATE	The date format you want printed or displayed on the Performance Profile. You can use any of the following additional formats: MMDDYY, DDMMYY, YYMMDD, MMDDYYYY, DDMMYYYY, YYYYMMDD.	MMDDYYYY
DATESEP	The character you want to use to separate date values. You can use any of the following additional characters: dash (-), period (.). These characters must be enclosed in single quote marks such as '!.	/
DSNAME	The high-order qualifier(s) of the sample data set name. This value can be blank or contain up to 26 characters. The DSNAME operand on the ADD and CHANGE operations can override it.	STROBE
DSNSUFIX	A low-order qualifier (suffix) for the sample data set name (default is blank). This value can contain from one to eight characters.	
EXPDATE	The sample data set expiration date. This value must be in the format yyyyddd (Julian date). If left blank, it is set to 0000000. This option and RETPD are mutually exclusive.	
FILTER	{DISABLE} A switch that controls whether the STROBE access filter checks your security package to determine the type of access privileges granted to a TSO user ID for issuing STROBE commands. STROBE will not check your security package unless you enable the filter by clearing the DISABLE value to "FILTER=". Note: If you have RACF or CA Top Secret installed and the TSO user ID is not defined to the security package, the filter permits access. If you have CA-ACF2 installed and the TSO user ID is not defined to the security package, the filter denies access. When the access filter is enabled, AutoSTROBE uses it to verify that the issuer of an AutoSTROBE request is authorized to access the program specified in the request.	DISABLE
HISTORY_PLEX	{YES NO} Set to NO if you do not want the history option to appear on the STROBE OPTIONS menu.	YES
HISTORY_USER	{YES NO} Set to NO if you do not want the default on the Performance Profile panels to be to create a measurement session history record as the Performance Profile is produced.	YES
JSTCB	{ALLOW ALWAYS NEVER} Determines whether STROBE can measure job steps that have multiple job step tasks. ALLOW permits measurement of such job steps when the user specifies the operand OPTIONS=NOQUIT upon adding a measurement request. ALWAYS causes STROBE to measure such job steps by forcing the NOQUIT option, regardless of whether it is specified on the ADD operation. NEVER prohibits measurement of all job steps that attach job step tasks. For more information, see "Programs That Attach Tasks as Job Step Tasks" on page 2-16	ALLOW
LINEPG	The number of lines to be printed on each page of the Performance Profile when it is produced through STROBE/ISPF. This value must be between 45 and 80, and it can be overridden with the LINEMAX reporting option. If you change the default, the new value will only apply when you generate a Performance Profile from the "Detail for a Performance Profile" panel.	60
LMSS	The name of the test License Management System subsystem you want to use to with a test version of STROBE. The subsystem must be CWSBssss, where ssss is a unique 4 character name of your choice.	

Table 3-2. STROBE Parameter Data Set Options

Keyword	Meaning	Default
LOG_MAX	The maximum number of tracks used for a log data set within a generation data group. Specify this parameter if you apply Extent Reduction to minimize the size of data sets. You must specify this value and assign it a value between 1 and 1000.	30
LOG_SPACE	{{(TRK,(1-9999,1-9999)) (CYL,(nnn,nnn)) (bbbb,(nnnn,nnnn))} The space allocation for a new log data set within a generation data group, when the allocation of the current log data set is not desired. Specify tracks, cylinders, or blocksize allocations.	(TRK,(10,10))
LOG_UNIT	The unit on which STROBE allocates the log data set.	SYSDA
LOG_VOL	The volume on which STROBE allocates the log data set.	
LOGWARN	The percentage of the current allocation of the STROBE log data set that must be used before STROBE issues an initial warning message, and the incremental percentage that causes STROBE to issue an additional message. Each value must be between 1 and 100. Enclose the values in parentheses and separate them by a comma. This parameter is used only on sequential log data sets and must be specified when the log data set is sequential.	(80,5)
LPALIB	{NO YES} Switch to determine whether STROBE accesses LPA libraries to obtain module mapping data. Most VSAM modules reside in LPA. To obtain control section mapping for these VSAM modules, you can set the LPALIB parameter to YES or use the MAPROGRAM parameter to point to the VSAM modules as described in the <i>STROBE MVS User's Guide</i> .	NO
MAXACT	The maximum number of measurement requests that can be concurrently active. This value must be between 10 and 1000, and it cannot exceed MAXREQ (defined below). If MAXREQ is set to a value below 20, the default value of MAXACT is reset to the same value.	20
MAXREQ	Maximum number of requests (active, queued, completed, or AutoSTROBE) that can be present in a system at any given time. This value must be between 10 and 999. If you are using the STROBE Advanced System Management Feature or are using XCF groups in a multisystem environment, increase the value to 400. (In a multisystem environment this parameter reflects all the requests in the STROBE XCF-group.) For consistent results, be sure to assign the same MAXREQ value to all members of a STROBE-XCF group. STROBE uses the value from the first member to join the group, applying this value to all other members as they join.	200
MGMTCLAS	Default SMS MGMTCLAS parameter used during sample data set allocation.	
MSASPROCNAME	Name of the Measurement Services Address Space (MSAS) startup procedure	STRMSAS
PROFILE	High-level profile ID for access to STROBE. The session management facility assigns the default value of \$STROBE internally. See "Define User Profiles to Control What Can be Measured" on page 4-2.	
REMV_OLD	Set to YES if you need to replace an executing STROBE environment that cannot be correctly removed because required run-time modules were deleted. Setting this parameter to YES causes the version of STROBE being installed to replace the environment of the existing version of STROBE if no active measurement is taking place in the existing environment. If active measurement sessions exist, STROBE initialization will fail and a message is issued. Note: If REMV_OLD is set to YES, common storage from the executing STROBE environment will not be re-used.	NO
RETAINC	Default number of additional days STROBE keeps a completed request on the queue. Day 1 starts at 00:00:00 hours of the day after the request is added. Valid values are 0 through 999. A value of 0 causes the queued request to be deleted at 23:59:59 hours on the day the request was created. A value of 999 indicates the request is kept indefinitely. For consistent results, be sure to assign the same RETAINC value to all members of a STROBE-XCF group. STROBE uses the value from the first member to join the group, applying this value to all other members as they join.	7

Table 3-2. STROBE Parameter Data Set Options

Keyword	Meaning	Default
RETAINQ	<p>Default number of additional days STROBE keeps a request on the queue before it completes. Day 1 starts at 00:00:00 hours of the day after the request was added. Valid values are 0 through 999. A value of 0 causes the queued request to be deleted at 23:59:59 hours on the day the request was created. A value of 999 indicates the request is kept indefinitely.</p> <p>For consistent results, be sure to assign the same RETAINQ value to all members of a STROBE-XCF group. STROBE uses the value from the first member to join the group, applying this value to all other members as they join.</p>	7
RETPD	Number of days that sample data sets are kept on the host. This value can be between 1 and 9999. This option and EXPDATE are mutually exclusive. You have to add the parameter to the data set if you want to use it, but it only should be added if you use DFSMS to manage your sample data sets.	
STORCLAS	Default SMS STORCLAS parameter to be used during sample data set allocation.	
STRAUTO	Name of the AutoSTROBE data set. This parameter must be specified unless the AUTOSTROBE parameter is set to NO. (The data set name cannot be longer than 44 characters)	STROBE. STRAHIST
STRMGR	<p>{HSM DMS DMS2 OTHER} Identifies the DASD management system installed at your site. STROBE uses this parameter in conjunction with ARCVID (page 3-15) to determine how archived sample data sets are processed. If DFHSM is installed, specify HSM. If DMS release 8.1.7 or greater is installed, specify DMS2. For HSM and DMS2 STROBE HDELETEs archived sample data sets and allocates new ones. For DMS release 8.1.6 and less, specify DMS and STROBE uncatalogs the sample data set and allocates a new data set. Otherwise, specify OTHER. STROBE forces a recall of archived data sets.</p> <p>If you do not specify a value for STRMGR, STROBE receives an allocation error when it attempts to access an archived sample data set.</p> <p>If your site uses FDR as the DASD management system with the defaults for MIGRAT and RECALL, place the STROBE high-level qualifier for sample data sets on FDR's EXCLUDE list so that the data sets are not archived by FDR.</p> <p>If you are using DMS release 8.1.7 and greater, specify DMS2. If you are using DMS release 8.1.6 or less, specify DMS.</p>	
SVC	The STROBE SVC. The default is SVC 67. The SVC number can be between 0 and 255, and must be type 2, 3, 4, or 5. The session manager does not accept SVC numbers in use by operating system SVCs unless SVCOVR is set to YES.	67
SVCOVR	{NO YES} Switch that determines whether STROBE uses an SVC number even if that SVC appears to be in use by another application.	NO
SWAP	{NO YES} Switch to determine whether the session manager is swappable (YES) or nonswappable (NO).	NO
TIME	The time format you want printed or displayed on the Performance Profile. You can use any of the following additional formats: ZHMMSS (24-hour format with leading zero suppressed), APMMSS (12-hour format with AM/PM indicator), ZPMMSS (12-hour with AM/PM indicator, leading zero suppressed).	HHMMSS (24-hour format)
TIMESEP	The character you want to use to separate time values. You can use any of the following additional characters: comma (,), period (.). These characters must be enclosed in single quote marks such as '!.	: (colon)
UNIT	Default direct access unit name used to catalog sample data sets. The UNIT operand on the ADD and CHANGE operations can override it.	SYSDA
VOLUME	Default direct access volume serial number for sample data sets. It can contain from one to six characters. The VOLUME operand on the ADD and CHANGE operations can override it.	

Table 3-2. STROBE Parameter Data Set Options

Keyword	Meaning	Default
XCFGROUP	The XCFGROUP= <i>groupname</i> parameter identifies the XCF group that this instance of STROBE will join when operating in a multisystem environment. If <i>groupname</i> is not specified, STROBE operates in local mode and the SYSTEM field will not appear on any of the STROBE ISPF panels. The <i>groupname</i> can be up to eight characters (valid characters are A-Z, 0-9, \$, #, and @). Do not begin <i>groupname</i> with the letters A through I, SYS, or UNDESIG because IBM XCF group names or system programmers may use these characters.	

Customize the Log Data Set

The log contains a record of the messages produced by the measurement and session management facilities. You can allocate it as a member of a generation data group (GDG) or as a non-GDG data set. Member ALLOCATE (or ALLOCTST) contains two job steps that create a GDG and another step (which is commented) that creates a non-GDG data set.

Figure 3-8. Customize the Log Data Set (Member ALLOCATE)

```

//*LOGNGDG EXEC PGM=IEFBRI4
//*SYSLOG DD DSN=STROBE.V03R00.LOG,
//*          SPACE=(TRK,(100,20)),DISP=(,CATLG),
//*          DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3024),
//*          UNIT=SYSDA,VOL=SER=
//IDCAMSI EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE GDG (
    NAME(STROBE.V03R00.LOG) -
    NOEMPTY -
    NOSCRATCH -
    LIMIT(10) -
  )
//*
//LOGGDG EXEC PGM=IEFBRI4
//SYSPRINT DD SYSOUT=*
//SYSLOG DD DSN=STROBE.V03R00.LOG.G0001V00,UNIT=SYSDA,
//          SPACE=(TRK,(10,10)),
//          DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3024),
//          DISP=(,CATLG),VOL=SER=
//*

```

Create a GDG and a Data Set for the Log

It is strongly recommended that you define the GDG and a log data set belonging to the first generation. STROBE maintains the log automatically, creating a new generation when the current log data set fills. To define the log, modify the IDCAMS1 and LOGGDG steps of member ALLOCATE (Figure 3-8 on page 3-19) as follows:

- in both job steps, change **STROBE.V03R00.LOG** to the name chosen for the Release 3.0 GDG log data set.
- in job step **LOGGDG**
 - change **UNIT=SYSDA** to the unit for your data set
 - if required, supply a value for **VOLSER**

CAUTION:

If you place the log data set on a system pack, change the reserve for the data set to an enqueue through your resource serialization package. See “Planning for Enqueues” on page 2-5.

Create a Non-GDG Log Data Set

You can define the STROBE log data set without a GDG, but this method requires you to manage the log data set with less assistance from STROBE. Define the log, in member ALLOCATE (or ALLOCTST) of your STROBE control data set (Figure 3-8 on page 3-19), as follows:

- Remove job steps IDCAMS1 and LOGGDG.
- In job step LOGNGDG
 - uncomment the job step
 - change **STROBE.V03R00.LOG** to the name chosen for the log data set
 - change **UNIT=SYSDA** to the unit for your data set
 - supply a value for **VOL=SER=**, if required
- Delete the LOG_MAX and LOG_SPACE parameters from the PARMLIB step.
- Add the LOGWARN parameter. See page 3-17 for more information about using the LOGWARN parameter.

Customize the Queue Data Set

The STROBE queue data set is a required VSAM key-sequenced data set that STROBE uses to store and retrieve information about requests. In job step QUEUEDS (Figure 3-9 on page 3-20) of member ALLOCATE (or ALLOCTST), define the data set as follows:

- change **STROBE.V03R00.QUEUE** to the name you are using for the Release 3.0 queue data set.
- supply a value for **VOLSER**

CAUTION:

If you place the queue data set on a system pack, change the reserve for the data set to an enqueue through your resource serialization package. See “Planning for Enqueues” on page 2-5.

Figure 3-9. Customize the Queue Data Set (Member ALLOCATE)

```
//QUEUEDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER -
(
  NAME(STROBE.V2R5M0.QUEUE) -
  CYL(2 2) -
  VOL(VOLSER) -
  BUFFERSPACE(112640) -
  INDEXED -
  CISZ(22528) -
  KEYS(54 0) -
  RECORDSIZE(1044 22528) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)
/* */
IF LASTCC = 0 -
  THEN LISTC ENT(STROBE.V2R5M0.QUEUE) ALL
/*
```

Customize the System Message Data Set

The STROBE system message data set is a required VSAM key-sequenced data set. To define the data set, in job step MSGDS of member ALLOCATE (or ALLOCTST) of your STROBE control data set (Figure 3-10)

- change **STROBE.V03R00.SYSMSG**s to the name of the Release 3.0 message data set

- supply a value for **VOLSER**

Figure 3-10. Customize the System Message Data Set (Member ALLOCATE)

```
//MSGDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER -
(
  NAME(STROBE.V03R00.SYSMSG) -
  CYL(2 2) -
  VOL(VOLSER) -
  INDEXED -
  CISZ(22528) -
  KEYS(54 0) -
  RECORDSIZE(1044 22528) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)
/* */
IF LASTCC = 0 -
  THEN LISTC ENT(STROBE.V03R00.SYSMSG) ALL
/*
```

Define the ALLOCATE Member Start Procedures

STROBE runs as a started task. Members ALLOCATE contains a model procedure that enables you to start STROBE from an operator's console or as a batch job. Modify job step SMPROC (Figure 3-11) of member ALLOCATE or ALLOCTST as follows:

- Change **SYS1.PROCLIB** to the name of the data set in your system procedure library that will contain the procedure.
- Change **STROBE.PARMLIB** to the name of the parameter data set.
- Change **STROBE.V03R00.LOG** to the name of the STROBE log data set. If the data set does not belong to a generation data group, remove the (0) reference.
- Change **STROBE.V03R00.QUEUE** to the name of the queue data set.
- Change **STROBE.V03R00.SYSMSG** to the name of the system message data set.

Member ALLOCATE creates procedures named STRBSM and STRBSSR. Member ALLOCTST creates procedures named STRTSM and STRTSSR.

Figure 3-11. Create the Procedure to Start STROBE (Member ALLOCATE)

```

//SMPROC EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSIN DD DATA,DLM=ZZ
./ ADD NAME=STRBSM
//STRBSM PROC
/* STROBE SESSION MANAGER EXECUTION PROCEDURE
/* BATCH JOB: //STRBSM JOB ...
/* // EXEC STRBSM
/* STARTED TASK: START STRBSM
/* TO SPECIFY SYSIN:
/* START STRBSM,DSN=ANY.DSN(MEMBER),DISP=SHR
/* USE THE FOLLOWING COMMANDS TO CONTROL
/* MODIFY STRBSM,....
/* STOP STRBSM
/*
//STRBSM EXEC PGM=STRBCSM,REGION=4096K,TIME=1440,PARM='ENVNEW=NO'

//PARMLIB DD DSN=STROBE.PARMLIB(STROBE),DISP=SHR
//SYSLOG DD DSN=STROBE.V03R00.LOG(0),DISP=SHR
//SYSQUEUE DD DSN=STROBE.V03R00.QUEUE,DISP=SHR
//SYSMSG DD DSN=STROBE.V03R00.SYSMSGGS.DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//ABNLIGNR DD DUMMY
//PROIGN DD DUMMY
//ULT@SOFF DD DUMMY
//VVHC$OFF DD DUMMY
//SVHC$OFF DD DUMMY
//SYSIN DD DDNAME=IEFRDER
./ ADD NAME=STRBSSR
//STRBSSR PROC
/*
/* STROBE SESSION REQUESTER PROCEDURE
/* FROM BATCH JOB:
/* EXEC PROC=STRBSSR
/* SYSIN DD *
/* {STROBE COMMAND SYNTAX}
/* OR
/* EXEC PROC=STRBSSR,PARM='COMMAND SYNTAX'
/*
//STROBE EXEC PGM=STRBCSR,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
./ ENDUP

```

Submit the Allocation Job

You can allocate the data sets after completing these steps:

1. Supply a **JOB** statement in member **ALLOCATE** or **ALLOCTST**.
2. Execute the job. All return codes should be zero.

Verify that the data sets were created.

Step 11. Edit Member ALLOCASM (or Member ALLOTSTA for Test)

The editing described in this section must be done only if you are using the STROBE Advanced Session Management Feature.

Warning: If you plan to use the Migration utility, Do not run the ALLOCASM step before you run the utility

Customize the Request Group Data Set

The request group data set is a VSAM key-sequenced data set used by the STROBE Advanced Session Management Feature. STROBE stores and retrieves information about Advanced Session Management request groups from the request group data set.

To customize the request data set allocation, make the following edits in job step GROUPDS (Figure 3-12) of member ALLOCASM (or ALLOTSTA):

- change **STROBE.V03R00.GROUP** to the name chosen for the Release 3.0 request group data set
- supply a value for **VOLSER**

Figure 3-12. Customize the Request Group Data Set (Member ALLOCASM)

```
//GROUPDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER -
(
  NAME(STROBE.V03R00.GROUP) -
  CYL(2 2) -
  VOL(VOLSER) -
  INDEXED -
  CISZ(22528) -
  KEYS(54 0) -
  RECORDSIZE(1044 22528) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)
/* */
IF LASTCC = 0 -
  THEN LISTC ENT(STROBE.V03R00.GROUP) ALL
/*
```

Customize the History Data Set

The STROBE history data set is an optional VSAM key-sequenced data set. The history data set is the repository for all measurement session history records. These records are used to track changes in the performance of a particular job step, transaction, or DBRM and to relate changes in performance to changes in operating costs.

To define the history data set, make the following changes in job step HISTDS of member ALLOCASM (or ALLOTSTA) of your STROBE control data set (Figure 3-13 on page 3-25)

- Change **STROBE.CNTL** to the name of your STROBE control data set
- Change **STROBE.STRHIST** to the name of the Release 3.0 STROBE Performance Profile history data set. If RACF is installed on the same system as the history log data set, anyone using the history function must have RACF update access authority for the history log data set. If you intend to continue using the same history data set, delete the JCL step that allocates the new history data set.
- Supply a value for **VOLSER**.
- Change **STROBE.STRHISTA** to the name of the alternate index data set for the history data set.
- Change **STROBE.STRHISTA.PATH** to the name of the path for the alternate index data set for the history data set.

In the step HISTLOG:

- Change **STROBE.STRHIST.LOG** to the name of the history log data set. If RACF is installed on the same system as the history log data set, anyone using the history function must have RACF update access authority for the history log data set.
- Change **UNIT=SYSDA** to the unit for the history log data set
- Supply a value for **VOL=SER=**, if required

For more information about setting CI size and space requirements, see “Direct Access Storage Requirements” on page 2-4.

Initializing the History Data Set

To initialize the history data set, perform the following steps in the HISTINIT job step of member ALLOCASM or ALLOTSTA shown in Figure 3-13 on page 3-25:

- Change **STROBLIB** to the name of your unauthorized STROBE library.
- Change **STROBE.STRHIST.LOG** to the name of the history log data set.
- Change **STROBE.STRHIST** to the name of the history data set.
- Change **STROBE.STRHISTA.PATH** to the path identifying the location of the alternate index data set for the history data set.
- Determine the cost identifier you wish to use to calculate operating costs. You can use either the cost of one CPU hour and the cost of 1000 I/Os, or the cost of 1000 service units, but not both.
- Specify the cost identifiers for the prime and non-prime processing periods. To do so, code one of the following sets of values for the **DEFAULTCOST** parameter:

To calculate cost by	Code values for
one CPU hour and the cost of 1000 I/Os	PRIMECPU, NONPRIMECPU, PRIMEIO, and NONPRIMEIO
-or-	
1000 service units	PRIMESU and NONPRIMESU

Note: CPU cost and I/O cost parameters must be set as a pair. To use the service units keywords, do not set values for CPU and I/O. Figure 3-13 on page 3-25 shows the **DEFAULTCOST** parameter with the appropriate keywords for service units. Ensure that these specifications do not exceed column 72.

- Change **PRIMESTART** and **PRIMEEND** to the appropriate values on the **DEFAULTCOST** parameter, as shown in Figure 3-13 on page 3-25. These values are the defaults that determine the prime processing period. Ensure that these specifications do not exceed column 72.

The default value for either of these parameters cannot be 0.

Figure 3-13. Customize the History Data Set (Member ALLOCASM)

```

//HISTDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//HINITREC DD DISP=SHR,DSN=STROBE.CNTL(HINITREC)
//SYSIN DD *
/**/
DEFINE CLUSTER -
(
NAME(STROBE.STRHIST) -
CYL(10 5) -
VOL(VOLSER) -
INDEXED -
BUFFERSPACE(112640) -
CISZ(22528) -
KEYS(62 18) -
RECORDSIZE(300 500) -
SHAREOPTIONS(4 3) -
UNIQUE -
)
/**/
DEFINE ALTERNATE INDEX -
(
NAME(STROBE.STRHISTA) -
RELATE(STROBE.STRHIST) -
VOL(VOLSER) -
CYL(5 2) -
BUFFERSPACE(112640) -
CISZ(22528) -
KEYS(64 0) -
SHAREOPTIONS(4 3) -
UPGRADE -
UNIQUE -
)
/**/
DEFINE PATH
(
NAME(STROBE.STRHISTA.PATH) -
PATHENTRY(STROBE.STRHISTA) -
UPDATE -
)
/**/
REPRO
INFILE(HINITREC)
OUTDATASET(STROBE.STRHIST)
/**/
IF LASTCC = 4 -
THEN SET MAXCC = 0

/**/
BLDINDEX INDATASET(STROBE.STRHIST)-
OUTDATASET(STROBE.STRHISTA)
/**/
IF MAXCC = 0
THEN DO
LISTC ENT(STROBE.STRHIST) ALL
LISTC ENT(STROBE.STRHISTA.*) ALL
END
//HISTLOG EXEC PGM=IEFBR14
//STRHLOG DD DSN=STROBE.STRHIST.LOG,
// SPACE=(TRK,(100,20)),DISP=(,CATLG,DELETE),
// DCB=(RECFM=VBA,LRECL=137,BLKSIZE=4096),
// UNIT=SYSDA,VOL=SER=
//HISTINIT EXEC PGM=STRHUTIL
//STEPLIB DD DISP=SHR,DSN=STROBLIB
//STRHLOG DD DISP=SHR,DSN=STROBE.STRHIST.LOG
//STRHIST DD DISP=SHR,DSN=STROBE.STRHIST
//STRHISTA DD DISP=SHR,DSN=STROBE.STRHISTA.PATH
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFAULTCOST=(PRIMESTART=0900,PRIMEEND=1700,
PRIMESU=100,NONPRIMESU=50)

```

Customize the AutoSTROBE Data Set

The AutoSTROBE data set is a VSAM key-sequenced data set used as a repository for data about programs that AutoSTROBE monitors.

To customize the AutoSTROBE data set allocation, make the following edits in job step AUTODS (Figure 3-14) of member ALLOCASM (or ALLOTSTA):

- Change **STROBE.STRAHIST** to your name for the AutoSTROBE data set.
- supply a value for **VOLSER**

Figure 3-14. Customize the AutoSTROBE Data Set (Member ALLOCASM)

```
//AUTODS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//HINITREC DD DISP=SHR,DSN=STROBE.CNTL(HINITREC)

//SYSIN DD *
DEFINE CLUSTER -
(
  NAME(STROBE.STRAHIST) -
  CYL(10 5) -
  VOL(VOLSER) -
  INDEXED -
  BUFFERSPACE(112640) -
  CISZ(22528) -
  KEYS(64 0) -
  RECORDSIZE(67 500) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)
/* */
REPRO
  INFILE(HINITREC)
  OUTDATASET(STROBE.STRAHIST)
/**/

IF LASTCC = 4 -
THEN SET MAXCC = 0
/*
IF MAXCC = 0
THEN LISTC ENT(STROBE.STRAHIST) ALL
/*
```

Define the ALLOCASM Member Start Procedures

STROBE runs as a started task. Member ALLOCASM contains a model procedure that enables you to start STROBE from an operator's console or as a batch job. Modify job step SMPROC (Figure 3-15 on page 3-27) of member ALLOCASM or ALLOTSTA, as follows:

- Change **SYS1.PROCLIB** to the name of the data set in your system procedure library that will contain the procedure. Make sure you use the same data set name in both **ALLOCATE** and **ALLOCASM** or in **ALLOCTST** and **ALLOTSTA**
- Change **STROBE.PARMLIB** to the name of the parameter data set.
- Change **STROBE.V03R00.LOG** to the name of the STROBE log data set. If the data set does not belong to a generation data group, remove the (0) reference.
- Change **STROBE.V03R00.QUEUE** to the name of the queue data set.
- Change **STROBE.V03R00.GROUP** to the name of the group data set.
- Change **STROBE.V03R00.SYSMSG** to the name of the system message data set.
- If you change the name of the MSAS startup procedure from **STRMSAS**, change the **ADD NAME** statement to reflect the name you selected.

Member ALLOCASM creates a procedure named **STRBSM** that overlays the one created by **ALLOCATE** and can be used with the STROBE Advanced Session Measurement Feature. Also member ALLOCASM creates a procedure named **STRBMSAS**. The fourth character of this procedure name and the **STRBSM** procedure must be identical.

Figure 3-15. Create the Procedure to Start STROBE (Member ALLOCASM)

```

//SMPROC EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSIN DD DATA,DLM=ZZ
./ ADD NAME=STRBSM
//STRBSM PROC
/* STROBE SESSION MANAGER EXECUTION PROCEDURE
/* BATCH JOB: //STRBSM JOB ...
/* // EXEC STRBSM
/* STARTED TASK: START STRBSM
/* TO SPECIFY SYSIN:
/* START STRBSM,DSN=ANY.DSN(MEMBER),DISP=SHR
/* USE THE FOLLOWING COMMANDS TO CONTROL
/* MODIFY STRBSM,....
/* STOP STRBSM
/*
//STRBSM EXEC PGM=STRBCSM,REGION=4096K,TIME=1440,PARM='ENVNEW=NO'

//PARMLIB DD DSN=STROBE.PARMLIB(STROBE),DISP=SHR
//SYSLOG DD DSN=STROBE.V03R00.LOG(0),DISP=SHR
//SYSQUEUE DD DSN=STROBE.V03R00.QUEUE,DISP=SHR
//SYSGROUP DD DSN=STROBE.V03R00.GROUP,DISP=SHR
//SYSMSG DD DSN=STROBE.V03R00.SYSMSG,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//ABNLIGNR DD DUMMY
//PROIGN DD DUMMY
//ULT@SOFF DD DUMMY
//VVHC$OFF DD DUMMY
//SVHC$OFF DD DUMMY
//SYSIN DD DDNAME=IEFRDER
./ADD NAME=STRMSAS
//STRMSAS PROC V=
/*
/* STROBE MEASUREMENT SERVICES PROCEDURE
/*
//IEFPROC EXEC PGM=STR&V.MSAS,REGION=1024K,TIME=1440
//STEPLIB DD DSN=APF.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*

```

Submit the Allocation Job

You can allocate the data sets after completing these steps:

1. Supply a **JOB** statement in member ALLOCASM or ALLOTSTA.
2. Execute the job. All return codes should be zero.
3. Verify that the data sets were created.

Step 12. Start the Address Space

You can start STROBE from an operator's console or by submitting a batch job that executes the procedure STRBSM from your system procedure library concatenation.

CAUTION:

To start STROBE, the Compuware License Management System (LMS) must be installed with the required license certificates imported into the license file for STROBE MVS for Sysplex Release 3.0 and any Features that are installed at your site. See the Compuware License Management System User Reference Manual for more information about LMS.

- To start STROBE from an operator's console, enter

```
START STRBSM
```

Note: If you modified ALLOCTST or ALLOTSTA to create the procedure that starts the address space, enter

```
START STRTSM
```

- To start STROBE from a SYSIN data set, enter

```
START STRBSM,DSN=command.file,DISP=SHR
```

where *command.file* is the name of the data set containing STROBE commands.

- To start STROBE from a batch job, construct job control statements that call the procedure STRBSM in your system procedure library concatenation. If you are not controlling access through a security package, supply the STROBE password with the PARM keyword of the EXEC statement. You can supply commands to STROBE in the input stream after the invocation of the procedure or in a data set defined with a STRBSM.SYSIN DD statement.

Note: It is recommended that you use a started task, rather than a batch job.

Step 13. Test the Installation

Test your installation of STROBE by using jobs provided in the control data set.

To perform this function	Refer to
Measure a sample program	"Measure a Sample Program", below
Create a Performance Profile	"Create a Performance Profile" on page 3-29
Optionally, create an indexed Performance Profile	"Create an Indexed Performance Profile (Optional)" on page 3-31

Measure a Sample Program

The STROBE control data set provides a batch job to run a test measurement session. The job creates a sample data set, lists the STROBE messages produced during the measurement session, and deletes the request. Submit one of the following members:

- TESTSTR (Figure 3-16 on page 3-28), if you have copied STROBE modules to a link list data set without renaming
- TESTNEW, if you have used member COPYTEST to copy and rename STROBE modules to a link list data set
- TESTAPE, if you have used member COPYTEST to copy renamed STROBE modules to an APF-authorized library that is not a link list data set

Figure 3-16. Measure a Sample Program (Member TESTSTR)

```
//STRBSSR PROC
//STROBE EXEC PGM=STRBCSR,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//
// MEASURE EXEC STRBSSR
ADD *,PGM=STRBSAM1
//GO EXEC PGM=STRBSAM1
//STEPLIB DD DISP=SHR,DSN=STROBLIB
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//HIBLOCK DD UNIT=SYSDA,SPACE=(CYL,(4,1),,CONTIG)
//LOBLOCK DD UNIT=SYSDA,SPACE=(CYL,(4,1),,CONTIG)
//NOBLOCK DD UNIT=SYSDA,SPACE=(CYL,(4,1),,CONTIG)
//*
//SMLIST EXEC STRBSSR
LIST *,DELETE
//*
```

Option 1: Use TESTSTR or TESTNEW

1. Edit the job control statements in member TESTSTR (Figure 3-16) or TESTNEW.
 - supply a **JOB** statement
 - change **STROBLIB** to the name of your unauthorized STROBE library
2. Execute the job.

Option 2: Use TESTAPF

1. Customize the job control statements in member TESTAPF (not shown).
 - supply a **JOB** statement
 - change **STROBLIB** to the name of your STROBE unauthorized library
 - change **STROBE.APF.LOADLIB** to the name of the APF-authorized library containing your renamed STROBE modules
2. Execute the job.

Testing in a Sysplex Environment

If you have installed STROBE on several systems within an MVS sysplex, you can first determine that STROBE is correctly functioning in the sysplex environment and then verify that the STROBE version you are using for the single point of control recognizes the other systems where STROBE is installed:

1. When you start session manager as described in “Step 12. Start the Address Space” on page 3-27 and you have installed STROBE in a sysplex environment, the following message is displayed:

```
STROBE ENVIRONMENT IS NOW CREATED ON SYSTEM sysname
STROBE-XCF GROUP MEMBERS ARE: sysname1...
```

2. To verify STROBE is installed on other systems contained in the sysplex and STROBE control version recognizes the other installed versions, perform the following steps after you install the ISPF panels and start STROBE:
 - From the ADD QUEUED panel, clear the SYSTEM field and press enter.
 - STROBE lists all of the systems contained within the sysplex where a version of STROBE is installed and you can submit a request to measure a job if it runs on the system you choose.

Create a Performance Profile

Member REPORT (Figure 3-17) contains a job that uses the STROBE procedure to produce a STROBE Performance Profile from the sample data set created from the test measurement.

Create an Indexed Performance Profile (Optional)

Member INDEXCOB (Figure 3-18 on page 3-32) contains a job that tests the STROBE system with a STROBE indexer program. It uses your installation's compile and link procedure to compile and link a sample source program (TESTPROG) and the STROBE STROX procedure to perform the indexing function that produces a map data set. After the compilation and indexing step, STROBE measures the test program and produces a sample data set. The map data set and sample data set are then used as input to the STROBE STROE procedure to create an indexed Performance Profile.

Figure 3-18. Member INDEXCOB

```

//STROX  PROC  MAPPGM=STRXCBF,PRTCLAS='*',DAUNIT=SYSDA,
//          STRLIB='STROBLIB',
//          LISTDSN='&LISTDSN',MAPDSN='&STRMAP',
//          MAPVOL=,MAPUNIT=SYSDA,MAPDISP=PASS,MAPLABL='(1,SL)'
//**
//** STROBE GENERALIZED INDEXER MAP PROCEDURE
//**
//MAP     EXEC  PGM=&MAPPGM,REGION=128K
//STEPLIB DD  DSN=&STRLIB,DISP=(SHR,PASS)
//SYSPRINT DD  SYSOUT=&PRTCLAS,DCB=(RECFM=FBA,LRECL=137,BLKSIZE=3018)
//SYSUDUMP DD  SYSOUT=&PRTCLAS
//STRUT1  DD  UNIT=&DAUNIT,SPACE=(1690,(100,100))
//STRUT2  DD  UNIT=&DAUNIT,SPACE=(1692,(100,100))
//LISTING DD  DSN=&LISTDSN,DISP=(OLD,PASS)
//STRMAP  DD  DSN=&MAPDSN,VOL=SER=&MAPVOL,UNIT=&MAPUNIT,
//          DISP=(NEW,&MAPDISP,DELETE),LABEL=&MAPLABL,
//          SPACE=(1692,(100,100),RLSE)
//
//          PEND
//**
//STROE  PROC  PRTCLAS='*',DAUNIT=SYSDA,
//          STRLIB='STROBLIB',
//          MAPDSN='NULLFILE',SAMDSN='&STRSAM',
//          SORTUT=300,REPNAME=,RESLTN=,SORTSIZ=,LINEMAX=,
//          DETAIL=,COMPRES=,NOPROC=,NODASD=,NOTRAN=,
//          ATTR=,NOATTR=,RPTPARM=,
//          WAITLOC=,PUBP=,DASD=
//**
//** STROBE REPORT
//**
//REP     EXEC  PGM=STROBEDT,REGION=4096,
//          PARM=('REPNAME=&REPNAME,RESLTN=&RESLTN',
//          'SORTSIZ=&SORTSIZ,LINEMAX=&LINEMAX,DETAIL=&DETAIL',
//          'COMPRES=&COMPRES,&NOPROC,&NODASD,&NOTRAN',
//          '&NOTASK,&ALLCSCT'&WAITLOC,&PAGELOC',
//          'ATTR=&ATTR,&NOATTR',
//          'PUBP=&PUBP,DASD=&DASD')
//STEPLIB DD  DSN=&STRLIB,DISP=(SHR)
//SYSPRINT DD  SYSOUT=&PRTCLAS
//SYSOUT   DD  SYSOUT=&PRTCLAS
//SYSUDUMP DD  SYSOUT=&PRTCLAS
//STRSAMPL DD  DSN=&PREFIX.&SUFFIX,DISP=(OLD,PASS)
//STRMAP  DD  DSN=&&STRMAP,DISP=(OLD,DELETE)
//SLSF001 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SLSF002 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SLSF003 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SLSF004 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SLSF005 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SLSF006 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SLSF007 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SLSF008 DD  DSN=&DDIODS1,DISP=(SHR,PASS)
//SYSUT1  DD  UNIT=&DAUNIT,SPACE=(1672,(400,400))
//SYSTEMP01 DD  SPACE=CYL(10,10),UNIT,UNIT=&DAUNIT
//SYSTEMP02 DD  SPACE=CYL(10,10),UNIT,UNIT=&DAUNIT
//SYSTEMP03 DD  SPACE=CYL(10,10),UNIT,UNIT=&DAUNIT
//SORTWK01 DD  SPACE=(1700,(&SORTUT),,CONTIG),UNIT=&DAUNIT
//SORTWK02 DD  SPACE=(1700,(&SORTUT),,CONTIG),UNIT=&DAUNIT
//SORTWK03 DD  SPACE=(1700,(&SORTUT),,CONTIG),UNIT=&DAUNIT
//SORTWK04 DD  SPACE=(1700,(&SORTUT),,CONTIG),UNIT=&DAUNIT
//SORTWK05 DD  SPACE=(1700,(&SORTUT),,CONTIG),UNIT=&DAUNIT
//SORTWK06 DD  SPACE=(1700,(&SORTUT),,CONTIG),UNIT=&DAUNIT
//
//          PEND
//** ***** *//
//** ***** *//
//** COBOL COMPILE AND LINK ***** *//
//** ***** *//
//** ***** *//
//COMPILE EXEC  COBVCL,
//          PARM.COB='PMAP,SOURCE,VERB,OPT,LOAD,NODECK,NOSTATE',
//          PARM.LKED='LIST,XREF,LET'
//COB.SYSPRINT DD  DSN=&LISTDSN,DISP=(MOD,PASS),UNIT=SYSDA,
//          SPACE=(2904,(400,200)),
//          DCB=(RECFM=FBA,LRECL=121,BLKSIZE=2904)
//COB.SYSIN DD  DSN=STROBE.CNTL(SAMPLEW),DISP=SHR
//LKED.SYSLMOD DD  DSN=&&GO(TESTPROG),DISP=(MOD,PASS),
//          UNIT=SYSDA,SPACE=(1024,(5,1,1))
//**
//INDEX   EXEC  STROX

```

Figure 3-19. Member INDEXCOB (Concluded)

```

/** ***** */
/**
/** MEASURE AND PRODUCE A REPORT ON THE NEWLY COMPILED PROGRAM */
/** ***** */
//STRBSSR PROC
//STROBE EXEC PGM=STRBCSR,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
/**
//MEASURE EXEC STRBSSR,COND=(4,LT)
ADD *,PGM=TESTPROG
/**
//GO EXEC PGM=TESTPROG
//STEPLIB DD DSN=*.COMPILE.LKED.SYSLMOD,DISP=(OLD,PASS)
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//HIBLOCK DD UNIT=SYSDA,SPACE=(CYL,(4,1),,CONTIG)
//LOBLOCK DD UNIT=SYSDA,SPACE=(CYL,(4,1),,CONTIG)
//NOBLOCK DD UNIT=SYSDA,SPACE=(CYL,(4,1),,CONTIG)
/**
//REP EXEC STROE,MAPDSN='&STRMAP',COND=(4,LT),
// SAMDSN='PREFIX.JOBNAME.S001D001'

```

1. Make the following edits to INDEXCOB:

- Supply a **JOB** statement.
- Change **STROBLIB** to the name of your unauthorized STROBE library.
- Change **STROBE.CNTL** to the name of your STROBE control data set.
- Change **PREFIX.JOBNAME.S001D001** to the name of the sample data set STROBE generates. The S001D001 qualifier may differ if you execute the job more than once.
- Change **COBVCL** to the name of your COBOL compile and link procedure.
- If you use run-time compiler libraries, concatenate a DD statement to STEPLIB in the GO step.
- If this is the test version, change **STRBCSR** to **STRTCSR**.

2. Execute the job.

Members INDEXFOR and INDEXPLI (not shown) provides the same functionality for testing a sample FORTRAN program and a PL/I program. Like INDEXCOB, INDEXFOR and INDEXPLI are included in the control data set.

Step 14. Install the ISPF Libraries

The following table gives an overview of the steps to take when installing the STROBE/ISPF components. Refer to the appropriate section for more information on each step.

Table 3-3. An Overview of the STROBE/ISPF Installation

For this step	Refer to
Edit and execute member SPFINSTL	"Create the Libraries" on page 3-34
Modify the STROBE options in the panel library	"Modify Option Menu Panel" on page 3-36
Enabling access to the STROBE data set packaging through ISPF.	"Accessing the STROBE Packaging Utility" on page 3-38
Optionally, modify the STROBE/ISPF and APMpower CLISTs	"Modify CLISTs" on page 3-40
Modify the ISPF/PDF utility panel	"Modify the ISPF/PDF Panels" on page 3-44
Test STROBE/ISPF and the APMpower MVS component	"Test CLISTs" on page 3-44

STROBE/ISPF Components

The control data set contains the following STROBE/ISPF components:

- Member SPFINSTL contains IEBUPDTE job control statements for installing the STROBE/ISPF and APMpower libraries.
- Member STROBE contains a CLIST that invokes ISPF/PDF services to execute STROBE/ISPF.
- Member APMPOWER contains a CLIST for the APMpower MVS component.
- The following members contain the data for the STROBE/ISPF libraries:

STRMLIB	STROBE/ISPF and APMpower message library.
STRPLIB	STROBE/ISPF and APMpower panel library.
STRSLIB	STROBE/ISPF and APMpower skeleton library.
STRMSGGS	STROBE/ISPF, APMpower and iSTROBE messages. The ISPF messages data set contains a member for each message generated by the STROBE system. STROBE/ISPF and APMpower users can access this data set whenever they need an explanation for a message.

Create the Libraries

Member SPFINSTL (Figure 3-20 on page 3-35), contains a job to create the libraries.

1. Make the following edits to member SPFINSTL:
 - supply a **JOB** statement
 - change **STROBLIB** to the name of your unauthorized STROBE library
 - change **STROBE.CNTL** to the name of your STROBE control data set
 - change the names for the ISPF panel (STROBE.ISPPLIB), ISPF skeleton (STROBE.ISPSLIB), and message (STROBE.ISPMLIB and STROBE.MESSAGES).
2. Execute the job. (A return code of 8 is acceptable for the DELETE step.)

Figure 3-20. Install the STROBE/ISPF Common Libraries (Member SPFINSTL)

```

//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE STROBE.ISPPLIB /* ISPF PANEL LIBRARY */
DELETE STROBE.ISPMLIB /* ISPF MESSAGE LIBRARY */
DELETE STROBE.ISPSLIB /* ISPF SKELETON JCL LIBRARY */
DELETE STROBE.MESSAGES /* STROBE MESSAGES */
//*****
/* EXECUTE A PROGRAM TO CREATE IEBUPDTE INPUT */
/*
//EPREP EXEC PGM=STRUPREP
//
//STEPLIB DD DSN=STROBLIB,
// DISP=SHR
//INPANEL DD DSN=STROBE.CNTL(STRPLIB),DISP=SHR
//OUTPANEL DD DSN=&&IPANL,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880)
//INMSGSD DD DSN=STROBE.CNTL(STRMLIB),DISP=SHR
//OUTMSGSD DD DSN=&&IMSGS,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880)
//INSKEL DD DSN=STROBE.CNTL(STRSLIB),DISP=SHR
//OUTSKEL DD DSN=&&ISKEL,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880)
//INMESS DD DSN=STROBE.CNTL(STRMSGSD),DISP=SHR
//OUTMESS DD DSN=&&MSGSD,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880)
//*****
/* NOW CREATE THE STROBE/ISPF LIBRARIES
//*****
/* PANEL LIBRARY
/*
//PLIB EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=STROBE.ISPPLIB,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(8880,(250,30,75),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880),
// UNIT=SYSDA,VOL=SER=
//SYSIN DD DSN=&&IPANL,DISP=(OLD,PASS)
//*****
/* MESSAGE LIBRARY
/*
//MLIB EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=STROBE.ISPMLIB,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(8880,(40,5,15),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880),
// UNIT=SYSDA,VOL=SER=
//SYSIN DD DSN=&&IMSGSD,DISP=(OLD,PASS)
//*****
/* SKELETON LIBRARY
/*
//SLIB EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=STROBE.ISPSLIB,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(8880,(20,5,5),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880),
// UNIT=SYSDA,VOL=SER=
//SYSIN DD DSN=&&ISKEL,DISP=(OLD,PASS)
//*****
/* STROBE MESSAGE LIBRARY
/*
//MESSAGES EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=STROBE.MESSAGES,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(8880,(250,30,75),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880),
// UNIT=SYSDA,VOL=SER=
//SYSIN DD DSN=&&MSGSD,DISP=(OLD,PASS)

```

Modify Option Menu Panel

Modify the STROBE Options menu (member STRBDFLT in Figure 3-21) of the ISPLLIB panel. The parameter values you set here also apply to APMpower and iSTROBE.

Figure 3-21. Modify the STROBE/ISPF Options Menu (Member STRBDFLT)

```

) ATTR DEFAULT (%+_)
) BODY EXPAND (\\)
%-\ STROBE - DEFAULT OPTIONS \-
%COMMAND ==>_OPTION
)PROC
&WORKUNIT = 'SYSDA'          /* UNIT NAME FOR WORK DATASETS      /*
&WORKVOL = ' '              /* VOLSER FOR WORK DATASETS        /*
&WORKPRI = '000010'        /* PRIMARY SPACE ALLOCATION (TRKS)  /*
&WORKSEC = '000032'        /* SECONDARY SPACE ALLOC. (TRKS)  /*
&STRDDISP = 'CATLG'        /* DEFAULT DISP OF SAMPLE DATASET /*
&STRBLOAD = 'STROBE.LOAD'  /* STROBE UNAUTHORIZED LOAD LIBRARY/*
&STRBETXT = 'STROBE.MESSAGES' /* STROBE MESSAGE TEXT LIBRARY  /*
&STRDSQL = 'STRBSPF'      /* 2ND LVL QUALIFIER FOR BROWSE DSN/*
&STRSORTL = ' '          /* SORTLIB LIBRARY DSN            /*
&STRDSYSN = ' '          /* SYSIN DATA SET FOR THE REPORTER /*
&STRSMS3 = 'NO'         /* ALLOC SAMPLE DS USING SMS PARMS /*
&STRSMS = TRUNC(&STRSMS3,1) /* USE 1ST CHAR ONLY (FOR PANELS) /*
&STRHIST = 'STROBE.STRHIST' /* STROBE HISTORY FILE           /*
&STRHISTA = STROBE.STRHISTA.PATH /* STROBE HISTORY FILE (ALT INDEX) /*
&STRHLOG = STROBE.STRHIST.LOG /* STROBE HISTORY LOG           /*
&STRBCSS = ' '          /* USER CSS LIBRARY              /*
&WORKSPAC = 'TRACK'      /* TEMPORARY DATA SET UNIT       /*
&SYSPRSEC = '000400'     /* TEMPORARY DATA SET SPACE ALLOC /*
&APMPower = 'Y'         /* APMPower INSTALLED: Y/N       /*
&ISTROBE = 'Y'         /* ISTROBE INSTALLED: Y/N       /*
&ISTRROOT = ' '        /* ROOT DIRECTORY FOR ISTROBE    /*
&IPADDR = ' '          /* IPADDR OF ROOT DIR. FOR ISTROBE /*
&ICODEPAG = 'IBM-STD'   /* CODE PAGE FOR XFER TO ASCII   /*
&STRXLTSW = 'Y'        /* SHOW TRANS. TABLE ON PANEL Y/N /*
&IXLATE = ' '          /* TRANSLATION TABLE           /*
&FTP = 'FTP'           /* NAME OF FTP PROGRAM FOR ISTROBE /*
&SELTLOC = '/usr/strobe/java' /* LOCATION OF CLASS SELECTOR    /*
&ZQ = &Z
IF (&OPTION = ' ')
  &ZQ = TRUNC(&OPTION, '.')
IF (&ZQ = &Z)
  &ZTRAIL = .TRAIL
  &OPTION = &ZQ
)END

```

Edit the following fields:

- **&STRBLOAD**—the name of your unauthorized STROBE library (default STROBE.LOAD).
- **&STRBETXT**—your name for the data set that contains STROBE/ISPF messages (default STROBE.MESSAGES).

You may also want to edit

- **&WORKUNIT**—unit name for work data sets used by STROBE, to a unit of your preference (default SYSDA).
- **&WORKVOL**—volume serial number for work data sets (default blank).
- **&WORKPRI**—primary space allocation for work data sets (default 10 TRACKS).
- **&WORKSEC**—secondary space allocation for work data sets (default 32 TRACKS for STRBDFLT).
- **&STRDSQL**—second-level qualifier for browse data sets created by STROBE/ISPF and APMpower (default STRBSPF).
- **&STRSORTL**—name of the library that contains the sort/merge program, if your sort/merge program is not link list resident (default blank).

- **&STRDDISP**—default disposition of sample data sets (CATLG (default) or KEEP).
- **&STRSMS3**—whether STROBE/ISPF applies SMS parameters to data sets that it creates (YES or NO (default)).

Note: Do not change the line containing of &STRSMS3 where the following value is required for use by ISPF:

```
&STRSMS = (TRUNC (&STRSMS, 1))
```

- **&STRBCSS**—name of the the Compuware Shared Services (CSS) load library where the DDIO utility modules are located.
- **&WORKSPAC**—unit type for temporary data set specification (default TRACK, you should specify BLOCK or CYLINDER if that is the unit type at your site).
- **&SYSPRSEC**—allocation, in units, of &WORKSPAC for temporary data set allocation (default 400 TRACKS).
- **&APMPOWER**—whether APMpower has been installed at this site (Y (default) or N).

The following parameters must be defined to enable STROBE to create Performance Profiles for iSTROBE.

- **&ISTROBE**—whether iSTROBE is available as a STROBE/ISPF panel choice (Y (default) or N).
- **&ISTRROOT**—the name of the root directory to which iSTROBE data files are sent.
- **&IPADDR**—the IP address of the host where iSTROBE is installed.
- **&ICODEPAG**—the name of the code page to be used for translation to the ASCII character set from the EBCDIC character set. Valid values are:
 - IBM-037 (default)
 - IBM-273: German
 - IBM-277: Danish
 - IBM-278: Swedish
 - IBM-280: Italy
 - IBM-284: Spanish
 - IBM-285: United Kingdom
 - IBM-297: French
 - IBM-423: Greek
 - IBM-424: Israel
 - IBM-500: Switzerland
 - IBM-870: Polish
 - IBM-875: Greek
- **&STRXLTSW**—whether translate table name field shows on STROBE-ISTROBE PERFORMANCE PROFILE OPTIONS panel (Y (default) or N)
- **&FTP**—name of FTP program installed on the MVS system that iSTROBE calls. (default FTP).
- **&SELTLOC**—name of the directory containing the Java selector jar file.
- **&IXLATE**—name of a customized code page translate table if you use one in addition to the standard IBM code page. This field must contain the data set name middle qualifier FTP uses to identify the customized code page.

iSTROBE FTP Communications

A skeleton JCL library is shipped with iSTROBE that includes FTP commands that enable the STROBE mainframe component to copy files to STROBE on the workstation. This JCL is located in the STRBPJCL member in the ISPSLIB skeleton library. You can change the FTP-related JCL if the FTP package configuration at your site has specific requirements. For example, the root directory of the file server with which you want to connect might

be different than the IP address used to connect. If it is, you can edit FTPSTEP in the STRBPJCL member to add a CD statement to point to the correct directory.

Figure 3-22. Editing FTP Commands (Member STRBPJCL)

```

)SEL &IXLATEUS = &Z
//FTPSTEP EXEC PGM=&FTP,COND(8,1e),PARM='IPADDRUS \EXIT'
)ENDSEL
)SEL &NETRCW -=&Z
//NETRC DD DSN=&NETCRW,DISP=SHR
)ENDSEL
//SYSFTPD DD *
CLIENTERRCODES TRUE
)SEL &CCXLATE -= &Z
CCTrans &CCXLATE
)ENDSEL
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//INPUT DD *
CD /LEVEL1 <- ADDED JCL LINE
CD /&OSTRRTUS/ISTROBE/PROFILES

```

Accessing the STROBE Packaging Utility

The STROBE packaging utility provides a fast and reliable method of preparing data sets, including sample and log data sets with variable-length records, for delivery to the technical support department. The XMIT utility is used by the packaging utility to maintain variable block format and built-in error detection can be provided.

By selecting sample and log data set names and other data sets from the utility's ISPF panels, you can generate a data set containing the specified sample, log, and other data sets that can be shipped to technical support. Multiple files can be transmitted through one output file that contains all of the input data sets and PDS members.

Note: The IBM TRSMAN compression utility, which is available from IBM web sites, can be optionally integrated with running the packaging utility. TRSMAN provides a method to compress the output file before you send it to technical support. Any file compressed by TRSMAN is assigned a name suffix of ".XM1".

To access the packaging utility from STROBE/ISPF, an EXEC named STRBPKG is shipped in STROBE.CNTL. If you have changed the name of STROBE.CNTL, you have to edit the member STROBE (Figure 3-24 on page 3-41) by replacing STROBE.CNTL with the name of the library where you keep STRBPKG. If you want to place STRBPKG in your SYSEXEC concatenation, comment out the activate and deactivate ALTLIB statements in the STROBE CLIST.

Two other members named OSTARXMT and OSTARREC are shipped in the STROBE control library and must either remain in the control library or be copied to the SYSEXEC concatenation along with STRBPKG.

In these members, you have to edit the following parameters:

- Change OSTARREC.CSUPP.SHARED.LOADLIB to the name of your unauthorized STROBE load library
- Change TRSMAN.CSUPP.SHARED.LOADLIB to the name of the library where you installed TRSMAN.

Output Data Set Formats

The STROBE packaging utility will produce data sets in three types of formats, depending on whether you ran the TRSMAN compression utility while creating the data sets.

- Files with an .XMT suffix are single MVS FB LRECL=88 files containing one or more MVS sequential or partitioned data sets with additional error detection data.
- Files with an .XM1 suffix are compressed versions of an .XMT file. RECFM=FB and LRECL=1024 are the format types of these files. .XM1 files are created when the TRSMAN utility is installed on your system.

Files with an .XM2 suffix are high compression versions of a single large sequential data set. Usually, the .XM2 format is appropriate for a file dump, but no error detection data is provided. RECFM=FB and LRECL=1024 are the format types for these files. .XM2 files are created when the TRSMAN utility is installed on your system and input to the packaging utility is a single, large, sequential data set

Installing the Java Targeting Selector (Java Feature Users Only)

The *STROBE Java Feature* enables you select and target for measurement Java applications running under the Java Virtual Machine (JVM). To specify targets, you can either specify what you want to measure through an ISPF panel text entry, or you can make a request for STROBE to list a set of Java targets. The request is made by entering “Y” on the “List Java Targets function available off of the STROBE - JAVA TARGETING panel. Entering “Y” causes STROBE to execute a Java utility that builds the list of the Java packages, classes and methods it can identify.

A jar file named *SELECTOR* is shipped as a object member in STROBE.CNTL. To install the Java target selector:

1. Change STROBE.CNTL to the name you want to use for your STROBE control data set.
2. Change SELECTR1.JAR to the name of a valid TSO dataset.
3. Execute JAVAINST to extract the SELECTOR jar file from the control library and copy it to a Unix System Services directory. The default USS directory is */ust/strobe/java*. (Compuware recommends that whoever installs the target selector have an active Unix Systems Services (USS) account.)
4. If you want to change the default directory location:
 - Edit the STRBDFLT member shown in Figure 3-21 on page 3-36 to change the variable “SELTLOC” value to name of the USS directory you want to use.
 - Update the HFSOUT DD statement of the COPY step of the JAVAINST member shown in Figure 3-23 on page 3-40 with the correct USS directory name.

Refer to the *STROBE Java Feature* for more information about using the Java target selector.

Figure 3-23. Install the Java Targeting Selector (Member JAVAINST)

```

/**
/**
/**
/**
/** RESTORE THE SELECTOR JAR FILE TO ORIGINAL FORMAT. THIS FILE WAS **
/** ORIGINALLY CREATED USING THE TSO TRANSMIT COMMAND **
/** **
/**
/**
/** RESTORE EXEC PGM=IKJEFT01,DYNAMNBR=20
/**
/**
/**
/** INFILE DSN: FILE OF 80-BYTE RECORDS CREATED VIA TSO TRANSFER **
/**
/** INFILE DD DISP=SHR,DSN=STROBE.CNTL(SELECTOR)
/**
/**SYSPRINT DD SYSOUT=*
/**SYSTSPRT DD SYSOUT=*
/**
/**
/** DATASET: OUTPUT FILE DSN, FULLY-QUALIFIED AND QUOTED **
/**
/**SYSTSIN DD *
RECEIVE INDD(INFILE) NODISPLAY
DATASET('SELECTR1.JAR')
/**
/**
/** COPY DATASET TO USS DIRECTORY **
/**
/**COPY EXEC PGM=IKJEFT01,DYNAMNBR=300,COND=EVEN
/**SYSTSPRT DD SYSOUT=*
/**INFILE DD DISP=SHR,DSN=SELECTR1.JAR
/**HFSOUT DD PATH='/usr/strobe/java/selector.jar',
/** PATHDISP=(KEEP,DELETE),
/** PATHOPTS=(ORDWR,OCREAT,OEXCL),
/** PATHMODE=(SIRWXU,SIXGRP,SIXOTH)
/**SYSPRINT DD SYSOUT=*
/**SYSTSIN DD DATA,DLM='>'
OCOPY INDD(INFILE) OUTDD(HFSOUT) BINARY PATHOPTS(USE)
/>
/**
/**
/** DELETE OUTFILE - CLEAN UP **
/**
/**DELETE EXEC PGM=IEFBR14
/**DD1 DD DIS=(OLD,DELETE).DSN=SELECTR1.JAR
/**
/**SYSPRINT DD SYSOUT=*
/**SYSTSPRT DD SYSOUT=*
/**

```

Modify CLISTS

Modify the TSO CLIST

Member STROBE in the STROBE control data set includes a TSO CLIST (Figure 3-24 on page 3-41) for executing STROBE/ISPF. The CLIST uses the ISPF/PDF LIBDEF service to allocate and concatenate application-level libraries.

Figure 3-24. TSO CLIST for Testing STROBE/ISPF (Member STROBE)

```

PROC 0
CONTROL MSG NOFLUSH NOPROMPT
/*****
/*PRE-ALLOCATE SQLAF TEMP FILES WITH UNIT PARAMETER OTHER THAN VIO
/*****
ALLOC FI (SYSTMP01 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
          DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
ALLOC FI (SYSTMP02 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
          DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
ALLOC FI (SYSTMP03 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
          DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*ALLOC FI (SYSTMP04 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
/*          DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*ALLOC FI (SYSTMP05 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
/*          DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*ALLOC FI (SYSTMP06 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
/*          DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*
/*****
/*DEFINE STROBE/ISPF LIBRARIES WITH SPECIFIED DDNAME
/*****
ISPEXEC LIBDEF ISPPLIB DATASET ID('STROBE.ISPPLIB')
ISPEXEC LIBDEF ISPMLIB DATASET ID('STROBE.ISPMLIB')
ISPEXEC LIBDEF ISPSLIB DATASET ID('STROBE.ISPSLIB')
/*
/*****
/*ACTIVATE STROBE/ISPF ALTLIB FOR REXX EXECs
/*****
ALTLIB ACT APPLICATION(EXEC) DATASET('STROBE.CNTL')
/*
/*****
/*EXECUTE STROBE/ISPF PROGRAM
/*****
ISPEXEC SELECT PGM(STRBISPF) NEWAPPL(STR) PASSLIB
/*
/*****
/*REMOVE STROBE/ISPF LIBRARIES
/*****
ISPEXEC LIBDEF ISPPLIB DATASET
ISPEXEC LIBDEF ISPMLIB DATASET
ISPEXEC LIBDEF ISPSLIB DATASET
/*
/*****
/* DEACTIVATE STROBE/ISPF REXX EXEC ALTLIB
/*****
ALTLIB DEA APPLICATION(EXEC)
/*
/*****
/*FREE PRE-ALLOCATED SQLAF TEMPORARY FILES
/*****
FREE F1(SYSTMP01 SYSTMP02 SYSTMP03)
/*FREE F1(SYSTMP04 SYSTMP05 SYSTMP06)

```

The CLIST invokes the ISPF LIBDEF service to define panel, message, and skeleton libraries, executes the STROBE program STRBISPF, and invokes the LIBDEF service again to free the libraries when you exit.

Make the following edits to member STROBE:

- Change the data set names of the STROBE/ISPF common libraries to match those that you have just installed.
- If you are testing with renamed STROBE modules, change the name of the STROBE/ISPF module STRBISPF in the SELECT PGM statement.
- If temporary datasets are not allowed to be allocated to VIO for TSO users, uncomment the allocation of the SYSTMPn datasets and enter a valid unit.
- Change STROBE.CNTL to the name you are using for the STROBE control data set in the ALTLIB activate and deactivate statements. As an alternative you can copy the REXX routines (STRBPKG for the data set packaging utility or STRBJTG for the Java targeting selector) into their SYSEXEC concatenations. In this case, you should comment out the ALTLIB activate and deactivate statements.

Use this CLIST in either of the following ways:

- Execute the CLIST from the command line of any ISPF/PDF panel.

```
TSO EXEC 'STROBE.CNTL(STROBE)'
```

- Copy member STROBE to a data set in your SYSPROC concatenation and modify your ISPF/PDF panels to invoke the CLIST, as described in “Modify the ISPF/PDF Panels” on page 3-44.

Modify the APMpower CLIST

Member APMPOWER in the STROBE control data set includes a TSO CLIST (Figure 3-25 on page 3-43) for executing the APMpower MVS Component. The CLIST uses the ISPF/PDF LIBDEF service to allocate and concatenate application-level libraries.

Figure 3-25. Modify the CLIST for Executing APMpower (Member APMPOWER)

```

CONTROL MSG NOFLUSH NOPROMPT
SET SYSOUTTRAP = 15
PROFILE
SET INDXLIM = &SYSOUTLINE
SET &NOMODE = NOMODE
SET &NOWTPM = NOWTPMSG
SET &NOINT = NOINTERCOM
SET &MODESW = 0
SET &WTPMSW = 0
SET &INTRSW = 0
SET INDX = 1
DO WHILE &INDX <= &INDXLIM
  SET PROFLINE = &&SYSOUTLINE&INDX
  SET TESTMODE = &SYSINDEX(&NOMODE,&PROFLINE)
  SET TESTWTPM = &SYSINDEX(&NOWTPM,&PROFLINE)
  SET TESTINTR = &SYSINDEX(&NOINT,&PROFLINE)
  SET &INDX = &INDX + 1
  IF &TESTMODE ^= 0 THEN +
    SET &MODESW = 1
  IF &TESTWTPM ^= 0 THEN +
    SET &WTPMSW = 1
  IF &TESTINTR ^= 0 THEN +
    SET &INTRSW = 1
END
PROFILE NOMODE NOINTERCOM NOWTPMSG
SET SYSOUTTRAP = 0
/*****
/*DEFINE STROBE/PFM LIBRARIES WITH SPECIFIED DDNAME
*****/
ISPEXEC LIBDEF ISPPLIB DATASET ID('STROBE.ISPPLIB')
ISPEXEC LIBDEF ISPMLIB DATASET ID('STROBE.ISPMLIB')
ISPEXEC LIBDEF ISPSLIB DATASET ID('STROBE.ISPSLIB')
/*****
/*PRE-ALLOCATE SQLAF TEMP FILES WITH UNIT PARAMETER OTHER THAN VIO
*****/
ALLOC FI (SYSTMP01 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
  DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
ALLOC FI (SYSTMP02 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
  DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
ALLOC FI (SYSTMP03 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
  DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*ALLOC FI (SYSTMP04 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
/*
  DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*ALLOC FI (SYSTMP05 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
/*
  DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*ALLOC FI (SYSTMP06 BLOCK(8876) BLKSIZE(8876) SPACE(200,50) +
/*
  DSORG(PS) RECFM(F) LRECL(0) NEW REU UNIT(SYSDA)
/*****
/*EXECUTE STROBE/PFM PROGRAM
*****/
ISPEXEC SELECT PGM(STRBSPFM) NEWAPPL(AMC) PASSLIB
/*
/*****
/*REMOVE STROBE/PFM LIBRARIES
*****/
ISPEXEC LIBDEF ISPPLIB DATASET
ISPEXEC LIBDEF ISPMLIB DATASET
ISPEXEC LIBDEF ISPSLIB DATASET
/*****
/* FREE PRE-ALLOCATED SQLAF TEMPORARY FILES
*****/
/*FREE FI(SYSTMP01 SYSTMP02 SYSTMP03)
/*FREE FI(SYSTMP04 SYSTMP05 SYSTMP06)
SET SYSOUTTRAP = 15
IF &MODESW = 0 THEN +
  PROFILE MODE
IF &WTPMSW = 0 THEN +
  PROFILE WTPMSG
IF &INTRSW = 0 THEN +
  PROFILE INTERCOM
SET SYSOUTTRAP = 0

```

The CLIST invokes the ISPF LIBDEF service to define panel, message, and skeleton libraries, executes the STROBE program STRBSPFM, and invokes the LIBDEF service again to free the libraries when you exit.

Make the following edits to the CLIST:

- Change the data set names of the STROBE/ISPF common libraries to match those that you have just installed.
- If you are testing with renamed STROBE modules, change the name of the APMpower module STRBSPFM in the SELECT PGM statement.
- If temporary datasets are not allowed to be allocated to VIO for TSO users, uncomment the allocation of the SYSTMPnn datasets and enter a valid unit.

Modify the ISPF/PDF Panels

You need to determine how to best modify your site's existing ISR@PRIM or USR@PRIM panel to invoke the STROBE CLIST; then create a new option on the panel that executes the CLIST.

CAUTION:

- Make sure the &ZTRAIL = .TRAIL statement appears in the PROC section of the panel.
- Remember to reapply the STROBE/ISPF changes after performing any SMP/E maintenance.

Test CLISTS

Verify that the STROBE/ISPF and APMpower CLISTS are working correctly.

Test the STROBE/ISPF CLIST

1. Execute the CLIST from the command line of any ISPF/PDF panel.

```
TSO EXEC 'STROBE.CNTL(STROBE)'
```

2. Select Option T from the Options menu to go to the STROBE/ISPF Tutorial, which includes an Add Active Request example.
3. Select Option 1 from the Tutorial Table of Contents.

Test the APMpower CLIST

To test or access the APMpower MVS component

1. Execute the CLIST from the command line of any ISPF/PDF panel.

```
TSO EXEC 'STROBE.CNTL(APMPOWER)'
```

2. At the option line, type

```
%logon
```

3. Press **Enter** and scan for the following text among the first few lines of the output.

```
LOGON SUCCESSFUL
```

4. At the option line, type

```
%logoff
```

5. Press **Enter** to display the message

```
SUCCESSFUL COMPLETION OF LOGOFF COMMAND
```

6. Press **Enter** again to return to the original ISPF/PDF panel.

Step 15. Install the SQL Analysis Feature

This section describes setting up the SQL Analysis Feature for iSTROBE and the DB2 Feature and for APMpower. If you are not licensed to use iSTROBE and the DB2 Feature or

APMpower, proceed to “Step 17. Install a TSO CLIST to Execute the Session Requester (Optional)” on page 3-52.

For this procedure	Refer to
Edit and execute member SQLINSTL	“Installation Procedures” on page 3-45
Bind the DBRMs	“Other Considerations” on page 3-46
Submit Grant statements in SPUFI	“Grant DB2 Authority” on page 3-50
Improve SQL Analysis Feature performance	“SQL Analysis Feature Performance (Optional)” on page 3-49
Establish dynamic EXPLAIN authority, if required	“Establish Dynamic EXPLAIN Authority” on page 3-50
Check whether the SQL Analysis Feature installed correctly	“Step 16. Verify the SQL Analysis Feature Installation” on page 3-50

MVS Component Parts

The SQL Analysis Feature MVS component is comprised of the following members in the STROBE control data set:

- SQLINSTL contains job control statements to build a DB2 version-specific load module.
- STRAIA51 and STRANA51 contain the DB2 Version 5 Release 1 object module. STRAIA61 and STRANA61 contain the DB2 Version 6 Release 1 object module. STRAIA71 and STRANA71 contain the DB2 Version 7 Release 1 object module.
- STRAN151 and STRAN251 contain the DB2 Version 5 Release 1 DBRMs. STRAN161 and STRAN261 contain the DB2 Version 6 Release 1 DBRMs. STRAN171 and STRAN271 contain the DB2 Version 7 Release 1 DBRMs.
- STRANBND contains job control statements to bind to DB2 subsystems.
- STRANGNT contains job control statements to grant users required DB2 privileges.
- SQLAVER contains job control statements to test the Feature through MVS.
- STRANSMP contains job control statements to test the Feature from APMpower.

Note: If you apply maintenance to your DB2 subsystem that affects load module DSNALI, relink the SQL Analysis Feature load module, STRANA##, where ## is the DB2 release number (51, 61 or 71) and re-execute the SQLINSTALL member.

Installation Procedures

Use member SQLINSTL (Figure 3-26 on page 3-46) to install the SQL Analysis Feature. To install the Feature:

1. Make the following edits to SQLINSTL:
 - supply a JOB statement
 - replace STROBLIB with the name of your unauthorized STROBE library
 - replace STROBE.CNTL with the name of your STROBE control data set
 - change the value of DB2LOAD to the name of the DB2 load library that contains member DSNTIAR for the appropriate release(s) of DB2. Make sure the DB2 load library is included in the link list or the load library must be specified in JCL language that you use to produce the Performance Profile.
 - APMpower installations without iSTROBE available should delete all LINKSQL## EXEC steps for any releases marked with the parameter IREL=I and any releases marked with the parameter IREL=N you are not using.

- iSTROBE installations without APMpower available should delete all LINKSQL## EXEC steps for any releases marked with the parameter IREL=N and any other DB2 releases marked with the parameter IREL=I you are not using.
- Installations with both APMpower and iSTROBE should delete any DB2 JCL steps that are not necessary.

2. Execute the job.

Figure 3-26. Install APMpower and iSTROBE SQL Analysis Feature (Member SQLINSTL)

```

/*
//LINK      PROC LPARM=NCAL,REL=,CNTL=,DB2LOAD=
//LKED      EXEC PGM=IEWL,REGION=1024K,
//          PARM='XREF,LIST,LET,&LPARM'
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD   DD DISP=SHR,DSN=STROBLIB
//STRANAB   DD DISP=SHR,DSN=&CNTL(STRA&IREL.A&REL)
//DB2LIB    DD DISP=SHR,DSN=&DB2LOAD
//SYSLIN    DD DISP=(SHR,PASS),DSN=&CNTL.(SQLOBJ&REL)
//          PEND
/*
//LINKSQL51 EXEC LINK,DSN='STROBE.CNTL',REL=51,IREL=N,OBJ=51,
//          DB2LOAD='DB2.DSN510.SDSNLOAD'
//LINKSQL61 EXEC LINK,DSN='STROBE.CNTL',REL=61,IREL=N,OBJ=61,
//          DB2LOAD='DB2.DSN610.SDSNLOAD'
//LINKSQL71 EXEC LINK,DSN='STROBE.CNTL',REL=71,IREL=N,OBJ=71,
//          DB2LOAD='DB2.DSN710.SDSNLOAD'
//LINKSQL5I EXEC LINK,DSN='STROBE.CNTL',REL=51,IREL=I,OBJ=5I,
//          DB2LOAD='DB2.DSN510.SDSNLOAD'
//LINKSQL6I EXEC LINK,DSN='STROBE.CNTL',REL=61,IREL=I,OBJ=6I,
//          DB2LOAD='DB2.DSN610.SDSNLOAD'
//LINKSQL7I EXEC LINK,DSN='STROBE.CNTL',REL=71,IREL=I,OBJ=7I,
//          DB2LOAD='DB2.DSN710.SDSNLOAD'

```

Other Considerations

To generate the SQL Analysis Feature Performance profile reports, a DB2 module must be loaded. If the DB2 modules are included in the link list, the procedure described next is not required. If the link list does not include DB2, you must add a STEPLIB statement to identify the DB2 load library for the version of DB2 available at your site. This statement should be added to the STRBPLOD member shipped with the STROBE. ISPSLIB skeleton library. Figure 3-27 shows STRBPLOD as it is originally shipped.

Figure 3-27. (Member STRBPLOD)

```

)SEL &STRBLOAD -= &Z
//STEPLIB      DD DISP=SHR,
//            DSN=&STRBLOAD
)ENDSEL

```

Figure 3-28 shows an edited version of STRBPLOD with the changes you could make to identify the DB2 load library.

Figure 3-28. Identifying a DB2 Load Library (Member STRBPLOD)

```

//STEPLIB      DD DISP=SHR,
//            DSN=DB.LOAD           <- ADDED JCL LINE
)SEL &STRBLOAD -= &Z             <- MOVED JCL LINE
//            DD DISP=SHR,         <- ADDED JCL LINE
//            DSN=&STRBLOAD
)ENDSEL

```

Bind the DB2 Version-Specific DBRMs

Member STRANBND (Figure 3-29 on page 3-47) contains the job that binds the DB2 version-specific DBRMs of the SQL Analysis Feature to the DB2 subsystem against which it executes.

BIND Considerations

The authorization ID that issues the BIND request must have one of the following DB2 privileges:

- SYSADM authority
- BINDADD authority
- BINDAGENT authority
- BIND authority for package collection STRANANA and plan STRANANA

You must BIND the DB2 version-specific DBRM members to each DB2 subsystem against which the SQL Analysis Feature will execute.

Figure 3-29. Bind for the DB2 Version-Specific DBRM (Member STRANBND)

```

/**
//BINDANA EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=DB2.EXIT.LIB
// DD DISP=SHR,DSN=DB2.LOAD.LIB
//DBRMLIB DD DISP=SHR,DSN=STROBE.CNTL
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//REPORT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSTSIN DD *
DSN SYSTEM(SSID)

BIND PACKAGE(STRANANA) -
MEMBER(STRAN1##) -
ACTION(REPLACE) -
ISOLATION(CS) -
QUALIFIER(SYSIBM) -
OWNER(00000000)

BIND PACKAGE(STRANANA) -
MEMBER(STRAN2##) -
ACTION(REPLACE) -
ISOLATION(CS) -
QUALIFIER(SYSIBM) -
OWNER(00000000)

BIND PACKAGE(STRANANA) -
MEMBER(STRANV24) -
ACTION(REPLACE) -
ISOLATION(CS) -
QUALIFIER(SYSIBM) -
OWNER(00000000)

BIND PLAN(STRANANA) -
PKLIST(STRANANA.*,STRANANP.*) -
ACTION(REPLACE) RETAIN -
ISOLATION(CS)
END
//

```

To BIND the DB2 version-specific DBRM members

1. Make the following edits to member STRANBND:
 - Supply a JOB statement.
 - Change DB2.EXIT.LIB to the name of the DB2 exit load library. (Delete this DD statement if you do not use exit libraries.)

- Change DB2.LOAD.LIB to the name of your DB2 load library for the specific release of DB2 against which you will execute.
 - Change SSID to the specific ID of the DB2 subsystem against which you issue BIND.
 - Change the STRAN1## and STRAN2## BIND parameters to the specific release of DB2 against which the SQL Analysis Feature executes.
 - STRAN151 and STRAN251 or
 - STRAN161 and STRAN261 or
 - STRAN171 and STRAN271.
- In languages other than English, the pound signs (##) may display as other characters.
- Change OOOOOOOO to the authorization ID of the application plan owner.

2. Execute the job.

Using Shadow Catalogs (Optional)

The SQL Analysis Feature can retrieve DB2 catalog information from shadow copies of DB2 catalog tables, if you bind the DBRMs used by the Feature into a package collection that can use the shadow catalog tables.

To use shadow catalogs

1. Add the following JCL to your edited copy of member STRANBND (Figure 3-29 on page 3-47):

```

BIND PACKAGE(COLSHADO) -
MEMBER(STRAN131) -
ACTION(REPLACE)
ISOLATION(CS) -
QUALIFIER(SYSIBM) -
OWNER(OOOOOOO)

BIND PACKAGE(COLSHADO) -
MEMBER(STRAN231) -
ACTION(REPLACE)
ISOLATION(CS) -
QUALIFIER(SYSIBM) -
OWNER(OOOOOOOOO)

BIND PACKAGE(COLSHADO) -
MEMBER(STRANV24) -
ACTION(REPLACE)
ISOLATION(CS) -
QUALIFIER(SYSIBM) -
OWNER(OOOOOOOOO)

BIND PLAN(STRANANA) -
PKLIST (STRANANA.* , COLSHADO.*) -
ACTION(REPLACE) RETAIN -
ISOLATION(CS)

END

```

2. Make the following edits
 - Change **COLSHADO** to the package set name you are using for the STRAN131 member.
 - Change **COLSHADO** to the package set name you are using for the STRAN231 member.
 - Change the package set **QUALIFIER (SYSIBM)** for the STRAN231 member to the creator ID of the DB2 catalog tables that make up the shadow catalog.

- Change the package set QUALIFIER (SYSIBM) for the STRANV24 member to the creator ID of the DB2 catalog tables that make up the shadow catalog.
 - Add the name of the package set name for the shadow catalogs to the PKLIST entry.
3. Repeat the first two steps for each DB2 release that your site supports.

SQL Analysis Feature Performance (Optional)

You can create additional indexes to improve the performance of the SQL Analysis Feature when it is retrieving DB2 catalog information. You can define Type 2 indexes on catalog tables. If you create the indexes shown in Figure 3-30 on page 3-49, the SQL Analysis Feature statements will invoke matching index scans when possible.

If you define the indexes, be sure to execute RUNSTATS against the tables and indexes and then rebind the SQL Analysis Feature to help ensure that the new indexes are used.

Figure 3-30. SQL Analysis Feature Indexes

```
--
-- THIS FILE CONTAINS DDL TO CREATE ADDITIONAL TYPE 2 INDEXES ON A DB2 v41 OR v51 CATALOG TO SPEED
-- RETRIEVAL OF CATALOG DATA BY SQLAF, WHERE:
--
-- INDEX-NAME.: NAMES A NEW INDEX THAT DOES NOT ALREADY EXIST AT THE CURRENT SERVER
--
-- USING-BLOCK: IDENTIFIES THE DATA SET MANAGEMENT STRATEGY
--
-- CREATE THE FOLLOWING 2 INDEXES IF YOU BIND DBRMS DIRECTLY TO PLANS. I.E, IF YOU DONT USE
-- PACKAGES EXCLUSIVELY
--
CREATE TYPE 2 INDEX INDEX-NAME
ON SYSIBM.SYSDBRM
(PLCREATOR,
 PLNAME,
 NAME)
USING USING-BLOCK
ERASE NO
CLOSE NO
--
CREATE TYPE 2 INDEX INDEX-NAME
ON SYSIBM.SYSSTMT
(PLCREATOR,
 PLNAME,
 NAME)
USING USING-BLOCK
ERASE NO
CLOSE NO
--
--CREATE THIS TYPE OF INDEXES IF YOU USE VIEWS
CREATE TYPE 2 INDEX INDEX-NAME
ON SYSIBM.SYSVIEWDEP
(DCREATOR,
 DNAME,
 USING USING-BLOCK
ERASE NO
CLOSE NO
--CREATE THE FOLLOWING TWO INDEXES IF YOU DEFINE REFERENTIAL CONSTRAINTS ON YOUR TABLES
--
CREATE TYPE 2 INDEX INDEX-NAME
ON SYSIBM.SYSFOREIGNKEYS
(CREATOR,
 TBNAME,
 USING USING-BLOCK
ERASE NO
CLOSE NO
--
CREATE TYPE 2 INDEX INDEX-NAME
ON SYSIBM.SYSRELS
(CREATOR,
 TBNAME,
 USING USING-BLOCK
ERASE NO
CLOSE NO
```

The following lists the tables on which the indexes are created and shows the types of processing for which the Feature will make use of the indexes.

Table 3-4. SQL Analysis Feature Optimizing Index Locations

Table	Processing Type
SYSDBRM	DBRMs bound directly to plans
SYSSTMT	Use existing PLAN_TABLE rows for DBRMs bound directly to plans
SYSVIEWDEP	SQL statements reference views
SYSFOREIGNKEYS	Table objects defined with referential constraints
SYSRELS	Table objects defined with referential constraints

Grant DB2 Authority

Member STRANGNT (Figure 3-31) contains the SQL statement that authorizes users to execute plan STRANANA. Submit GRANT statements in SPUFI using member STRANGNT. If you do not wish to grant PUBLIC access, run a GRANT for each DB2 authorization ID that is allowed access.

Figure 3-31. Grant Statement for SPUFI (Member STRANGNT)

```
-- GRANT EXECUTE AUTHORITY ON PLAN STRANANA
GRANT EXECUTE ON PLAN STRANANA TO PUBLIC;
```

Establish Dynamic EXPLAIN Authority

Users of the SQL Analysis Feature may need authority to run EXPLAIN, depending upon the options specified at report time. For more information, see the description of dynamic EXPLAIN privileges in the APMpower Reference Guide or the STROBE User's Guide.

Step 16. Verify the SQL Analysis Feature Installation

STROBE provides you with two versions of an SQL Analysis Feature installation verification tool to demonstrate that the Feature load modules and DB2 objects have been installed correctly. One version permits the STROBE installer to determine if the Feature installation was successful. The other version creates a sample data set that you can use to create a Performance Profile from the SQL Analysis Feature user interface.

Verifying the Installation

STROBE provides an installation verification tool to demonstrate that the load modules and DB2 objects have been installed correctly. The tool creates a report showing which DB2 version-specific load modules are installed, analyzes the SQL Analysis Feature plan for the correct DBRMs, and lists the DB2 execute authorizations.

To verify the installation, edit and run the JCL in member SQLAFVER (Figure 3-32 on page 3-51) against all DB2 subsystems where the SQL Analysis Feature will be used:

1. Make the following edits to member SQLAFVER:
 - change STROBE.CNTL to the name of your PDS containing the copy of the STROBE distribution tape
 - change DB2.LOAD.LIB to the name of your DB2 load library that contains the DSNTIAR and connection modules
 - change STROBLIB to the name of your STROBE load library that contains the SQL Analysis Feature load modules

2. Each time you verify a different DB2 subsystem, change the SSID to the ID of the DB2 subsystem you are targeting for the test.
3. Execute the job and read the SYSPRINT output to make sure the BIND(s) worked for the DBRM versions you are using.

Figure 3-32. Execute the Verification Utility (Member SQLAFVER)

```

//VERIFY  PROC CNTL=,DB2LOAD=,STRBLIB=
//GO      EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=4096K
//STEPLIB DD DISP=SHR,DSN=&DB2LOAD
//        DD DISP=SHR,DSN=&STRBLIB
//STRLIB  DD DISP=SHR,DSN=&STRBLIB
//DBRMLIB DD DISP=SHR,DSN=&CNTL
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//        PEND
//*
//VERIFY  EXEC VERIFY,CNTL='STROBE.CNTL',
//        DB2LOAD='DB2.LOAD.LIB',STRBLIB='STROBLIB'
//SYSTSIN DD *
DSN SYSTEM(SSID)
RUN PROGRAM(STRANVER) PLAN(STRANANA)
END
/*
//

```

Verifying Product Use

STROBE enables you to also verify that the SQL Analysis Feature is operating and running correctly by using APMpower or iSTROBE. Two JCL members called STRANSMP (Figure 3-33 on page 3-52) and STRANSMI (not shown) create a workload that can be measured and provide a sample data set to generate a Performance Profile for APMpower or iSTROBE. STRANSMI runs the STRANVER program just described. It executes STRANVER 50 times. You can edit the STRANSMI member to increase or decrease the DB2 workload by altering the number of RUN statements in it.

1. Make the following edits to member STRANSMP:
 - replace STROBE.CNTL with the name of your STROBE control data set
 - change DB2.LOAD.LIB to the name of your DB2 load library containing the DSNTIAR and connection modules
 - change STROBLIB to the name of your STROBE load library containing the SQL Analysis Feature load modules
 - change SSID to the ID of the DB2 subsystem against which you will run the SQL Analysis Feature.
2. Execute the job.
3. Using the sample data set created by member STRANSMP, use STROBE/ISPF to create the iSTROBE or APMpower Performance Profile.
4. Download and view the measurement data with either APMpower or iSTROBE.

Figure 3-33. Create a Sample with SQLAF (Member STRANSMP)

```

//*
//VERIFY PROC CNTL=,DB2LOAD=,STRBLIB=
//* STRBCSR IS THE DEFAULT PROGRAM NAME. PLEASE VERIFY THAT
//* THIS NAME IS CORRECT WHEN USING THE TEST VERSION OF STROBE
//STROBE EXEC PGM=STRBCSR
//STEPLIB DD DSN=&STRBLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//GO EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=4096K
//STEPLIB DD DISP=SHR,DSN=&DB2LOAD
// DD DISP=SHR,DSN=&STRBLIB
//STROBLIB DD DISP=SHR,DSN=&STRBLIB
//DBRMLIB DD DISP=SHR,DSN=&CNTL
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
// PEND
//
//VERIFY EXEC VERIFY,CNTL='STROBE.CNTL',
// DB2LOAD='DB2.LOAD.LIB',STRBLIB='STROBLIB'
//STROBE.SYSIN DD *
ADD *,STEP=GO,GOMIN=1,SAMPLES=15000
//GO.SYSTSIN DD *
 DSN SYSTEM(SSID)
// DD DISP=SHR,DSN=&CNTL(STRANSMI)

```

Step 17. Install a TSO CLIST to Execute the Session Requester (Optional)

Member SRCLIST (Figure 3-34) of your STROBE control data set contains a TSO CLIST that you can execute to call the session requester. This CLIST uses the TSOEXEC command so that the session requester is invoked as an authorized program. Copy member SRCLIST to your CLIST library.

Figure 3-34. TSO CLIST to Execute the Session Requester (Member SRCLIST)

```

PROC 0
ALLOC F(SYSPRINT) DA(*) REUSE
ALLOC F(SYSIN) DA(*) REUSE
TSOEXEC STRBCSR
FREE F(SYSPRINT SYSIN)
END

```

Step 18. Put into Production (Optional, if you had a test version)

When you are ready to put STROBE into production:

1. Stop all STROBE address spaces.
2. If you have installed the release in the link list by renaming the STROBE modules, remove the renamed modules.
3. If you tested the release from an authorized library that is not in the link list concatenation, switch the parameter data set from CVLIB=APFLIB to CVLIB=LINKLIST.
4. Copy the release to the link list with member COPY.
5. Perform an LLA refresh.
6. Test the production release with member TESTSTR.

You can enable STROBE to start at IPL by defining it to your SYS1.PARMLIB concatenation. Add the following command to member COMMNDxx:

```
COM='S STRBSM'
```

Migration Utility

If you run the STROBE migration utility, you can use measurement requests created with any previous release of STROBE MVS for Sysplex with STROBE Release 3.0. There are two versions of the utility. The first (member MIGRATEQ) takes the queue data set of your old STROBE version and revises it so Release 3.0 can read the contents of the completed measurement requests. The second version (member MIGRATEG) is for Advanced Session Management Feature users and takes the group data set of your old STROBE version and revises it so Release 3.0 can read its contents. The third version (MIGRATEA) is for AutoSTROBE users and takes the AutoSTROBE data set of your old STROBE version and revises it so Release 3.0 can read its contents.

Note: If you are using the Advanced Session Management Feature to track STROBE measurements with history data sets, you are not required to migrate the history data set(s) to use them with STROBE Release 3.0.

Upgrading to STROBE MVS for Sysplex Release 3.0

The JCL members MIGRATEQ, MIGRATEG and MIGRATEA are set up to perform the necessary data set conversion process. MIGRATEQ converts queue data sets, MIGRATEG converts group data sets, and MIGRATEA converts AutoSTROBE data sets. MIGRATEG only has to be run if you are using STROBE Advance Session Management to issue groups of measurement requests. MIGRATEA only needs to be run if you have AutoSTROBE archived data or basis data that you want to retain for use with STROBE Release 3.0. If the previous version of STROBE from which you are migrating used the same XCF Group name (multisystem) or SMF ID (local mode) and STROBE version ID (production or test) as you plan to use with Release 3.0, you only have to provide your old and new queue (and group) data set names as explained next.

Note: If you are switching to a new type of STROBE environment, you have to provide some additional parameter information described in “Migrating Data Sets Across Different Environments” on page 3-57.

Migrating Queue Data Sets

The following explains how to migrate queue data sets:

1. Use the following MVS console command to stop the STROBE session manager if it is running:
 - STOP STRBSM (production version)
 - STOP STRTSM (test version)
2. Make the following edits to the MIGRATEQ member (Figure 3-35 on page 3-55)
 - Change STROBLIB to the name of your unauthorized STROBE Release 3.0 load library.
 - Change VOLSER to the name of the volume where you want to install the test queue and group data sets.
 - Change STROBE.V2RxMx.QUEUE to the name you used for the queue data set in the release of STROBE you want to migrate.
 - Change STROBE.V03R00.QUEUE to the name of your Release 3.0 queue data set.
3. Execute the job.

Note: Release 3.0 provides a corresponding test version member named MIGRTSTQ. If you created a test version of STROBE as described in “Step 12. Start the Address Space” on page 3-27, you should use the MIGRTSTQ member to migrate queue data sets to the test STROBE version. The test migration utility member only works for completed requests to avoid any possibility of contention between two versions of STROBE on the same system attempting to issue the same queued measurement request.

Migrating Group Data Sets (Advanced Session Management Only)

The following explains how to migrate group data sets:

1. Use the following MVS console command to stop the STROBE session manager if it is running:
 - STOP STRBSM (production version)
 - STOP STRTSM (test version)
2. Make the following edits to the MIGRATEG member (Figure 3-36 on page 3-56)
 - Change **STROBLIB** to the name of your unauthorized STROBE Release 3.0 load library.
 - Change **VOLSER** to the name of the volume where you want to install the test queue and group data sets.
 - Change **STROBE.V2RxMx.GROUP** to the name you used for the group data set in the release of STROBE you want to migrate.
 - Change **STROBE.V03R00.GROUP** to the name of your Release 3.0 group data set
3. Execute the job.

Release 3.0 provides a corresponding test version member named MIGRTSTG. If you created a test version of STROBE as described in “Step 12. Start the Address Space” on page 3-27, you should use the test version member to migrate group data sets to the test STROBE version.

Migrating the AutoSTROBE Data Set (Advanced Session Measurement Only)

1. Use the following MVS console command to stop the STROBE session manager if it is running:
 - STOP STRBSM (production version)
 - STOP STRTSM (test version)
2. If you want to use an AutoSTROBE archived and basis data from a previous release with STROBE Release 3.0, you have to back it up and then migrate the data set. Make the following edits to the MIGRATEA member (Figure 3-37 on page 3-57)
 - Change STROBLIB to the name of your unauthorized STROBE Release 3.0 load library.
 - Change STROBE.STRAHIST to the name of your AutoSTROBE data set.
 - Change STROBE.AHIST.BKUP to the name of your AutoSTROBE backup data set.
3. Execute the job.

Release 3.0 provides a corresponding test version member named MIGRTSTA. If you created a test version of STROBE as described in “Step 12. Start the Address Space” on page 3-27, you should use the test version member to migrate AutoSTROBE data sets to the test STROBE version.

Figure 3-35. Migration Utility (Member MIGRATEQ)

```

//ALLOCWK EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEFINE CLUSTER -
    (
        NAME(STROBE.WFILE1.QUEUE) -
        CYL(2 2) -
        VOL(VOLSER) -
        BUFFERSPACE(112640) -
        INDEXED -
        CISZ(22528) -
        KEYS(54 0) -
        RECORDSIZE(1044 22528) -
        SHAREOPTIONS(3 3) -
        UNIQUE -
    )
    DEFINE CLUSTER -
    (
        NAME(STROBE.WFILE2.QUEUE) -
        CYL(2 2) -
        VOL(VOLSER) -
        BUFFERSPACE(112640) -
        INDEXED -
        CISZ(22528) -
        KEYS(54 0) -
        RECORDSIZE(1044 22528) -
        SHAREOPTIONS(3 3) -
        UNIQUE -
    )
/*
//ALLOCNW EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DELETE STROBE.V2R5M0.QUEUE
    IF LASTCC = 8 THEN SET MAXCC = 0
    DEFINE CLUSTER -
    (
        NAME(STROBE.V03R00.QUEUE) -
        CYL(2 2) -
        VOL(VOLSER) -
        BUFFERSPACE(112640) -
        INDEXED -
        CISZ(22528) -
        KEYS(54 0) -
        RECORDSIZE(1044 22528) -
        SHAREOPTIONS(4 3) -
        UNIQUE -
    )
/*
//MIGRATE EXEC PGM=STRBCMIG,REGION=4M,COND(4,LT),
//
    PARM ='MIGRATE'
/*
    PARM ='MIGRATE,INREL=0250,OUTREL=0300'
//STEPLIB DD DISP=SHR,DSN=STROBLIB
//SYSPRINT DD SYSOUT=*
//SNAP DD SYSOUT=*
//REPORT DD SYSOUT=*
//SYSQUEUE DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.V2RXMX.QUEUE
//NEWQUEUE DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.V03R00.QUEUE
/*
//WF1QUEUE DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.WFILE1.QUEUE
//WF2QUEUE DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.WFILE2.QUEUE
/*
//DELETWRK EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DELETE STROBE.WFILE1.QUEUE
    DELETE STROBE.WFILE2.QUEUE

```

Figure 3-36. Migration Utility (Member MIGRATEG)

```

//ALLOCWK EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEFINE CLUSTER -
    (
        NAME(STROBE.WFILE1.GROUP) -
        CYL(2 2) -
        VOL(VOLSER) -
        BUFFERSPACE(112640) -
        INDEXED -
        CISZ(22528) -
        KEYS(54 0) -
        RECORDSIZE(1044 22528) -
        SHAREOPTIONS(3 3) -
        UNIQUE -
    )
    DEFINE CLUSTER -
    (
        NAME(STROBE.WFILE2.GROUP) -
        CYL(2 2) -
        VOL(VOLSER) -
        BUFFERSIZE (112640) -
        INDEXED -
        CISZ(22528) -
        KEYS(54 0) -
        RECORDSIZE(1044 22528) -
        SHAREOPTIONS(3 3) -
        UNIQUE -
    )
/*
//ALLOCNW EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE STROBE.V2R5M0.GROUP
IF LASTCC = 8 THEN SET MAXCC = 0
    DEFINE CLUSTER -
    (
        NAME(STROBE.V03R00.GROUP) -
        CYL(2 2) -
        VOL(VOLSER) -
        BUFFERSPACE(112640) -
        INDEXED -
        CISZ(22528) -
        KEYS(54 0) -
        RECORDSIZE(1044 22528) -
        SHAREOPTIONS(4 3) -
        UNIQUE -
    )
/*
//MIGRATE EXEC PGM=STRBCMIG,REGION=4M,COND(4,LT),
//
    PARM ='MIGRATE'
/*
    PARM ='MIGRATE,INREL=0250,OUTREL=0300'
//STEPLIB DD DISP=SHR,DSN=STROBE.LOADLIB
//SYSPRINT DD SYSOUT=*
//SNAP DD SYSOUT=*
//REPORT DD SYSOUT=*
//SYSGROUP DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.V2RxMx.GROUP
//NEWGROUP DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.V03R00.GROUP
/*
//WF1GROUP DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.WFILE1.GROUP
//WF2GROUP DD DISP=OLD,AMP=('BUFND=31'),DSN=STROBE.WFILE2.GROUP
/*
//DELETWRK EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE STROBE.WFILE1.GROUP
DELETE STROBE.WFILE2.GROUP

```

Figure 3-37. Migration Utility (Member MIGRATEA)

```

//*STROBE JOB 0,STROBE
//*
//*****//
//*                                     //
//*   BACKUP THE AUTOSTROBE DATASET   //
//*                                     //
//*****//
//HISTABCK EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  /* */
  /* */
  /* DELETE PREVIOUS BACKUP DATASET */
  /* */
  DELETE STROBE.STRAHIST.BKUP
  IF LASTCC =8 THEN SET MAXCC = 0
  /* */
  /* ALLOCATE THE BACKUP DATASET
  /* */
  DEFINE CLUSTER -
    (
      NAME(STROBE.STRAHIST.BKUP) -
      CYL(10 5) -
      VOL(VOLSER) -
      INDEXED -
      BUFFERSPACE (112640) -
      CISZ(22528) -
      KEYS (64 0) -
      RECORDSIZE(67 500) -
      SHAREOPTIONS(4 3) -
      UNIQUE -
    )
  /* */
  /* BACKUP AUTOSTROBE HISTORY DATASET
  /* */
  REPRO
    INFILE(STROBE.STRAHIST)
    OUTDATASET(STROBE.STRANHIST.BKUP
  /*

//*****//
//*                                     //
//*   RUN MIGRATION OF AUTOSTROBE DATASET   //
//*                                     //
//*****//
MIGRATEA PROC
//*
//* AUTOSTROBE HISTORY MIGRATION PROCEDURE
//*
//HISTAUMG EXEC PGM=STRHAUMG,REGION=1024K,TIME=1440
//SYSQUEUE DD DISP=SHR,DSN=STROBE.V03R00.QUEUE
//STRHISTA DD DISP=SHR,DSN=STROBE.STRAHIST
//REPORT DD SYSOUT=*
//SNAP DD SYSOUT=*
//SYSPRIN DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*

```

Migrating Data Sets Across Different Environments

If characteristics differ between your current STROBE environment and the environment where you are installing STROBE Release 3.0, you need to set certain parameters in the migration utility JCL to enable the migration utility to create a data set that can be used in the new environment. Table 3-5 on page 3-58 shows these parameters. The following are types of data set migration where these parameters need to be applied.

- local to XCF mode or XCF to local mode
- no Advanced Session Management to Advanced Session Management

If you want to copy queue or request group data sets from one MVS system to another, the migration utility provides a COPY parameter that makes copies of Release 3.0 STROBE data sets so they can be used on another MVS system with a different XCFGROUP or SMF ID.

Migration Requirements

The migration utility parameters you need to supply are based on the following information:

- STROBE version release number from which you want to migrate data sets. The default previous release number for the migration utility is Release 2.5.0. If you are migrating from an earlier STROBE for Sysplex version, you have to specify the release by editing the MIGRATEQ, MIGRATEG, or MIGRATEA JCL.
- STROBE version release version ID—if value changed from default values of B for production or T for test
- MVS system environment—the SMF ID and XCF group names at your site

CAUTION:

Before performing a data set migration you should use the utility and specify a LIST parameter to verify what actually exists in the data sets of your current STROBE environment. STROBE produces a report you can use to make sure that all data sets migrated from the current environment to the newly installed environment. The following is an example of the JCL you specify for the LIST parameter.

```
//MIGRATE EXEC PGM=STRBCMIG,REGION=4M,COND(4,LT),
// PARM='LIST,LINES=55'
```

Migration Utility Parameters

The STROBE Release 3.0 migration utility requires parameters that describe what type of data sets you were using with your previous version of STROBE. The following table describes the parameters.

Table 3-5. Parameters for the Migration Utility

Parameter	Purpose
COPY	Copies records without migrating. The input release (INREL) defaults to the most current release (0250) when paired with the COPY parameter and requires an INXCFGRP or INSMFID parameter. The COPY parameter is mutually exclusive with the MIGRATE parameter.
INREL	Identifies the 4-character input release number. The default is 0240 with MIGRATE, 0250 for COPY.
INSMFID	The 4-character identifier of the input SMFID (A-Z or 0-9); mutually exclusive with INXCFGRP.
INVERID	Identifies the 1-character identifier of the input version ID (A-Z, or 0-9); requires the INSMFID parameter, mutually exclusive with INXCFGRP. The default is B.
INXCFGRP	Identifies the 1- to 8-character input XCF group name (A-Z, 0-9, @, # and \$); mutually exclusive with INSMFID or INVERID.
LINE	Specifies how many lines should appear on a report.
LIST	Produces a report of the records in the queue and/or group data sets. Specifying LIST without the MIGRATE or COPY parameters will produce a report of the input queue and group data sets. Specifying LIST with the MIGRATE or COPY parameters will produce a report of the output queue and group data sets.
MIGRATE	Migrates from one release to the next. The input release (INREL) defaults to the previous release (0240). The MIGRATE parameter is mutually exclusive with the COPY parameter.
OUTREL	Identifies the current version of STROBE to where you are migrating (3.0). This parameter only needs to be specified when you are migrating to a version other than 3.0.
OUTSMFID	Identifies the 4-character output SMFID (A-Z and 0-9); mutually exclusive with OUTXCFGRP.

Table 3-5. Parameters for the Migration Utility

Parameter	Purpose
OUTVERID	Identifies the 1-character output version ID (A-Z or 0-9). Requires the OUTSMFID parameter. The default is T with TESTQDS parameter, B in all other cases. Mutually exclusive with OUTXCFGRP.
OUTXCFGRP	Identifies the 1-8 character output XCF group name (A-Z, 0-9, @, #, and \$); mutually exclusive with OUTSMFID or OUTVERID. Sets VERID to x'00' in all selected queue request elements.
TESTQDS	Creates a test queue data set. This parameter requires INXCFGRP or INSMFID and OUTXCFGRP or OUTSMFID. Only valid when MIGRATE is specified. The OUTVERID value defaults to "T" when the TESTQDS parameter is specified.

Setting Migration Utility Parameters

Depending on the task you want to perform with the migration utility, you must specify certain combinations of the parameters in Table 3-5 on page 3-58. This section provides some examples of the parameters you need to specify for the types of the migration supported by the utility. The examples are for a production data set to a production data set migration.

If you are migrating from a STROBE Version 2 or 3 environment that is identical to the environment for STROBE Release 3.0, you do not have to specify any additional parameters to the ones described in "Migrating Queue Data Sets" on page 3-53 and if you are using the STROBE Advanced Session Management Feature, the ones described in "Migrating Group Data Sets (Advanced Session Management Only)" on page 3-54.

CAUTION:

The examples within this section may not describe *all* parameters required for the exact type of data set migration you are performing. For example, the INVERID and OUTVERID parameters only need to be specified in certain cases. If you are migrating from an old production version to a new production environment and you have not changed the default version id value of "B" for either version, you do not have to specify the version ID. If you want to install STROBE and assign it a version id of "X", then you need to specify "X" as the value for OUTVERID. Make sure you specify every parameter that is relevant for the environment you are migrating *from* and *to*.

Local to XCF Environment STROBE Migration

If your previous release of STROBE ran on a single local MVS system and you are installing STROBE Release 3.0 in a multi-system environment, you need to

- Migrate the queue data set to the system that will be used as the STROBE single point of control. (In a multisystem environment you can use only one queue data set.)
- If using the STROBE Advanced Session Management Feature, migrate your group data set elements to a single group data set on the same system as the queue data set so all instances of STROBE within the sysplex can use the same groups.

You do not have to specify the INVERID because it has the correct default value of "B",

```
//MIGRATE EXEC PGM=STRBCMIG,REGION=4M,COND(4,LT),
// PARM='INSMFID=SYS1,OUTXCFGROUP=GROUP1'
```

To migrate from an XCF environment to a local installation, use the INXCFGROUP and OUTSMFID parameters. INVERID is not required because STROBE does not use a version ID with XCF group names.

Migrating to STROBE Release 3.0 with Advanced Session Management

If your copy of STROBE has the STROBE Advanced Session Management Feature, make sure the Release 3.0 group data set is re-allocated and empty before running the migration utility.

Copying Measurement Requests

You can use the migration utility to copy STROBE Release 3.0 measurement requests that ran in one MVS environment so STROBE can use them on another MVS system. The following example copies completed measurement requests from a system with SMFID "SYS1" to a system with a "SYS5" SMFID. The default for the INREL parameter with COPY is 0300.

```
//MIGRATE EXEC PGM=STRBCMIG,REGION=4M,COND(4,LT),
// PARM='COPY,INSMFID=SYS1,OUTSMFID=SYS5'
```

Listing Migration Utility Information

You can produce a screen display listing of the activity generated by the migration utility. The listing shows the parameters you specified for the utility and information about the data sets you are migrating and their contents. The listing can be produced for all types of STROBE migrations with the LIST parameter designated as shown in the following example:

```
//MIGRATE EXEC PGM=STRBCMIG,REGION=4M,COND(0,NE),
// PARM='LIST,INSMFID=SYS1,OUTSMFID=SYS5'
```

To use the LIST parameter, you must create a data set to contain the report information. And you do not have to allocate the temporary files for the group and queue data sets.

Chapter 4.

Customizing STROBE

This chapter discusses

- setting up security using RACF, CA-Top Secret, or CA-ACF2
- writing user-exit programs
- modifying STROBE procedures
- customizing module descriptors
- loading data collectors
- editing the CICS System Task Table

Setting Up Security

STROBE has an access filter you can enable that interfaces with RACF, CA-Top Secret, or CA-ACF2 to validate the issuer's authority to perform certain functions. The access filter works in conjunction with one of these security products and optionally, a user-exit routine, to determine the types of measurement requests that STROBE accepts.

The default environment for STROBE is for the access filter to be disabled. To enable the access filter, see "Customize the Parameter Data Set" on page 3-12.

How Access Checking Works

When the access filter is disabled, no attempt is made to find a security package and check the access privileges for the TSO user ID issuing the STROBE command. When the access filter is enabled through the parameter data set, STROBE checks the TSO user ID of issuing the command. This check determines whether the user has the authority to:

- Measure or monitor the requested job.
- Issue STROBE administrator commands.
- Start the STROBE session manager as a batch job.

If STROBE determines that any of the following is true for the TSO user ID, the access filter allows the command to be executed without checking the privileges defined through the security package:

- STROBE previously determined the TSO user ID has administrator authority.
- The user's TSO ID matches the string of characters beginning the job name to be measured. For example, if your TSO ID is USER001, you can measure a job named USER001T.
- For commands such as LIST and CHGGRP, the TSO ID of the user making the command issued the original ADD or ADDGROUP command.

If STROBE finds the TSO ID does not meet any of these conditions, it calls the security package at your site. To validate ADD or ADDGROUP commands, STROBE calls security from the measured address space (as opposed to the session manager or session requester address space), just before the measurement is to be initialized. To validate other types of requests such as LIST or CHGGRP, STROBE calls security from the TSO session or job that issues the command.

When a resource is not defined to the security product, the filter

- permits access when the security product is RACF or CA-Top Secret
- denies access when the security product is CA-ACF2. For more information, see the *CA-ACF2 Administrator Guide*.

Enabling and Disabling the Access Filter

STROBE displays the value of the FILTER parameter when you start the session manager. You can also use the SHOW DEFAULT command to display the status of the access filter. Control the filter as follows:

- The access filter by default is disabled.
- To enable the access filter and indicate STROBE should check access rules defined for CA-ACF2, RACF or CA-Top Secret, set the FILTER parameter value to "FILTER= ".

Defining Access Authority

Security products require a set of rules to determine who has access to particular resources. These rules comprise what is called the user profile. Security implementation under RACF, CA-Top Secret, or CA-ACF2 requires rules that define which privileges are granted to the user IDs that will make STROBE measurement requests. The privileges control

- the user IDs that are allowed to make measurement requests
- the names of jobs that can be measured
- the types of jobs that can be measured

You can also grant user IDs special privileges to enable the following capabilities

- Granting MANAGER privileges to a user ID allows the ID to start STROBE from a batch job.
- Granting ADMIN privileges to a user ID allows the ID to perform functions such as maintaining STROBE data sets, viewing status information and testing user-exit programs.

Define User Profiles to Control What Can be Measured

To implement the STROBE access filter, you establish a set of generic rules that STROBE uses to validate requests. The syntax for the rules includes

- A high-level qualifier that identifies the rule as belonging to STROBE. The default is \$STROBE, but you can supply another value in the parameter data set using the PROFILE parameter. (See page 3-17 for information about the PROFILE parameter.)
- A system identifier equal to the one-to eight-character MVS system name of the target system for the request.
- An identifier that defines the target address space type for the request. The values are:
 - 'job' for batch jobs
 - 'stc' for started tasks
 - 'tsu' for TSO users
 - 'appc' for advanced program-to-program communication
 - 'omvs' for OpenEdition
- A one- to eight-byte jobname mask or an asterisk (*) that restricts access to a jobname or set of jobnames. The value depends on the security system at your site.

Define Privileged Access

For privileged access, STROBE requires that the rules have a keyword of MANAGER or ADMIN.

- MANAGER is \$STROBE.MANAGER
- ADMIN is \$STROBE.ADMIN

MANAGER restricts who can start STROBE from a batch job. ADMIN controls access to all the other system administrator functions that STROBE requires.

Defining RACF Access Rules

RACF has the following requirements:

- User profiles must be in a general resource class called FACILITY.
- The name profile must start with a dollar sign (\$).

To define access rules for RACF, take the following steps:

Step 1. Initialize the resource facility with three SETROPTS commands.

SETROPTS CLASSACT(FACILITY)	Activates the FACILITY resource class
SETROPTS GENERIC(FACILITY)	Includes generic profiles in FACILITY class
SETROPTS GENLIST(FACILITY)	Provides a performance boost for generic profiles

Step 2. Use the RDEFINE statement to establish a generic profile of the types of jobs that can be measured. Repeat the following statement for the five job types: job, stc, tsu, appc, omvs.

```
RDEFINE FACILITY $STROBE.sysname.jobtyp.* UACC(NONE)
```

The high level qualifier of \$STROBE must match the value in the PROFILE keyword of your parameter data set. Refer to Table 4-1 for a definition of RDEFINE variables.

Step 3. Use the PERMIT command to selectively grant users access to specific jobs. Establish read access for a user or group, repeating for every user or group that requires access to the jobs. Refer to Table 4-1 for a definition of PERMIT variables.

```
PERMIT $STROBE.sysname.jobtyp.jobname CLASS(FACILITY)ID(userid|grp) ACCESS(READ)
```

Step 4. Use the RDEFINE command to establish privileged access for MANAGER and ADMIN functions.

- Specify a \$STROBE.MANAGER profile to control batch starts of the STROBE address space.

```
RDEFINE FACILITY $STROBE.MANAGER UACC(NONE)
```

- Specify a \$STROBE.ADMIN profile to control access to STROBE administrative functions.

```
RDEFINE FACILITY $STROBE.ADMIN UACC(NONE)
```

Step 5. Use the PERMIT command to add specific users or groups to the access list for MANAGER or ADMIN authority, repeating for every user or group that requires access.

```
PERMIT $STROBE.authtype CLASS(FACILITY) ID(userid|grp) ACCESS(READ)
```

Table 4-1. RDEFINE and PERMIT Keywords

Keyword	Function
<i>sysname</i>	Provide the MVS system name, as specified in the IEASYSxx member of SYS1.PARMLIB.
<i>jobtyp</i>	Name the type of target address space (job for batch job, stc for started tasks, tsu for TSO user IDs, appc for advanced program-to-program communication, omvs for OpenEdition).
<i>jobname</i>	Job name or set of job names with a common prefix of targets to which RACF allows access. Identify names with a common prefix with an asterisk (*) after the prefix. For example: - MEASURE1 specifies access only to the job name MEASURE1. - M* specifies access to all jobs whose jobnames begin with M.
<i>userid</i>	Provide the user ID of the submitter of the request.
<i>grp</i>	Specify the RACF group to which the user ID of the request submitter is defined.
<i>authtype</i>	Specify the STROBE authorization type (ADMIN or MANAGER) for access to STROBE administrative functions.

Initializing the Examples

Each of the RACF examples assumes that the following commands have been issued once at the beginning of the access definition:

```
SETROPTS CLASSACT(FACILITY)
SETROPTS GENERIC(FACILITY)
SETROPTS GENLIST(FACILITY)
```

Example 1: Access to a Specific System

The following commands give users belonging to the group STROBERS access to all address spaces running on system SYSA:

```
RDEFINE FACILITY $STROBE.SYSA.* UACC(NONE)
PERMIT $STROBE.SYSA.* CLASS(FACILITY) ID(STROBERS) ACCESS(READ)
```

Example 2: Access to a Set of Jobs

The following commands give the user IDs SUSAN and JERRY access to all batch jobs on system SYSB whose jobnames begin with RED.

```
RDEFINE FACILITY $STROBE.SYSB.JOB.RED* UACC(NONE)
PERMIT $STROBE.SYSB.JOB.RED* CLASS(FACILITY) ID(SUSAN) ACCESS(READ)
PERMIT $STROBE.SYSB.JOB.RED* CLASS(FACILITY) ID(JERRY) ACCESS(READ)
```

Example 3: Privileged Access to a Session Manager

The following commands authorize the user ID SMITH to start STROBE from a batch job and the user ID JONES to administer STROBE.

```
RDEFINE FACILITY $STROBE.MANAGER UACC(NONE)
RDEFINE FACILITY $STROBE.ADMIN UACC(NONE)
PERMIT $STROBE.MANAGER CLASS(FACILITY) ID(SMITH) ACCESS(READ)
PERMIT $STROBE.ADMIN CLASS(FACILITY) ID(JONES) ACCESS(READ)
```

Defining CA-Top Secret Access Rules

CA-Top Secret resembles RACF in structure and requires you to build similar definitions. However, you also define a user resource class that contains STROBE profiles.

The following procedure defines access rules for CA-Top Secret. Use the values in Table 4-2 for the ADD and PERMIT commands. Issue the commands from the command line during a TSO session.

Step 1. Define a resource class in the resource definition table (RDT); the RESCODE is in hexadecimal notation.

```
TSS ADD(RDT) RESCLASS(STROBE) RESCODE(xx) ATT(DEFPROT, LONG)
```

Step 2. Specify the types of access allowed.

```
ACLST(NONE, READ, UPDATE)
```

Step 3. Assign ownership of the STROBE profile. The high level qualifier of \$STROBE must match the value in the PROFILE keyword of your parameter data set.

```
TSS ADD(ownerid) STROBE($STROBE.)
```

Step 4. Specify user access to target jobs of identified types on identified systems; repeat for each user that needs access.

```
TSS PERMIT(userid|department|division)
STROBE($STROBE.sysid.jobtype.jobnamemask) ACCESS(READ)
```

Step 5. Specify user access to MANAGER and ADMIN profiles; repeat for each user who requires access to STROBE administrative functions.

```
TSS PERMIT (userid) STROBE($STROBE.authtype) ACCESS(READ)
```

Table 4-2. TSS ADD and TSS PERMIT Keywords

Keyword	Function
<i>sysid</i>	Provide the MVS system name as specified in the IEASYSxx member of SYS1.PARMLIB.
<i>jobtyp</i>	Specify the type of target address space (job for batch job, stc for started tasks, tsu for TSO user IDs, appc for advanced program-to-program communication, omvs for OpenEdition).
<i>jobnamemask</i>	Identify the job name or set of job names with a common prefix of targets to which CA-Top Secret allows access. Identify names with a common prefix with an asterisk (*) after the prefix. For example: - MEASURE1 specifies access only to the job named MEASURE1 - M* specifies access to all jobnames that begin with M.
<i>userid</i>	Specify the user ID of the submitter of the request.
<i>grp</i>	Identify the group to which the user ID of the request submitter is defined.
<i>authtype</i>	Specify the STROBE authorization type (ADMIN or MANAGER) for access to management functions.

Initializing the Examples

The CA-Top Secret examples assume that the following commands have been issued to define a resource class with the hex identifier 4A:

```
TSS ADD(RDT) RESCLASS(STROBE) RESCODE(4A) ATT(DEFPROT, LONG)
ACLIST(NONE, READ, UPDATE)
```

Example 1: Access to a Specific System

The following command defines and allows users belonging to the department (or division) STROBERS access to all address spaces running on system SYSA:

```
TSS PERMIT(STROBERS) STROBE($STROBE.SYSA.) ACCESS(READ)
```

Example 2: Access to a Set of Jobs

The following commands give the user IDs WOLFE and DAVIS access to all batch jobs on system SYSB with jobnames beginning with RED.

```
TSS PERMIT(WOLFE) STROBE($STROBE.SYSB.JOB.RED*) ACCESS(READ)
TSS PERMIT(DAVIS) STROBE($STROBE.SYSB.JOB.RED*) ACCESS(READ)
```

Example 3: Privileged Access to Session Manager

The following commands authorize the user ID TIBBETTS to start STROBE from a batch job and the user ID JCDALY to maintain STROBE.

```
TSS PERMIT(TIBBETTS) STROBE($STROBE.MANAGER) ACCESS(READ)
TSS PERMIT(JCDALY) STROBE($STROBE.ADMIN) ACCESS(READ)
```

Defining CA-ACF2 Access Rules

Whenever the access filter is enabled, STROBE passes the resource rules to CA-ACF2. Therefore, if you have enabled the access filter, you must define rules to CA-ACF2. If you have not defined the rules, CA-ACF2 allows STROBE users to measure only those jobs that begin with their user ID.

Two different CA-ACF2 Releases can have common syntax for defining the resource rules, but different procedures for defining resource types. For additional information, refer to the *CA-ACF2 Administrator Guide*.

Note: You must always define the STROBE session manager to CA-ACF2 as a Multi User Single Address Space System (MUSASS). If AutoSTROBE is enabled, you must define started task STRBMSASas an ACF2 logon id.

Defining a Resource Type for Release 6.2 and 6.3

Take the following steps to define a resource type if you have CA-ACF2 Release 6.2 or 6.3

Step 1. Add the following CLASMAP control record:

```
SET CONTROL(GSO)
INSERT CLASMAP RSRCTYPE(STR) RESOURCE(STROBE)
INSERT CLASMAP RSRCTYPE(STR) RESOURCE(STROBE)
```

Step 2. Add the following INFODIR record:

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(D-RSTR)
```

Step 3. Issue the following command to refresh modified GSO records:

```
F ACF2,REFRESH(CLASMMAP,INFODIR)
```

Step 4. Issue the following command to rebuild the resident directory:

```
F ACF2,REBUILD(STR),CLASS(R)
```

Defining Resource Rules

The following commands define the resource rules for access to STROBE. Use the values in Table 4-3 for command variables. Enter the commands on the command line in a TSO session.

Step 1. Set the resource type to STR with the following command:

```
SET RESOURCE(STR)
```

Step 2. Use the \$KEY command to specify user access to target jobs of identified types on identified systems. Repeat this step until all STROBE users have access to the appropriate jobs.

```
$KEY($STROBE.sysid.jobtype.jobnamemask) UID(userid|grp) TYPE(STR)
SERVICE(READ)
```

Step 3. Use the \$KEY command to specify MANAGER and ADMIN profiles. Repeat this step for all user who require access to STROBE administrative functions.

```
$KEY($STROBE.authtype) UID(userid|grp) TYPE(STR) SERVICE(READ)
```

Table 4-3. \$KEY Keywords

Keyword	Function
<i>sysid</i>	Identify the MVS system name as specified in the IEASYSxx member of SYS1.PARMLIB.
<i>jobtyp</i>	Indicate the type of target address space (job for batch job; stc for started tasks; tsu for TSO user IDs; appc for advanced program-to-program communication; omvs for OpenEdition).
<i>jobnamemask</i>	Identify the name (or set of names with a common prefix) of targets to which CA-ACF2 allows access. Use an asterisk (*) after the prefix to identify names with a common prefix. For example: - MEASURE1 specifies access to the job with this name. - M* specifies access to all jobs whose jobnames begin with M.
<i>userid</i>	Specify the user ID of the submitter of the request.
<i>grp</i>	Identify the group to which the user ID of the request submitter is defined.
<i>authtype</i>	Indicate the authorization type (ADMIN or MANAGER) for access to STROBE administrative functions.

Initializing the Examples

The following CA-ACF2 examples assume that you have defined a resource type of STR with a SUBSYS name of STROBE, as detailed above.

Example 1: Access to a Specific System

The following commands give access to all address spaces running on system SYSA to users with IDs that begin with "STR":

```
SET RESOURCE(STR)
$KEY($STROBE.SYSA.*) UID(STR-----) TYPE(STR) SERVICE(READ)
```

Example 2: Access to a Set of Jobs

The following commands give the user IDs JMARTIN and ANDERSON access to all batch jobs on system SYSB whose jobnames begin with RED.

```
$KEY($STROBE.SYSB.JOB.RED*****) UID(JMARTIN) TYPE(STR)SERVICE(READ)
$KEY($STROBE.SYSB.JOB.RED*****) UID(ANDERSON) TYPE(STR)SERVICE(READ)
```

Example 3: Privileged Access to Session Manager

The following commands authorize the user ID SANDY to start STROBE from a batch job and the user ID COUGHLIN to maintain STROBE.

```
$KEY($STROBE.MANAGER) UID(SANDY) TYPE(STR) SERVICE(READ)
$KEY($STROBE.ADMIN) UID(COUGHLIN) TYPE(STR) SERVICE(READ)
```

Implementing Security in a Multisystem Environment

In a multisystem environment, STROBE allows a user to submit a request from one system that will run on or relates to measurements on another system. For example:

- A user can issue a request from system A to measure a job that is currently running on system B.
- A user can issue a request from system A to target a measurement session for several systems where the designated job may subsequently run.
- A user can issue a request, such as CHANGE or LIST, from system A that is directed to a measurement session that is running on another system or is targeted to multiple systems.

Security Database Recommendation

STROBE access filter processing in a multisystem environment includes authorization checks against the security database of the system on which the request is entered and the system on which the request runs. If the authorization check fails on the system where the request is entered, STROBE issues a warning message and allows the request to proceed. If the authorization check fails on the system where the request is supposed to run, the request is rejected.

It is recommended that all systems with members in a STROBE-XCF group have the same security database. Similarly, if you disable the access filter on one system, you should disable it on all systems in the STROBE XCF-group.

Restricting Access on Particular Systems

You can prevent all STROBE activity (other than the ability of users to measure their own jobs) on specific systems with a RACF profile.

```
RDEFINE FACILITY $STROBE.sysname.* UACC(NONE)
```

You can also restrict certain users or groups of users from specific systems.

Implementing Security for Automatic Profile Creation

To use the automatic Performance Profile option, ensure that the job statement information at the bottom of the STROBE - CREATE AN AUTO PERFORMANCE PROFILE panel is valid, and that you have specified the USER parameter. To prevent exposing user logon passwords, the PASSWORD keyword is not allowed on the job statement.

Also, ensure that the user ID associated with the STROBE session manager is granted access by any security product you are using to submit jobs on behalf of any STROBE user.

- When STROBE runs as a started task, the user ID of the started task needs authority to submit jobs on behalf of STROBE users.
- When STROBE is started with a batch job submitted by a user with MANAGER authority, the user ID of the submitter needs authority to submit jobs on behalf of STROBE users.

For example, in RACF, define the user ID associated with the STROBE session manager as a surrogate user for any STROBE user who intends to automatically generate Performance Profiles. For information about surrogate job submission and controlling job submission on MVS, see the *RACF Security Administrator's Guide*.

If you are using CA-ACF2 or CA-Top Secret, see either the *CA-ACF2 System Programmer's Guide* or the *CA-Top Secret Batch and STC Guide* for information about enabling a job to submit another job with a different user's authority.

Providing User-Exit Programs

If you have a specific requirement that the access filter does not accommodate, you can create a STROBE user-exit program to perform functions, such as:

- determining whether a user is authorized to measure certain jobs
- validating installation-dependent keyword values for a measurement request
- enforcing installation standards
- supplying missing keyword values

The user-exit program enables you to control and modify commands issued to STROBE as well as the controls provided by the STROBE access filter. The exit program can enforce exceptions to the more general access permitted by the access filter and the security product.

STROBE Processing

STROBE uses request values in the following order:

- values supplied by the user-exit program, or, if not available,
- values entered by the user, or, if not available,
- default values from the STROBE parameter data set.

Processing ADD and ADDGRP Commands

STROBE processes ADD and ADDGRP commands as follows:

1. STROBE calls the user-exit program from the TSO or session requester address space to check the request. If necessary, the user-exit program makes modifications.
2. STROBE validates the modification made by the user-exit program. If incorrect, STROBE rejects the command and displays an error message. If correct, the request gets added to the measurement queue.

3. Immediately before the measurement becomes active, STROBE takes the following actions from the measured address space:
 - Calls the access filter (if enabled) to verify that the requester has the correct access authority if the access filter is enabled.
 - Calls the user-exit program. The program checks the current environment and status and decides whether to let the measurement request continue.

Processing CHANGE and CHGGRP Commands

STROBE processes CHANGE and CHGGRP commands as follows:

1. STROBE calls the access filter (if enabled) from the TSO or session requester address space to check that the requester has the correct access authority.
2. STROBE calls the user-exit program to check the request. If necessary, the user-exit program makes modifications.
3. STROBE validates the modification made by the user-exit program. If incorrect, STROBE rejects the command and displays an error message. If correct, the request gets added to the measurement queue.
4. If this is a change for an active request, STROBE calls the access filter from the measured address space to verify that the requester has the correct access authority.

Processing DELETE or DELGRP Commands

STROBE processes DELETE and DELGRP commands as follows:

1. STROBE calls the access filter (if enabled) from the TSO or session requester address space to check that the requester has the correct access authority.
2. STROBE calls the user-exit program to check the request. If necessary, the user-exit program makes modifications.
3. STROBE validates the modification made by the user-exit program. If incorrect, STROBE rejects the command and displays an error message. If correct, the request is deleted.

Processing LIST and LISTGRP Commands

For LIST ALL and LISTGRP, STROBE calls the access filter (if enabled) from the TSO or session requester address space to check that the requester has the correct access authority. (There is no call to the user-exit program.)

For LIST by JOB, TYPE, or USER

1. STROBE calls the access filter (if enabled) from the TSO or session requester address space to check that the requester has the correct access authority.
2. STROBE calls the user-exit program to check the request. If necessary, the user-exit program makes modifications.
3. STROBE validates the modification made by the user-exit program. If incorrect, STROBE rejects the command and displays an error message. If correct, the request gets added to the measurement queue.
4. If this is a change for an active request, STROBE calls the access filter (if enabled) from the measured address space to verify that the requester has the correct access authority.

Processing SEND Commands

STROBE processes SEND commands as follows:

1. STROBE calls the access filter (if enabled) from the TSO or session requester address space to check that the requester has the correct access authority.
2. STROBE calls the user-exit program from the TSO or session requester address space to check the request. If necessary, the user-exit program makes modifications.
3. STROBE validates the modification made by the user-exit program. If incorrect, STROBE rejects the command and displays an error message. If correct, the request gets added to the measurement queue.
4. Immediately before the measurement becomes active, STROBE calls the access filter (if enabled) from the measured address space to verify that the requester has the correct access authority.

User Exits in a Multisystem Environment

User-exit routines should be the same on all systems in the STROBE-XCF group. If the user-exit routines are not the same on the originating system and on the target system, and both are called, the changes made by the exit routine on the target system may override the changes made by the exit routine on the originating system.

For an ADD or ADDGROUP request, if the user-exit routine on the originating system cancels the request, then the exit routine on the target system is not called.

The PRMUSER exit routine parameter allows the first invocation of the user exit to pass data to subsequent invocations of the exit for a given measurement request. In a multisystem environment, these two invocations might be on different systems. Therefore, the PRMUSER field should not pass any system-dependent data, such as an address.

Implementing a User-Exit Program

To implement a STROBE user-exit program:

1. Write your own program or modify one of the STROBE sample members to suit your installation requirements. The control data set includes two members that supply source for different types of user-exit programs.
 - SMEXIT, which requires an interface to one of the security products
 - SMEXIT2, which does not require an interface to a security product
 Appendix A displays a sample user-exit program based on SMEXIT2.
2. Assemble and link edit your user-exit program, naming the load module **STRBCXAC**.

Note: The STROBE program STRBCSR loads and calls user-exit program STRBCXAC. The distribution tape contains a dummy module (BR14) that allows all security functions to be handled by the access filter (if enabled) and the installation security product. The user-exit program replaces this module.

3. Copy the module to your STROBE authorized load library.

Writing a User-Exit Program

STROBE allows the user-exit program to modify certain request-specific fields. With this capability, your installation can override parameter values or supply missing parameters, such as STORCLAS, as requests are processed.

Parameters That Cannot Change

The following table lists parameters that provide required information to the user-exit program. They cannot be modified.

Table 4-4. Parameters the User-exit Program Cannot Modify

Parameter	Function
PRMJOBBSR	The address of an eight-byte character field containing the jobname or TSO user ID that submitted the current command. This field is blank during initialization (when PRMCMD=I).
PRMJOB	The address of an eight-byte character field containing the target jobname specified in the request.
PRMCALL	The address of a one-byte character field containing the calling mode of the user, denoting whether the request is being made from batch, TSO, or a started task. CL1'J' BATCH CL1'T' TSO CL1'S' STARTED TASK STROBE supplies a value for this field during initialization (when PRMCMD=I).
PRMCMD	The address of a one-byte character field containing the command being processed by STROBE upon this call to the exit program. Because some fields may be blank during initialization, check this field before manipulating any of the others. CL1'A' ADD CL1'C' CHANGE CL1'S' SEND CL1'L' LIST CL1'D' DELETE CL1'I' Measurement task initialization
PRMACEID	The address of an eight-byte character field containing the RACF, CA-ACF2, or CA-Top Secret user ID associated with the TSO/batch job that issued the command. STROBE retrieves this field from the ACEE control block when the command is issued. This field may be blank if a system task added the request.
PRMREQ	The address of a one-byte character field containing the type of request. CL1'Q' QUEUED REQUEST CL1'A' ACTIVE REQUEST
PRMSTEP	The address of the list containing the stepnames to be measured, The first eight bytes of this list is a header in the form - number of entries (two bytes) - length of an entry (two bytes) - a total length of table (four bytes) The header is immediately followed by the first entry. Each entry is a 16 byte stepname. STROBE supplies a value for this field during initialization (when PRMCMD=I).
PRMSNUM	The address of the list containing the step numbers to be measured, The first eight bytes of this list is a header in the form - number of entries (two bytes) - length of an entry (two bytes) - a total length of table (four bytes) The header is immediately followed by the first entry. Each entry is a 4 byte step number. STROBE supplies a value for this field during initialization (when PRMCMD=I).

Table 4-4. Parameters the User-exit Program Cannot Modify

Parameter	Function
PRMPROG	<p>The address of the list containing the program names to be measured, The first eight bytes of this list is a header in the form</p> <ul style="list-style-type: none"> - number of entries (two bytes) - length of an entry (two bytes) - a total length of table (four bytes) <p>The header is immediately followed by the first entry. Each entry is a 8 byte program name. STROBE supplies a value for this field during initialization (when PRMCMD=I).</p>
PRMSYSL	The address of a list of systems as specified in the SYSTEMS parameter of the request.
PRMEXCL	The address of a list of systems as specified in the EXCLUDE parameter of the request.
PRMSTDTE	<p>The address of a table containing the dates on which the scheduled request will be initiated in unsigned, packed form CYYDDDS where</p> <ul style="list-style-type: none"> - C is century (0 for current or 1 for next) - YY is year - DDD Julian day - S is sign. <p>The table has an eight-byte header in the following format:</p> <ul style="list-style-type: none"> - a two-byte field for the number of entries in the list - a two-byte field containing the size of each entry - a four-byte field for the size of the table, including the header
PRMSTTIM	<p>The address of a table containing the start times at which the scheduled request will be initiated in unsigned, packed format HHMMSSHTS where</p> <ul style="list-style-type: none"> - HH is hour - MM is minute - SS is second (always zero) - H is hundredth of a second (always zero) - T is thousandth of a second (always zero) - S is sign <p>The table has an eight-byte header in the following format:</p> <ul style="list-style-type: none"> - a two-byte field with the number of entries in the list - a two-byte field with the size of each entry - a four-byte field with the size of the table, including the header
PRMSTRTY	The address of a four-byte binary field that contains the number of retry attempts.
PRMSTINC	The address of a four-byte binary field that contains the retry interval.
PRMPJCL	<p>The address of the list of JCL as specified in the RJCLTEXT parameter of the request. The first eight bytes of this list is a header in the form</p> <ul style="list-style-type: none"> - number of entries (two bytes) - length of an entry (two bytes) - a total length of table (four bytes) <p>The header is immediately followed by the first entry. Each entry is an 80-character JCL card.</p>
PRMPRD	<p>The address of the list of JCL related data in the RJCLFILE parameter of the request. The first eight bytes is a header in the form</p> <ul style="list-style-type: none"> - number of entries (two bytes) - length of an entry (two bytes) - a total length of table (four bytes) <p>The header is immediately followed by the first entry. Each entry is 100 characters of data related to the creation of the JCL for automatic profile creation.</p>

Parameters That Can Change

The following table lists the parameters that can be modified by the user-exit program.

Table 4-5. Parameters the User-exit Program Can Modify

Parameter	Function
PRMDSN	The address of a 44-byte character field containing the sample data set prefix specified on the ADD or CHANGE operation, or blanks if no data set name was specified. The prefix can contain as many as (34-joblen) bytes, where joblen is the length of the jobname. When modifying this field, the user-exit program should clear it to spaces first, and pass a fully-qualified sample data set prefix padded with blanks.
PRMUNIT	The address of an eight-byte character field containing the unit name of the sample data set or blanks. When modifying this field, the user-exit program should left-justify it and pad it with blanks.
PRMVOL	The address of a six-byte character field containing the volume serial number of the sample data set or blanks. When modifying this field, the user-exit program should left-justify it and pad it with blanks.
PRMUSER	The address of an eight-byte user field, initially set to binary zeros, that your exit program can modify on the first call for the measurement request. STROBE returns it as modified on all subsequent calls. You can store the address of a data area in this field. Make sure that the referenced data area is in commonly addressable storage, because subsequent calls will come from different address spaces. After using it, you can free the storage for the data area during the measurement task initialization call.
PRMMSG	The address of a fullword containing the address of a message for the user. Begin the message with a two-byte length field, followed by as many as 120 bytes of text. STROBE truncates messages greater than 120 bytes on the right. The severity code associated with the message depends on the return code that the exit issues when it returns to STROBE.
PRMSAMP	The address of a four-byte binary field containing the sample count supplied by the user. If the user did not supply a value, STROBE supplies the default. The user exit must ensure the integrity of its modifications to this field. The exit interface enforces limits of 1000 to 150,000.
PRMLIMIT	The address of a four-byte binary field containing the limit count supplied by the user. If the user did not supply a value, STROBE supplies the default. The user exit must ensure the integrity of its modifications to this field because the exit interface does not. The exit interface enforces limits of 1 to 99.
PRMLTYP	The address of a one-byte character field containing the type of action STROBE is to take when it reaches the LIMIT count. STROBE supplies a value of 'C' if an attempt to modify this field results in a value other than Q, S, or C. CL1'Q' QUIT CL1'S' STOP CL1'C' NOLIMIT OR CONTINUE
PRMGOMIN	The address of a four-byte binary field containing the GOMIN value supplied by the user. If the user did not supply a value, STROBE supplies the default. The user exit must ensure the integrity of its modifications to this field. The exit interface enforces limits of 1 to 1440.
PRMRETNQ	The address of a four-byte binary field containing the number of days (a day is 24 hours from the time the request was added) STROBE keeps a queued, non-completed request before removing it from the queue. The exit interface enforces limits of 0 to 999.
PRMRETNC	The address of a four-byte binary field containing the number of days (a day is 24 hours from the time the request was added) STROBE keeps a completed request after completion before removing it from the queue. The exit interface enforces limits of 0 to 999.
PRMSUFF	The address of an eight-byte character field that contains either the sample data set name suffix as specified in the parameter data set or blanks if no data set name was specified.

Table 4-5. Parameters the User-exit Program Can Modify

Parameter	Function
PRMSCLAS	The address of an eight-byte character field containing the STORCLAS parameter as submitted on the request or blanks. When it modifies this field, the user-exit program should left-justify it and pad it with blanks.
PRMDCLAS	The address of an eight-byte character field containing the DATACLAS parameter as submitted on the request or blanks. When it modifies this field, the user-exit program should left-justify it and pad it with blanks.
PRMMCLAS	The address of an eight-byte character field containing the MGMTCLAS parameter as submitted on the request or blanks. When it modifies this field, the user-exit program should left-justify it and pad it with blanks.
PRMSTRDD	The address of an eight-byte character field containing the STRBDD parameter for module mapping libraries as submitted on the request, or blanks. When it modifies this field, the user-exit program should left-justify it and pad it with blanks.

Return Codes

On return to STROBE, the exit should set a return code in register 15.

Return Code	Significance
0	STROBE continues processing the command
Any other	STROBE rejects the command

Return codes and their associated error message severity codes are as follows:

Return Code	Severity Code
0	I (Information)
1-4	W (Warning)
5-8	E (Error) or C (Conditional error)
9+	S (Severe Error)

Figure 4-1 on page 4-16 shows list of parameters that are included with the SMEXIT and SMEXIT2 samples and how they might appear in a user-exit program.

Figure 4-1. STROBE User-exit Program Parameters

PRMCXAC	DSECT		STROBE EXIT PARMLIST DSECT
PRMJOBSR	DS	A	A(JOBNAME OF CALLER)
PRMJOB	DS	A	A(JOBNAME OF REQUEST)
PRMCALL	DS	A	A(MODE OF CALLER)
PRMBATCH	EQU	C'J'	BATCH JOB
PRMTSO	EQU	C'T'	TSO
PRMSTASK	EQU	C'S'	STARTED TASK
PRMCMD	DS	A	A(COMMAND)
PRMADD	EQU	C'A'	ADD COMMAND
PRMCHG	EQU	C'C'	CHANGE COMMAND
PRMSEND	EQU	C'S'	SEND COMMAND
PRMDEL	EQU	C'D'	DELETE COMMAND
PRMLIST	EQU	C'L'	LIST COMMAND
PRMMTI	EQU	C'I'	MEASUREMENT TASK INITIALIZATION
PRMDSN	DS	A	A(DATA SET NAME PREFIX)
PRMUNIT	DS	A	A(UNIT NAME)
PRMVOL	DS	A	A(VOLUME SERIAL NUMBER)
PRMUSER	DS	A	A(USER FIELD)
PRMMSG	DS	A	A(A(MESSAGE))
PRMACEID	DS	A	A(OWNERID)
PRMREQ	DS	A	A(TYPE OF REQUEST)
PRMQUED	EQU	C'Q'	ADD QUEUED
PRMACT	EQU	C'A'	ADD ACTIVE
PRMSTEP	DS	A	A(STEP NAME LIST)
PRMSNUM	DS	A	A(STEP NUMBER LIST)
PRMPROG	DS	A	A(PROGRAM NAME LIST)
PRMSAMP	DS	A	A(SAMPLE COUNT)
PRMLIMIT	DS	A	A(LIMIT)
PRMLTYP	DS	A	A(LIMIT ACTION)
PRMQUIT	EQU	C'Q'	QUIT
PRMSTOP	EQU	C'S'	STOP
PRMCONT	EQU	C'C'	NOLIMIT OR CONTINUE
PRMGOMIN	DS	A	A(GOMIN)
PRMRETNQ	DS	A	A(RETAINQ)
PRMRETNC	DS	A	A(RETAINC)
PRMSUFF	DS	A	A(DSN SUFFIX)
PRMSCLAS	DS	A	A(STORCLAS)
PRMMCLAS	DS	A	A(MGMTCLAS)
PRMDCLAS	DS	A	A(DATACLAS)
PRMSTRDD	DS	A	A(STRBDD)
PRMPJCL	DS	A	A(JCL)
PTMPRD	DS	A	A(PROFILE RELATED DATA)
PRMLN	EQU	*-PRMCXAC	LENGTH OF STROBE EXIT PLIST

Linking a User-Exit Program

A user-exit program must be linked as reentrant. To link a user-exit program

1. Code the following statements:

```
INCLUDE myexit
NAME STRBCXAC(R)
```

where *myexit* is the name of the object module produced by your assembly of the user exit.

2. Copy the load module to your authorized STROBE library.

Testing a User-Exit Program

Before you make the user-exit program available to your installation, you should test it with a range of commands that exercise all the supported functions. To assist in your test, STROBE includes the following debugging command that can be issued by users with ADMIN authority:

```
TEST EXITON
```

This command displays to SYSPRINT and to the STROBE log a hex dump of the parameter list, both before and after the user-exit program is called. The dump contains a list of addresses followed by the data areas pointed to in the list.

Because this facility causes STROBE to display large amounts of data for all requests entered by all users, immediately issue the following command to cancel production of the dump for subsequent requests.

```
TEST EXITOFF
```

Modifying STROBE Procedures

Member PROCS of your STROBE control data set contains a library of STROBE procedures. Appendix C, "Symbolic Parameters for STROBE Job Control Procedures" provides detailed definitions of the symbolic parameters used in the procedures. Review the procedures to determine which ones are useful for your installation. You can modify the STROBE procedures and copy them to your system procedure library.

Functions of the Procedures

Note: Execute, as used in the following table, is a two-step process: 1) add a measurement request and 2) run the target program.

Procedure	Function
STROBEzz	Compile, index, link, execute, and report
STROzz	Compile and index
STROBE	Execute and report on target program
STROB	Execute
STROE	Report
STROX	Generalized indexer map procedure
STROXN	List a NATURAL 4GL program
STROXNAT	List and index a NATURAL 4GL program
STROXE	Generalized index and report procedure
STRBDSPY	Sample data set utility
zz stands for one of the following target program language identifiers: <ul style="list-style-type: none"> AF (assembler) CI (C/370) CS (SAS/C) CU (COBOL) C2 (COBOL II and AD/CYCLE COBOL/370) PF (PL/I) CX (CA-Optimizer COBOL post-processor) X2 (CA-OptimizerII COBOL post-processor) FG (FORTRAN G-level) FH (FORTRAN H-level and V5) 	

In the procedures STROBEzz, STROBE, and STROB, MEASURE steps add a measurement request through the session requester and require input data. GO steps cause execution of the target program.

Changing the Procedures

Make the following changes to the STROBE procedures (member PROC5):

- Change the “*/” preceding each IEBUPDTE control statement to “./”.
- Change the STROBLIB parameter to point to your STROBE load library.
- Change the PROCLIB parameter to point to your procedure library.

Submit the job.

In addition, review the unit and region parameters to verify that they are appropriate for your installation.

Procedures Invoking Language Processors

The procedures most likely to require further customization are those that invoke the language processors—STROBEzz and STROzz. You can modify these procedures to match your installation standards, or you can modify copies of your installation's existing procedures to incorporate the STROBE components. Do not suppress the compiler listing options (SOURCE, PMAP, and VERB in COBOL; LIST and ESD in assembler; and so on), because the STROBE indexing programs, which process the compiler SYSPRINT data sets, require both the source text listing and the object text listing.

Report Parameters

You may want to change the SORTSIZ and LINEMAX report parameters.

SORTSIZ controls the amount of main storage available to the system sort. SORTSIZ can be from 12000 to 999999. To allow the sort program to claim all available main storage, use 999999. To use your sort system default, specify a null value. The larger the value, the more rapidly the sort program operates, provided that there is sufficient real storage to preclude excessive paging.

If your installation has a sort product that uses a secondary space allocation, you can change data definitions for sort work data sets in the reporting facility. You may need to add a SORTLIB data definition statement.

LINEMAX specifies the number of lines printed on each page of the STROBE Performance Profile. This value must be between 45 and 80.

Customizing the Module Descriptor Program

The STROBE module descriptor program, STREMODS, provides function descriptors for MVS system and subsystem modules (modules whose names begin with the prefixes IGG, IDA, DFS, and so on). STREMODS is link-edited into the STROBE library and loaded by the STROBE reporting facility.

You can add module names and function descriptors for your own programs and subsystems and change descriptors for module names already in the STREMODS program.

Adding and Changing Module Descriptors

Member MODCSECT of your STROBE control data set, shown in Figure 4-2 on page 4-19, provides a sample control section that you can use to add and change module descriptors.

Add your module name to the eight-character field and its corresponding descriptors to the 24-character field. For example:

```
DC CL8'modname',CL24'module descriptor'
```

The module name field must be exactly eight characters long. The descriptor field must be exactly 24 characters long. The number of entries you can add to a single control section or the number of control sections that you can create is limited only by the size of your region.

Assemble your descriptor programs. Link edit each program into your unauthorized STROBE library, naming the programs STREMOD1, STREMOD2, and so on.

Figure 4-2. Module Descriptor Program (Member MODCSECT)

```
//STREMOD1 EXEC PGM=ASMA90,
//          PARM='OBJECT,NODECK',REGION=256K
//SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(800,400))
//SYSUT2 DD UNIT=SYSDA,SPACE=(1700,(800,400))
//SYSUT3 DD UNIT=SYSDA,SPACE=(1700,(800,400))
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=&&SYSGO,UNIT=SYSDA,DISP=(MOD,PASS),
//          SPACE=(3200,(80,40)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN DD *
STREMOD1 CSECT
DC CL8'      ',CL24'      '
END

/*
/**
//LKED EXEC PGM=IEWL,REGION=256K,COND=(0,NE),
//          PARM='XREF,LIST,LET,REUS'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSMOD DD DISP=SHR,
//          DSN=STROBLIB
//SYSLIN DD DISP=(SHR,PASS),DSN=&&SYSGO
//          DD *
NAME STREMOD1(R)
```

Link Editing a New Copy of STREMODS

After you have separately assembled and link edited your module descriptor programs, you can include them in the STREMODS load module currently in the STROBE library. Member MODLINK, shown in Figure 4-3 on page 4-20, provides linkage editor control statements to link edit the STREMODS module.

Figure 4-3. Link Edit the STREMODS Program (Member MODLINK)

```
//LKED EXEC PGM=IEWL,REGION=256K,COND=(0,NE),
// PARM='XREF,LIST,LET,REUS'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD DD DISP=SHR,
// DSN=STROBLIB
//SYSLIN DD *
ORDER STREMODS,STREMFRS
ORDER STREMBAS
ORDER STREMC
ORDER STREMCIC
ORDER STREMCOB
ORDER STREMCSP
ORDER STREMDB2
ORDER STERMEZB
ORDER STREMFRT
ORDER STREMGFU
ORDER STRE MIDM
ORDER STRE MIEF
ORDER STRE MIGW
ORDER STRE MIMS
ORDER STRE MJAV
ORDER STRE MLE
ORDER STRE MMQ
ORDER STRE MNAT
ORDER STRE MPLI
ORDER STRE MRSC
ORDER STRE MUXS
ORDER STRE MOD1 YOUR MODULE DESCRIPTOR CONTROL SECTION
ORDER STRE MLAS
INCLUDE SYSLMOD(STREMOD1) YOUR MODULE DESCRIPTOR CONTROL SECTION
INCLUDE SYSLMOD(STREMODS)
ENTRY STREMODS
NAME STREMODS(R)
```

This control section	References these module descriptors
STREMC	C/370 and SAS/C
STREMCIC	CICS
STREMCOB	COBOL
STREMCSP	CSP
STREMDB2	DB2
STERMEZB	TCP/IP System Services
STREMFRT	FORTRAN
STREMGFU	Unix System Services HFS Services
STRE MIDM	CA-IDMS
STRE MIEF	IEF
STRE MIGW	Unix System Services
STRE MIMS	IMS
STRE MJAV	Java
STRE MLE	Language environment for MVS

This control section	References these module descriptors
STREMMQ	MQSeries
STREMNAT	ADABAS/NATURAL
STREMPLI	PL/I
STREMRSC	CICS
STREMUXS	UNIX System Services

Add INCLUDE and ORDER statements for each program you want to include, following this sequence.

- STREMODS is the first.
- STREMFERS is the second.
- STREMLAS is the last.
- All other control sections come between STREMFERS and STREMLAS.

To override any of the STROBE-defined descriptors, include your module just after STREMFERS. To have your module searched after all STROBE-defined descriptors, add it just before STREMLAS.

Customizing the Loading of Data Collectors

When STROBE measures a target program, it determines whether it should invoke any of the available data collectors. STROBE searches link list for a module that has a prefix of STRB and a suffix that contains the first four characters of the target program name. The STROBE data collectors use the IBM- or vendor-supplied name for the subsystems that they support:

Subsystem	Data Collector Name	Alias
ADABAS	STRBNATL	STRBADAB, STRBADAR, STRBNATB
CICS	STRBDFHS	
DB2	STRBDSNY	
CA-IDMS	STRBIDMS	
IEF	STRBDIEF	
IMS	STRBDFSR	STRBDFSM
MQSERIES	STRBDCSQ	
NATURAL	STRBNATL	STRBADAB, STRBADAR, STRBNATB

If you use the STROBE DB2 data collector, STROBE invokes it for all measurement sessions. If the DB2 data collector determines that the target region does not support DB2, the data collector disables itself.

If you use nonstandard names to initialize any of these subsystems, link edit the data collector again, supplying the alias you are using; you cannot rename the data collector.

CICS System Task Table

If you are using the STROBE *STROBE CICS Feature* to measure CICS transactions, you can specify whether or not a transaction is to be detailed. STROBE uses a table called the system task table to determine which transactions are initiated by the CICS system instead of user applications. If a transaction is listed in the table, STROBE does not provide attribution for it or profile it.

At times you may want STROBE to measure a system transaction. If a system transaction is used to broadcast messages to CICS users' terminals, you can remove the name of the transaction from the table. Or you can add transactions to the table. If you are using a third party product and its transactions are appearing in the Performance Profile reports, you can add the transaction name(s) to the table and STROBE will not attribute or profile them.

Figure 4-4 on page 4-22 shows an example of the CICS System Task Table. This table is located in the member named STRBCTSK that is provided on your STROBE installation tape.

Figure 4-4. CICS System Task Table (Member 9STRBCTSK)

```
CICSTRAN TRANID=CATA
CICSTRAN TRANID=CATD
CICSTRAN TRANID=CATR
CICSTRAN TRANID=CAUT
CICSTRAN TRANID=CBRC
CICSTRAN TRANID=CCMF
CICSTRAN TRANID=CDBC
CICSTRAN TRANID=CDBD
CICSTRAN TRANID=CDBF
CICSTRAN TRANID=CDBI
CICSTRAN TRANID=CDBO
CICSTRAN TRANID=CDBQ
CICSTRAN TRANID=CDBT
CICSTRAN TRANID=CEBR
.
.
.
CICSTRAN TRANID=CECI
```

To edit the table, use the same format as shown in the example.

1. To add a transaction to be excluded for measurement, specify its four character name following "TRANID=".
2. To remove a transaction from the table causing STROBE to measure it, delete the entire line from the table that specifies its name.

Note: If you add a transaction name, the name must be placed in the list in ascending (top to bottom) alphabetical order. For example to add a transaction named CCYZ to the example table, you would add the specification between transaction names CCMF and CDBC.

3. To enable STROBE to access the table and use it to determine which transactions to measure, use the JCL in member STRBCTSJ to assemble and link (Figure 4-5 on page 4-23).

Figure 4-5. CICS System Task Table (Member STRBCTSJ)

```

//STEP0      EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
            DELETE STROBE.TEMP.LOADNONVSAM

            SET MAXCC=0
//*
//STEP1      EXEC PGM=IEFBR14
//STRLIB    DD DSN=STROBE.TEMP.LOAD,

            DISP=(NEW,CATLG,DELETE),
            SPACE=(TRK,(60,15,20)),DCB=(BLKSIZE=8096,RECFM=U),
            UNIT=SYSDA,VOL=SER=
//*
//LINK      PROC LPARM=NCAL
//LKED      EXEC PGM=HEWLH096,REGION=1024K,
            PARM='XREF,LIST,LET,&LPARM',
            COND=(0,LT)
//LKED      EXEC PGM=IEWL,REGION=1024K,
            PARM='XREF,LIST,LET,&LPARM',
            COND=(0,LT)
//*
//SYSPRINT  DD SYSOUT=*
//SYSUTI    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIN    DD DDNAME=SYSIN
//OLDMOD    DD DSN=STROBLIB,DISP=SHR
//NEWCSECT  DD DSN=&&OBJ,DISP=(OLD,DELETE)
//SYSLMOD   DD DISP=SHR,
            DSN=STROBE.TEMP.LOAD
//
//          PEND
//*
//STEP2      EXEC PGM=ASMA90,
            PARM='XREF(SHORT),RENT,OBJECT'
//*ASM      EXEC PGM=IEV90,
            PARM='XREF(SHORT),RENT,OBJECT'
//SYSLIB    DD DSN=STROBE.CNTL(CICSTRAN),DISP=SHR
//SYSIN     DD DSN=STROBE.CNTL(STRBCTSK),DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE(1700,(800,400))
//SYSUT2    DD UNIT=SYSDA,SPACE(1700,(800,400))
//SYSUT3    DD UNIT=SYSDA,SPACE(1700,(800,400))
//SYSPUNCH  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSLIN    DD DSN=&&OBJ(STRBCTSK),
            DISP=(NEW,PASS,DELETE),
            SPACE=(TRK,(60,15,20)),
            DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB)
//*
//STEP3      EXEC LINK,
            LPARM='AMODE=31,RMODE=ANY,RENT'
//LKED.SYSLIN DD *
            INCLUDE NEWCSECT(STRBCTSK)
            REPLACE STRBCTSK(STRBCTSK)
            INCLUDE OLDMOD(STRBDFHS)
            ORDER STRBDFHS
            ENTRY STRBDFHS
            NAME STRBDFSH(R)
//
//*

```

- supply a **JOB** statement
 - change **STROBE.CNTL** to the name of the STROBE control data set
 - change **STROBE.TEMP.LOAD** to the name of the STROBE temporary load library
 - change **STROBLIB** to the name of the STROBE load library built from the installation tape.
 - make any additional changes to customize the job control statements for your installation
4. Execute the job.
 5. Copy the STRBCTSJ load module from your STROBE temporary load library to the link list library.
 6. Perform an LLA REFRESH.

Chapter 5. STROBE Administration and Maintenance

This chapter describes the STROBE administration and maintenance tasks. It also contains command syntax for the STROBE commands you use to administer and maintain STROBE.

Administering STROBE

The STROBE administrator performs the following types of tasks:

- starts, stops, and restarts STROBE
- establishes and maintains the STROBE environment
- monitors and maintains the various STROBE data sets
- manages request groups owned by other users

You issue commands to the session manager or the session requester, depending on the type of administrative task.

- **Session manager commands** initiate STROBE measurement requests and are generally issued from the operator's console or from a batch job. After you start the session manager procedure, you can control STROBE by issuing commands to the session manager. For example, you can obtain status information about the session manager and data sets you are using. Or you could request the session manager to stop the environment so you can perform some system maintenance and start it again when the maintenance is completed.
- **Session requester commands** pass requests to the session manager and can be issued from a batch job, from the STROBE command language, or from STROBE/ISPF. These commands request the session manager to perform a STROBE operation for you. For example, a session requester command can add a measurement request, modify a request, or display information about STROBE status.

Submitting Commands to the Session Manager from an Operator's Console

You issue commands to STRBSM (the session manager program) through the operator's console, using the MVS system MODIFY command. For example:

```
MODIFY STRBSM,ENV DISABLE
MODIFY STRBSM,LIST
MODIFY STRBSM,SHOWPTF
```

Submitting Commands to the Session Requester with a Batch Job

To supply a privileged command to STROBE, submit a batch job that executes STRBCSR (the session requester program). Submit the job using a STROBE password, or a user ID with ADMIN authority. You can establish ADMIN authority through the security product at your installation. Refer to "Setting Up Security" on page 4-1, for instructions on setting up ADMIN authority.

Figure 5-1 shows the job control statements for a user who submits the SHOWPTF and CLEAN QUEUE=0 commands through the session requester.

If STROBE resides in an APF-authorized library that is not in the link list concatenation, include the name of the library in a STEPLIB DD statement that includes only APF-authorized data sets.

Figure 5-1. JCL for Submitting Commands Through the Session Requester

```
//STRBCSR EXEC PROC=STRBCSR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
SHOWPTF
CLEAN QUEUE=0
```

Managing the STROBE Environment

As you are using STROBE, you may have times when you need to start and stop STROBE and then re-start it. One situation that you can encounter is having to stop STROBE when measurement requests are active. The commands you use to stop STROBE will leave the STROBE environment in different states. The next sections describe how to start and stop STROBE and the possible environments that can exist.

Starting STROBE

To start a fresh STROBE environment from an operator's console with no outstanding measurement requests running, issue the following command to the session manager:

```
START STRBSM
```

To start STROBE from a batch job, invoke the session manager as shown in Figure 5-2. This JCL is created in the ALLOCATE member as shown in Figure 3-11 on page 3-22.

Figure 5-2. JCL for Starting STROBE

```
//STRBSM EXEC PGM=STRBCSM,REGION=4096,TIME=1440
//PARMLIB DD DSN=STROBE.PARMLIB(STROBE),DISP=SHR
//SYSLOG DD DSN=STROBE.V03R00.LOG(0),DISP=SHR
//SYSQUEUE DD DSN=STROBE.V03R00.QUEUE,DISP=SHR
//SYSMSGSGS DD DSN=STROBE.V03R00.SYSMSGSGS,DISP=SHR
//SYSGROUP DD DSN=STROBE.V03R00.GROUP,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//ABNLIGNR DD DUMMY
//ULT@OFF DD DUMMY
//SYSIN DD DDNAME=IEFRDER
```

Stopping STROBE

Compuware recommends that you use the MVS STOP command whenever possible to stop the STROBE address space, and terminate the execute of STROBE. If the results of using the STOP command aren't as you expected, you should use the MVS CANCEL command, which stops all measurement requests. If using the CANCEL command is unsuccessful, you can issue the MVS TERMINATE command, which leaves the measurement requests active.

You should note that using the CANCEL or TERMINATE command can produce unpredictable results in both the the session requester address space, the measured

address space or both address spaces because issuing either command causes STROBE to perform many different tasks. Therefore, you should wait a reasonable period of time before you issue any of the commands described in the next three sections.

Stopping STROBE with the STOP Command

To stop the STROBE address space, and terminate the execution of STROBE, issue the following command from the operator's console:

```
STOP STRBSM
```

STROBE will not terminate if you issue a STOP command on a system where there are active measurement sessions or session requesters. STROBE will issue message STR6441W or message STR6112E, which indicate what is active.

You can use this information to issue a SEND QUIT command to cancel the session requesters and stop the measurement. Then you can re-issue the STOP command and the STROBE environment will come down cleanly.

Stopping STROBE with the CANCEL Command

If there are active measurement requests that cannot be QUIT, Compuware recommends that you issue an MVS CANCEL command to STROBE. All session requesters receive a U611 ABEND and STROBE will attempt to end all active measurement requests and terminate.

At times, STROBE is unable to end all active requests and the environment cannot be entirely removed. By default, STROBE attempts to use the environment it finds when it is restarted.

If you want STROBE to create a new environment, you should specify a parameter called ENVNEW with the START command as shown in the example:

```
START STRBSM,PARM='ENVNEW=YES'
```

Setting ENVNEW to YES causes STROBE to create a new environment. Any measurement requests that had remained with the old environment will now fail.

The default value for ENVNEW is NO, meaning that a new environment is not created and the old one is used during attempts to restart STROBE.

Stopping STROBE with the TERMINATE Command

If you need to bring down STROBE when measurement sessions are still active and you want to keep them active, issue the following command from an operator's console:

```
MODIFY STRBSM,TERMINATE
```

Active measurement requests will continue to measure. All session requesters will receive a U611 ABEND. Then to re-start, issue the following command:

```
START STRBSM
```

By default, the remaining environment is reused, and the measurements remain active. If you find the remaining environment cannot be reused, restart STROBE by issuing the following command:

```
START STRBSM,PARM='ENVNEW=YES'
```

Stopping a STROBE XCF Group

Use the following procedure to stop a STROBE XCF group:

1. To stop a STROBE XCF group, you first have to end all instances of STROBE measurement sessions in the group. To do this, you can:
 - Stop submitting new requests and wait for active requests to complete.
 - Use the LIST TYPE=ACTIVE command to determine which measurements are active and then use the SEND command with the QUIT option to terminate them.
 - Prevent queued requests from becoming active by disabling the attach filter with an ENV DISABLE command.
2. Issue the STOP STRBSM command for each instance of STROBE that is to be stopped.

If you missed any of the active measurement sessions in Step 1, the STOP command is rejected and STROBE issues message STR6411W which lists the request numbers of the active sessions. Stop those sessions with one of the methods listed in Step 1 and re-issue the STOP command.

Measurement Services Address Space (MSAS)

The Measurement Services Address Space (MSAS) handles functions that are not applicable to the session manager or a specific STROBE measurement. AutoSTROBE runs in MSAS. If the STROBE Advanced Session Measurement Feature is installed and AutoSTROBE is active (the AUTOSTROBE parameter is set to YES), the STROBE session manager starts MSAS when STROBE is initialized.

Stopping and Starting MSAS

When STROBE terminates, the MSAS also is stopped. You can stop the MSAS but leave STROBE operating by issuing the following MVS command:

```
P msasname
```

where “msasname” is the name of the MSAS procedure.

To be able to restart the MSAS address space using the session manager, you must issue the following STROBE command:

```
F STRxSM.STARTMSAS
```

where STRxSM is the name of the session manager address space qualified by its version letter.

Starting and Stopping AutoSTROBE

To stop AutoSTROBE without stopping the MSAS or the session manager, issue the following MVS command:

```
F msasname,STOPAUTO
```

To restart AutoSTROBE, issue this MVS command:

```
F msasname,STARTAUTO
```

where “msasname” is the name of the MSAS procedure.

Starting and Stopping AutoSTROBE Autoloader Processing

To stop the AutoSTROBE Autoloader without stopping AutoSTROBE, the MSAS or the session manager, issue the following MVS command:

```
MODIFY msasname,STRTACND
```

To restart the AutoSTROBE Autoloader, issue this MVS command:

```
MODIFY msasname,STOPACND
```

where “msasname” is the name of the MSAS procedure.

Establishing and Modifying the STROBE Environment

To establish the environment, STROBE performs the following tasks when the address space begins as a started task:

- loads the programs STRBCLPA and STRBCLPB into the CSA
- adds the STROBE SVC to the system SVC table
- if the environment is enabled, modifies the attach SVC (SVC 42) address to point to the STROBE attach filter, a routine within STRBCLPA

To establish the STROBE environment, the session management facility modules must execute in an authorized state. They must reside in an authorized link list data set or an APF-authorized library and be link-edited with an authorization code of 1.

Examples

To create the STROBE environment from an operator’s console, issue the following MVS system command:

```
START STRBSM
```

To disable the attach filter from an operator’s console, issue the following MVS system command:

```
MODIFY STRBSM,ENV DISABLE
```

To disable the attach filter from the SYSIN input stream in a batch job, issue the following STROBE command:

```
ENV DISABLE
```

Determining Session Manager Status

Use the STATUS command to gather the following information about the session manager:

- creation date
- number of requests by type
- maximum number of total and active requests
- whether the attach filter is enabled or disabled
- amount of storage the session manager is occupying
- SVC in use
- names of the log, queue, message, and request group data sets and their percent of fullness
- operating system level
- user IDs of current users of the session manager

Use the SHOW command to view the following information:

- ABEND history of STROBE since the session manager started.
- Amount of common storage area used by STROBE.

- STROBE default values supplied through the parameter data set and currently in storage. In addition, the session management facility displays
 - current release
 - PTF level
 - tape volume serial number
- Current enqueues held by STROBE
- Status of the SVCs. If you do not supply a list of numbers, the session manager displays the status of the SVC it is using
- Names of all active systems within the STROBE-XCF group.

Note: STROBE displays only the names of the systems to which the user has access.

Managing STROBE Data Sets

Periodically review the status of STROBE data sets to make sure that they have sufficient space and to clean them up or copy them to backup data sets if they are reaching capacity.

Reorganize a data set as follows:

- make a backup copy
- delete the data set and reallocate it with the same space specification
- rebuild the data set from the backup copy

If reorganizing does not free enough space, expand the data set as follows:

- make a backup copy of the data set
- delete the data set and reallocate it with a larger space specification
- rebuild the data set from the backup copy

Managing the Log Data Set

STROBE cannot execute when the log data set is full. To make sure that space is available, issue the SHOW LOG command to display the status of the log.

```
SHOW LOG
```

You can issue this command through the MVS system MODIFY command on the operator's console; through batch execution of the session requester; or through STROBE/ISPF (where the command is .SHOW LOG).

STROBE displays the following status information:

- name of the log data set
- percentage of space used
- DCB characteristics
- space allocation
- number of extents

If the log data set belongs to a GDG, STROBE displays the following information for each data set in the group (to a limit of ten):

- name of the log data set
- date and time STROBE allocated the data set

Managing a GDG Log Data Set

When you install STROBE with a log data set that belongs to a GDG, STROBE takes care of the maintenance. It dynamically allocates a new generation of the log data set when

the current generation reaches the threshold set by the LOG_MAX value in the parameter data set. You rarely need do anything else to make sure the log data set is available.

Use the LOGSWITCH command to force the allocation of a new generation before the current log data set is full. Submit the command with the start of the session manager or as a command through the operator's console. From an operator's console, enter:

```
MODIFY STRBSM,LOGSWITCH
```

STROBE closes the current data set and allocates a new one with a higher generation number.

CAUTION:

Only use the LOGSWITCH command to manage GDG log data sets. Do not use the LOGCLEAR command.

When STROBE allocates a new log data set automatically or through a LOGSWITCH command, it applies the following values from the parameter data set:

- LOG_SPACE
- LOG_UNIT
- LOG_VOL

If these values are not in the parameter data set, STROBE uses the allocation information from the current in-use log. Other characteristics for newly allocated data sets are determined by Automatic Class Selection (ACS) code that is written by the system administrator at your site. The session manager supports allocations in blocks, tracks, or cylinders.

Managing a Non-GDG Log Data Set

STROBE helps you manage the log when you have not defined it to a GDG. The session manager sends a warning message to the operator's master console based on values coded for the LOGWARN keyword in the parameter data set. STROBE sends the first message when the log contains a higher percentage than the threshold percentage value. STROBE sends an additional message each time the log exceeds the incremental percentage value. When the log fills, STROBE issues a message with each attempt to write to the log.

For example, if you code LOGWARN=(80,5) in the parameter data set, STROBE sends a warning message at the following intervals:

- when the log data set exceeds 80% full
- when the log data set exceeds 85%, 90%, and 95% full
- each time STROBE attempts to write to the log after it fills

Based on these messages, you can modify your automated facility for handling console messages to schedule the execution of the STROBE utilities for managing the log data set.

STROBE provides a command (LOGCLEAR) and a utility program (STRBCLOG) to manage non-GDG log data sets.

Using the LOGCLEAR Command

Use the LOGCLEAR command to clear and reuse the log dataset, as follows:

```
MODIFY STRBSM,LOGCLEAR
```

Submit the command with the start of the session manager or as a command through the operator's console. STROBE does not preserve the original contents of the log.

CAUTION:

Only use the LOGCLEAR command for non-GDG log data sets. Do not use the LOGSWITCH command.

Using the Log Utility Program

To copy the log to a backup data set and clear the log, submit the STROBE Log Utility program from your unauthorized STROBE load library. The utility enqueues exclusively on the log resource, so that STROBE cannot write to the log when it is being cleared. The STRBCLOG program (Figure 5-3) copies all DCB characteristics from the current log and accepts overrides on the DD statement for BLKSIZE to allow for efficient block sizes on the backup.

Figure 5-3. Copying the Log to a Backup Data Set (Member (STRBCLOG))

```
//LOGUTIL EXEC PGM=STRBCLOG
//STEPLIB DD DSN=STROBLIB,DISP=SHR
//SYSUT1 DD DSN=STROBE.LOG,DISP=SHR
//SYSUT2 DD DSN=BACKUP.STROBE.LOG,DISP=(NEW,CATLG),
//        SPACE=(TRK,(10,10),RLSE),UNIT=SYSDA,
//        DCB=BLKSIZE=6048
```

Managing the Queue Data Set

Use the following commands to manage the STROBE queue data set. The full syntax for these commands is provided in “Syntax for Administrative Commands” on page 5-17.

- STATUS
- SHOW QUEUE
- CLEAN
- LIST

To view the characteristics of the queue and other data sets, issue the following command from the operator's console:

```
SHOW QUEUE
```

STROBE displays the following information:

- data set name
- percentage of space used
- number of extents
- RBA statistics
- available FREE SPACE
- number of records inserted, updated, and deleted
- number of control interval and control area splits
- number of records in the data set

Checking the Status

Use the STATUS command to determine the amount of space remaining in the STROBE queue data set. Alternatively, look in the log, which provides messages about the size of the data set contents. When you customize the STROBE parameter data set during installation, use the MAXREQ parameter to establish the maximum number of requests allowed in the queue data set. If you are using the STROBE Advanced Session Management Feature, these requests include queued, completed, active, and deferred requests. The default number of requests allowed is 200. STROBE issues a warning message with each request whenever the queue size has reached 75% of the maximum value and accepts no further requests after the maximum is reached.

To determine the number of requests, invoke the STATUS command or the SHOW QUEUE command. Then issue the CLEAN command to remove completed requests, queued requests, or both, based on the number of days these requests have been on the queue.

Expanding the Queue Data Set

Periodically, you may need to reorganize or expand the data set, as follows:

1. Delete all unwanted requests.
2. Stop STROBE.
3. Reorganize or expand the data set.
4. Restart STROBE.

Deleting all Queued Requests

To delete all queued requests, regardless of how long they have been on the queue, use the following command:

```
CLEAN QUEUE=0
```

Deleting Specific Requests

To delete specific request categories, first issue the LIST command (with specific arguments) and then issue the CLEAN command (with the same arguments). For example, to target requests queued for four or more days and requests that have been completed for three or more days, and then to delete those requests, use the following commands:

```
LIST QUEUE=4,COMPLETE=3
CLEAN QUEUE=4,COMPLETE=3
```

Synchronizing the Queue Data Set

In a multisystem environment, XCF communications occasionally fail. If this happens, internal representations of requests can lose synchronization with their counterparts on the queue data set. For example, a request marked “Completed” on one member of the STROBE-XCF group is marked “Queued” on another. Message STR6814E alerts you to this problem.

You can correct the problem in either of the following ways:

- Issue the RESYNC command, specifying the request number of the problem request. You can issue this command from any system in the STROBE-XCF group. For example, to refresh request number 453 with information from the queue data set, enter

```
RESYNC 453
```

- Stop all members in the STROBE-XCF group and restart STROBE as described in “Stopping a STROBE XCF Group” on page 5-3.

Managing the System Message Data Set

Periodically issue a STATUS or a SHOW QUEUE command to determine how much space remains in the STROBE system message data set. Alternatively, you can look in the log for messages that indicate how much space is left in the data set. During periods of especially heavy usage, the system message data set might become full. If the data set fills, STROBE issues a warning message and operates in a degraded mode. Messages might be lost until you reorganize the data set. To correct this type of situation, take one or both of the following actions:

- Temporarily reduce the amount of information in the system data set, for example by asking frequent STROBE users to list their measurement requests. This action deletes messages that have not been previously viewed, and subsequently will only be available from the log.

- Reorganize or expand the system message data set to reduce the chances of it becoming full.

Expanding the System Message Data Set

You may need to reorganize or expand the data set periodically using standard VSAM utility programs. Do the following:

1. Stop STROBE.
2. Reorganize or expand the data set.
3. Restart STROBE.

Managing the History Data Set

Periodically, check to see how much space remains in the STROBE history data set. During periods of especially heavy usage, the history data set might become full. If the data set fills, STROBE issues a warning message and operates in a degraded mode. At this point, you will not be able to collect measurement session history until you expand or reorganize the data set. To correct this type of situation, take one or both of the following actions:

- Remove information from the history data set. Delete any measurement session history records that you no longer need. This process is described in Chapter 3 of the *STROBE MVS User's Guide with Advanced Session Management*.
- Reorganize or expand the history data set.

Reorganizing the History Data Set

To reorganize the STROBE history data set and its associated alternate index, execute the reorganization utility provided on the STROBE installation tape.

To execute the reorganization utility, make the following edits to the JCL shown in Figure 5-4:

- Change **STROBE.STRHIST** to the name of your STROBE history data set.
- Change **STROBE.STRHISTA** to the name of your STROBE history alternate index.
- Change **STROBE.STRHIST.COPY** to the name of the exported copy of the STROBE history data set.
- Change **STROBE.STRHISTA.COPY** to the name of the exported copy of the STROBE alternate index history data set.
- Change **SYSDA** to the unit name you want to assign for the copy data sets.
- Adjust the space allocations for each dataset to approximate the size of each VSAM dataset being exported.

The copy data sets can be deleted once this operation is successful.

Figure 5-4. History Data Set (Member HSTREORG)

```

/*STROBE JOB 0,STROBE

/*
//HSTREORG EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//HISTCOPY DD DISP=(,CATLG,DELETE),DSN=STROBE.STRHIST.COPY,
//          UNIT=SYSDA,SPACE=(CYL,(15,5),RLSE)
//HISTACPY DD DISP=(,CATLG,DELETE),DSN=STROBE.STRHISTA.COPY,
//          UNIT=SYSDA,SPACE=(CYL,(15,5),RLSE)

//SYSIN DD*
EXPORT STROBE.STRHIST -
        OUTFILE(HISTCOPY) -
        TEMPORARY
/**/
EXPORT STROBE.STRHISTA -
        OUTFILE(HISTACPY) -
        TEMPORARY
/**/
IMPORT -
        INFILE(HISTCOPY) -
        OUTDATASET(STROBE.STRHIST)
/**/
IMPORT -
        INFILE(HISTACPY) -
        OUTDATASET(STROBE.STRHISTA)
/**/
IF MAXCC=0
  THEN DO
          LISTC ENT(STROBE.STRHIST) ALL
          LISTC ENT(STROBE.STRHISTA.*) ALL
  END

```

Expanding the History Data Set

You may need to expand the data set periodically using standard VSAM utility programs. Only use these programs: do not use a third party product. Perform the following steps:

1. Create new data sets for the history data set and the alternate index, increasing the sizes of each.
2. Use the IDCAMS REPRO command to populate the existing history data set.
3. Use the IDCAMS BLDINDEX command to create the alternate index.

Printing the History Log Data Set

The STROBE history manager maintains a log of changes made to the history data set. Should this log data set become full, execute the history log utility to print, delete, and reallocate the history log data set.

To execute the utility, perform the following steps:

1. Make the following edits to member HSTPDLOG shown in Figure 5-5:
 - Change **STROBE.STRHIST.LOG** to the name of your STROBE history log data set.
 - Change **SYSDA** to the unit assigned for the history data set.
 - Change **VOL=SER=**, if required
 - Adjust the space allocations as necessary.
2. Execute the job.

Figure 5-5. History Data Set (Member HSTPDLOG)

```

/*STROBE JOB 0,STROBE
/*
// PRINT EXEC PGM=IEBGENER
// SYSPRINT DD DUMMY
// SYSIN DD DUMMY
// SYSUT2 DD SYSOUT=*
// SYSUT1 DD DSN=STROBE.STRHIST.LOG,
//
// UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,KEEP)
/*
//HISTLOG EXEC PGM=IEFBR14
//STRHLOG DD DSN=STROBE.STRHIST.LOG,
// SPACE=(TRK,(100,20),DISP=(,CATLG,DELETE),
// DCB=(RECFM=VBA,LRECL=137,BLKSIZE=4096),
// UNIT=SYSDA,VOL=SER=
/*

```

Managing the AutoSTROBE Data Set

You may need to expand the AutoSTROBE data set periodically using standard VSAM utility programs. Only use these programs: do not use a third party product. Perform the following steps:

1. Create a new data set for the AutoSTROBE data set.
2. Use the IDCAMS REPRO command to populate the existing AutoSTROBE data set and Compuware recommends backing up the AutoSTROBE data set using the IDCAMS utility every seven days.

Managing the Request Group Data Set

This task applies only to users of the STROBE STROBE Advanced Session Management Feature.

You should monitor the request group data set and delete or reevaluate request groups that are used infrequently. STROBE users can manipulate and view only their own request groups. However, as STROBE system administrator, you can

- list all the elements in the request group data set by owner ID or by group name
- delete any element or group in the request group data set
- transfer ownership of a request group
- copy a request group
- change the parameters of an element in a request group or add a new parameter
- submit a request group regardless of whether you created the group

You should set policies for your site controlling the length of time a request or request group can remain idle in the request group data set. View the request groups at regular intervals, paying attention to the last submit date. Note request groups that exceed your site's policies and delete them.

The following commands help you to manage the request group data set.

This command	Allows you to
ADDGRP	Create a request group or add an element to an existing group
LISTGRP	View the contents of a request group
DELGRP	Delete a group or a number of group

This command	Allows you to
DELETE	Delete an element from a request group
COPYGRP	Transfer ownership of a request group and/or copy the group
CHGGRP	Add a new parameter for a request group element or change the value of a parameter for an existing element
SUBMIT	Send a request group to initiate a set of measurement requests

The syntax of these commands is described in the section called “Syntax for Administrative Commands” on page 5-17.

Expanding the Request Group Data Set

If your site is using the STROBE Advanced Session Management Feature, you need to reorganize or expand the request group data set when it reaches capacity. (Message STR6810W indicates when the request group data set is getting full.) A short term solution is to list the groups and delete any request groups and elements you do not need. As a long term solution, reorganize or expand the data set using standard VSAM utility programs as follows:

1. Delete all unwanted request groups.
2. Stop STROBE.
3. Reorganize or expand the data set.
4. Restart STROBE.

Resequencing Request Element Numbers

As elements are added to a request group, STROBE assigns them each a unique sequence number. The maximum sequence number that STROBE assigns is 999. Once STROBE assigns that number, you cannot add any more request elements to the request group even though the actual number of elements in the group is less than 999. You can renumber the request elements, reclaiming unused sequence numbers, by copying and renaming the request group.

Transferring Ownership of a Request Group

You must have STROBE administrative authority to manipulate request groups that you do not own. As administrator, you can give one STROBE user access to another user's request group by assigning the person as its owner. To do so, use the COPYGRP command as follows:

```
COPYGRP GROUP=name, NEWGROUP=newname, GRPOWNER=currentowner,
[NEWOWNER=newowner]
```

It is not necessary to rename the group at this time, although you can do so. If you do not specify a new owner, ownership of the group is transferred to you.

Controlling Request Group Display

STROBE enables you to limit what request group names display in the STROBE - REQUEST GROUP PROCESSING panel (Figure 5-6). If you have administrator privileges, the panel will contain an “Owner Mask” field. If you want to view the groups created by a specific owner ID:

1. Enter “Y” in the REQUEST GROUP PROCESSING field of either the STROBE-ADD ACTIVE or ADD QUEUED REQUEST panels. The STROBE-REQUEST GROUP PROCESSING panel appears.

- In the "Owner Mask" field, enter the user ID that created the group(s) you want to view. You can also use an "*" asterisk as a wildcard indicator. For example, entering "WPA*" would display all request groups created by user IDs beginning with "WPA".

Figure 5-6. Controlling Request Group Processing

```

----- STROBE - REQUEST GROUP PROCESSING -----
COMMAND ==>                                     SCROLL ==> PAGE

Group Name ==>                                (New Group Name) Owner Mask ==> WPA*
Change Sort Criteria ==> N                    (Y or N)      (Limit Display of Groups)
-----
Line Options:
C - Copy Group      E - Edit Group Elements    U - Submit Group
D - Delete Group    L - List Group Elements      T - Submit Group w/Schedule

                                     (MM/DD/YYYY)
- OPTION - --- GROUP NAME --- --- OWNER --- ELEMENT CNT --- LAST SUBMIT --- LAST EDIT ---
-         DB2_GRP1_PROD      WPAABD      18      09/30/2002  07/21/2002
-         DB2_GRP1_TEST      WPAABD       7              07/11/2002
U         DB2_CICS_CON1      WPAABD      15      08/03/2002  07/21/2002
-         CICSGRP1          WPAABD       2      09/06/2002  07/23/2002

```

Maintaining STROBE

STROBE is maintained through PTF update tapes and individual transmissions of PTFs.

Creating the Maintenance Data Set

To load a STROBE fix tape, submit the MAINTNCE job after making the following edits:

- Change **STROBE.FIX.CNTL** to the name of your maintenance control dataset
- Change **VVVVVV** to the name the volser of the STROBE fix tape.
- Submit the job.

Figure 5-7. Loading Maintenance Data Set (Member MAINTNCE)

```

//MAINT EXEC PGM=IEBCOPY,
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(2))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(2))
//IN DD DSN=STROBE.SYSTEM,
// DISP=OLD,UNIT=CART,VOL=SER=VVVVVV
//OUT DD DSN=STROBE.FIX.CNTL,
// UNIT=SYSDA,VOL=SER=,DISP=(NEW,CATLG,DELETE),
// SPACE=(3120,(6000,500,20))
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT

```

PTF Update Tapes and FTP Files

Between releases of STROBE, you may receive PTFs to correct problems. The PTFs are supplied on tape in machine-readable format. Apply these PTFs to keep your STROBE system up to date and prevent the recurrence of corrected problems.

The tape is a standard-labelled tape, sequentially numbered. It is accompanied by instructions for applying PTFs.

The tape may contain the following:

- procedures to expand modules requiring additional patch area
- IMASPZAP procedures to apply machine-readable fixes to load modules
- program temporary fixes
- SQLAF object code
- additional members with appropriate instructions.

To install the PTF tape, unload it to your system using IEBCOPY. Then complete the following tasks using the new control data set.

1. You need to apply the appropriate level member as determined by your current PTF level. The current level is indicated on the Measurement Session Data report in the field named PTF LVL. The second number following the slash in the field value is the PTF level. The members use a naming convention of FROMnnnn where nnnn is the PTF level. Therefore you want to apply the member with the name where nnnn is the number one above your current maintenance level. For example, if the number following the slash is 0023 you want to apply member FROM0024.
2. Edit the member you have selected.
 - Remove any PTFs you may have applied
 - Supply a JOB statement
3. Change 'STROBLIB' to your unauthorized STROBE library
4. Stop STROBE with the following command


```
STOP STRBSM
```
5. Execute the job.
6. Submit the COPY or COPYTEST member.
7. Submit the following command:


```
LLA REFRESH
```
8. Re-start STROBE.

Individual PTFs

You may also receive, through fax, e-mail, or postal mail, individual PTFs that need to be applied to STROBE.

Maintaining the SQL Analysis Feature

The STROBE fix tape contains SQL Analysis Feature object and DBRMs.

Replace the SQL Analysis Feature MVS Component Object

When you receive a STROBE fix tape, use the SQLAFRPL member (Figure 5-8 on page 5-17) from your STROBE.CNTL library to copy the updated SQL Analysis Feature objects into your STROBE.CNTL library. To replace existing SQL Analysis Feature and DBRM objects

1. Make the following edits to member SQLAFRPL:
 - change STROBE.CNTL to the name of the PDS that contains your copy of the STROBE distribution tape
 - Change STROBE.FIX.CONTROL to the the name of the maintenance data set
 - change UNIT=SYSDA to another unit name according to your installation environment
2. Execute the job. This job

- renames the existing SQL Analysis Feature and DBRM object members in your STROBE.CNTL library
- copies the new SQL Analysis Feature and DBRM object members into your STROBE.CNTL library from the distribution tape

For example, object member STRANA31 is renamed to SSRANA31 and the new object member is copied from the distribution tape as STRANA31.

If you receive a non-zero return code, contact your Compuware systems engineer for technical support.

Relink the SQL Analysis Feature MVS Component

Execute the SQLINSTL job to relink the SQL Analysis Feature MVS component. See “Step 15. Install the SQL Analysis Feature” on page 3-44 for instructions.

Bind the DB2 Version-Specific DBRM

Execute the STRANABND job to bind the new version of the SQL Analysis Feature MVS component to a DB2 subsystem. See “Other Considerations” on page 3-46 for instructions.

Figure 5-8. Replace SQL Analysis Feature objects (Member SQLAFRPL)

```

//STEP01 EXEC PGM=IEBCOPY, PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD UNIT=SYSDA, SPACE=(TRK, (2, 2))
//SYSUT4 DD UNIT=SYSDA, SPACE=(TRK, (2, 2)),
//IN DD DSN=STROBE.FIX.CNTL, DISP=SHR
//OUT DD DSN=&&SQLAFON, DISP=(NEW, PASS)
// UNIT=SYSDA, SPACE=(TRK, (25, 25, 5)),
// DCB=(LRECL=80, BLKSIZE=3120, RECFM=FB, DSORG=PO)
//SYSIN DD *
COPY INDD=IN, OUTDD=OUT
/*
//STEP02 EXEC PGM=IEBCOPY, REGION=2M, COND=(0, NE)
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, (2))
//SYSUT4 DD UNIT=SYSDA, SPACE=(CYL, (2))
//INDD1 DD DISP=SHR, DSN=STROBE.CNTL
//OUTDD1 DD DSN=&&STRBCNTL, DISP=(NEW, PASS),
// UNIT=SYSDA, VOL=SER=,
// SPACE=(TRK, (25, 25, 5)), DCB=*. INDD1
//SYSIN DD *
COPY OUTDD=OUTDD1, INDD=((INDD1, R))
SELECT MEMBER=((STRAIA51, SSRAIA51))
SELECT MEMBER=((STRAIA61, SSRAIA61))
SELECT MEMBER=((STRAIA71, SSRAIA71))
SELECT MEMBER=((STRANA51, SSRANA51))
SELECT MEMBER=((STRANA61, SSRANA61))
SELECT MEMBER=((STRANA71, SSRANA71))
SELECT MEMBER=((STRAN151, SSRAN151))
SELECT MEMBER=((STRAN251, SSRAN251))
SELECT MEMBER=((STRAN161, SSRAN161))
SELECT MEMBER=((STRAN261, SSRAN261))
SELECT MEMBER=((STRAN171, SSRAN171))
SELECT MEMBER=((STRAN271, SSRAN271))
SELECT MEMBER=((STRANV24, SSRANV24))
/*
//STEP03 EXEC PGM=IEBCOPY, REGION=2M, COND=(0, NE)
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, (2))
//SYSUT4 DD UNIT=SYSDA, SPACE=(CYL, (2))
//SYSPRINT DD SYSOUT=*
//OUTDD1 DD DISP=SHR, DSN=STROBE.CNTL
//INDD1 DD DISP=OLD, DSN=&&STRBCNTL
//SYSIN DD *
COPY OUTDD=OUTDD1, INDD=((INDD1, R))
/*
//STEP04 EXEC PGM=IEBCOPY, REGION=2M, COND=(0, NE)
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, (2))
//SYSUT4 DD UNIT=SYSDA, SPACE=(CYL, (2))
//SYSPRINT DD SYSOUT=*
//INDD1 DD DISP=(OLD, PASS), DSN=&&SQLAFON
//OUTDD1 DD DISP=SHR, DSN=STROBE.CNTL
//SYSIN DD *
COPY OUTDD=OUTDD1, INDD=((INDD1, R))
SELECT MEMBER=STRAIA51
SELECT MEMBER=STRAIA61
.
.
SELECT MEMBER=STRAN171
SELECT MEMBER=STRAN271
SELECT MEMBER=STRANV24

```

Syntax for Administrative Commands

This section describes the commands you use to administer STROBE. In the command syntax descriptions, keywords are capitalized and variables appear in italicized lower case. Brackets [] enclose optional operands. Alternatives within operands are grouped within braces { }, and separated by a vertical bar |. Do not code brackets, braces, or vertical bars.

Note: Users with ADMIN authority can enter these administrative commands from the ISPF panel command line. A period must precede the command name.

The ADDGRP Command

The ADDGRP command creates a request group or adds an element to it.

ADDGRP	<i>job</i> , GROUP= <i>groupname</i>
--------	--------------------------------------

JOB

The name of the job step that you want to measure. See Chapter 3 of the *STROBE MVS User's Guide with Advanced Session Management* for information about the syntax for specifying what you want to measure.

GROUP

The name of the group you are creating or the name of the group to which you are adding a request group element.

The CLEAN Command

The CLEAN command deletes requests from the queue data set.

CLEAN	{ QUEUE= <i>qqq</i> COMPLETE= <i>ccc</i> QUEUE= <i>qqq</i> , COMPLETE= <i>ccc</i> }
-------	---

QUEUE

The QUEUE operand allows you to delete queued requests that are older than the number of days specified in *qqq*, where *qqq* is a one- to three-digit value between 0 and 999.

COMPLETE

The COMPLETE operand allows you to delete the completed requests that are older than the number of days specified in *ccc*, where *ccc* is a one- to three-digit value between 0 and 999.

The COPYGRP Command

The COPYGRP command copies a request group and assigns the new group a new owner. The command also can be used to copy the contents of a group to another group. If you only specify the GROUP and NEWGROUP parameters, the elements in the first group are copied to a second group and ownership of the new group is assigned to you.

COPYGRP	GROUP= <i>name</i> , NEWGROUP= <i>newname</i> , GRPOWNER= <i>currentowner</i> , [NEWOWNER= <i>newowner</i>]
---------	--

GROUP

The current name of the request group you are copying.

NEWGROUP

The new name of the request group, or, if you are not changing names, the current name of the request group.

GRPOWNER

The owner of the request group to be copied.

NEWOWNER

The owner of the new request group. If you do not specify this parameter, the group ownership is transferred to you.

The CHGGRP Command

The CHGGRP command adds a new parameter(s) for an element in a request group or changes the value of an existing parameter(s) for a request group element.

CHGGRP	<i>elementnumber</i> , GROUP= <i>newname</i> , GRPOWNER= <i>groupowner</i> , PARAMETER= <i>parametervalue</i>
--------	--

ELEMENTNUMBER

The number of the group element to be changed.

GROUP

The name of the request group.

GRPOWNER

The owner of the request group containing the element to be changed. (You do not have to specify this parameter if you created the request group.)

PARAMETER

The parameter(s) you want to add or delete for the element. To delete a parameter, do not specify a value. To change an existing parameter, specify a new value for it.

The DELGRP Command

The DELGRP command deletes a group. With administrative authority, you can delete a group you did not create by supplying the group owner name.

DELGRP	<i>group</i> , GRPOWNER= <i>groupowner</i>
--------	--

GROUP

The name of the request group to be deleted.

GRPOWNER

The owner of the request group you want to delete.

The DELETE Command

The DELETE command deletes a request group element. With administrative authority, you can delete an element in a request group you do not own.

DELETE	<i>elementnumber</i> , GROUP= <i>groupname</i> , GRPOWNER= <i>groupowner</i>
--------	--

ELEMENTNUMBER

The number of the element to be deleted.

GROUP

The name of the request group containing the element to be deleted.

GRPOWNER

The owner of the request group containing the element to be deleted.

The ENV Command

The ENV command establishes, changes, or displays the status of the STROBE environment. This command can only be issued from the operator's console.

ENV	CHECK= <i>svcno</i> {DISABLE ENABLE} , DISABLE ENABLE STATUS
-----	---

CHECK

CHECK=*svcno* determines whether the specified SVC appears to be in use. STROBE considers an SVC to be unused if it is a type 5 SVC or if its associated address in the SVC table points to a dummy entry. See "Determining Session Manager Status" on page 5-5 for more information.

DISABLE

Specifies that the STROBE attach filter be placed in the disabled state. Measurement sessions in progress can continue and measurement requests can still be submitted. STROBE services a request to measure an active job step, but it does not service a queued request unless the attach filter is enabled before the target job step is executed.

ENABLE

Specifies that the STROBE attach filter be placed in the enabled state.

STATUS

Requests a message giving the status of the STROBE session manager. See "Determining Session Manager Status" on page 5-5 for more information.

The LIST Command

The LIST command displays requests and messages that currently exist in the STROBE system. You can specify syntax for the LIST command that will provide many types information about STROBE. The following explains the LIST command syntax.

The following is the STROBE command language syntax for the LIST command, which shows information about STROBE measurement requests that have not been deleted:

```
LIST  {{jobname|rrrr|*}|{jobname|rrrr}|({jobname|rrrr,jobname|rrrr,|rrrr-
      rrrr,...})|
      SYSTEM={*ALL|systemname|(systemname,...)}
      EXCLUDE={systemname|(systemname,...)}
      TYPE={QUEUED|ACTIVE|COMPLETE|ALL|DEFERRED}|
      OWNERID=ownerid
      {QUEUE=qqq|COMPLETE=ccc|QUEUE=qqq,COMPLETE=ccc}
      [,QUEUE=qqq]
      [,COMPLETE=ccc]
      [,DEFERRED=ddd]
      [,DELETE]
```

Controlling LIST Operands

The operands in the following table control the attributes of the LIST command. The SYSTEM and EXCLUDE operands pertain to all LIST command formats. The TYPE and OWNERID operands pertain to format 2 and format 3. The QUEUE, COMPLETE, and DELETE operands pertain to formats 1 through 3. Format 4 uses only QUEUE and COMPLETE.

SYSTEM	SYSTEM specifies the systems for which you would like to view the status of measurement requests.
EXCLUDE	EXCLUDE specifies the systems for which you do not want to view the status of measurement requests.
TYPE	TYPE specifies the type of requests to be listed: <ul style="list-style-type: none"> • QUEUED for all queued requests • ACTIVE for all active requests • COMPLETE for all completed requests • ALL for all requests of any status • DEFERRED for all deferred requests • AUTOSTROBE for all AutoSTROBE requests
OWNERID	ownerid is either the TSO user ID or a job name of a batch job that submitted a measurement request
QUEUE	The QUEUE=qqq operand allows you to limit the display to those queued requests that have been queued longer than the number of days specified in qqq.
COMPLETE	The COMPLETE=ccc operand allows you to limit the display to those completed requests that have been completed longer than the number of days specified in ccc.
DELETE	The DELETE operand allows you to delete completed requests after any pending messages have been listed.

Specifying the List Format

The LIST command allows you to view the status of measurement requests in four different formats:

- selected requests
- requests with a given status
- requests according to a specified user ID

- all requests and their status

In a multisystem environment, the default for each of the LIST formats is all of the systems in the sysplex. For example, if you specify the following LIST command,

```
LIST MYJOB
```

STROBE returns status information for each request on any system in the sysplex with the job name specification of "MYJOB".

If you would like to limit the scope of the LIST command, use the SYSTEM or the EXCLUDE parameters with any of the LIST formats.

Format 1: List Selected Requests

Format 1 of the LIST command lists status, request operands, and pending (as yet unlisted) messages for selected STROBE measurement requests. If a request is active, LIST also displays the sampling status and the number of samples collected. It also allows you to delete completed requests after their messages have been displayed. For example:

LIST 236,DELETE	Shows the request information for request 236, then deletes the request
LIST (MYJOB,44)	Shows request information for the jobname MYJOB and for request 44
LIST MY*	Shows requests submitted for jobs whose names begin with the characters "MY"

Format 2: List Requests with a Given Status

Format 2 of the LIST command lists status, request operands, and pending messages for all requests of a specified status. If a request is active, LIST also displays the sampling status and the number of samples collected. It also allows you to delete completed requests after their messages have been listed. For example:

```
LIST TYPE=ACTIVE Shows all active requests
```

Format 3: List Requests by Ownerid

Format 3 of the LIST command lists the request number, job name, and the status for each request submitted under the specified TSO user ID or job name. If a request is active, LIST also displays the sampling status and the number of samples collected. It also allows you to delete completed requests after their messages have been listed. For example:

```
LIST OWNERID=MYT* Shows all requests submitted by a TSO user ID or job names beginning with the characters "MYT"
```

To view requests submitted under TSO user IDs or job names that begin with a common prefix, specify the common characters followed by an asterisk (*), which is the global search character.

Format 4: Show the Status of All Requests

Format 4 of the LIST command lists the request number, job name, and the status (queued, active, deferred, or complete) for all requests. For example:

LIST QUEUE=002	Shows all queued requests that are more than two days old, giving request number, job name, and status
LIST QUEUE=005,COMPLETE=005	Shows all queued and completed requests that are more than five days old, giving request number, job name, and status

The LISTGRP Command

The LISTGRP command displays the request groups that currently exist in the STROBE system and the elements of a request group if one's name is specified.

LISTGRP	[<i>group</i> ,GRPOWNER= <i>groupowner</i>]
---------	---

GROUP

The name of the request group for which the contained elements should be listed. If the same name exists for groups created by different owners, all elements in the groups with identical names are listed.

GRPOWNER

The owner of the group. If groups with the same name exist, you can use the GRPOWNER parameter to designate which group's elements you want to list. If you specify just this parameter, only groups belonging to the person associated with this GRPOWNER identification are listed. You can also specify an asterisk (*) as a global search indicator. For example, if you set to GRPOWNER to COM*, all groups created by a person with an owner ID starting with the characters "COM" will be listed.

The LMREFRESH Command

The LMREFRESH command informs STROBE that a new license for a STROBE product such as a new feature has been enabled by the License Management System. You can use this command to activate the Feature without having to restart session manager. This command can only be issued from the operator's console as an MVS MODIFY command.

LMREFRESH	
-----------	--

The LOGCLEAR Command

The LOGCLEAR command deletes the contents of a non-GDG log data set.

CAUTION:

The LOGCLEAR command cannot be issued through a batch job.

LOGCLEAR	
----------	--

The LOGSWITCH Command

The LOGSWITCH command forces the allocation of a new GDG log data set before the current one reaches its threshold.

CAUTION:

The LOGSWITCH command cannot be issued through a batch job.

LOGSWITCH	
-----------	--

The RESYNC Command

The RESYNC command causes the state of a measurement request to be synchronized with the state of the request as recorded in the queue data set.

RESYNC	REQUEST NUMBER
--------	----------------

The SHOW Command

The SHOW command displays information about the STROBE system.

SHOW	(ABENDS CSASIZE ENQUEUE SVCS QUEUE LOG DEFAULTS)
------	--

ABENDS

The ABEND history of STROBE since the session manager started. (ADMIN privileges are required to use this parameter.)

CSASIZE

The amount of common storage area used by STROBE. (ADMIN privileges are required to use this parameter.)

ENQUEUE

The current enqueues held by STROBE. (ADMIN privileges are required to use this parameter.)

MEMBERS

Other members of the XCF group that contains STROBE as a member.

SVCS=(*nnn,nnn,...*)

The status of the SVCs listed. If you do not supply a list of numbers, the session manager displays the status of the SVC it is using. (ADMIN privileges are required to use this parameter.)

QUEUE

The status of the STROBE data sets. (ADMIN privileges are required to use this parameter.)

LOG

The contents of the STROBE log data set.

DEFAULTS

The STROBE default values supplied through the parameter data set and currently in storage. In addition, the session management facility displays

- current release of STROBE
- STROBE PTF level
- STROBE tape volume serial number

The SHOWPTF Command

The SHOWPTF command displays the PTF level of your STROBE system.

SHOWPTF	
---------	--

SHOWPTF (no parameters) displays the session management facility components.

SHOWPTF	{TYPE=ISPF rr jobname}
---------	------------------------

TYPE=ISPF

Displays the STROBE/ISPF components. It can be entered on the command line of any STROBE/ISPF panel. For example:

```
.SHOWPTF TYPE=ISPF
```

{rr|jobname}

Shows the measurement task components used for the measurement request, identified by request number or jobname.

The STATUS Command

The STATUS command displays information about the session manager.

STATUS	
--------	--

The SUBMIT Command

The SUBMIT command can be used with administrative privileges to submit groups owned by other STROBE users by supplying the GRPOWNER parameter.

SUBMIT	group,GRPOWNER=groupowner
--------	---------------------------

GROUP

The name of the request group to be submitted.

GRPOWNER

The owner of the request group being submitted.

The TERMINATE Command

The TERMINATE command terminates the session manager and cancels all session requesters, but leaves active measurements active. This command can only be issued from the operator's console as an MVS MODIFY command.

TERMINATE	
-----------	--

The TEST Command

The TEST command is a debugging command to use when testing the user-exit program. It displays a hex dump of the user-exit parameter list to SYSPRINT and to the STROBE log data set before and after the user exit is called.

TEST	{EXITON EXITOFF}
------	------------------

EXITON

Dumps the parameter list to SYSPRINT and the log data set.

EXITOFF

Cancels the production of dumps for subsequent requests.

Chapter 6.

Error Diagnosis and Trouble Reporting

This chapter briefly describes error messages, return codes, and ABENDs that STROBE produces. For detailed descriptions of the messages and codes, see *STROBE MVS Messages*. The chapter also explains what information is required if you need to report a problem with STROBE, and how to determine the release level of STROBE.

Error Messages

STROBE writes error messages to indicate exceptional conditions, displaying the message number and text in the following destinations:

- the log data set
- the session requestor SYSPRINT data set
- the user's terminal, if STROBE is communicating with an online user.

For a complete list of messages, see *STROBE MVS Messages*.

Return Codes

When the target program or subsystem terminates during a measurement session, STROBE returns the return code set by the target program (if it has terminated normally) or the completion code (if it has terminated abnormally).

If STROBE detects errors during indexing or creation of a Performance Profile, STROBE sets the return code and writes an error message to SYSPRINT.

For a complete list of return codes, see *STROBE MVS Messages*.

ABENDs

Measuring with STROBE does not affect the execution of the target program. A measured step terminates abnormally only when the user program itself has terminated abnormally.

When a measurement task module terminates abnormally, STROBE writes abend information to the sample data set, the log data set, and the SYSPRINT data set. The SHOW ABEND command allows users to monitor abends in the STROBE environment.

For a complete list of ABENDs, see *STROBE MVS Messages*.

Diagnostic Program

STRBDSPY is a diagnostic program that lists portions of a sample data set. If STROBE terminates abnormally, STRBDSPY writes information about the termination to the sample data set. Occasionally, additional information from STRBDSPY may help to explain unusual data in the STROBE Performance Profile.

You can use STRBDSPY to obtain a listing of the diagnostics. Because the listing of an entire sample data set may be quite large, STRBDSPY accepts a parameter specifying the portion of the data set to be printed. The parameter, enclosed in single quotation marks, specifies the starting sample number and the number of samples to print—two numbers separated by a comma. If you omit the parameter or do not use the required format, STRBDSPY uses a default of '1,2500'.

Use these sample job control statements for executing STRBDSPY and modify the values appropriately.

```
//DSPY      EXEC PGM=STRBDSPY,PARM='1,2500'
//STEPLIB   DD    DSN=stroblib,DISP=SHR
//STRSAMPL  DD    DSN=prefix.jobname.S001D001,DISP=SHR
//SYSPRINT  DD    SYSOUT=*
```

A procedure named STRBDSPY, which executes the program, accompanies the STROBE job control procedures in member PROCS of the STROBE control data set.

Reporting Problems to Technical Support

If you suspect a STROBE program or documentation problem, report the trouble to Technical Support. The information required from you depends on where the problem has occurred.

For measurement task and data collection errors, supply

- a log data set
- STRBDSPY listing or a copy of the sample data set on tape
- a dump, if one was produced
- Performance Profile, if one was produced

For session management errors, supply

- a log data set
- a dump, if one was produced

For STROBE/ISPF errors, supply

- a log data set
- a screen print of the STROBE/ISPF panel displaying the error message
- the option chosen and panel identification
- an account of the sequence of events leading up to the error
- complete error message numbers

For reporting errors, supply

- Performance Profile
- STRBDSPY listing or a copy of the sample data set on tape
- a dump, if one was produced

For indexing errors, supply

- Performance Profile
- dump, if one was produced
- compiler SYSPRINT listing showing source program and the accompanying map file listing


```

*      OUTPUT CONDITIONS                                00000431
*
*      ON EXIT, REGISTER 15 CONTAINS A RETURN CODE.    00000441
*
*      RETURN CODES (DECIMAL)                          00000451
*
*      0 - COMMAND WILL BE EXECUTED; ANY MESSAGE RETURNED TO 00000461
*           STROBE WILL BE ISSUED WITH A SEVERITY CODE OF 'I'. 00000471
*
*      1-4 - COMMAND WILL NOT BE EXECUTED; ANY MESSAGE RETURNED TO 00000481
*           STROBE WILL BE ISSUED WITH A SEVERITY CODE OF 'W'. 00000491
*
*      5-8 - COMMAND WILL NOT BE EXECUTED; ANY MESSAGE RETURNED TO 00000501
*           STROBE WILL BE ISSUED WITH A SEVERITY CODE OF 'E'. 00000511
*
*      9+ - COMMAND WILL NOT BE EXECUTED; ANY MESSAGE RETURNED TO 00000521
*           STROBE WILL BE ISSUED WITH A SEVERITY CODE OF 'S'. 00000531
*
*
*      00000541
*      00000541
*      00000541
*      00000551
*      00000551
*      00000561
*      00000571
*      00000581
*      00000591
*      00000601
*      00000611
*
*      ----WHAT THIS SAMPLE EXIT ACCOMPLISHES----- 00000621
*
*      THIS EXIT LIMITS AUTHORITY TO SCHEDULE MEASUREMENT SESSIONS 00000631
*      TO JOBS WITH A PROJECT CODE THAT MATCHES ANY IN A TABLE OF 00000641
*      ACCEPTABLE CODES. IT ALSO SUPPLIES THE TSO USERID AS THE 00000651
*      SAMPLE DATA SET NAME PREFIX WHEN THE USER MAKES THE REQUEST 00000661
*      FROM TSO AND DOES NOT SPECIFY A PREFIX. 00000671
*
*      00000681
*      00000691
*
*      THE PROJECT CODE OF THE JOB ISSUING THE REQUEST IS A FIVE- 00000701
*      CHARACTER STRING WITHIN THE JOBCARD ACCOUNTING FIELD. IT 00000711
*      BEGINS AT A DIFFERENT LOCATION WITHIN THIS FIELD FOR TSO 00000721
*      THAN FOR BATCH JOBS. 00000731
*
*      00000741
*
*      IF THE SEARCH FOR THE PROJECT CODE FAILS TO FIND A MATCH IN 00000751
*      THE TABLE, THEN THE EXIT REJECTS THE COMMAND AND RETURNS AN 00000761
*      ERROR MESSAGE. IF THE SEARCH SUCCEEDS, CONTROL RETURNS TO 00000771
*      STROBE TO ACCEPT THE COMMAND AND PROCEED NORMALLY. 00000781
*
*      00000791
*
*      ALTERNATIVELY, IF THIS EXIT IS CALLED G1700 00000761
*      AT THE TIME OF MEASUREMENT TASK INITIALIZATION, G1700 00000761
*      IT CAN BE USED TO EXCLUDE CERTAIN PROGRAMS G1700 00000761
*      FROM BEING MEASURED. THIS ROUTINE SHOULD BE USED WHEN G1700 00000761
*      AN ADD STEP=*ALL HAS BEEN USED TO ADD THE JOB TO THE G1700 00000761
*      MEASUREMENT QUEUE. (IN THIS CASE, PROGRAM NAME FIELD G1700 00000761
*      WOULD NOT OTHERWISE BE AVAILABLE). G1700 00000761
*      SEE ADDITIONAL NOTES SURROUNDING ROUTINE 'CXAC0600' G1700 00000771
*      AND TABLE 'PGMTABLE'. G1700 00000771
*
*      ----- 00000801
*
*      EJECT 00000811
*
*      ----- 00000821
*
*      R0      EQU  0      WORK REGISTER 00000831
*      R1      EQU  1      WORK REGISTER 00000841
*      R2      EQU  2      AVAILABLE 00000851
*      R3      EQU  3      WORK REG; POINTS TO JSCB G1700 00001971
*      RPARM   EQU  4      A(STROBE PARAMETER LIST) 00000871
*      RJCT    EQU  5      A(JOB CONTROL TABLE) 00000881
*      RACT    EQU  6      A(ACCOUNT CONTROL TABLE) 00000891
*      R7      EQU  7      AVAILABLE 00000901
*      R8      EQU  8      AVAILABLE 00000911
*      R9      EQU  9      USED IN CXAC0600 ROUTINE G1700 00000921
*      RTBL    EQU 10      A(PROJECT CODE TABLE) 00000931
*      R10     EQU 10      USED IN CXAC0600 ROUTINE G1700 00000941
*      R11     EQU 11      USED IN CXAC0600 ROUTINE G1700 00000941
*      RBASE   EQU 12      PROGRAM BASE 00000951
*      R13     EQU 13      A(SAVE AREA) 00000961
*      R14     EQU 14      WORK REGISTER 00000971
*      R15     EQU 15      WORK REGISTER 00000981
*
*      ----- 00000991
*
*      RC00    EQU  0      COMMAND IS TO BE ACCEPTED (WITH MESSAGE SEVERITY I) 00001001
*      RC04    EQU  4      COMMAND IS TO BE REJECTED (WITH MESSAGE SEVERITY W) 00001011

```

```

RC08 EQU 8 COMMAND IS TO BE REJECTED (WITH MESSAGE SEVERITY E) 00001021
RC12 EQU 12 COMMAND IS TO BE REJECTED (WITH MESSAGE SEVERITY S) 00001031
*----- 00001041
*-----G1700 00001471
* THIS MACRO IS USED TO BUILD THE PGMNAME TABLES G1700 00001481
*-----G1700 00001491
MACRO G1700 00001501
PGM &PGMNAME G1700 00001511
LCLA &JLEN G1700 00001521
AIF (T'&PGMNAME EQ 'O').ENDIT G1700 00001531
&JLEN SETA K'&PGMNAME G1700 00001541
AIF (&JLEN LT 9).LENOK G1700 00001551
MNOTE 8,'INVALID PGMNAME, NAME &PGMNAME IS &JLEN CHARACTERS' G1700 00001561
MEXIT G1700 00001571
.LENOK ANOP G1700 00001581
DC AL1(&JLEN-1),C'&PGMNAME' G1700 00001591
AGO .ENDJ G1700 00001601
.ENDIT ANOP G1700 00001611
DC X'FF' G1700 00001621
.ENDJ ANOP G1700 00001631
MEND G1700 00001641
EJECT G1700 00001651
STRBCXAC CSECT 00001061
STRBCXAC AMODE 31 G2100 00001061
STRBCXAC RMODE ANY G2100 00001061
USING STRBCXAC,R15 00001071
SAVE (14,12) SAVE ENTRY REGS 00001081
B CXAC0100 BRANCH AROUND SAVE AREA 00001091
CXACNAME DC C'STRBCXAC' MODULE NAME 00001101
CXACDATE DC C'&SYSDATE' ASSEMBLY DATE 00001111
CXACTIME DC C'&SYSTIME' ASSEMBLY TIME 00001121
CXACSAVE DC 18F'0' SAVE AREA 00001131
SPACE 00001141
CXAC0100 DS OH 00001151
DROP R15 00001161
LR RBASE,R15 PROGRAM BASE ADDR 00001171
USING STRBCXAC,RBASE 00001181
LR RPARM,R1 A(PLIST) 00001191
USING PRMCXAC,RPARM 00001201
LR R2,R13 SAVE CALLER'S SVEAREA PTR G2100 00001211
LA R1,ENDWKAR-WORKAREA LOAD LENGTH FOR GETMAIN G2100 00001211
GETMAIN RC,LV=(R1) OBTAIN STORAGE FOR WORKAR G2100 00001211
LR R13,R1 POINT R13 TO SAVEAREA G2100 00001221
USING WORKAREA,R13 PROVIDE ADDRESSABILITY G2100 00001231
ST R13,8(0,R2) CHAIN FORWARD G2100 00001241
ST R2,4(0,R13) CHAIN BACK G2100 00001241
SPACE 00001251
***** 00001261
* ENSURE THAT ONLY ADD, CHANGE AND DELETE COMMANDS ARE EXAMINED. 00001271
***** 00001281
L R1,PRMCMD A(COMMAND) 00001291
CLI 0(R1),PRMADD IF 'ADD' COMMAND 00001301
BE CXAC0150 THEN CONTINUE 00001311
CLI 0(R1),PRMCHG IF 'CHANGE' COMMAND 00001321
BE CXAC0150 THEN CONTINUE 00001331
CLI 0(R1),PRMDEL IF 'DELETE' COMMAND 00001341
BE CXAC0150 THEN CONTINUE 00001351
CLI 0(R1),PRMNTI IF MEASUREMNT TASK INIT G1700 00001341
BE CXAC0600 THEN CHECK PROGRAM NAME G1700 00001351
B CXACRC00 EXIT FOR ALL OTHER COMMANDS 00001361
SPACE 00001371
***** 00001381
* ENSURE THAT ONLY TSO AND BATCH JOBS ARE EXAMINED. 00001391
***** 00001401
CXAC0150 DS OH 00001411
L R1,PRMCALL A(CALLER'S MODE) 00001421
CLI 0(R1),PRMTSO IF CALLED FROM 'TSO' 00001431
BE CXAC0200 THEN CONTINUE 00001441
CLI 0(R1),PRMBATCH IF CALLED FROM 'BATCH' 00001451

```

```

BE      CXAC0220          THEN CONTINUE          00001461
B       CXACRC00          EXIT FOR ALL OTHER MODES 00001471
SPACE                                     00001481
***** 00001491
* OBTAIN THE MODE-DEPENDENT OFFSET TO THE JOB ACCOUNTING INFORMATION 00001501
* 00001511
* FOR TSO THE ACCOUNTING FIELD IS FORMATTED AS FOLLOWS: 00001521
* 00001531
* ACTACCNT DS  OCL144 00001541
* +0 DC AL1(LENGTH OF FIRST (ONLY) JOB ACCT FIELD) 00001551
* +1 DC XL2 -- THESE TWO BYTES NOT USED BY EXIT 00001561
* +3 DC XL5'PROJECT CODE' 00001571
* 00001581
* FOR BATCH THE ACCOUNTING FIELD IS FORMATTED AS FOLLOWS: 00001591
* 00001601
* ACTACCNT DS  OCL144 00001611
* +0 DC AL1(LENGTH OF FIRST JOB ACCT FIELD) 00001621
* +1 DC XL2 -- THESE TWO BYTES NOT USED BY EXIT 00001631
* +3 DC AL1(LENGTH OF SECOND JOB ACCT FIELD) 00001641
* +4 DC XL5'PROJECT CODE' 00001651
* 00001661
***** 00001671
CXAC0200 DS  OH 00001681
        LA R15,ACTACCNT+3-IEFAACTB TSO PROJ CODE FIELD OFFSET 00001691
        B CXAC0300 REJOIN COMMON CODE 00001701
SPACE 00001711
CXAC0220 DS  OH 00001721
        LA R15,ACTACCNT+4-IEFAACTB BATCH PROJ CODE FIELD OFFSET 00001731
SPACE 00001741
***** 00001751
* ---OBTAIN JCT AND ACT ADDRESSES--- 00001761
***** 00001771
CXAC0300 DS  OH 00001781
        L R1,PSATOLD-PSA A(TCB) 00001791
        USING TCB,R1 00001801
        L R1,TCBJSCB A(JSCB) 00001811
        USING IEZJSCB,R1 00001821
        L R1,JSCBACT A(ACTIVE JSCB) 00001831
        L RJCT,JSCBJCT A(JCT PREFIX) 00001841
        DROP R1 00001851
        LA RJCT,16(,RJCT) A(JCT DATA AREA) 00001861
        USING INJMJCT,RJCT 00001871
        XR RACT,RACT CLEAR 00001881
        ICM RACT,B'0111',JCTACTAD A(ACT PREFIX) 00001891
        LA RACT,16(,RACT) A(JCT DATA AREA) 00001901
* USING IEFAACTB,RACT NOT USED; DEL; AVOID DIAG G1700 00001911
SPACE 00001921
***** 00001931
* SEARCH PROJECT CODE TABLE USING THIS PROJECT CODE 00001941
***** 00001951
        LA R15,0(R15,RACT) A(PROJECT CODE) IN ACCT'G FIELD 00001961
        LA RTBL,PCTABLE A(TABLE OF VALID PROJECT CODES) 00001971
        LA R0,PCT#ENT #PROJECT CODES TO BE SEARCHED 00001981
SPACE 00001991
CXAC0400 DS  OH 0002001
        CLC 0(L'PCTABLE,R15),0(RTBL) IF PROJECT CODE FOUND 0002011
        BE CXAC0500 THEN ACCEPT CMD+CHECK DSNAME 0002021
        LA RTBL,L'PCTABLE(,RTBL) ELSE A(NEXT TABLE ENTRY) 0002031
        BCT R0,CXAC0400 CONTINUE UNTIL E-0-T 0002041
SPACE 0002051
***** 0002061
* PROJECT CODE NOT FOUND - CREATE 'COMMAND REJECTED' MESSAGE 0002071
***** 0002081
        MVC BLDMSGAR,MSG0001 MOVE MESSAGE TO WORKAREA G2100 0002091
* MOVE INVALID PROJ CODE TO MESSAGE G2100 0002091
        MVC BLDMSGAR+MSG0001C-MSG0001(L'MSG0001C),0(R15) G2100 0002091
        LA R15,BLDMSGAR-WORKAREA DISPLACEMENT G2100 0002101
        AR R15,R13 ADD WORKAREA POINTER G2100 0002101
        L R1,PRMMSG A(STROBE MSG ADDR PARM) G2100 0002111

```

```

          ST   R15,0(0,R1)          SET WITH A(MESSAGE)          G2100 00002121
          LA   R15,RC08             INFORM STROBE TO REJECT COMMAND 00002131
          B    CXACRTN              RETURN TO STROBE                00002141
          SPACE                     00002151
*****
* SUPPLY DSNNAME PREFIX FOR TSO
*****
CXAC0500 DS   0H                    00002191
          L    R1,PRMCLL             A(CALLER'S MODE)                00002201
          CLI  0(R1),PRMTSO          IF MODE NE 'TSO'              00002211
          BNE  CXACRC00              THEN EXIT                       00002221
          L    R1,PRMDSN             A(DSNNAME PREFIX)              00002231
          CLI  0(R1),X'40'           IF ALREADY SUPPLIED           00002241
          BNE  CXACRC00              THEN DON'T MODIFY              00002251
          USING INJMJCT,RJCT                    G1700 00001871
          MVC  0(L'JCTJNAME-1,R1),JCTJNAME TSO USER ID -> PREFIX 00002261
          SPACE                     00002271
CXACRC00 DS   0H                    00002281
          LA   R15,RC00             INFORM STROBE TO ACCEPT COMMAND 00002291
          SPACE                     00002301
*****
* RETURN TO STROBE
*****
CXACRTN  DS   0H                    00002341
          L    R8,4(0,R13)           SAVE CALLER'S SAVEAREA AD G2100
          LTR  R15,R15               TEST RET.CODE                 G2100
          BNZ  EXITNOW               DO NOT ISSUE FREEMAIN          G2100
          LA   R9,ENDWKAR-WORKAREA   LOAD LENGTH                G2100
          FREEMAIN RU,LV=(R9),A=(R13) FREE STORAGE                G2100
          LA   R15,RC00              RESTORE RET.CODE           G2100
EXITNOW  DS   0H                    00002351
          LR   R13,R8                A(CALLER'S SAVE AREA)        G2100
          RETURN (14,12),,RC=(15)   RETURN TO CALLER          00002361
          EJECT                      00002371
CXAC0600 DS   0H                    G1700 00002371
*****
***                                     G1700 00002631
***                                     G1700 00002641
*** ON INITIALIZATION, GET PROGRAM NAME OF THIS STEP G1700 00002651
*** SO WE CAN PREVENT CERTAIN PROGRAMS FROM BEING MEASURED G1700 00002651
***                                     G1700 00002661
*****
* CHECK TO SEE IF THE TABLE OF PROGRAM NAMES TO BE EXCLUDED HAS G1700 00002691
* ANY ENTRIES. G1700 00002691
          LA   R1,ENDPGMTB-PGMTABLE G1700 00002701
          CH   R1,=H'1'              TBL LEN 0 OR 1 FOR 'FF' ?G1700 00002691
          BNH  CXACRC00              RETURN CONTROL, EMPTY TBL G1700 00002691
          L    R1,PSATOLD-PSA        LOAD USER TCB ADDRESS     G1700 00002691
          USING TCB,R1                PROVIDE ADDRESSIBILITY    G1700 00002701
          L    R3,TCBJSCB            LOAD JSCB ADDRESS         G1700 00002711
          USING IEZJSCB,R3           PROVIDE ADDRESSIBILITY    G1700 00002701
          L    R9,=A(PGMTABLE)       LOAD PROGRAM TABLE ADDR G1700 00002711
          L    R8,=A(ENDPGMTB)       POINT PAST PROGRAM TABLE G1700 00002711
          XR   R11,R11               CLEAR REGISTER (LA LATER) G1700 00002711
TESTPGM  DS   0H                    G1700
          CLI  0(R9),X'FF'           TABLE END SENTINEL?        G1700 00002721
          BE   CXACRC00              YES, ALL DONE                 G1700 00002721
          IC   R11,0(0,R9)           GET LENGTH CODE OF ENTRY G1700 00002711
          EX   R11,COMPARE           EXCLUDE THIS PROG NAME?  G1700 00002721
          BE   PGMNOGO              YES, EXCLUDE THIS PGM          G1700 00002721
* NO, CONTINUE THRU TABLE UNTIL END TABLE G1700 00002721
          LA   R9,2(R11,R9)          BUMP UP TO NEXT ENTRY     G1700 00002721
          CR   R9,R8                 AT END?                   G1700 00002721
          BNH  TESTPGM              NO, CONTINUE LOOKUP          G1700 00002721
          B    CXACRC00              YES, RETURN CONTROL           G1700 00002721
* PUT OUT ERROR MESSAGE AND RETURN G1700 00002741
PGMNOGO DS   0H                    G1700 00002721
          LA   R15,BLDMSGAR-WORKAREA DISPL WITHIN WORKAREA        G1700 00002101
          AR   R15,R13               ADD WORKAREA POINTER       G1700 00002101
          MVC  0(MSG0007P+2,R15),MSG0007 MVE MSG TEXT TO WORKAR G1700 00003171

```

```

MVC MSG0007X-MSG0007(8,R15),JSCBPGMN MOVE PGM NAME G1700 00003171
L R1,PRMMSG G1700 00003191
ST R15,0(0,R1) STORE THE MESSAGE ADDR G1700 00003201
* LET RETURN CODE STOP PROCESSING G1700 00002741
LA R15,RC08 SET RETURN CODE G1700 00003211
B CXACRTN G1700 00003221
COMPARE CLC JSCBPGMN(0),1(R9) EXCLUDE THIS PGM? G1700 00002721
EJECT G1700 00003231
***** G1700 00005991
*** G1700 00006001
*** ---TABLE OF PROGRAM NAMES TO BE EXCLUDED --- G1700 00006011
*** G1700 00006021
*** USE THE PGM MACRO TO DEFINE PROGRAM NAMES TO BE EXCLUDED. G1700 00006031
*** PGMNAMES MAY BE FROM 1 TO 8 CHARACTERS LONG, AND ONLY THE G1700 00006041
*** CHARACTERS CODED WILL BE CHECKED AGAINST THE ACTUAL G1700 00006051
*** PROGRAM NAME TO BE EXECUTED IN THIS STEP. G1700 00006051
*** FOR EXAMPLE, TO EXCLUDE PROGRAM NAMES STARTING WITH 'ABC' G1700 00006051
*** CODE ONLY THE 'ABC'. G1700 00006051
*** TO END THE TABLE, CODE PGM MACRO WITH NO OPERANDS. G1700 00006051
*** G1700 00006001
***** G1700 00006141
PGMTABLE DS OD G1700 00006151
* EMPTY TABLE IN SAMPLE; TO ACTIVATE, CHANGE ASTERISK TO SPACE G1700 00006161
* IN COLUMN ONE AND CHANGE 'SAMPL' TO YOUR PROGRAM NAME DATA. G1700 00006161
* PGM SAMPL EXAMPLE EXCLUDE PGMNAMES SAMPLXXX G1700 00006161
* PGM IEFB EXAMPLE EXCLUDE PGMNAMES IEFBXXX G1700 00006161
PGM 00006221
ENDPGMTB EQU * G1700 00006221
***** 00002381
* ---TABLE OF VALID PROJECT CODES--- 00002391
***** 00002401
PCTABLE DC CL5'STA80' 00002411
DC CL5'STB80' 00002421
DC CL5'STC80' 00002431
DC CL5'STD80' 00002441
DC CL5'STE80' 00002451
DC CL5'STF80' 00002461
DC CL5'STI80' 00002471
DC CL5'STL80' 00002481
DC CL5'STM80' 00002491
DC CL5'STP80' 00002501
PCT#ENT EQU (*-PCTABLE)/L'PCTABLE 00002511
SPACE 00002521
***** 00002531
* 'COMMAND REJECTED' MESSAGE 00002541
***** 00002551
MSG0001 DS 0XL122 MESSAGE AREA 00002561
DC AL2(MSG0001P) MESSAGE LENGTH 00002571
MSG0001T DS 0CL120 MESSAGE TEXT AREA 00002581
DC CL8'STRBCXAC' NAME OF EXIT PROGRAM 00002591
DC CL1' ' FILLER 00002601
MSG0001C DC CL5' ' PROJECT CODE 00002611
DC CL1' ' FILLER 00002621
DC C'==> PROJECT CODE NOT VALID FOR STROBE RELEASE 8.0 JOBS.' 00002631
MSG0001P EQU *-MSG0001T LENGTH OF MSG PARM 00002641
MSG0007 DS 0XL122 G1700 00007011
DC AL2(MSG0007P) G1700 00007021
MSG0007T DS 0CL120 G1700 00007031
DC CL9'STRBCXAC ' G1700 00007041
MSG0007C DC C'ACCESS DENIED, PROGRAM EXCLUDED -- XXXXXXXX ' G1700 00007051
ORG MSG0007C+35 ' G1700 00007051
MSG0007X DS CL8 ' G1700 00007051
MSG0007P EQU *-MSG0007T G1700 00007061
LTOrg 00002651
EJECT 00002661
***** 00002671
* ---STROBE EXIT PARAMETER LIST DSECT--- 00002681
* 00002691
* PRMJOB CV 00002701

```

```

* THE ADDRESS OF AN EIGHT-BYTE CHARACTER FIELD CONTAINING THE 00002711
* JOBNAME OR TSO USERID THAT SUBMITTED THE CURRENT COMMAND. 00002721
* THIS FIELD IS BLANK DURING INITIALIZATION (WHEN PRMCMD=I). 00002731
* 00002741
* PRMJOB 00002751
* THE ADDRESS OF AN EIGHT-BYTE CHARACTER FIELD CONTAINING THE 00002761
* TARGET JOBNAME SPECIFIED IN THE REQUEST. 00002771
* 00002781
* PRMCALL 00002791
* THE ADDRESS OF A ONE-BYTE CHARACTER FIELD CONTAINING THE 00002801
* CALLING MODE OF THE USER, DENOTING WHETHER THE REQUEST IS 00002811
* BEING MADE FROM BATCH, TSO, OR A STARTED TASK. 00002821
* 00002831
* NOTE: THIS FIELD IS NOW FILLED IN DURING INITIALIZATION 00002841
* (WHEN PRMCMD=I) AND IT WILL BE THE MODE OF THE JOB BEING 00002851
* MEASURED. 00002861
* 00002871
* CL1'J' BATCH 00002881
* CL1'T' TSO 00002891
* CL1'S' STARTED TASK 00002901
* 00002911
* PRMCMD 00002921
* THE ADDRESS OF A ONE-BYTE CHARACTER FIELD CONTAINING THE 00002931
* COMMAND PROCESSED BY STROBE UPON THIS CALL TO THE EXIT 00002941
* PROGRAM. BECAUSE SOME FIELDS MAY BE BLANK DURING 00002951
* INITIALIZATION, YOU SHOULD CHECK THIS PARAMETER BEFORE 00002961
* MANIPULATING OF THE OTHERS. 00002971
* 00002981
* CL1'A' ADD 00002991
* CL1'C' CHANGE 00003001
* CL1'S' SEND 00003011
* CL1'L' LIST 00003021
* CL1'D' DELETE 00003031
* CL1'I' MEASUREMENT TASK INITIALIZATION 00003041
* 00003051
* PRMDSN (MAY BE MODIFIED) 00003061
* THE ADDRESS OF A 44-BYTE CHARACTER FIELD CONTAINING EITHER 00003071
* THE SAMPLE DATA SET PREFIX SPECIFIED ON THE ADD OR CHANGE 00003081
* OPERATION, OR BLANKS IF NO DATA SET NAME WAS SPECIFIED. THE 00003091
* PREFIX CAN CONTAIN AS MANY AS (34 - JOBLEN) BYTES, WHERE JOBLEN 00003101
* IS THE LENGTH OF THE JOBNAME. WHEN MODIFYING THIS FIELD, THE USER 00003111
* EXIT PROGRAM SHOULD CLEAR IT FIRST, THEN PASS A FULLY-QUALIFIED 00003121
* SAMPLE DATA SET PREFIX PADDED WITH BLANKS. 00003131
* 00003141
* PRMUNIT (MAY BE MODIFIED) 00003151
* THE ADDRESS OF AN EIGHT-BYTE CHARACTER FIELD CONTAINING THE 00003161
* UNIT NAME OF THE SAMPLE DATA SET OR BLANKS. 00003171
* WHEN MODIFYING THIS FIELD, THE USER EXIT PROGRAM SHOULD CLEAR IT 00003181
* FIRST, THEN PASS UNIT NAME PADDED WITH BLANKS. 00003191
* 00003201
* PRMVOL (MAY BE MODIFIED) 00003211
* THE ADDRESS OF A SIX-BYTE CHARACTER FIELD CONTAINING THE 00003221
* VOLUME SERIAL NUMBER OF THE SAMPLE DATA SET OR BLANKS. 00003231
* WHEN MODIFYING THIS FIELD, THE USER EXIT PROGRAM SHOULD CLEAR IT 00003241
* FIRST, THEN PASS VOLUME SERIAL NUMBER PADDED WITH BLANKS. 00003251
* 00003261
* PRMUSER (MAY BE MODIFIED) 00003271
* THE ADDRESS OF AN EIGHT-BYTE USER FIELD, INITIALLY SET TO 00003281
* BINARY ZEROS, WHICH YOUR EXIT PROGRAM CAN MODIFY ON THE 00003291
* FIRST CALL FOR THE MEASUREMENT REQUEST. STROBE RETURNS 00003301
* IT AS MODIFIED ON ALL SUBSEQUENT CALLS. 00003311
* 00003321
* YOU CAN USE THIS FIELD TO CONTAIN THE ADDRESS OF A DATA 00003331
* AREA. YOU MUST ENSURE, HOWEVER, THAT THE REFERENCED DATA 00003341
* AREA IS IN COMMONLY ADDRESSABLE STORAGE, BECAUSE SUBSEQUENT 00003351
* CALLS WILL COME FROM DIFFERENT ADDRESS SPACES OR MAY BE GONE 00003361
* AFTER AN IPL. AFTER USING IT, YOU CAN FREE THE STORAGE FOR 00003371
* THE DATA AREA DURING THE MEASUREMENT TASK INITIALIZATION CALL. 00003381
* 00003391

```

```

* ONE POSSIBLE WAY TO USE THIS FIELD IS TO VERIFY THAT A 00003401
* STROBE USER MEASURES ONLY THE JOBS OWNED BY THAT USER. 00003411
* WHEN STROBE CALLS THE EXIT FOR AN ADD OPERATION, THE EXIT 00003421
* CAN SAVE AN ACCOUNTING FIELD. THEN, WHEN THE TARGET JOB 00003431
* STEP BECOMES ACTIVE AND STROBE AGAIN CALLS THE EXIT 00003441
* PROGRAM, THE EXIT PROGRAM CAN CHECK THE ACCOUNTING FIELD OF 00003451
* THE TARGET JOB AND REJECT THE REQUEST IF THE USER HAS NO 00003461
* AUTHORITY TO MEASURE THE TARGET JOB STEP. 00003471
* 00003481
* PRMSG (MAY BE MODIFIED) 00003491
* THE ADDRESS OF A FULLWORD CONTAINING THE ADDRESS OF A 00003501
* MESSAGE FOR THE USER. BEGIN THE MESSAGE WITH A TWO-BYTE 00003511
* LENGTH FIELD, FOLLOWED BY UP TO 120 BYTES OF TEXT. MESSAGES 00003521
* GREATER THAN 120 BYTES WILL BE TRUNCATED. THE SEVERITY CODE 00003531
* ASSOCIATED WITH THE MESSAGE DEPENDS ON THE RETURN CODE 00003541
* ISSUED BY THE EXIT WHEN IT RETURNS TO STROBE. 00003551
* 00003561
* PRMACEID 00003571
* THE ADDRESS OF AN EIGHT-BYTE CHARACTER FIELD CONTAINING THE 00003581
* RACF/ACF2/TOPSECRET USERID ASSOCIATED WITH THE TSO/BATCH JOB 00003591
* THAT ADDED THE REQUEST. THIS FIELD IS RETRIEVED FROM THE 00003601
* ACEE CONTROL BLOCK WHEN THE REQUEST IS ADDED. THIS FIELD MAY 00003611
* BE LOW VALUES IF THE REQUEST WAS ADDED FROM A SYSTEM TASK. 00003621
* 00003631
* PRMREQ 00003641
* THE ADDRESS OF A ONE-BYTE CHARACTER FIELD CONTAINING THE 00003651
* TYPE OF REQUEST 00003661
* 00003671
* CL1'Q' ADD QUEUED REQUEST 00003681
* CL1'A' ADD ACTIVE REQUEST 00003691
* 00003701
* 00003711
* PRMSTEP 00003721
* THE ADDRESS OF THE LIST OF STEP NAMES SPECIFIED IN THE 00003731
* STEP PARAMETER OF THE REQUEST. THE FIRST EIGHT BYTES 00003741
* OF THIS LIST IS A HEADER IN THE FORM 00003751
* 00003761
* - NUMBER OF ENTRIES (TWO BYTES) 00003771
* - (LENGTH OF AN ENTRY (TWO BYTES) 00003781
* - TOTAL LENGTH OF THE LIST (FOUR BYTES) 00003791
* 00003801
* THE HEADER IS IMMEDIATELY FOLLOWED BY THE FIRST ENTRY 00003811
* EACH ENTRY IS A SIXTEEN BYTE BINARY FIELD CONTAINING 00003821
* A STEP NUMBER OF INTEREST. 00003831
* THE HEADER SHOWS ZERO ENTRIES IF THIS PARAMETER WAS NOT CODED 00003841
* NOTE: THIS FIELD IS FILLED IN DURING INITIALIZATION 00003851
* (WHEN PRMCMD=I) 00003861
* 00003871
* PRMSNUM 00003881
* THE ADDRESS OF THE LIST OF STEP NUMBERS SPECIFIED IN THE 00003891
* NUMBER PARAMETER OF THE REQUEST. THE FIRST EIGHT BYTES 00003901
* OF THIS LIST IS A HEADER IN THE FORM 00003911
* 00003921
* - NUMBER OF ENTRIES (TWO BYTES) 00003931
* - LENGTH OF AN ENTRY (TWO BYTES) 00003941
* - TOTAL LENGTH OF THE LIST (FOUR BYTES) 00003951
* THE HEADER IS IMMEDIATELY FOLLOWED BY THE FIRST ENTRY 00003961
* EACH ENTRY IS A FOUR BYTE BINARY FIELD CONTAINING 00003971
* THE STEP NUMBER. 00003981
* THE HEADER SHOWS ZERO ENTRIES IF THIS PARAMETER WAS NOT CODED 00003991
* NOTE: THIS FIELD IS FILLED IN DURING INITIALIZATION 00004001
* (WHEN PRMCMD=I) 00004011
* 00004021
* PRMPROG 00004031
* THE ADDRESS OF THE LIST OF THE PROGRAM NAMES SPECIFIED IN THE 00004041
* PROGRAM PARAMETER OF THE REQUEST. THE FIRST EIGHT BYTES 00004051
* OF THIS LIST IS A HEADER IN THE FORM 00004061
* 00004071
* - NUMBER OF ENTRIES (TWO BYTES) 00004081

```

```

*           - LENGTH OF AN ENTRY (TWO BYTES)                                00004091
*           - TOTAL LENGTH OF THE LIST (FOUR BYTES)                        00004101
*                                                                 00004111
*           THE HEADER IS IMMEDIATELY FOLLOWED BY THE FIRST ENTRY.          00004121
*           EACH ENTRY IS A EIGH CHARACTER PROGRAM NAME                    00004131
*           THE HEADER SHOWS ZERO ENTRIES IF THIS PARAMETER WAS NOT CODED  00004141
*           NOTE: THIS FIELD IS FILLED IN DURING INITIALIZATION            00004151
*           (WHEN PRMCMD=I)                                                00004161
*                                                                 00004171
*           PRMSAMP      (MAY BE MODIFIED)                                  00004181
*           THE ADDRESS OF A FOUR-BYTE BINARY FIELD CONTAINING THE         00004191
*           SAMPLE COUNT AS SPECIFIED ON THE REQUEST OR THE DEFAULT IF     00004201
*           THE PARAMETER WAS NOT CODED.                                    00004211
*                                                                 00004221
*           PRMLIMIT    (MAY BE MODIFIED)                                  00004231
*           THE ADDRESS OF A FOUR-BYTE BINARY FIELD CONTAINING THE LIMIT   00004241
*           COUNT AS SPECIFIED ON THE REQUEST OR THE DEFAULT IF THE        00004251
*           PARAMETER WAS NOT CODED.                                        00004261
*                                                                 00004271
*           PRMLTYP     (MAY BE MODIFIED)                                  00004281
*           THE ADDRESS OF A ONE-BYTE CHARACTER FIELD CONTAINING THE       00004291
*           TYPE OF ACTION STROBE IS TO TAKE WHEN THE LIMIT COUNT HAS     00004301
*           BEEN REACHED.                                                 00004311
*                                                                 00004321
*           CL1'Q'      QUIT                                               00004331
*           CL1'S'      STOP                                               00004341
*           CL1'C'      NOLIMIT OR CONTINUE                                00004351
*                                                                 00004361
*           PRMGOMIN    (MAY BE MODIFIED)                                  00004371
*           THE ADDRESS OF A FOUR-BYTE BINARY FIELD CONTAINING THE GOMIN   00004381
*           VALUE AS SPECIFIED ON THE REQUEST OR THE DEFAULT IF THE        00004391
*           PARAMETER WAS NOT CODED.                                        00004401
*                                                                 00004411
*           PRMRETNQ    (MAY BE MODIFIED)                                  00004421
*           THE ADDRESS OF A FOUR-BYTE BINARY FIELD CONTAINING THE RETAIN  00004431
*           QUEUED VALUE AS SPECIFIED ON THE REQUEST OR THE DEFAULT IF THE  00004441
*           PARAMETER WAS NOT CODED.                                        00004451
*                                                                 00004461
*           PRMRETNC    (MAY BE MODIFIED)                                  00004471
*           THE ADDRESS OF A FOUR-BYTE BINARY FIELD CONTAINING THE RETAIN  00004481
*           COMPLETE VALUE AS SPECIFIED ON THE REQUEST OR THE DEFAULT IF THE 00004491
*           PARAMETER WAS NOT CODED.                                        00004501
*                                                                 00004511
*           PRMSUFF     (MAY BE MODIFIED)                                  00004521
*           THE ADDRESS OF A EIGHT-BYTE CHARACTER FIELD CONTAINING         00004531
*           EITHER THE SAMPLE DATA SET NAME SUFFIX AS SPECIFIED IN THE    00004541
*           DEFAULTS MODULE, OR BLANKS IF NO DATA SET NAME WAS           00004551
*           SPECIFIED. WHEN MODIFYING THIS FIELD, THE USER EXIT          00004561
*           PROGRAM SHOULD CLEAR IT FIRST, THEN PASS A SUFFIX VALUE       00004571
*           PADDED WITH BLANKS.                                           00004581
*                                                                 00004591
*           PRMSCLAS    (MAY BE MODIFIED)                                  00004601
*           THE ADDRESS OF A EIGHT-BYTE CHARACTER FIELD CONTAINING THE     00004611
*           STORCLAS VALUE AS SPECIFIED IN THE REQUEST OR THE DEFAULT IF THE 00004621
*           PARAMETER WAS NOT CODED.                                        00004631
*           WHEN MODIFYING THIS FIELD, THE USER EXIT PROGRAM SHOULD CLEAR  00004641
*           IT FIRST, THEN PASS A STORCLAS VALUE PADDED WITH BLANKS.      00004651
*                                                                 00004661
*           PRMMCLAS    (MAY BE MODIFIED)                                  00004671
*           THE ADDRESS OF A EIGHT-BYTE CHARACTER FIELD CONTAINING THE     00004681
*           MGMTCLAS VALUE AS SPECIFIED IN THE REQUEST OR THE DEFAULT IF THE 00004691
*           PARAMETER WAS NOT CODED.                                        00004701
*           WHEN MODIFYING THIS FIELD, THE USER EXIT PROGRAM SHOULD CLEAR  00004711
*           IT FIRST, THEN PASS A MGMTCLAS VALUE PADDED WITH BLANKS.      00004721
*                                                                 00004731
*           PRMDCLAS    (MAY BE MODIFIED)                                  00004741
*           THE ADDRESS OF A EIGHT-BYTE CHARACTER FIELD CONTAINING THE     00004751
*           DATACLAS VALUE AS SPECIFIED IN THE REQUEST OR THE DEFAULT IF THE 00004761
*           PARAMETER WAS NOT CODED.                                        00004771

```



```

PRMRETNC DS      A          A(RETAINC)                00006161
PRMSUFF  DS      A          A(DSN SUFFIX)            00006171
PRMSCLAS DS      A          A(STORCLAS)              00006181
PRMMCLAS DS      A          A(MGMTCLAS)              00006191
PRMDCLAS DS      A          A(DATACLAS)              00006201
PRMSTRDD DS      A          A(STRBDD)                00006211
PRMSYSL  DS      A          A(SYSTEM LIST)           G1501 00006221
PRMEXCL  DS      A          A(EXCLUDE LIST)          G1501 00006231
PRMSTDTE DS      A          A(STARTTIME DATE LIST)   G1501 00006241
PRMSTTIM DS      A          A(STARTTIME TIME LIST)   G1501 00006251
PRMSTRTY DS      A          A(STARTTIME RETRY COUNT) G1501 00006261
PRMSTINC DS      A          A(STARTTIME INCREMENT)   G1501 00006271
PRMPJCL  DS      A          A(JCL)                  G1601 00006281
PRMPRD   DS      A          A(PROFILE RELATED DATA) G1601 00006291
PRMLLEN  EQU     *-PRMCXAC  LENGTH OF STROBE EXIT PLIST 00006301
EJECT                                         00006311
*****
* WORKAREA DSECT                               G2100 00006321
***** G2100 00006331
***** G2100 00006341
WORKAREA DSECT          GETMAINED AREA FOR REENTERANCY G2100
SAVEAREA DS      18F    SAVE AREA                    G2100
BLDMSGAR DS      CL122  MESSAGE TEXT AREA             G2100
ENDWKAR  EQU     *      LABEL USED TO GENERATE LEN OF AREA G2100
EJECT                                         00006311
***** 00006321
* MVS DSECTS                               00006331
***** 00006341
* CODE EXTRA 'DSECT' STATEMENT, AS IEFAJCTB EXPANSION DOES G1700 00006351
* NOT GENERATE ONE.                               G1700 00006351
DSECT                                         G1700 00006351
PRINT NOGEN                                   00006351
IEFAJCTB          MAPS JCT, ACT                00006361
IEZJSCB           MAPS JSCB                    00006371
IHAPSA           MAPS PSA                      00006381
IKJTCTB         MAPS TCB                      00006391
END                                                       00006401

```

MVS System IEFU85 Exit

If you are using the STROBE Unix System Services Feature, you have to enable the recording of SMF type 92 records. The following is a sample MVS system IEFU85 exit program which can be used to avoid recording excess SMF type 92 record data from UNIX System Services. An example of this JCL can be found in the member IEFU85.

```

*-----
*SAMPLE IEFU85 SMF EXIT
*
* THIS EXIT WILL DISABLE THE RECORDING OF OPENMVS FILE SYSTEM
* ACTIVITY TYPE 92 RECORDS.
* THIS CODE IS FUNCTIONAL, HOWEVER IT IS AN EXAMPLE ONLY. NOTE THAT
* AN FRR RECOVERY ROUTINE IS NOT ESTABLISHED, (IBM RECOMMENDS ESTABLISHING
* AN FRR- REFER TO THE MVS/ESA INSTALLATION EXITS MANUAL).
* ALSO, A SAVE AREA IS NOT PROVIDED IN THIS EXAMPLE. (AVOIDING A STORAGE
* OBTAIN/FREEMAIN ON EACH INVOCATION.)
*
* ENTRY POINTS -
* IEFU85
* INPUT
* REGISTER 1 POINTS TO THE ADDRESS OF THE SMF RECORD TO BE PROCESSED.
* OUTPUT
* REGISTER 15:
* 0- WRITE THE RECORD TO THE SMF DATA SET
* 4- DO NOT WRITE TO THE RECORD

```

```

* EXTERNAL REFERENCES - NONE
* EXITS,NORMAL - RETURN TO CALLER
* EXITS,ERROR - NONE
* ATTRIBUTES - STANDARD
* CHARACTER CODE DEPENDENCY - NONE

```

```

*-----
          EJECT
R0      EQU 0
R1      EQU 1
R2      EQU 2
R3      EQU 3
R4      EQU 4
R5      EQU 5
R6      EQU 6
R7      EQU 7
R8      EQU 8
R9      EQU 9
R10     EQU 10
R11     EQU 11
R12     EQU 12
R12     EQU 13
R14     EQU 14
R15     EQU 15
          EJECT
IEFU85  CSECT
          USING *,R15
          B      START          BRANCH AROUND MOD NAME
          DC     C'IEFU85 '      IDENT FOR DUMPS
          DS     0H              FORCE HALF WORD ALIGNMENT
START    STM    R14,R12,12(R13)  STANDARD
          DROP   R15
          LR     R12, R15        ENTRY LINKAGE
          USING IEFU85, R12
*
          LA    R15,0            ASSUME RC 0
          L     R2,0(,R1)        -> SMF RECORD
          CLI   5(R2),92 -      TYPE 92 RECORD?
          BNE   EXIT            NO
EXIT4    LA    R15,4            YES, SET RC TO 4 DO NOT WRITE
*
EXIT     DS     0H
          RETURN (14,12),RC=(15)
          LTORG
          END

```


Appendix B.

STROBE Data Sets

This appendix contains information about the STROBE data sets.

The Control Data Set

The contents of the control data set, which is created from the distribution tape, are shown below.

ALLOCASM	Allocate the STROBE group data sets and create a procedure to start STROBE (when not maintaining multiple versions)
ALLOCATE	Allocate the STROBE queue data sets and create a procedure to start STROBE (when not maintaining multiple versions)
ALLOCTST	Allocate the STROBE data sets and create a procedure to start STROBE (when maintaining multiple versions)
ALLOTSTA	Allocate the STROBE group data sets and create a procedure to start STROBE (when maintaining multiple versions)
APMPOWER	CLIST to bring up APMpower
COPY	Copy the STROBE data sets to a link list data set (when not maintaining multiple versions)
COPYTEST	Copy the STROBE data sets to a link list (when maintaining multiple versions)
GENSAM	Copy the sample data set to tape
HSTPDLOG	Print, delete, reallocate STROBE history log data set
HSTRELOG	Reorganize STROBE history data set and alternate index
INDEXCOB	Test COBOL indexing
INDEXFOR	Test FORTRAN indexing
INDEXPLI	Test PL/I indexing
INSTALL	Install the STROBE system on an unauthorized library
MAINTNCE	Unload STROBE fix tapes
MIGRATEG	Execute migration utility (group data sets)
MIGRATEQ	Execute migration utility (queue data sets)
MIGRTSTG	Execute test migration (group data sets)
MIGRTSTQ	Execute test migration (queue data sets)
MODCSECT	Assemble and link edit the skeleton module descriptor program
MODLINK	Link edit the customized STROBE module descriptor program
OSTARREC	Data set packaging utility
OSTARXMT	Data set packaging utility
PANEPTF	STROBE CA-Panexec Feature instructions

PROCS	Use cataloged procedures to execute STROBE functions (installation is optional)
REPORT	Produce a STROBE Performance Profile from the STROBE test job
SPFINSTL	Install the STROBE/ISPF common libraries
SQLAFRPL	Replace SQL and DBRM objects in the existing STROBE.CNTL (used for maintenance)
SQLAFVER	Verify installation of the SQL Analysis Feature
SQLINSTL	Install the SQL Analysis Feature
SQLOBJ nn	Link edit control statements for STRANANA where nn equals the DB2 Version you are using
SRCLIST	TSO CLIST to execute the session requester
STRAN Ann	object module for SQL Analysis Feature where nn indicates the version of DB2
STRAI Ann	object module used by iSTROBE and the SQL Analysis Feature where nn indicates the version of DB2
STRANBND	SQL Analysis Feature DB2 BIND JCL
STRANGNT	SQL Analysis Feature GRANT statement for STRANANA
STRANRDJ	Link edit the SQL Analysis Feature threshold module
STRANRDM	Macro to create a table of threshold values for SQL Analysis Feature
STRANSMI	Contains RUN and END statements that are input to STRANSMP
STRANSMP	Creates a sample data set that can be used to test SQL Analysis Feature
STRANV24	DBRM for SQL Analysis Feature verification program
STRAN1 nn	DBRM for SQL Analysis Feature where nn indicates the version of DB2
STRAN2 nn	DBRM for SQL Analysis Feature where nn indicates the version of DB2
STRBANCE	Default source ADABAS/NATURAL Configurations module
STRBANCJ	ADABAS/NATURAL Configurations module assemble and link JCL
STRBANCM	Macro to generate ADABAS/NATURAL Configuration module
STRBCLOG	Administer the log data set
STRBCTSJ	CICS System Task Table
STRBCTSK	CICS System Task Table
STRBPKG	Data set packaging utility
STROBE	CLIST to execute STROBE/ISPF
TESTAPF	Test a new version of STROBE from an APF-authorized library
TESTNEW	Test a new version of STROBE (when maintaining multiple versions)
TESTSTR	Test STROBE (when not maintaining multiple versions)
TRNCLEAN	STROBE training lab exercise clean-up job
TRNDSN	STROBE training lab exercise allocation job
TRNPDS	STROBE training lab exercise allocation job
TRNPGMXX	STROBE training lab exercise training program JCL/COBOL
TRNPLIXX	STROBE training lab exercise JCL/PLI
TRN2COB1	STROBE training lab exercise object code/COBOL1

TRN2COB2	STROBE training lab exercise object code/COBOL2
TRN2PLI	STROBE training lab exercise object code/PLI

Object Modules

Object modules and associated linkage editor control statements for the STROBE system are in members L1OBJ and L2OBJ.

L1OBJ

Member L1OBJ contains object modules and linkage editor control statements to produce the following load modules.

STRANVER	SQL Analysis Feature installation verification
STRBCMIG	STROBE data set migration utility
STRBDSPY	Sample data set display program
STRBSAM1	Test program
STRBSNAM	Module mapping table
STREBALE	Sample data set decode routine
STREBALO	Sample data set decode routine
STREMODS	Module descriptor program
STREUTIL	Support subroutines for indexers and reporter
STROBEDS	STROBE/ISPF display format Profile editor
STROBEDT	STROBE reporter
STRUPREP	Part of the STROBE/ISPF installation utility

Member L1OBJ also contains object modules and linkage editor control statements for any of the STROBE features available at your installations.

STRETHRS	Threshold functions for Advanced Session Management
STRXCBX	CA-Optimizer indexer
STRXASF	Assembler indexer
STRXC	C indexer
STRXCBF	COBOL indexer (aliases STRFCBU, STRFCBF)
STRXFTG	FORTRAN G indexer
STRXFTH	FORTRAN H and VS indexer
STRXNAT	ADABAS/NATURAL indexer
STRXPLF	PL/I indexer
STRXCSP	CSP indexer
STRXIEF	IEF indexer
STRHEXPT	STROBE history export routine
STRHPRNT	STROBE history print routine
STRHUTIL	STROBE history utility routine
STRHCOMP	STROBE history comparison routine

L2OBJ

Member L2OBJ contains object modules for re-entrant STROBE measurement modules.

OSTAREDC	Data set packaging utility
STRBADLI	DLI attributor
STRBALS	Linkage stack attributor
STRBASVC	Attribute SVC usage
STRBATTR	Attribution driver
STRBCLOG	Log clear program
STRBCOPT	Sampling control with STROBE
STRBCSM	STROBE session manager
STRBCSR	STROBE session requester
STRBCINI	Measurement request initialization
STRBCLPA	STROBE run-time support (for extended CSA)
STRBCLPB	STROBE run-time support (in CSA)
STRBCPH0	Measurement initialization
STRBCRB	End-of-job step processing
STRBCTXR	CICS transaction profiling data collector
STRBCXAC	Dummy user exit
STRBGMIM	STROBE generalized capture module
STRBISPF	STROBE/ISPF
STRBPC1	PC routines
STRBSLMX	STROBE License Management System module
STRBSPFM	APMpower MVS component
STRBTUSR	Instrumentation
STRBUT24	Measurement task support modules below 16MB line
STRBVPH2	Measurement task main driver and CPU sampler
STRBVPH3	Sample data set write module
STRBMAPS	Module mapping
STRBVM11	Standalone data collector module
STRBVM12	Standalone data collector module
STRBVPH1	Measurement task initialization

Member L2OBJ also contains the object modules for the STROBE feature programs that are linked as re-entrant.

STRBAUSI	AutoSTROBE user interface exit
STRBACTR	AutoSTROBE user termination exit
STRBAUTO	AutoSTROBE main control module
STRBAEVT	AutoSTROBE event control module
STRBAMON	AutoSTROBE monitor control module

STRBAUHM	AutoSTROBE data set management module
STRBACSP	CSP attributor for STROBE CSP Feature
STRBACSQ	MQSeries attributor
STRBADFH	CICS attributor
STRBADBC	CICS/DBCTL data collector
STRBADML	CA-IDMS DML statement attributor
STRBAIEF	COOL:GEN attributor
STRBACPP	C/C++/XPLink attributor
STRBAE15	AutoSTROBE SMF Candidate utility
STRBAE35	AutoSTROBE SMF Candidate utility
STRBALJA	Java attributor table
STRBJDC1	"Just In Time" JITted Java attributor
STRBJDC2	Interpreted Java attributor
STRBJVMA	Java Virtual Machine attributor
STRBALC	C attributor
STRBALCB	COBOL attributor
STRBALPL	PL/I attributor
STRBALS	Attribution for language library routines
STRBANCM	STROBE ADABAS/NATURAL configurations module
STRBASSA	IMS DLI attribution call utility
STRBAVSM	VSAM attributor
STRBDCSQ	MQSeries data collector
STRBDC nm	CICS Sub-Data Collector (where nm is the CICS release number)
STRBDF nm	CICS Performance Supplement (where nm is the CICS release number)
STRBDFHS	CICS data collector program
STRBDFSD	STROBE IMS Attributor
STRBDFSM	STROBE IMS Feature
STRBDFSR	STROBE IMS Feature
STRBDM10	CA-IDMS R10.2 batch DML statement attributor
STRBDM12	CA-IDMS R12.0 batch DML statement attributor
STRBDSNA	DBRM attributor for the STROBE DB2 Feature
STRBDSNS	DB2 module
STRBIDMS	STROBE CA-IDMS Feature
STRBID10	STROBE CA-IDMS release 10.2 data collector
STRBID12	STROBE CA-IDMS release 12.0 data collector
STRBMSAS	Measurement services module
STRBNATL	ADABAS/NATURAL data collector
STRBPANE	CA-Panexec Feature
STRBPR10	CA-IDMS 10.2 attributor

STRBPR12	CA-IDMS 12.0 attributor
STRBTPX	CICS transaction profiling data collector
STRH	STROBE history manager
STRXADS	STROBE ADS/O Dialog indexer

Other Members

The control data set also contains test program source, STROBE/ISPF, and STROBE/PFM libraries, and documentation.

CICSTRAN	CICS System Task Table
HINITREC	Default record values for STROBE measurement data history
IEFU85	Sample system exit for handling SMF 92 records
SMEXIT2	STROBE user exit source
SMEXIT	STROBE user exit source for sites with security packages
SAMFORT	Sample FORTRAN program
SAMPLEW	Sample COBOL program
SAMPLI	Sample PL/I program
STRBPASS	Macro to generate the STROBE password

STROBE/ISPF Common Libraries

The control data set also contains the following libraries.

STRMLIB	STROBE/ISPF message library
STRMSGS	STROBE message text library
STRPLIB	STROBE/ISPF panel library
STRSLIB	STROBE/ISPF skeleton library

The Parameter Data Set

The STROBE parameter data set is a member of a fixed block, 80-byte, partitioned data set that you specify during STROBE installation. It provides installation-dependent values that govern the function of STROBE at your site. The parameter values are described in Table 3-3 on page 3-33.

The Log Data Set

The log data set contains an historical record of all activity initiated through the session manager. It can help you track the activity and the history of individual requests. The log also provides a valuable record in the event of problems with STROBE.

All messages in the log data set begin with the date and time, the MVS system name, the jobname of the issuer of the STROBE operation, and, when available, the request number and jobname associated with the request.

Data Set Characteristics

The log data set is a sequential data set that must reside on a direct access device. The DCB specifies a record format of VBA. The record length is 137. The block size can range from 141 to the maximum size for the device (default 3024). If you do not want to use the defaults, specify the block size when you allocate the data set. (Always select a moderate block size, because large block sizes result in wasted space.) A primary allocation of 10 tracks with a 10-track secondary allocation is sufficient for most installations.

The log data set is always allocated with a disposition of share to allow multiple users of STROBE to be active at the same time. It should never reside on a system pack.

When STROBE creates its environment, it acquires the data set name of the log from the SYSLOG data definition statement supplied in the startup procedure.

CAUTION:

Do not rename or reallocate the log data set while the STROBE environment is active. STROBE uses the data set name for all subsequent invocations.

Figure B-1 shows the attributes you specify for log data sets.

Figure B-1. Log Data Set Characteristics

```
SYSLOG DD DSN=STROBE.V2R5M0.LOG.G0001V00,
          DISP=(,CATLG)
          DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3024),
          SPACE=(TRK,(10,10)),
          UNIT=SYSDA
```

Processing Considerations

During installation, you can define the log as a generation data set or a conventional data set. It is recommended that you define the log as a generation data set.

As a Generation Data Set

If the log is a generation data set, STROBE automatically allocates and switches to a new data set when the log becomes full. STROBE/ISPF provides a facility to identify, read, and print from previous generations of the log data set.

As a Non-Generation Data Set

If the log is a conventional data set, the session manager tests the first LOGWARN value on the STROBE parameter file and issues a message when the file accumulates the specified percentages of allocation. Each time the log exceeds the second value of LOGWARN, the session manager issues another warning message. When the log becomes full, STROBE will not execute until you issue the LOGCLEAR command or invoke the STROBE Log Utility to clear the log. See “Managing the Log Data Set” on page 5-6. The STROBE Log Utility (STRBCLOG) allows you to copy and clear all records from the current log data set to a backup data set. The utility enqueues exclusively on the log resource so that STROBE cannot write to the log when it is being cleared.

The Queue Data Set

The queue data set enables the session manager to preserve and restore requests across initial program loads of MVS. The session manager maintains the current highest request

number and the log history in the queue data set. After an IPL, STROBE starts numbering new requests by increasing the saved request number by one.

The session manager updates the queue data set each time a request is added or modified. If an IPL occurs before a request has been processed, the request is recreated when the STROBE environment is established after the IPL. The session manager removes requests from the queue only when they exceed their retention period or when a user deletes them.

Data Set Characteristics

The queue data set is a VSAM key-sequenced data set. If you need to reallocate the data set, do so when the STROBE environment does not exist. The data set is defined at install time through the IBM utility IDCAMS with the attributes shown in Figure B-2.

Figure B-2. Queue Data Set Characteristics

```

DEFINE CLUSTER -
(
  NAME(STROBE.V2R5M0.QUEUE) -
  CYL(2 2) -
  VOL(VOLSER) -
  BUFFERSPACE(112640) -
  INDEXED -
  CISZ(22528) -
  KEYS(54 0) -
  RECORDSIZE(1044 22528) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)

```

CAUTION:

Do not reallocate or rename the queue data set while the STROBE environment is active.

Multisystem Considerations

In a multisystem environment, the queue data set is a central repository for all STROBE measurement requests in the XCF-group. Each instance of STROBE maintains its own copy (the local work queue) of the requests contained in the queue data set. It keeps the local work queue current by reading the new or changed information in the queue data set and updating the local work queue whenever it receives an XCF signal indicating that the queue data set has changed.

Note: With a shared AutoSTROBE file, the queue data set needs to be shared by the same system.

Processing Considerations

When you initialize the STROBE environment, STROBE verifies that the data set allocated to the data definition name SYSQUEUE exists and is a VSAM data set. STROBE terminates if it determines that the data set does not exist or is not a VSAM data set.

Use the same queue data set for all systems that share DASD in a complex. Each session manager operates independently of the others through a unique identifier for each system. "Planning for Enqueues" on page 2-5 discusses serialization requirements.

Size Considerations

In planning the size requirements for the queue data set, allow 3K for each request plus 1K for growth. At a control interval size of 22K, plan on

- 5 requests per control interval
- 10 requests per track
- 150 requests per cylinder

The System Message Data Set

The system message data set contains any messages or information associated with requests on the queue data set.

Data Set Characteristics

The system message data set is a VSAM key-sequenced data set. It is defined at installation through the IBM utility IDCAMS with the attributes shown in Figure B-3.

Figure B-3. System Message Data Set Characteristics

```

DEFINE CLUSTER -
(
  NAME(STROBE.V2R5M0.SYSMSG) -
  CYL(1 0) -
  VOL(VOLSER) -
  INDEXED -
  CISZ(22528) -
  KEYS(54 0) -
  RECORDSIZE(1044 22528) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)

```

Processing Considerations

STROBE processing periodically removes data that is no longer needed from the message data set.

Size Considerations

In planning the size requirements for the system message data set, allow 3K for each request plus 1K for growth. At a control interval size of 22K, plan on

- 5 requests per control interval
- 10 requests per track
- 150 requests per cylinder

The Request Group Data Set

The STROBE Advanced Session Management Feature requires the use of the request group data set. The request group data set enables the session manager to preserve the contents of STROBE Advanced Session Management request groups.

The session manager updates the request group data set each time a request group is added or modified. The session manager removes requests from the request group only when a user or the STROBE administrator deletes them.

Data Set Characteristics

The request group data set is a VSAM key-sequenced data set. The data set is defined at install time through the IBM utility IDCAMS with the attributes shown in Figure B-4.

Figure B-4. Request Group Data Set Characteristics

```

DEFINE CLUSTER -
(
  NAME(STROBE.V2R5M0.GROUP) -
  CYL(2 2) -
  VOL(VOLSER) -
  INDEXED -
  CISZ(22528) -
  KEYS(54 0) -
  RECORDSIZE(1044 22528) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)

```

CAUTION:

Do not reallocate or rename the request group data set while the STROBE environment is active.

Processing Considerations

When the STROBE environment is created during the initialization of the session manager, the session manager verifies that the data set allocated to the data definition name SYSGROUP exists and is a VSAM data set. If it determines that the data set does not exist or is not a VSAM data set, the session manager terminates.

Use the same request group data set for all systems that share DASD in a complex. (A unique identifier for each system allows each session manager to operate independently.)

Serialization Requirements

If you place the request group data set on a system pack, first change the reserve for the data set to an enqueue through your resource serialization package. Refer to "Planning for Enqueues" on page 2-5.

Size Considerations

The size of the request group data set depends on the number of requests, including scheduled requests. A multisystem environment increases the number of requests in the data set. In planning the size requirements for the request group data set, allow 2K for each request plus 1K for growth.

The History Data Set

The STROBE Advanced Session Management Feature optionally uses the history data set. The history data set is the repository for measurement session history, and enables the analysis of performance trends for particular job steps, DBRMs, or transactions.

The history manager updates the history data set each time a measurement session history record is created. The history manager removes measurement session history records from the history data set only when a user or the STROBE administrator deletes them.

History Data Set Characteristics

The history data set is a VSAM key-sequenced data set. The data set is defined at install time through the IBM utility IDCAMS with the attributes shown in Figure B-5.

Figure B-5. History Data set Characteristics

```

DEFINE CLUSTER -
(
  NAME(STROBE.STRHIST) -
  CYL(10 5) -
  VOL(VOLSER) -
  INDEXED -
  BUFFERSPACE(112640) -
  CISZ(22528) -
  KEYS(62 18) -
  RECORDSIZE(300 380) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)

```

Serialization Requirements

If you place the history data sets on a system pack, first change the reserve for the data set to an enqueue through your resource serialization package. Refer to “Planning for Enqueues” on page 2-5.

Size Considerations

The size of the history data set depends on the number of measurement session history records being created. In planning the size requirements for the history data set, allow 1K for every two measurement session history records, plus 1K for growth.

The History Log Data Set

The history log data set creates the history log.

Figure B-6. History Log Data Set Characteristics

```

STRHLOG DD DSN=STROBE.STRHIST.LOG,
           SPACE=(TRK,(100,20)),DISP=(,CATLG,DELETE),
           DCB=(RECFM=VBA,LRECL=137,BLKSIZE=4096),
           UNIT=SYSDA,VOL=SER=

```

The AutoSTROBE Data Set

The AutoSTROBE data set is a VSAM key-sequenced data set used by the STROBE Advanced Session Management Feature that enables you to use its AutoSTROBE function. It is used to store the basis data that STROBE collects and is used for the establishment of thresholds that initiates AutoSTROBE measurement.

AutoSTROBE Data Set Characteristics

The AutoSTROBE data set is a VSAM key-sequenced data set. The data set is defined at install time through the IBM utility IDCAMS with the attributes shown in Figure B-7.

Figure B-7. Data Set Characteristics

```

*
DEFINE CLUSTER -
(
  NAME(STROBE.STRAHIST) -
  CYL(10 5) -
  VOL(VOLSER) -
  INDEXED -
  BUFFERSPACE(112640) -
  CISZ(22528) -
  KEYS(64 0) -
  RECORDSIZE(67 500) -
  SHAREOPTIONS(4 3) -
  UNIQUE -
)
/

```

Serialization Requirements

If you place the AutoSTROBE data sets on a system pack, first change the reserve for the data set to an enqueue through your resource serialization package. Refer to “Planning for Enqueues” on page 2-5.

Size Considerations

The size required for the AutoSTROBE data set depends on the number of programs for which monitoring data is to be retained. This calculation should accommodate the archived basis data for programs that are not currently being monitored because their AutoSTROBE request has been deleted. Allow 3.7K for each such program.

The Sample Data Set

The sample data set is a variable-length blocked sequential data set with a block size between 8192 and 32,760 bytes and a single logical record per block. When a measurement session is completed, STROBE closes the data set, and then reopens it for update. The first record, containing sample counts and starting and ending times, is updated and rewritten. For this reason, always assign the sample data set to a direct access device.

The STROBE measurement task dynamically allocates a sample data set in the target program's address space. The data definition name is determined by the operating system. Each sample data set is assigned a data set name of the following form:

prefix.jobname.SsssDddd

where

- *prefix* is a period-delimited string of data set name qualifiers that you can specify at installation as a system default. The user can override this prefix with the DSNAME operand on the ADD operation. You can also write a STROBE user-exit program to change or verify this prefix.
- *jobname* is the name of the job in which the measured program executes.
- *sss* is a sequence number of the request measuring a given job—S001 for the first request, S002 for the second, and so on. This number is incremented only when STROBE encounters a new request for a job with the same jobname as one that it previously sampled, and the prior request is still stored by STROBE.
- *ddd* is a sequence number of the data set within the request. It also denotes the number of measurement sessions conducted under the request.

STROBE reuses existing data sets that meet the established naming conventions. Otherwise, it allocates new data sets.

Sample Data Set Characteristics

Figure B-8 shows the attributes you specify for sample data sets.

Figure B-8. Sample Data Set Characteristics

```
ddname DD DSN=prefix.jobname.SsssDddd,
        DISP=(NEW,[user specified or CATLG]),
        DCB=(RECFM=VB,LRECL=8188,
            BLKSIZE=[installation-specified or 8192],
            BUFNO=[installation-specified or 2])
        SPACE=(BLKSIZE,(1024,3072)),
        UNIT=[user- or installation-specified],
        VOL=SER=[user- or installation-specified or null],
```

You can specify the data set name prefix, UNIT, VOL, BLKSIZE, BUFNO, and EXPDT as system defaults when you install STROBE. The ADD and CHANGE commands can override the data set name prefix, UNIT, and VOL.

Disposition of the Sample Data Set

You can specify the data set disposition on the ADD and CHANGE commands. The default is CATLG. You can supply a JOBCAT or STEPCAT data definition statement in the job control statement for the measured job and also specify a disposition of CATLG. In this case, the catalog entry for the sample data set appears in the catalog specified on the JOBCAT or STEPCAT statement. When you execute the reporter program to produce the STROBE Performance Profile, specify the same JOBCAT or STEPCAT statement you specified on the measured job or specify UNIT and VOL on the sample data set data definition statement.

The Map Data Set

Map data sets produced by the indexers contain data gathered from the SYSPRINT output of the assembler or a compiler. This data relates programmer-assigned statement names to absolute hexadecimal addresses, relative to the base of a control section. When the reporter produces the STROBE Performance Profile, it correlates this data with the data from a sample data set. Map data sets can be sequential or can be members of a previously defined partitioned data set. They have fixed-length, 36-character records, with a block size of 1692 bytes.

The STROBE Performance Profile

The Performance Profile is a collection of reports that details the activity of an online subsystem or batch processing program. Users can specify the Performance Profile as either wide or narrow.

Wide

The Performance Profile, which is normally assigned to a SYSOUT data set, has fixed-length, 121-character records with an ANSI carriage control character in the first position of each record. The block size is 2904 bytes.

Narrow

The Performance Profile has fixed-length, 80-character records.

Temporary Data Sets

The reporter, indexers, APMpower MVS component, STROBE/ISPF and the AutoSTROBE SMF Candidate Utility use temporary, sequential data sets.

The data definition names for the temporary data sets used by the STROBE indexers are STRUT1 and STRUT2. In addition, some of the indexers use sort files.

The reporter uses a temporary data set named SYSUT1. The DCB characteristics needed for these data sets are contained in the DCBs in the programs. Each is assigned a temporary data set name and has the implied disposition of (NEW,DELETE) in the STROBE procedures that call the programs. The reporter also requires sort work data sets.

STROBE/ISPF creates work data sets for foreground profiles with LRECL=80 for narrow format, and LRECL=121 for wide format. These data sets are located on a unit and volume you select at installation time. For creating Performance Profiles from measurement session history, STROBE/ISPF uses a temporary data set with the following naming convention:

tsoid.2ndlvlqualifier.HISTnnnn

- *tsoid* is the TSO user ID of the person creating the Performance Profile
- *2ndlvlqualifier* is the second level qualifier as defined in the STROBE defaults module, STRBDFLT
- *HISTnnnn* is the sequence number of the temporary file, where *nnnn* is a number from 0001 to 9999.

This file is used to temporarily store the Performance Profile parameters that were used when the measurement session history record was created. When you request to re-create the Performance Profile from one of the measurement session history panels, STROBE/ISPF uses this temporary file as input to the reporter. The file is deleted after it is used.

The temporary data set naming convention for the AutoSTROBE SMF Candidate Utility is as follows:

tsoid.2ndlvlqualifier#.CANDIDAT

- *tsoid* is the TSO user ID of the person creating the output file to be read by the Candidate Utility
- *2ndlvlqualifier* is the second level qualifier as defined in the STROBE defaults module, STRBDFLT followed by the logical terminal number (#).
- CANDIDAT indicates to the utility that this file is the output file to use.

Appendix C.

Symbolic Parameters for STROBE Job Control Procedures

The following symbolic parameters are used with the job control procedures that are supplied with the STROBE system.

Symbolic Parameter	Meaning of the Parameter	Procedures that Use the Parameter
ADAPARM=	ADABAS parameter file name	STROXN STROXNAT
AINDX	ADABAS higher level index name	
ASMBLR=ASMA90	Name of assembler (defaults to high level assembler)	STROAF SROBEAF
ATTR=	Option to specify attribution detail	STROB STROBE STROBEzz
CALLIB=	Linked-in subroutine library	STROBEC5
CBASE=BASE	C library stubs	STROBECI
CLINK=LINK	C dynamic library	
COMBASE=COMBASE	Common run-time stubs	
COMHD='EDC.'	Prefix for common library	
COMLINK=COMLINK	Common dynamic run-time library	
COMVER='V2R1M0'	Version of C common library	
COMPARM=NODECK	Compilation options	STROzz
COMPARM='NODECK,NO XREF,NOSEQ,OPT,DUMP,NOADV'		STROCX2
COMPREG=	Compilation region size	STROBEzz STROzz
COMPRES=	Modules and control sections to be reported on as pseudo-procedures; supply in format: ((xxx=.yyyyyy),aaa=.bbb))	STROE STROBE STROXE STROBEzz
CPARM=	Compiler option	STROCI STROBECI
CPROG='EDCCOMP'	Name of IBM C/370 compiler	
CVER='V2R1M0'	Version of C compiler	
DASD=	Option to condense cylinders that use less than a given percentage of total DASD in the DASD Usage by Cylinder report	STROE STROBE STROXE STROBEzz

Symbolic Parameter	Meaning of the Parameter	Procedures that Use the Parameter
DAUNIT=SYSDA	Unit assignment for all direct access devices	STROE STROBE STROcx STROBEzz STROX STROXE STROXN STROXNAT
DETAIL=	Modules and control sections normally reported on as pseudo-procedures, for which detailed execution reports are to be produced. If more than one is specified, a parenthesized, comma-delimited list is used.	STROE STROBE STROXE STROBEzz
DISPNO=2500	Number of samples to print from sample data set	STRBDSPY
EDCHDRS=' .SEDCHDRS'	C system headers	STROCI
EDCMSGS=' .SEDCMSGS'	C compiler message	STROBECI
ENTRY=MAIN	Module entry point	STROBECS
ENV=STD	Environment in which module runs	
FORTPGM=IEKAA00	Name of FORTRAN compiler (defaults to FORTRAN H compiler)	STROFH STROBEFH
GODSN='&SYSLMOD'	Step library containing target program load module	STROB STROBE
GOPROG=USERPROG	Target program load module name	STROB STROBE STROBEzz
GOREG=	Target program region size	
GOUNIT=	Target program step library unit assignment	STROB STROBE
GOVOL=	Target program step library for volume serial number	
INFILE=	Input files for compiler	STROBECI
INDXPRM=	Indexer parameters	STROXNAT
LANG=EDCMSGE	Message language	STROCI STROBECI
LINEMAX=	Number of lines per page for the STROBE Performance Profile min 45, max 80	STROE STROBE STROXE STROBEzz
LISTDSN='LISTDSN'	Name for SYSPRINT data sets	STROX STROXN STROXE
LKED='XREF,LIST,LET'	Linkage editor options	STROBEzz, except for STROBECI and STROBECS
LPARM='LIST,MAP'	Linkage editor options	STROBECI STROBECS
LSTATYP=TRK	Listing allocation type	STROXN
LSTPRIM=3	Listing primary allocation	
LISTSEC=1	Listing secondary allocation	
MACLIB=	Macro libraries	STROCS STROBECS
MAPDISP=CATLG	Map data set disposition	STROX STROzz STROXNAT

Symbolic Parameter	Meaning of the Parameter	Procedures that Use the Parameter
MAPDSN=NULLFILE	Map data set name	STROE STROBE
MAPDSN='STRMAP'		STROX STROzz
MAPDSN=MAPDSN		STROXNAT
MAPPGM=STRXCBF	Name of indexer program	STROX STROXE
MAPLABL='(1,SL)'	Map data set LABEL parameter	STROX
MAPUNIT=SYSDA	Map data set unit assignment	STROCX
MAPVOL=	Map data set volume serial	STROXNAT
NATBNUC=	NATURAL batch nucleus name	STROXN
NATREG=2048K	NATURAL region size	STROXNAT
NINDX=	NATURAL higher level index name	
NPARAM=	NATURAL run parameter	
NOATTR=	Option to suppress attribution reports (DB2 only)	STROE STROBE
NODASD=	Option to suppress DASD Usage by Cylinder reports	STROXE STROBEzz
NOPROC=	Option to suppress Program Usage by Procedure report	
NOTRAN=	Option to suppress Transaction Usage report	
OBJLIB=	Object library	STROBECS
PL1PGM=IEMAA	Name of PL/I compiler (defaults to PL/I F compiler)	STROPF STROBEPF
PREF=SASC	Link-edited SYSLIB prefix	STROBECS
PREFIX=	Sample data set prefix (defined as prefix for your installation plus target job step name) Supply in format prefix.jobname	STROBE STROBEzz
PRTATYP=TRK	CMPRINT allocation type	STROXNAT
PRTCLAS='**'	SYSOUT print class	All procedures
PRTPRIM=3	CMPRINT primary allocation	STROXNAT
PRTSEC=1	CMPRINT secondary allocation	
PUBP=	Option to condense procedures that exhibit less than a given percentage of CPU activity in the Program Usage by Procedure report	STROE STROBE STROXE STROBEzz
REPNAME=	Performance Profile subtitle, up to 40 characters (defaults to program name)	STROE STROBE STROXE STROBEzz
RESLTN=	Report resolution	
RPTPARM=	Additional reported options not available through other symbolic parameters	STROE STROBE STROBEzz
SAMDISP=CATLG	Sample data set disposition	STROB
SAMDSN='STRSAM'	Sample data set name	STROE STROXE STRBDSPY

Symbolic Parameter	Meaning of the Parameter	Procedures that Use the Parameter
SAMUNIT=	Sample data set unit name	STRBDSPY
SAMPLABL=	Sample data set label	
SAMVOL=	Sample data set volume serial	
SORTSIZ=	Amount of main storage for COBOL sort	STROE STROBE STROXE STROBEzz
SORTUT=300	Space allocation for sort work data sets for reporter	STROE STROBE STROXE STROBEzz STROX
STARTNO=1	Starting sample number in sample data set	STRBDSPY
STRLIB='STROBLIB'	STROBE load module library name	STROE STROBE STRBDSPY STROzz STROX STROCE STROXNAT
SUFFIX=S001D001	Sample data set suffix	STROBE STROBEzz
SYSLIB=	C resident library	STROBECS
TRANLIB=	C transient library	
SYSUDMP=DUMMY	Dump data set	STROXNAT
UTBLOCK=300	Space allocation for work files	STROzz STROBEzz
VSCCHD='EDC.'	Prefix for system files	STROCI STROBECI
WAITLOC=	Option to show the location of wait in the Wait Time by Module report	STROE STROBE STROXE STROBEzz

Appendix D.

SQL Analysis Feature Threshold Set

The SQL Analysis Feature uses a set of values to determine which items to include in the Opportunities report. You can override some of these default values to satisfy requirements specific to a particular DB2 subsystem or application:

The following are the thresholds that you can change and their default values:

Table D-1. SQL Analysis Threshold Rules

Macro Variable	Default	Description
MNTBUTS	1	Upper limit for the number of tables defined in unsegmented (simple) tablespaces
SMTBRSZ	32	Lower limit for table row size (4K page)
LGTBRSZ	256	Upper limit for table row size
SMTSPSZ	4	Lower limit for tablespace page size (4K page)
LGTSPSZ	32	Lower limit for tablespace page size
SMTBPAG	7	Lower limit for number of used tablespace pages
MNTBROW	250000	Upper limit for number of table rows
MNTBPGS	10000	Upper limit for number of pages for a table defined in a nonpartitioned tablespace
MNINCNT	4	Upper limit for number of indexes defined for a table
MNINLVL	4	Upper limit for number of levels in an index
MNINLEF	1000	Upper limit for number of leaf pages in an index
INCLRAT	95	Lower limit for cluster ratio of a clustered index
MNCOLFN	2	Upper limit for number of column functions in an SQL statement
MNJNCNT	4	Upper limit for a number of joins in a SQL statement
MNTSFAR	2	Upper limit of percent of table pages relocated far from their original page
MNTSNER	5	Upper limit of percent of table pages relocated near their original page
MNINFAR+	2	Upper limit for percent of data rows located far from their optimal position for a clustering index
MNINNER	5	Upper limit for percent of data rows located near their optimal position for a clustering index

The SQL Analysis Feature provides these default values automatically. If you do want to change the defaults, you must edit and execute the STRANRDJ member shown in Figure D-1. It assembles and link edits member STRANRDM to create a module the SQL Analysis Feature can use.

Figure D-1. Assemble and Link Rule Threshold Set (Member STRANRDJ)

```

//ASM      EXEC   ASMHCL
//          PARM.C='XREF(SHORT),LOAD,OBJECT',
//
//PARM.L='AMODE=ANY,RMODE=24,LIST,XREF,MAP,NORENT'
//C.SYSLIB DD DSN=STROBE.CNTL,DISP=SHR
//C/SYSLIN DD *
           STRANRDM MNTBUTS=1,           X
           SMTBRSZ=32                    X
           LGTBRSZ=256,                  X
           SMTSPSZ=4,                    X
           LGTSPSZ=32,                   X
           SMTBPAG=7,                    X
           MNTBROW=250000,                X
           MNTBPGS=10000,                 X
           MNINCNT=4,                    X
           MNINLV=4,                     X
           MNINLEF=1000,                  X
           INCLRAT=95,                   X
           MNCOLFN=2,                    X
           MNTSFAR=2,                    X
           MNTSNER=5,                    X
           MNINFAR=2,                    X
           MNINNER=5,                    X
           END
/*
/**
//L.SYSLIN DD
//          DD *
           NAME MMMMMMM (R)
/**
//L.SYSMOD DD DSN=STROBELIB,DISP=SHR,UNIT=,SPACE=,DCB=

```

To create a set of threshold values:

1. Change the default threshold value in member STRANRDJ to the desired values; all values must be positive integers
2. Supply a **JOB** statement
3. Change **STROBE.CNTL** to the name of your unauthorized STROBE control data set
4. Change **STROBELIB** to the name of the STROBE unauthorized load library
5. Change **MMMMMMM** to the name of the load module containing your custom defaults
6. Make any additional changes to customize the job control statements for your installation.
7. Execute the job.

You can define multiple rule sets for use with different DB2 applications and subsystems. For example, for a large theoretical database, the following rules might apply:

- Since the number of tables in the database contain at least a million rows, the acceptable limit for a large table is a 1.5 million rows. Set **MNTBROW=1500000**.
- Because of number of rows, you want to limit the use of column functions to no more than a single function in any one SQL statement. Set **MNCOLFN=1**.
- Since several of frequently used tables have clustered indexes defined for them, you want to ensure the cluster ration for these indexes does not fall below 85 percent. Set **INCLRAT=85**.

After you create a rule threshold that includes these criteria, STRANRDM appears as shown in Figure D-2.

Figure D-2. Defining Rule Sets (Member STRANRDM)

```
STRANRDM  MNTBUTS=1,           X
           SMTBRSZ=32          X
           LGTBRSZ=256,        X
           SMTSPSZ=4,          X
           LGTSPSZ=32,         X
           SMTBPAG=7,          X
           MNTBROW=150000,     X
           MNTBPGS=10000,      X
           MNINCNT=4,          X
           MNINLVL=4,          X
           MNINLEF=1000,       X
           INCLRAT=85,         X
           MNCOLF=1,           X
           MNTSFAR=2,          X
           MNTSNER=5,          X
           MNINFAR=2,          X
           MNINNER=5,          X
END
```


Index

A

abend condition, data sets showing information, 6-1
 access filter interfaces supported, 4-2
 ADABAS/NATURAL Feature
 customizing configurations module, 3-5
 indexer. See also STRXNAT, B-3
 installation checklist, 2-21
 linking configurations module, 3-8
 STRBADAB, B-5
 ADABAS/NATURAL Feature, IMS requirements, 3-6
 ADD command, CA-Top Secret keywords, 4-5
 address spaces
 described, 1-3
 measured program's, 2-13
 session manager's, 2-13
 starting, 3-27
 stopping STROBE's, 5-2
 address spaces using VIO, TCAM or TSO
 how STROBE attaches measurement task, 2-16
 administrative commands, issuing, 5-1
 ALLOCASM member
 editing requirements, 3-22
 ALLOCATE, B-1
 ALLOCATE member
 editing requirements, 3-22, 3-27
 ALLOCTST, B-1
 ALLOCTST member
 editing requirements, 3-22, 3-27
 ALLOTSTA, B-1
 alternate index path, history data set, 3-23
 APMpower
 maintaining SQL Analysis Feature, 5-15
 testing MVS component, 3-44
 APMPOWER CLIST, B-1
 APMPOWER CLIST, 3-42
 APMPOWER member, 3-42
 APMpower PDF data set, planning for, 2-3
 archiving
 data sets, 2-6
 previous STROBE versions, 3-3
 test STROBE versions, 3-3
 ARCVID, meaning and value, 3-15
 Assembler Indexer, B-3
 attach filter, 1-5
 ATTFLTR, meaning and value, 3-15
 attribution driver, B-4
 attributor, reporting errors, 6-2
 authorization ID for SQL Analysis Feature, 2-10
 authorized libraries, STROBE component requirements, 2-3
 AUTOACTION, meaning and value, 3-15
 AUTOCANDIDATE, meaning and value, 3-15
 AUTOCANELAP, meaning and value, 3-15
 AUTOCANEXCP, meaning and value, 3-15
 AUTOCANFREQ, 3-15

AUTOCANFREQ, meaning and value, 3-15
 AUTODBMIN, meaning and value, 3-15
 AUTOELAPPERCNT, meaning and value, 3-15
 AUTOEVENT1, meaning and value, 3-15
 AUTOIOEVENT2, meaning and value, 3-15
 AUTOIOEVENT3, meaning and value, 3-15
 AUTOIOEVENT4, meaning and value, 3-15
 AUTOIOEVENT5, meaning and value, 3-15
 automatic Performance Profile, planning, 2-8
 AutoStrobe, 1-9, B-11
 AutoStrobe data set, 3-25
 Autostrobe data set, customizing, 3-25
 AutoStrobe data set, managing, 5-12
 AUTOSTROBE, meaning and value, 3-15
 AutoStrobe, starting and stopping through MSAS, 5-4
 AUTOTCBPERCNT, meaning and value, 3-15

B

BASELIN, meaning and value, 3-15
 batch job, using to start Strobe, 3-28
 BatchPipes support, 2-16
 BIND values for SQL Analysis Feature, 2-10
 BLKSIZE, meaning and value, 3-15
 BMC Data Accelerator, restrictions, 2-18
 BMC UNLOAD restrictions, 2-19
 BPXBATCH
 measuring activity, 2-11
 buffers, storage space required, 2-14
 BUFNO, meaning and value, 3-16

C

C Indexer, B-3
 CA-ACF2
 defining generic access rules, 4-6
 defining job access rules, 4-8
 defining session manager access rules, 4-8
 defining system access rules, 4-7
 syntax keyword definitions, 4-7
 CA-ACF2 restrictions, 2-17
 CA-InterTest restrictions, 2-17
 CA-Optimizer Indexer, STRXCBX, B-3
 CA-Panexec restrictions, 2-18
 CA-Top Secret
 defining access rules, 4-5
 defining job access security rules, 4-6
 defining session manager access rules, 4-6
 specific system access rules example, 4-6
 checkpoint/restart, MVS restriction, 2-16
 CHGGRP command, 5-19
 ELEMENTNUMBER parameter, 5-19
 GROUP parameter, 5-19
 GRPOWNER parameter, 5-19
 PARAMETER parameter, 5-19
 CICS
 attributor, B-5
 data collector program, B-5
 Performance Supplement, B-5
 system control block name, 3-6

CICS Sub-Data Collector, B-5
 CICS Sysplex Manager (CP/SM), 2-17
 CICS System Task Table, 4-22
 CICS/DBCTL, data collector, B-5
 CICSTRAN, B-6
 CLEAN command, 5-8–5-9
 COMPLETE parameter, 5-18
 QUEUE parameter, 5-18
 CLIST, modifying STROBE/ISPF's, 3-40, 3-42
 COBOL language Indexer, B-3
 commands
 LIST, 5-21
 COMMNDxx command, starting STROBE at IPL, 3-53
 common control blocks, 2-14
 Common Storage Area (CSA), 2-14
 COMPLETE parameter, described, 5-21
 control data set
 contents, B-1
 creating, 3-3
 direct access storage requirements, 2-4
 control data set members
 SPFINSTL, 3-34
 STROBE, 3-34
 Control-D, Control-M systems, using with STROBE,
 2-19
 conventional data set, B-7
 COPY, B-1
 COPY member, editing requirements, 3-10
 COPYGRP command, 5-18
 GROUP parameter, 5-18
 GRPOWNER parameter, 5-19
 NEWGROUP parameter, 5-18
 NEWOWNER parameter, 5-19
 COPYTEST, B-1
 COPYTEST member, editing requirements, 3-10
 CSA. See Common Storage Area (CSA)
 CSP
 Attributor, B-5
 Indexer, B-3
 CVEXIT, B-6
 CVLIB, meaning and value, 3-16

D

data collector
 identifying, 4-21
 reporting errors, 6-2
 Data set packaging utility, accessing from ISPF, 3-38
 data set packaging utility, maintaining two releases,
 3-9
 data sets
 allocating for STROBE, 3-12
 monitoring, 5-6
 naming conventions, 2-4
 required by STROBE, 1-3
 DATACLAS, meaning and value, 3-16
 DATE, meaning and value, 3-16
 DATESEP, meaning and value, 3-16
 DB2 Feature storage requirements, 2-13
 DB2 load libraries, identifying for the SQL Analysis
 Feature, 3-46
 DBRM
 Attributor, B-5
 consideration for binding, 3-47
 DELETE command, 5-19
 ELEMENTNUMBER parameter, 5-20

 GROUP parameter, 5-20
 GRPOWNER parameter, 5-20
 DELETE parameter, described, 5-21
 DELGRP command, 5-19
 GROUP parameter, 5-19
 GRPOWNER parameter, 5-19
 development environment features, 1-9
 diagnosing errors in STROBE, 6-1
 direct access storage requirements, 2-3
 distribution tape requirements, 3-3
 DLI attributor, B-4
 DSNAME, meaning and value, 3-16
 DSNSUFx, meaning and value, 3-16
 dynamic EXPLAIN authority, 3-50

E

ECSA. See Extended Common System Area (ECSA)
 enqueue/dequeue facility, planning, 2-5
 ENV command, 5-5, 5-20
 CHECK parameter, 5-20
 DISABLE parameter, 5-20
 ENABLE parameter, 5-20
 STATUS parameter, 5-20
 environment
 of STROBE, 1-3
 environment, modifying STROBE's, 5-5
 ENVNEW parameter, 5-3
 expanding the history data set, 5-11–5-12
 EXPDATE, meaning and value, 3-16
 EXPLAIN authority, when required, 3-50
 Extended Common System Area (ECSA), 1-3

F

FILTER, meaning and value, 3-16
 FOCUS restrictions, 2-18
 FORTRAN G Indexer, B-3
 FORTRAN H and VS Indexer, B-3
 FTP communications
 configuring for iSTROBE, 3-37

G

GDG log data set, maintaining, 5-6
 GDG, non-. See non-generation data group (non-
 GDG)
 GDG. See generation data group (GDG)
 generation data group (GDG), 3-19
 generation data set, B-7
 generic access rules, syntax definitions, 4-2
 GENSAM, B-1
 GRANT statements, submitting, 3-50

H

Hiperbatch restrictions, 2-18
 history data set
 expanding, 5-11–5-12
 initializing, 3-24

- maintaining, 5-10
- reorganizing, 5-10
- history data set, customizing, 3-23
- history log data set, printing, 5-11
- history manager, described, 1-9
- history, measurement session, 2-11
- HISTORY_PLEX, meaning and value, 3-16
- HISTORY_USER, meaning and value, 3-16
- HSM restrictions, 2-17
- HSTPDLOG, B-1
- HSTPDLOG member, 5-11
- HSTRELOG, B-1
- HSTREORG member, 5-10

I

- IBM code page
 - translation for iSTROBE characters, 3-37
- IDMS DML statement attributor, B-5
- IEBUPDTE, 4-18
 - copying distribution tape, 3-4
- IEF Indexer, B-3
- IEFU85, B-6
- IMS DLI attribution call utility, B-5
- IMS Feature storage requirements, 2-13
- IMS requirements for the ADABAS/NATURAL Feature, 3-6
- INDEXCOB, B-1
- INDEXCOB member, 3-31
- Indexers
 - reporting errors, 6-2
 - testing, 3-31
- INDEXFOR, B-1
- INDEXPLI, B-1
- initialization, B-4
- INSTALL, B-1
- INSTALL member, 3-4
- installation checklist, ADABAS/NATURAL Feature, 2-21
- installation step checklist, 3-1, 3-12, 3-28
- installing
 - checklist for APMpower SQL Analysis Feature, 2-22
 - planning for security, 2-7
 - STROBE/ISPF, 3-33
 - TSO CLIST, 3-52
- installing STROBE
 - copying distribution tape, 3-3
 - customizing parameter data set, 3-12
 - testing programs, 3-3
- IP address
 - used for iSTROBE, 3-37
- IPL, using to start STROBE, 3-53
- ISPF
 - direct access storage requirements, 2-4
 - modifying CLIST, 3-40, 3-42
 - testing, 3-44
- ISPF/PDF utility panel, modifying, 3-44
- iSTROBE
 - FTP communications, 3-37
 - FTP program name, 3-37
 - IBM code page, 3-37
 - IP address requirement, 3-37

J

- Java target selector, installation, 3-39
- JAVAINST member, 3-40
- job access
 - defining CA-ACF2 security rules, 4-8
 - defining CA-Top Secret security rules, 4-6
- job control procedures, 4-17
 - customizing, 4-18
 - functions, 4-18
 - invoking indexing, 4-18
 - modifying Reporter parameters, 4-18
- job control statements, binding, 3-47
- job control statements, editing for installation, 3-4
- job step tasks, measuring when a program is already executing, 2-16
- JSTCB, meaning and value, 3-16

L

- L3OBJ
 - contents, B-4
- language
 - library routine attribution, 1-8
 - processors, test programs for, 3-31
- language features storage requirements, 2-13
- language processors
 - procedures invoking, 4-18
- LFAS. See Limited Function Address Spaces (LFAS)
- License Management System, 2-12, 3-2
- Limited Function Address Spaces (LFAS), 2-16
- LINEMAX report parameter, described, 4-18
- LINEPG, meaning and value, 3-16
- link editing, STREMODS, 4-19
- link list data set
 - copying modules to, 3-8
 - refreshing before testing STROBE, 3-11
- link list library, direct access storage requirements, 2-4
- linkage editor control statements
 - for installation, B-3
 - for STROBE/ISPF installation, B-5
- LIST command, 5-9
 - examples of, 5-21
 - parameters of, 5-21
- LISTGRP command, 5-20, 5-23
 - GROUP parameter, 5-23
 - GRPOWNER parameter, 5-23
- LMREFRESH command, 5-23
- LMSS, meaning and value, 3-16
- log data set
 - allocating, 3-19
 - as a conventional data set, B-7
 - as a generation data set, B-7
 - characteristics, B-7
 - copying for backup, 5-8
 - described, 1-3
 - direct access storage requirements, 2-4
 - maintaining, 5-6
 - requirements, B-6
 - specifications, B-7
- LOG_MAX, meaning and value, 3-17
- LOG_SPACE, meaning and value, 3-17
- LOG_UNIT, meaning and value, 3-17
- LOG_VOL, meaning and value, 3-17

LOGCLEAR command, 5-23
 LOGSWITCH command, 5-24
 LOGWARN, meaning and value, 3-17
 LPALIB, meaning and value, 3-17

M

maintaining a GDG log data set, 5-6
 maintaining a non-GDG log data set, 5-7
 maintaining the history data set, 5-10
 maintaining two STROBE releases, 3-9
 MAINTNCE, B-1
 map data sets, 3-31
 described, 1-5
 planning for, 2-3
 requirements, B-13
 MAXACT, meaning and value, 3-17
 MAXREQ, meaning and value, 3-17
 measured job step, data set naming requirements, 2-4
 measured program's address space, 2-13
 measurement requests
 initialization. See also STRBCINI, B-4
 Measurement Services Address Space (MSAS), 1-9
 Measurement Services Address Space (MSAS), starting and stopping, 5-4
 Measurement Services Address Space, size requirements, 2-14
 measurement session history, 2-11
 measurement task, 2-16, B-12
 measurement task storage requirements, 2-13
 measurement tasks, reporting errors in, 6-2
 message data sets
 size requirements, B-9
 MGMTCLA, meaning and value, 3-17
 MIGRATEG, B-1
 MIGRATEG member
 editing requirements, 3-60
 MIGRATEQ, B-1
 MIGRATEQ member
 editing requirements, 3-60
 migration utility, 3-53
 MIGRTSTG, B-1
 MIGRTSTQ, B-1
 MIM, restrictions, 2-18
 MODCSECT, B-1
 MODLINK, B-1
 module descriptor programs, B-3
 changing, 4-18
 STREMODS, B-3
 modules, copying to link list data set, 3-8
 MSASPROCNAME, meaning and value, 3-17
 multisystem environment
 security database recommendation, 4-8

N

naming conventions for data sets, 2-4
 NATURAL region definitions, 3-6
 non-GDG data set
 creating, 3-20
 maintaining, 5-7
 NOQUIT value, OPTIONS parameter, 2-16

O

OBJECT member of STROBE control data set, contents, B-3
 object modules
 needed for installation, B-3
 needed for STROBE/ISPF installation, B-5
 operator's console, submitting session manager commands, 5-1
 OPTIMIZE(2) compiler option restrictions, 2-19
 OPTIONS menu, changing, 3-36
 OPTIONS parameter, NOQUIT value, 2-16
 OS/390 restrictions, 2-18
 OSARTXMT member, 3-38
 OSTAREDC, B-4
 OSTARREC, B-1
 OSTARREC member, 3-38
 OSTARXMT, 3-38, B-1
 OSTARXMT member, 3-38
 OWNERID parameter, described, 5-21

P

packaging utility, access requirements, 3-38
 PANEPTE, 2-10, B-1
 parameter data set
 customizing during installation, 3-12
 described, 1-3
 direct access storage requirements, 2-4
 planning for, 2-3
 requirements, B-6
 Strobe startup process use, 2-2
 parameters
 described
 COMPLETE, 5-21
 DELETE, 5-21
 OWNERID, 5-21
 QUEUE, 5-21
 TYPE, 5-21
 partitioned data set (PDS), 2-2
 PASSWORD parameter, supplying, 3-4
 PDS. See partitioned data set (PDS)
 PDSMAN LPA option
 effect on STROBE, 2-18
 Performance Profile
 data set requirements, 2-3
 producing from test, 3-29
 requirements, B-13
 PERMIT command
 CA-Top Secret keywords, 4-5
 RACF keywords, 4-4
 PL/I Indexer, B-3
 previous STROBE versions, archiving and deleting, 3-3
 printing the history log data set, 5-11
 problems, reporting to Compware, 6-2
 PROCS, B-2
 parameters for STROBE job control, 4-17
 PROFILE, meaning and value, 3-17
 program temporary fix (PTF)
 described, 5-14
 programs for finding STROBE errors, 6-1

Q

- queue data set
 - allocating, 3-20, 3-22, 3-25, B-11
 - characteristics, B-8
 - described, 1-3
 - direct access storage requirements, 2-4
 - expanding, 5-10
 - migrating between versions, 3-53
 - monitoring, 5-8–5-9
 - multisystem considerations, B-8
 - obtaining status, 5-8
 - planning for, 2-3
 - processing, B-8, B-10
 - requirements, B-7
- QUEUE parameter, described, 5-21

R

- RACF
 - defining access rules, 4-3
 - session manager access rules example, 4-4
 - set of jobs access rules example, 4-4
 - specific system access rules example, 4-4
- RDEFINE command, keywords, 4-4
- READ access, requirements, 2-7
- reclaiming sequence numbers, 5-13
- REMV_OLD, meaning and value, 3-17
- reorganizing the history data set, 5-10
- REPORT, B-2
- REPORT member
 - using to create a Performance Profile, 3-29
- Reporter
 - changing LINEMAX in PROCS, 4-18
 - changing sort storage size in PROCS, 4-18
 - reporting errors, 6-2
- Reporter program
 - produce Performance Profile from test, 3-29
- reporting facility, described, 1-5
- reporting, problems to Compuware, 6-2
- request group data set
 - described, 3-22
 - maintaining, 5-12
 - migrating between versions, 3-53
 - serialization of, B-10
 - size requirements, B-10–B-12
- request group elements
 - reclaiming unused sequence numbers, 5-13
 - renumbering, 5-13
- reserves, converting to enqueues, 2-5
- RETAINC, meaning and value, 3-17
- RETAINQ, meaning and value, 3-18
- RETPD, meaning and value, 3-18
- return codes, described, 6-1
- rules, defining for security packages and STROBE access, 4-2

S

- SAMFORT, B-6
- sample data set
 - buffers, 2-14

- characteristics, B-13
- disposition, B-13
- requirements, B-12
- sample user exit program, modifying, 4-11
- SAMPLEW, B-6
- SAMPLI, B-6
- SAS restrictions, 2-18
- security
 - defining access rules, 2-7, 4-2
 - multisystem environment considerations, 4-8
 - overview, 4-2
- serialization of the request group data set, B-10–B-11
- session manager
 - address space used, 2-13
 - described, 1-2
 - reporting errors, 6-2
- session manager access
 - defining access rules for RACF, 4-4
 - defining CA-ACF2 security rules, 4-8
 - defining CA-Top Secret security rules, 4-6
- session requester
 - submitting commands to it, 5-1
- shadow catalogs, using with SQL Analysis Feature, 3-48
- SHOW command, 5-24
 - ABENDS parameter, 5-24
 - CSASIZE parameter, 5-24
 - DEFAULTS parameter, 5-25
 - ENQUEUE parameter, 5-24
 - LOG parameter, 5-24
 - QUEUE parameter, 5-24
 - SVCS parameter, 5-24
- SHOWPTF command, 5-25
- SmartBatch restrictions, 2-17
- SMEXIT, B-6
- SMEXIT user exit program
 - described, 4-11
 - parameters of, 4-12–4-15
- SMEXIT2, B-6
- SMEXIT2 user exit program
 - described, 4-11
 - parameters of, 4-12–4-15
- sort/merge utility program, 2-13
- SORTSIZ report parameter, described, 4-18
- source file, of the distribution tape, B-6
- source modules, B-6
- SPFINSTL, B-2
- SPFINSTL member, 3-34
- SQL Analysis Feature component parts, 3-45
- SQL Analysis Feature MVS component
 - binding to new version, 5-16
 - installation verification, 3-50
 - maintaining, 5-15
 - optimizing performance, 3-49
 - planning, 2-10
 - relinking, 5-16
 - replacing, 5-15
- SQL Analysis Feature, creating threshold sets, D-1
- SQL Analysis Feature, identifying DB2 load libraries, 3-46
- SQLAFRPL, B-2
- SQLAFVER, B-2
- SQLINSTL, B-2
- SQLOB, B-2
- SRCLIST, B-2
- STATUS command, 5-5, 5-25
- stopping STROBE address space, 5-2
- storage

- direct access requirements, 2-3
- for sample data set buffers, 2-14
- required for address spaces, 2-13
- STORCLAS, meaning and value, 3-18
- STRAIA51, 3-45
- STRAIA61, 3-45
- STRAIA71, 3-45
- STRAIAnn, B-2
- STRAN123, B-2
- STRAN151, 3-45
- STRAN223, B-2
- STRAN251, 3-45
- STRANA51, 3-45
- STRANAnn, B-2
- STRANGNT, B-2
- STRANGNT member, 3-50
- STRANRDJ, B-2
- STRANRDJ member, D-2
- STRANRDM, B-2, D-2
- STRANSM(, B-2
- STRANSMP, B-2
- STRANV24, B-2
- STRANVER, B-3
- STRAUTO, meaning and value, 3-18
- STRBACSP, B-5
- STRBACSQ, B-5
- STRBACTR, B-4
- STRBADAB, B-5
- STRBADAR, B-5
- STRBADBC, B-5
- STRBADFH, B-5
- STRBADLI, B-4
- STRBADML, B-5
- STRBAEVT, B-4
- STRBAIEF, B-5
- STRBALC, B-5
- STRBALCB, B-5
- STRBALPL, B-5
- STRBALSJ, B-4
- STRBALSJ, B-5
- STRBAMON, B-4
- STRBANCE, B-2
- STRBANCJ, B-2
- STRBANCJ, editing requirements, 3-6, 4-22
- STRBANCM, B-2, B-5
- STRBANCM, editing requirements, 3-6
- STRBASSA, B-5
- STRBASVC, B-4
- STRBATTR, B-4
- STRBAUHM, B-5
- STRBAUSI, B-4
- STRBAUTO, B-4
- STRBAVSM, B-5
- STRBCINI, B-4
- STRBCLOG, B-2, B-4
- STRBCLOG program, 5-8
- STRBCLPA, B-4
- STRBCLPB, B-4
- STRBCMIG, B-3
- STRBCOPT, B-4
- STRBCPH0, B-4
- STRBCRB, B-4
- STRBCSM, B-4
- STRBCSR, B-4
- STRBCTSJ, B-2
- STRBCTSJ member, 4-23
- STRBCTSK, B-2
- STRBCTSK member, 4-22
- STRBCTXR, B-4
- STRBCXAC, 4-11, B-4
- STRBDCrr, B-5
- STRBDCSQ, B-5
- STRBDFHD, B-5
- STRBDFHS, B-5
- STRBDFnn, B-5
- STRBDFSD, B-5
- STRBDFSM, B-5
- STRBDFSR, B-5
- STRBDM10, B-5
- STRBDM12, B-5
- STRBDSNA, B-5
- STRBDSNS, B-5
- STRBDSPY, B-3
- STRBDSPY diagnostic program
 - described, 6-1
 - job control statementsr, 6-2
- STRBGMIM, B-4
- STRBID10, B-5
- STRBID12, B-5
- STRBIDMS, B-5
- STRBISPF, B-4
- STRBMAP, B-4
- STRBMSAS, B-5
- STRBNATL, B-5
- STRBPANE, B-5
- STRBPASS, B-6
- STRBPC1, B-4
- STRBPJCL member, 3-38
- STRBPKG, B-2
- STRBPLOD member, 3-46
- STRBPR10, B-5
- STRBPR12, B-6
- STRBSAM1, B-3
- STRBSLMX, B-4
- STRBSM procedure, 3-27
- STRBSNAM, B-3
- STRBSPFM, B-4
- STRBTUSR, B-4
- STRBTPX, B-6
- STRBUT24, B-4
- STRBVM11, B-4
- STRBVM12, B-4
- STRBVPH1, B-4
- STRBVPH2, B-4
- STRBVPH3, B-4
- STREBAL0, B-3
- STREBALE, B-3
- STREMODS, B-3
- STREMODS module descriptor program
 - described, 4-18
- STREMODS module descriptor programs
 - link editing new copy of, 4-19
- STRETHRS, B-3
- STREUTIL, B-3
- STRFCBF, B-3
- STRFCBU, B-3
- STRH, B-6
- STRHCOMP, B-3
- STRHEXPT, B-3
- STRHPRNT, B-3
- STRHUTIL, B-3
- STRMGR, meaning and value, 3-18
- STRMLIB, B-6
- STRMLIB member, 3-34
- STRMSGS, B-6
- STRMSGS member, 3-34

- STRNBND, B-2
 - STROBE
 - control statements, module for processing, B-4
 - environment of, 1-2
 - job control data set, B-2
 - load library, 4-18
 - modifying environment, 5-5
 - session requester, B-4
 - starting from a batch job, 3-28
 - starting its address spaces, 3-27
 - STROBE address space, size requirements, 2-14
 - STROBE command language
 - described, 1-7
 - submitting to session requester, 5-1
 - syntax for STROBE administration, 5-17
 - STROBE environment, components, 1-3
 - Strobe environments, managing, 5-2
 - STROBE facilities
 - accessing, 1-6
 - reporting, 1-5
 - session management, 1-2
 - strobe libraries, direct access storage requirements, 2-4
 - STROBE member, described, 3-34
 - Strobe packaging utility, output data set formats, 3-38
 - STROBE-XCF Group, 1-6
 - STROBE/ISPF
 - and STROBE/PFM libraries source, B-6
 - display format Profile editor. See also STROBEDS, B-3
 - editing job control statements, 3-34
 - message library. See also STRMLIB
 - message text library. See also STRMSGs
 - panel library. See also STRPLIB
 - reporting errors, 6-2
 - skeleton library. See also STRSLIB
 - STROBE CV
 - log data set name, 2-5
 - queue data set name, 2-5
 - system message data set name, 2-5
 - STROBEDS, B-3
 - STROBEDT, B-3
 - STROBLIB, 4-18
 - STROX procedure, 3-31
 - STRPLIB, B-6
 - STRPLIB member, 3-34
 - STRSLIB, B-6
 - STRSLIB member, 3-34
 - STRUPREP, B-3
 - STRXADS, B-6
 - STRXASF, B-3
 - STRXC, B-3
 - STRXCBF, B-3
 - STRXCBX, B-3
 - STRXCSP, B-3
 - STRXFTG, B-3
 - STRXFTH, B-3
 - STRXIEF, B-3
 - STRXNAT, B-3
 - STRXPLF, B-3
 - SUBMIT command, 5-25
 - GROUP parameter, 5-25
 - GRPOWNER parameter, 5-25
 - submitting
 - GRANT statements, 3-50
 - SVC, meaning and value, 3-18
 - SVCOVR, meaning and value, 3-18
 - SWAP, meaning and value, 3-18
 - symbolic parameters, used with JCL procedures, C-1
 - syntax
 - for LIST command, 5-21
 - SYSHIST job step, 3-23
 - SYSPRIN
 - data sets, 4-18
 - SYSPRINT data sets, writing information to, 6-1
 - system access
 - defining CA-ACF2 security rules, 4-7
 - defining CA-Top Secret security rules, 4-6
 - defining RACF security rules, 4-4
 - restricting with RACF, 4-8
 - system control block, naming for CICS, 3-6
 - system programming tasks, 1-1
- ## T
- Targeting, Java selector installation, 3-39
 - TCAM, how STROBE attaches measurement tasks, 2-16
 - temporary data sets
 - planning for, 2-3
 - requirements, B-14
 - TERMINATE command, 5-26
 - terminating a request, using the QUIT command, 2-16
 - TEST command, 5-26
 - EXITOFF parameter, 5-26
 - EXITON parameter, 5-26
 - test program
 - source, B-6
 - test STROBE versions, archiving and deleting, 3-3
 - TESTAPF, B-2
 - TESTAPF member
 - using to test STROBE, 3-29
 - testing
 - APMpower MVS Component, 3-44
 - STROBE/ISPF, 3-44
 - user exit programs, 4-16
 - with TESTAPF member, 3-29
 - with TESTNEW member, 3-29
 - with TESTSTR member, 3-29
 - TESTNEW, B-2
 - TESTNEW member
 - using to test STROBE, 3-29
 - TESTSTR, B-2
 - TESTSTR member
 - using to test STROBE, 3-29
 - threshold values, changing for SQL Analysis Feature, D-2
 - TIME, meaning and value, 3-18
 - TIMESEP, meaning and value, 3-18
 - transactions, preventing measurement, 4-22
 - Transrelate Utility Manager restrictions, 2-18
 - TRN2COB1, B-2
 - TRN2COB2, B-3
 - TRNCLEAN, B-2
 - TRNDSN, B-2
 - TRNPDS, B-2
 - TRNPGMXX, B-2
 - TRNPLI, B-3
 - TRNPLIXX, B-2
 - trouble reporting in STROBE, 6-1
 - TSO CLIST, installing, 3-52
 - TSOEXEC command, invoking session requester as authorized, 3-52
 - two Strobe releases, maintaining, 3-9
 - TYPE parameter, described, 5-21

U

- unauthorized library, direct access storage requirements, 2-4
- UNIT, meaning and value, 3-18
- upgrading, maintaining two STROBE releases, 2-12
- user exit
 - described, 4-9
 - linking the program, 4-16
 - purpose and functions, 2-8
 - sample, A-1
 - source, B-6
 - testing the program, 4-16
- user's TSO address space
 - storage requirements, 2-14

V

- virtual storage requirements, 2-14
- VOLUME, meaning and value, 3-18

X

- XCFGROUP, meaning and value, 3-19