

**EMC Solutions Enabler
for z/OS**
Version 6.0

IMPLEMENTATION GUIDE

**P/N 300-000-884
REV A06**

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.emc.com

Copyright © 2002, 2003, 2004, 2005 EMC Corporation. All Rights Reserved.

Printed January, 2005

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

Trademark Information

EMC², EMC, Symmetrix, Celerra, CLARiiON, CLARAlert, Documentum, HighRoad, Legato, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CentraStar, CLARevent, Connectrix, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Enginuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, RepliCare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Preface	ix
----------------------	----

Chapter 1 Pre-Install Considerations

Introduction	1-2
Before You Begin	1-3
Choose an Installation/Configuration User Account	1-3
Gather Customization Information.....	1-3
Install ResourcePak Base	1-5
Review the Release Notes.....	1-5

Chapter 2 Installing Solutions Enabler

Overview of Installation	2-2
The Tape	2-2
Install Procedure	2-5
Step1: Unload the RIMLIB Dataset from the Tape	2-5
Step 2: Customize the JCL	2-5
Step 3: Run the Jobs	2-10
Step 4: Complete the Installation.....	2-13
Starting Over	2-14

Chapter 3 Configuring the Mainframe for Solutions Enabler

Configuration Overview	3-2
SYMAPI Server Architecture	3-2
SYMAPI Server Security Preparation	3-5
OMVS Segment Requirement.....	3-5
Started Task User Identity	3-5
Error Attempting to Create USS Directory /var/symapi ..	3-6

Solutions Enabler Configuration	3-7
CA TCPAccess Support	3-10
SYMAPI Database Support	3-11
Database Compression and Compatibility	3-12
Server Log	3-13
Gatekeeper Devices	3-14
Solutions Enabler Files	3-15
Network Services File	3-18
Environment Variable Definition File	3-20
Configuring for Local Time Zone	3-20
Options File	3-21
Authorizing Control Operations	3-23

Chapter 4 Using the SYMAPI Server

Starting the Server	4-2
Running the Server as a Batch Job	4-2
Running the Server as a Started Task	4-2
SYMAPI SV Command Operands	4-2
Controlling the Server	4-5
Display Server	4-5
Display Sessions	4-5
Display Variable	4-6
List Sessions	4-7
Setenv	4-7
Setvar	4-7
Shutdown	4-8
Special Circumstances during Server Execution	4-9
Stopping the Server	4-11
Stopping a Server Running as a Batch Job	4-11
Stopping a Server Running as a Started Task	4-11
Considerations When Stopping the Server	4-11

Appendix A Solutions Enabler Messages

Message Format	A-2
Messages	A-4

Index	i-1
--------------------	-----

Figures

2-1	RIMLIB Member List	2-6
2-2	EXEC SEMJCL Member	2-6
2-3	SEMWIN1 Pop-Up Window	2-7
2-4	Enter Information Screen	2-9
3-1	SYMAPI Server Threads	3-4
3-2	Example JCL	3-8

2-1	Contents of the Install Tape	2-3
3-1	#SEMAGNT DD Statements	3-9
3-2	SYMAPI Files	3-15
3-3	Solutions Enabler Avoidance and Selection Files	3-16

As part of its effort to continuously improve and enhance the performance and capabilities of the EMC product line, EMC periodically releases new versions of both the EMC Enginuity Operating Environment and EMC Solutions Enabler. Therefore, some functions described in this guide may not be supported by all versions of Enginuity or Solutions Enabler currently in use. For the most up-to-date information on product features, see your product release notes.

If a Solutions Enabler feature does not function properly or does not function as described in this guide, please contact the EMC Customer Support Center for assistance.

Audience This guide describes how to install EMC Solutions Enabler software and configure your mainframe as a server to support the Solutions Enabler offerings. Readers of this guide are expected to be familiar with the following topics:

- ◆ Symmetrix operation
- ◆ The z/OS mainframe type environment

Organization This guide is organized as follows:

Chapter 1, *Pre-Install Considerations*, presents an overview of EMC Solutions Enabler for z/OS mainframe types and identifies requirements and considerations before you begin installation and configuration.

Chapter 2, *Installing Solutions Enabler*, presents an overview of the installation process and describes how to install EMC Solutions Enabler on your mainframe system.

Chapter 3, *Configuring the Mainframe for Solutions Enabler*, describes how to configure the mainframe for Solutions Enabler offerings.

Chapter 4, *Using the SYMAPI Server*, discusses how to start the SYMAPI Server, how to examine its running status and adjust its behaviors, and how to stop it.

Appendix A, *Solutions Enabler Messages*, describes the messages that can be returned by the Solutions Enabler associated products.

Related Documentation

Related documents include:

- ◆ *EMC Solutions Enabler Version 6.0 Support Matrix*
- ◆ *EMC Solutions Enabler Version 6.0 Release Notes*
- ◆ *EMC ResourcePak Base for z/OS Product Guide*

Conventions

The following conventions are used in this manual.

Every use of the word SYMCLI means Symmetrix CLI.

Note: A note calls attention to any item of information that may be of special importance to the reader.



CAUTION

A caution contains information essential to avoid damage or degraded integrity to storage of your data. The caution might also apply to protection of your software or hardware.

Typographical Conventions

EMC uses the following type style conventions in this guide:

AVANT GARDE	Keystrokes
Palatino, bold	◆ Dialog box, button, icon, and menu items in text ◆ Selections you can make from the user interface, including buttons, icons, options, and field names
<i>Palatino, italic</i>	◆ New terms or unique word usage in text ◆ Command line arguments when used in text ◆ Book titles
<i>Courier, italic</i>	Arguments used in examples of command line syntax.

Courier	System prompts and displays and specific filenames or complete paths. For example: working root directory [/user/emc]: c:\Program Files\EMC\Symapi\db
---------	---

Courier, bold	◆ User entry. For example: sympoll -p ◆ Options in command line syntax
--------------------------	--

z/OS type conventions

The syntax conventions used in this guide are:

- ◆ CAPITALIZATION = must be typed
- ◆ [] = optional entry
- ◆ | = alternative parameter value
- ◆ UNDERSCORED = default value or menu selection

Where to Get Help

For questions about technical support, call your local sales office or service provider.

If you have a valid EMC service contract, contact EMC Customer Service at:

United States: (800) 782-4362 (SVC-4EMC)

Canada: (800) 543-4782 (543-4SVC)

Worldwide: (508) 497-7901

Follow the voice menu prompts to open a service call and select the applicable product support.

If you are a partner in the EMC Developers Program, refer to the EMC Powerlink Website at:

<http://powerlink.emc.com>

Sales and Customer Service Contacts

For the list of EMC sales locations, please access the EMC home page at:

<http://emc.com/contact/>

For additional information on the EMC products and services available to customers and partners, refer to the EMC Powerlink Website at:

<http://powerlink.emc.com>

Your Comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send a message to **techpub_comments@emc.com** with your opinions of this guide.

Your technical enhancement suggestions for future development consideration are welcome. To send a suggestion, log on to <http://powerlink.emc.com>, follow the path **Support, Contact Support**, and choose **Software Product Enhancement Request** from the Subject menu.

Pre-Install Considerations

This chapter presents an overview of the EMC Solutions Enabler offerings and components for mainframe support. It also identifies requirements and considerations before initiating any install operation.

- ◆ Introduction 1-2
- ◆ Before You Begin 1-3

Introduction

The EMC® Solutions Enabler for z/OS product provides all the software to establish a Solutions Enabler SYMAPI Server environment on the following mainframe types:

- ◆ z/OS
- ◆ OS/390

Throughout this manual, every use of the word z/OS also implies the OS/390 operating system.

The Solutions Enabler SYMAPI Server allows clients to map devices and data objects for the management of your storage complex. Through the SYMAPI Server, clients can monitor device configuration and status of devices that constitute the storage environment.

Clients can also perform SRDF® and TimeFinder® Manager control operations. Note, however, that when this optional `usermod` is applied, control operations run *unsecured* on the Solutions Enabler SYMAPI Server.

Before You Begin

Before installing the EMC Solutions Enabler software, do the following:

- ☐ Choose an Installation/Configuration User Account
- ☐ Gather information necessary for customizing product installation and execution JCL streams.
- ☐ Install EMC ResourcePak[®] Base if it is not already installed.
- ☐ Review the EMC Solutions Enabler release notes.

Choose an Installation/Configuration User Account

You must choose a TSO account in your system to run the installation jobs that have an OMVS segment defined in the security database. Since Solutions Enabler runs with the IBM Language Environment option POSIX(ON), the software requires that you either have a base OMVS segment defined or have access to an installation default profile. The OMVS segment is required by jobs #08SLMF, #09IVP, #10ECCIN, and #SEMAGNT.

For more information on defining OMVS segments, see the IBM publication *z/OS Security Server RACF Security Administrators' Guide*.

Gather Customization Information

Prepare for the installation by gathering the following information:

- ◆ Solutions Enabler dataset name prefix

Choose the prefix for all the product data sets to be allocated for the installation. The prefix includes the High Level Qualifier and all secondary qualifiers except the last. For example, if you choose the default EMC.SSEM600 as the prefix, you will allocate EMC.SSEM600.LOADLIB, EMC.SSEM600.PARMLIB, and so on.

This should generally be the same prefix as the one you choose when you copy the contents of the RIMLIB data set from the installation tape. The #01ALLOC job copies some members from the RIMLIB to the PARMLIB. If you use the recommended method to run the SEMJCL customization utility, the RIMLIB prefix is filled for you automatically.

- ◆ SMP/E dataset name prefix

Identify the prefix for the SMP/E datasets of the environment into which you have installed or will install the EMC ResourcePak Base (EMCSCF). The default value is `EMC.SMPE`, which is the default for the ResourcePak Base product.

- ◆ SCF subsystem ID

The EMCSCF server address space uses a z/OS subsystem identifier (SSID) to make itself known to applications which use its services. Solutions Enabler will use the SCF SSID you specify to communicate with the SCF. The default is `EMC`.

- ◆ SCF linklib prefix

Identify the prefix for the product datasets into which you have installed or will install the EMC ResourcePak Base (EMCSCF) Version 5.3 or higher. The default value is `EMC.SSCF530`, which is the default for the ResourcePak Base product, Version 5.3.0. The EMCSCF Linklib will be added to the STEPLIB DD statement of the Solutions Enabler execution JCL.

- ◆ Disk unit name and volume serial

Choose the unit name and a corresponding disk volume serial where you will install the Solutions Enabler product datasets. The default for unit name is `SYSDA`, and there is no default for the volume serial.

- ◆ Tape unit name and volume

Choose the unit name for the tape device where the Solutions Enabler distribution tape will be mounted for the `#01ALLOC` and `#05RECEV` jobs. The default unit name is `CART`. The tape volume serial is filled in with `SEMvrn`, and should not be changed unless you have been given specific instructions by EMC Customer Support.

Install ResourcePak Base

Solutions Enabler requires the use of EMC ResourcePak Base Version 5.3.0 at a minimum. Read and follow the instructions in the *EMC ResourcePak Base for z/OS Product Guide*.

If you have already installed ResourcePak Base V5.3.0 or higher as part of another product installation, you do not need to re-install it. However, you should ensure that all recommended maintenance is applied. Refer again to the ResourcePak Product Guide for information on how to obtain the latest maintenance.

Review the Release Notes

For the latest information about this software release, you should read the *Solutions Enabler for z/OS Release Notes*. In addition, the *EMC ControlCenter Release Notes* describe detailed hardware and software requirements.

This chapter discusses how to install EMC Solutions Enabler on your mainframe. It includes the following topics:

- ◆ Overview of Installation2-2
- ◆ Install Procedure2-5

Overview of Installation

The Solutions Enabler software can be installed on an z/OS type mainframe to operate as a SYMAPI Server. The SYMAPI Server for z/OS directly connects to and manages the Symmetrix[®] storage system. A SYMAPI Server serves EMC ControlCenter[™] or Solutions Enabler clients on open system platforms by way of a TCP/IP network.

The Tape

The SMP/E formatted 3480 installation tape is standard-labeled and has a VOLSER in the following format:

SEMvzm

where:

v = version

r = revision

m = modification

For example, Solutions Enabler V6.0 has the volser *SEM600*.

The tape contains six datasets that provide and support the install function management ID, FMID(SSEM600). This function installs the Solutions Enabler product.

Table 2-1 lists the tape contents, including the dataset names and associated JCLs. The first dataset on the tape is RIMLIB, which contains sample JCL for all subsequent installation steps and sample parameter files for product customization.

Table 2-1 Contents of the Install Tape

Dataset	Contents
EMC.SSEMvrm.RIMLIB	Unloaded Partitioned dataset containing the following members:
#ASMHA	If job #05RECEV fails with: "GIM23401T the pProgram IEV90 was required for SMP/E but was not available" Run #ASMHA to define IEV90, then re-run #05RECEV.
#01ALLOC	Creates all needed datasets for installing the product.
#02DFZON	This job is not used in V6.0.
#03REPRO	This job is not used in V6.0.
#04DDDEF	Defines the DD definitions for all three SMP/E zones.
#05RECEV	Gets the SYSMODS and HOLDDATA, and gets the FMID functions for the products.
#06APPLY	Installs the products into the target libraries.
#07OPT	This job is currently not needed for any ControlCenter component and is not included in this release.
#08SLMF	Runs the Solutions Enabler License Management Facility (symlmf) in batch mode.
#09IVP	Verifies the installation.
#10ECCIN	Creates and initializes a SYMAPI database for use by #SEMAGNT.
#11ACCP	Updates the distribution libraries.
#12CNTRL	Enables some or all of the ControlCenter SRDF and TimeFinder functions to hosts outside of the mainframe environment. Use this job with caution. For more information, refer to <i>Authorizing Control Operations</i> on page 3-23.
#13SMPOP	Sample job that can be used later to receive and apply APARS.
#14MSACL	Sample job that can be used to get the unique host id. for a mainframe host or Lpar. The unique host id is needed for every host that will access a Symmetrix System which has been enabled for access control.

Table 2-1 Contents of the Install Tape (continued)

Dataset	Contents	
EMC.SSEMvrm.RIMLIB (continued)	#99RESTR	Utility job to execute SMP/E RESTORE command.
	#99REJCT	Utility job to execute SMP/E REJECT command.
	#SEMAGNT	Sample startup JCL for the Solutions Enabler for z/OS server.
	SEMEDIT	Sample TSO edit macro to customize all installation jobs in the library.
	SEMJCL	A utility which helps to customize your JCL to edit installation jobs. (See section below.)
	SEMWIN1	The pop-up window for SEMJCL.
	ANRTCP00	For the Computer Associates TCP_Access product. This member contains sample control statements that allow the remote SYMAPI Server to use TCPACCESS IP stack. This member will be copied to the PARMLIB by #01ALLOC.
	SYMENV00	Sample member containing some SYMAPI environment variable definitions. This member will be copied to the PARMLIB by #01ALLOC.
	SYMNET00	Sample configuration file for network services definitions. This member will be copied to the PARMLIB by #01ALLOC.
EMC.SSEMvrm.HOLDDATA	Not used in V6.0.	
EMC.SSEMvrm.SMPMCS	Unloaded sequential dataset containing the SMP/E MCS statements needed to install the products. FMID (SSEMvrm) installs the Solutions Enabler for z/OS software.	
EMC.SSEMvrm.F1	SMP/E relfile that contains the JCLIN for SSEMvrm.	
EMC.SSEMvrm.F2	SMP/E relfile that contains the product load modules for SSEMvrm.	
EMC.SSEMvrm.EMCIVP	A SYMAPI database provided for the installation verification program.	

Install Procedure

This procedure installs both the EAS driver component and the Solutions Enabler software required for the operation of the mainframe SYMAPI Server.

If, while installing Solutions Enabler, you decide that you want to back out and start the installation over, you can do so up until you run job #11ACCP. For instructions, refer to *Starting Over* on page 2-14.

Step1: Unload the RIMLIB Dataset from the Tape

To unload the RIMLIB from the installation tape:

1. Create, customize, and submit the following JCL job to unload the RIMLIB from tape:

```
//JOB CARD
//COPY01 EXEC PGM=IEBCOPY,REGION=4M
//INDD1 DD DSN=EMC.SSEMvrm,.RIMLIB,DISP=OLD,
// UNIT=(TapeDrive,,DEFER),VOL=SER=Volser,LABEL=(1,SL)
//OUTDD1 DD DSN=DS-Prefix.RIMLIB,DISP=(NEW,CATLG),
// UNIT=SYSALLDA,SPACE=CYL,(2,1,20)
//SYSPRINT DD SYSOUT=X
//SYSIN DD *
COPY OUTDD=OUTDD1,INDD=INDD1
//
```

The first job (#01ALLOC) of the install uses dataset *DS-Prefix*.RIMLIB.

2. When requested, mount the Solutions Enabler tape in the drive.

Step 2: Customize the JCL

Solutions Enabler includes a REXX exec program, SEMJCL, to expedite the JCL customization process by allowing you to create a site specific TSO edit macro in your CLIST library and then running it against every member of the RIMLIB whose name starts with a pound sign (#).

If you prefer to manually customize the JCL, customize the # prefixed members as necessary, and then continue with *Step 3: Run the Jobs* on page 2-10.

To use SEMJCL, do the following.

1. Go to the **Utilities.DSList** of the main TSO menu and type the first few qualifiers of your RIMLIB dataset name.

The RIMLIB displays as part of the DSLIST.

2. Scroll to the RIMLIB dataset and type **m** in the command field.

The member list for the RIMLIB dataset (Figure 2-1) displays.

Menu Options View Utilities Compilers Help				
DSLIST - Data Sets Matching EMC.SSEM600.RIMLIB				
Command ==>			Row 1 of 1	
			Scroll ==> CSR	
Command - Enter "/" to select action			Tracks %Used XT Device	
m	EMC.SSEM600.RIMLIB		15	41 1 3390
***** End of Data Set list *****				

Figure 2-1 RIMLIB Member List

3. Scroll to the SEMJCL member in the RIMLIB, and then type **EXEC** in the input area to the left of the member name (Figure 2-2).

Menu Functions Confirm Utilities Help				
DSLIST EMC.SSEM600.RIMLIB				
Command ==>			Row 00001 of 00033	
			Scroll ==> CSR	
Name	Prompt	Size	Created	Changed ID
\$\$\$README				
\$EXTRACT				
#ASMHA				
#SEMAGNT				
#01ALLOC				
#01EALLC				
#02DFZON				
#03REPRO				
#04DDDEF				
#04EDDDF				
#05RECEV				
#05RECEV				
#06APPLY				
#08SLMF				
#09IVP				
#10ECCIN				
#11ACCP				
#12CNTRL				
#13SMP0P				
#14MSACL				
#15ECLN				
#99REJCT				
#99RESTR				
ANRTCP00				
HOLDDATA				
SEMEDIT				
exec SEMJCL				
SEMWIN1				
SMPMCS				
SSEM600				
SYMENV00				
SYMNET00				
SYMOPT00				
*XEnd**				

Figure 2-2 EXEC SEMJCL Member

This executes the SEMJCL exec, which displays the customization screen (Figure 2-3).

```

Menu  Functions  Confirm  Utilities  Help
-----
E  Customize EMC Solutions Enabler 6.0.0 Tape Package Install JCL
C  PF3 To Cancel.
  ENTER To customize the installation JCL.

Data set name Prefix:  EMC.SSEM600
SMP/E Data set prefix: EMC.SMPE
SCF Subsystem Id:     EMC
SCF Linklib Prefix:   EMC.SSCF530
Disk unit name:       SYSDA      Disk Volume Serial:
Tape unit name:       CART        Tape Volume Serial: SEM600

Enter JOB card below. ('%MEMBER%' is replaced by the member name):
//ADMIN1A JOB ACCT,'EMC SEM 6.0.0',
// CLASS=A,             <-- CHANGE IF NEEDED
// MSGCLASS=A,          <-- CHANGE IF NEEDED
// NOTIFY=ADMIN1        <-- CHANGE IF NEEDED

Default CLIST Library:  EMC.SSEM600.RIMLIB
Default Edit macro name: SEMX600

#14MSACL
#15ECLN
#99REJCT
#99RESTR
ANRTCP00
EMCMLIST
HOLDDATA
SEMEDIT
SEMJCL      *RC=0
SEMWIN1
SEMWIN1E
SEMX600
SMPMCS
SMPMCSE
SSEM600
SYMENV00
SYMNET00
SYMOPT00
**End**

```

Figure 2-3 SEMWIN1 Pop-Up Window

4. Enter your site specific information according to the following:

Tip

To cancel the SEMJCL, press PF3.

- a. In the **Data set name Prefix** field, enter the high level qualifier and any additional qualifiers to be used when allocating new Solutions Enabler datasets.

For restrictions on SYMAPI database allocations, refer to *SYMAPI Database Allocation Guidelines and Restrictions* on page 3-11. The sample database and your runtime database will use the prefix specified here.

- b. In the **SMP/E Data set prefix** field, enter the prefix of the SPM/E datasets where EMC ResourcePak Base is installed.
- c. In the **SCF Subsystem Id** field, enter the subsystem name of the SCF address space. The default is *EMC*.

- d. In the **SCF Linklib Prefix** field, enter the prefix of the SCF load module library corresponding to the subsystem you entered above.
- e. In the **Disk unit name** field, enter a valid unit name defined at your site to be used in the UNIT= operand when allocating new Solutions Enabler datasets. The default is `SYSDA`.
- f. In the **Disk Volume Serial** field, enter the volume serial number of the DASD volume where the new Solutions Enabler datasets will be allocated.
- g. In the **Tape unit name** field, enter a valid unit name where the Solutions Enabler installation tape should be mounted. The default is `CART`.
- h. In the **Tape Volume Serial** field, enter the volume serial number of the Solutions Enabler installation tape. The default is `SEMvrm`.

A default job card is filled in, including a place holder for accounting field, programmer name value, `CLASS=A`, `MSGCLASS=A`, and `NOTIFY` operands. The `JOBNAME` and `NOTIFY=` operands use the TSO ID of the user running the `SEMJCL` process.

If you use `%member%` in the jobname field in the job card, the `RIMLIB` member name will be used as the job name.

- i. In the **Job Card Information** field, specify up to four statements for your jobcard (Figure 2-4).

The syntax of the statements is not validated until the editing job is done.

```

Menu  Functions  Confirm  Utilities  Help
-----
E      Customize EMC Solutions Enabler 6.0.0 Tape Package Install JCL
C      PF3 To Cancel.
      ENTER To customize the installation JCL.

      Data set name Prefix:  EMC.SSEM600
      SMP/E Data set prefix:  EMC.SMPE
      SCF Subsystem Id:      EMC
      SCF Linklib Prefix:    EMC.SSCF530
      Disk unit name:        3390      Disk Volume Serial:  EMC002
      Tape unit name:        CART      Tape Volume Serial:  SEM600

      Enter JOB card below, ('%MEMBER%' is replaced by the member name):
      //ADMIN1A JOB 1234,'EMC SEM 6.0.0',
      //  CLASS=A,              <-- CHANGE IF NEEDED
      //  MSGCLASS=X,           <-- CHANGE IF NEEDED
      //  NOTIFY=ADMIN1         <-- CHANGE IF NEEDED

      Default CLIST Library:  EMC.SSEM600.RIMLIB
      Default Edit macro name: SEMX600

      #14MSACL
      #15ECLN
      #99REJCT
      #99RESTR
      ANRTCP00
      EMCMLIST
      HOLDDATA
      SEMEDIT
EXEC  SEMJCL      *RC=0
      SEMWIN1
      SEMWIN1E
      SEMX600
      SMPMCS
      SMPMCSE
      SSEM600
      SYMENV00
      SYMNET00
      SYMOPT00
      **End**

```

Figure 2-4 Enter Information Screen

- j. Notice that the **CLIST Library** field is filled in for you with the name of the RIMLIB dataset. This field is not normally modifiable.
- k. Notice that the **macro name** field is filled in for you with the name SEMX600. This field is not normally modifiable.
- l. Press ENTER.

SEMJCL generates an edit macro and uses the ISPF editor to edit the installation jobs.

At this point in the procedure, all of the installation jobs have been edited with site specific information and are ready to run.

Step 3: Run the Jobs

Run each of the following jobs:

◆ #01ALLOC

Creates all needed datasets for installing the product.

This job also copies the IVP database off the tape. Sample configuration members are copied from the RIMLIB into the Solutions Enabler PARMLIB.

◆ #02DFZON and #03REPRO

These jobs are no longer used since the SMP/E environment has been setup for the installation of EMC ResourcePak Base.

◆ #04DDDEF

Defines the DD definitions for all three SMP/E global zones.

◆ #05RECEV

Gets the SYSMODS and HOLDDATA. It also gets the FMID function, `FMID(SSEMvrm)`, which delivers the Solutions Enabler for z/OS software.

If job #05RECEV fails with the message:

```
GIM23401T the pProgram IEV90 was required for SMP/E
but was not available
```

Run #ASMHA to define IEV90, and then re-run #05RECEV.

◆ #06APPLY

Does a selective apply for the function received in the previous job:

```
apply select(SSEMvrm)
```

At this point you have installed the load library members into the target load library. The next few jobs execute programs in the load library, which have additional requirements. Be sure to check each program's requirements before submitting each job.

◆ #08SLMF

Runs the Solutions Enabler License Management Facility (**symlmf**) in batch mode. You must use an editor to customize the input, entering the license keys from the key cards which were received with your Solutions Enabler package.

The **symlmf** program normally runs in batch in z/OS, and the input to the program is specified in the **SYSIN DD** statement. The statements there satisfy the dialog that **symlmf** would normally have with an interactive user on non-z/OS platforms.

The dialog sequence is as follows:

1. At the following prompt, enter **Y** to begin the registration process:

Do you want to enter a registration key? **Y**

2. At the following prompt, enter the 19-byte key value as specified on the key card:

Enter the license key:

3. At the following prompt, enter **Y** to register another key value, or **N** to complete the registration process:

Do you want to enter a registration key? **N**

Entering **N** causes **symlmf** to finish updating the license file and end the job step. The sample input below shows the appearance of the **SYSIN DD** statement coded to enter two keys:

```
000045 //SYMLMFI EXEC PGM=SYMLMF, PARM='POSIX(OFF)/'
000046 //STEPLIB DD DSN=EMC.SSEM600.LOADLIB,DISP=SHR
000047 //SYMLIC DD DSN=EMC.SSEM600.LICENSE,DISP=OLD
000048 //SYSPRINT DD SYSOUT=*
000049 //SYSOUT DD SYSOUT=*
000050 //SYSIN DD *
000051 Y
000052 0000-1111-2222-3333
000053 Y
000054 3333-2222-1111-0000
000055 N
000056 /*
```

From this point on, the Solutions Enabler load library must be APF-authorized. The EMCSCF linklib will have been APF-authorized for SCF to operate. Use the desired method at your site to authorized the Solutions Enabler load library.

Also, the user who runs jobs from this point must have an OMVS segment defined. For more information, refer to *Before You Begin* on page 1-3.

The ResourcePak Base (EMCSCF) address space must be active and must specify the same subsystem identifier (SSID) as the one specified on the JCL Customization panel.

◆ #09IVP

Runs the installation verification program. The installation verification program exercises the Solutions Enabler for z/OS against a known Symmetrix database in off-line mode. This installation step just shows that all components have been installed and are working together.

When this job completes, the last line in the SYSPRINT will be:
Total number of test performed>>>xxxx

Where *xxxx* is a value that varies depending on the Symmetrix configuration. On your z/OS system, if you run the offline tests only, the value of *xxxx* is 248.

If you choose the online tests, you must make the Solutions Enabler load library APF authorized, and EMCSCF must be active.

◆ #10ECCIN

This job creates and loads the database that is supplied to EMC ControlCenter clients. Job #10ECCIN attempts to discover every Symmetrix system connected to your mainframe host. If there are many Symmetrix systems connected, this job may run for a considerable period of time. If there are Symmetrix systems that you do not want the ControlCenter client to view, you may exclude them from the discover process.

It is only necessary to initialize one SYMAPI database file for use with the server. Job #10ECCIN will initialize the dataset described by the SYMAPI01 DD statement. The dataset referenced by SYMAPI01 is the same as the default database referenced by the SYM\$DB statement in the #SEMAGNT job. #10ECCIN can initialize more than one database during a single run, but it is not necessary since the server can manage multiple sessions sharing the same database file.

To increase the number of databases initialized by #10ECCIN, copy one or more of the //SYMAPIxx DD cards, being sure to increment the 2-digit number on the DD card and the dataset name.

For instance, to initialize a seventh database file, add a DD card like this:

```
//SYMAPI07 DD DSN=DS-PREFIX.SYMAPI07,VOL=SER=VOLID,
              UNIT=SYSDA,SPACE=(TRK,(15,15)),
              DCB=(LRECL=11500,BLKSIZE=23000,RECFM=FBS,DSORG=PS),
              DISP=(NEW,CATLG,DELETE)
```

In the #10ECCIN job, there is a DD statement named SYM\$AVD. This is the Symmetrix avoid file and it expects a list of Symmetrix serial numbers. If this list is provided, then information about those Symmetrix systems is not stored in the database.

For example, to avoid Symmetrix 000000000001, the JCL should look like this:

```
//SYM$AVD DD *
000000000001
/*
```

IMPORTANT: All 12-digits of the serial number are required.

Step 4: Complete the Installation

To complete the installation:

1. Perform all other customizing and any testing as required. A sample startup job (#SEMAGNT) can be customized and run on a z/OS system. Note that you can either run SEMAGNT as a batch job or convert it to run as a started task.

When the tests are successfully completed, continue to the next step.

2. Customize and run job #11ACCPT. This job accepts the function management ID SSEMvzm into the distribution zone.
3. If SRDF and TimeFinder control functions are to be made available to hosts outside of z/OS, an optional zap must be applied. The zap is located in the RIMLIB as job #12CNTRL. See the job itself for installation details, and *Authorizing Control Operations* on page 3-23.

Your Solutions Enabler installation is now complete. Next, you need to establish your server environment by performing the configuration and setup procedures explained in Chapter 3.

Starting Over

Solutions Enabler requires a pre-existing SMP/E environment. Because the environment is shared with other EMC products, it is important that you install the product according to your site standards. If, while installing the product, you decide that you want to back out and start the installation over, you can do so until you run job #11ACPT.

There are two utility jobs in the RIMLIB that allow you to back out of an installation. Both are customized by the SEMJCL process along with other installation JCL. The members are:

- ◆ #99RESTR — executes the SMP/E RESTORE command, which reverses the effect of an APPLY function. Use this job if you have successfully run #06APPLY and want to back out of that step.
- ◆ #99REJCT — executes the SMP/E REJECT command, which reverses the effect of a RECEIVE function. Use this job if you have successfully run #05RECEV and want to back out of that step. You cannot REJECT an FMID that has been applied. You must RESTORE it before REJECTing it.

#99RESTR and #99REJCT are not normally used in the installation process. You should only use these jobs to redo your installation.

Configuring the Mainframe for Solutions Enabler

This chapter discusses how to configure the z/OS mainframe for Solutions Enabler offerings. It includes the following topics:

- ◆ Configuration Overview3-2
- ◆ SYMAPI Server Security Preparation3-5
- ◆ Solutions Enabler Configuration3-7
- ◆ Authorizing Control Operations3-23

Configuration Overview

The Solutions Enabler installation process installs software and sample JCL which you can modify to suit the needs of your site. The SYMAPI Server will execute according to the configuration you choose. This chapter describes how to modify the JCL and parameter files to meet your site's requirements.

SYMAPI Server Architecture

The SYMAPI Server executes as a job step task specified on the EXEC PGM= statement in a job stream. The JCL can specify the TCP port on which the server will listen for connection requests, types of events to record in a server log, and other runtime variable data.

The *main thread* of the server initializes for TCP/IP operations, prepares the server log, opens communication with the operator console, and then waits for connection requests. Until a connection request arrives from a client, the main thread is the only thread executing in the address space. There are no pre-started tasks or threads.

The server uses POSIX thread technology supported by Unix System Services component of z/OS. When a connection request arrives from a client program, the main thread starts a new thread which will be dedicated to communicating with the client and executing SYMAPI requests on behalf of the client. The main thread is free to continue to listen for new connections and to accept instructions from the operator.

The thread dedicated to the session is called the *session thread*. It will receive and execute all API requests and return responses to the client. When the client calls the SymExit() function, the session thread executes SymExit remotely, and then closes the TCP/IP connection to the client. The thread itself then terminates, freeing memory and other resources to be available for other session threads.

The session may also be terminated by the server automatically when it detects a repeated succession of failed calls to the system to receive data from the client. This will occur if the client program ends without calling SymExit. Such an end may be normal or abnormal, but the server cannot detect which. It simply notices that it can no longer communicate with the client and ends the session.

While session threads come and go, the main thread continues to accept connection requests until the operator enters a command to initiate the server shutdown process. The operator has the choice to end the server safely, where the server will wait for all current sessions to terminate on their own, or to end the server immediately, where the server will simply terminate all current threads without giving them a chance to end on their own. The former method is preferred, when there is time to let sessions continue till they are done. The latter method can be used in an emergency, especially when a catastrophic condition occurs that requires a system IPL.

Each session has a sequentially assigned number called the *session number*. Each session also has an associated *thread number*. The session number and thread number will always be the same in this release of the software. The operator can use the thread number or the session number when referring to a session in commands such as `SHOW SESSION -num session_number`. The main thread is always assigned thread number 0.

Figure 3-1 on page 3-4 shows the SYMAPI Server main thread and several session threads.

The main thread will respond to the `SymInit()` request from a client and start a session thread, passing the connection on to the session thread. The main thread also accepts commands from the system operator who enters commands through a hardware console or using software product that can emulate the standard console.

Figure 3-1 shows the SYMAPI Server main thread with a network connection to a client and with a connection to the system console. The network connection is made when a client calls `SymInit()`, resulting in a new session to the server. The server starts a new thread to handle the session and returns to wait for more connections and to listen for operator commands. All session threads and the main thread write to the server log.

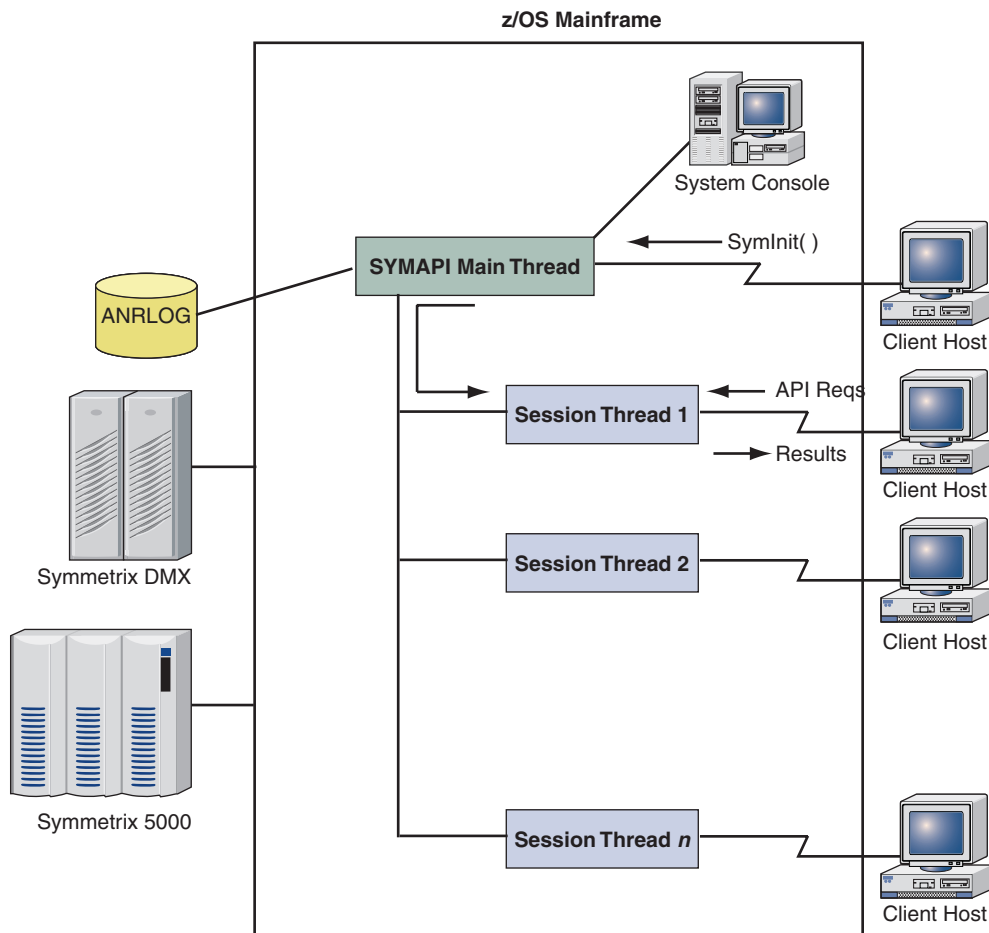


Figure 3-1 SYMAPI Server Threads

SYMAPI Server Security Preparation

OMVS Segment Requirement

When you are configuring Solutions Enabler JCL and your system to execute the SYMAPI Server, you may need to add definitions to your local security system.

If you are using IBM RACF, you may see message ICH408I when the server initializes. If you do, you must define an OMVS segment for the user or users who will run the server job. The following sample message assumes the job name and step name of the server are

SEMAGENT:

```
*ICH408I JOB(SEMAGENT) STEP(SEMAGENT) CL(PROCESS)
OMVS SEGMENT NOT DEFINED
```

If you are running the server as a started task, the user identity associated with the STC must have an OMVS segment defined.

For information on defining an OMVS segment for each user, refer to the IBM publication *Security Server RACF Security Administrator's Guide*.

Started Task User Identity

The SYMAPI Server is installed to be run as a batch job, but it can also be customized to run as a started task.

If you choose to run the server as a started task, it must have a user identity associated with it. You can assign a user identity to the server using the `RDEFINE` command or the started task table `ICHRIN03`. An example of the `RDEFINE` command is shown below assigning the user `SEMAGENT` to all started tasks whose names start with `SEMAGENT`.

```
RDEFINE STARTED SEMAGENT.* UACC(NONE)
STDATA(USER(SEMAGENT)) OWNER(SYS1)
```

If you use the `ICHRIN03` table to associate started task names with user identities, refer to the IBM publication *Security Server RACF System Programmer's Guide* for instructions on how to prepare this table.

Error Attempting to Create USS Directory /var/symapi

Solutions Enabler jobs #09IVP, #10ECCIN, and #SEMAGNT may attempt to create directories in the USS file system. Users executing these jobs without write permission for the directory /var/symapi, may see messages similar to those shown in the example below. The failure to create the directory /var/symapi does not cause any problem for the Solutions Enabler application, since the directory structure is not used during the execution of SYMAPI requests.

```
09.01.57 ICH408I USER(user) GROUP(group) NAME(user name) 035
035      /var/symapi CL(DIRACC ) FID(01C8C6E2F0F0F200010D000000000003)
035      INSUFFICIENT AUTHORITY TO MKDIR
035      ACCESS INTENT(-W-) ACCESS ALLOWED(OTHER R-X)
035      EFFECTIVE UID(0000888888) EFFECTIVE GID(0000000900)
```

In a future release, Solutions Enabler will begin to exploit the USS file system. At that time, the user executing Solutions Enabler jobs will be required to have write permission to a configurable USS directory.

Solutions Enabler Configuration

The following JCL is from the #SEMAGNT member of the RIMLIB dataset. It shows the JCL without local customization, as would be performed by executing SEMJCL.

```
//      JOBCARD
//*-----*
/* SAMPLE STARTUP JCL FOR THE EMC SOLUTIONS ENABLER FOR Z/OS
/*
/* * EDIT INSTRUCTIONS:
/* * THIS JOB IS EDITED BY THE SEMJCL PROCEDURE.  DS-PREFIX AND
/* * SCF-PREFIX WILL BE CHANGED TO THE VALUES YOU SPECIFY
/* * WHEN YOU RUN SEMJCL.
/*
/* * SYMAPISV PARM= OPERANDS:
/* * - &LEPARMS SHOULD BE BLANK UNLESS DIRECTED BY EMC CUSTOMER
/* * SUPPORT OR ENGINEERING.
/* * - THE -PORT OPERAND CAN BE OMITTED.  CURRENTLY SPECIFIES THE
/* * DEFAULT PORT 2707.
/* * - THE -LOG OPERAND SPECIFIES A LIST OF KEYWORDS INDICATING WHAT
/* * TYPES OF SERVER EVENTS SHOULD BE LOGGED.  REFER TO THE
/* * INSTALLATION AND CONFIGURATION MANUAL. 'NONE' INDICATES THAT
/* * THE SERVER WILL RUN VERY QUIETLY.
/* * - 'START' INDICATES THAT THE SERVER SHOULD BE STARTED.
/*-----*
// SET LEPARMS=' '
//SYMAPISV EXEC PGM=SYMAPISV,REGION=0M,
//          TIME=1440,
//          PARM='&LEPARMS/-PORT 2707 -LOG NONE START'
//STEPLIB  DD DSN=DS-PREFIX.LOADLIB,DISP=SHR
//          DD DSN=SCF-PREFIX.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSABEND DD SYSOUT=*
/*-----*
/* SYM$OPT IS THE SYMAPI OPTIONS FILE.
/*-----*
//SYM$OPT  DD DSN=DS-PREFIX.PARMLIB(SYMOPT00),DISP=SHR
/*-----*
/* SYM$LIC IS THE SYMAPI LICENSE KEY FILE.
/*-----*
//SYM$LIC  DD DSN=DS-PREFIX.LICENSE,DISP=SHR
/*-----*
/* SYM$DB IS THE SYMAPI DEFAULT DATABASE. NOTE WELL DISP=OLD.
/*-----*
//SYM$DB   DD DSN=DS-PREFIX.SYMAPI01,DISP=OLD
/*-----*
/* SYM$ENV SHOULD BE USED TO SET YOUR TIMEZONE USING THE TZ= SETTING.
/* OTHERWISE IT SHOULD BE USED ONLY UNDER THE DIRECTION OF EMC
```

```

/* * CUSTOMER SUPPORT.
/* *-----*
//SYMS$ENV DD DSN=DS-PREFIX.PARMLIB(SYMENV00),DISP=SHR
/* *-----*
/* * SCF$SCF-SSID CAUSES USE OF EMC SYMMETRIX CONTROL FACILITY SUBSYSTEM
/* * SCF-SSID. IF YOU CHANGE THE SUBSYSTEM ID OF THE SCF SERVER, OR
/* * INSTALL A NEW VERSION, CHANGE 'SCF-SSID' TO THE NEW SUBSYSTEM NAME.
/* *-----*
//SCF$SCF-SSID DD DUMMY
/* *-----*
/* * ANRLOG CONTAINS SERVER LOG MESSAGES. THE DD STATEMENT IS OPTIONAL.
/* * IF ANRLOG DD STATEMENT IS NOT SPECIFIED, THE SERVER LOG MESSAGES
/* * WILL BE DIRECTED TO THE SYSPRINT DD STATEMENT.
/* *-----*
//ANRLOG DD SYSOUT=*
/* *-----*
/* * SYMS$LOG CONTAINS SYMAPI LOG MESSAGES.
/* *-----*
//SYMS$LOG DD SYSOUT=*
/* *-----*
/* * SYMS$AVD CONTAINS SYMMETRIX AVOID STATEMENTS.
/* *-----*
//SYMS$AVD DD DUMMY
/* *-----*
/* * SYMS$GAVD CONTAINS GATEKEEPER AVOID STATEMENTS
/* *-----*
//SYMS$GAVD DD DUMMY
/* *-----*
/* * SYMS$GSEL CONTAINS GATEKEEPER SELECT STATEMENTS
/* *-----*
//SYMS$GSEL DD DUMMY
/* *-----*
/* * SYMS$INQ LIMITS SYMAPI ACTIONS TO THE SPECIFIED LIST OF DEVICES.
/* *-----*
//SYMS$INQ DD DUMMY

```

Figure 3-2 Example JCL

Descriptions of statements specific to the Solutions Enabler environment are listed in Table 3-1.

Table 3-1 #SEMAGNT DD Statements

DD Statement	Description
ANRLOG	Defines the destination of server log messages. If ANRLOG is omitted, server log messages are written to SYSPRINT.
STEPLIB	Defines the SYMAPI Server link library. The link library must be APF authorized.
SYM\$AVD	Contains the Symmetrix Avoid list. This is a list of Symmetrix serial numbers from which the SYMAPI will avoid collecting information.
SYM\$DB	Defines the default SYMAPI database. For more information, see <i>Server Default Database Locking</i> on page 3-12.
SYM\$ENV	Contains the C runtime environment variables. This file must be either a sequential dataset or a member of a partitioned dataset. This file must only be used with the direction of the EMC Customer Support Center.
SYM\$INQ	Contains the Symmetrix device inquiry file list. It lists the devices (by VOLSER name) to which the discovery process should confine its action.
SYM\$GAVD	Contains the Symmetrix gatekeeper Avoid list. It lists volume serial numbers of devices the SYMAPI should avoid using as a gatekeeper device. Typically, this list contains important system volumes (such as SYSRES, Page, JES Spool) that should not be used as a communications portal for the Symmetrix system.
SYM\$GSEL	Contains the Symmetrix gatekeeper selection list.
SYM\$LIC	Contains the Solutions Enabler license information.
SYM\$LOG	Contains SYMAPI log messages. Typically, this statement should be allocated to a JES SYSOUT file.
SYM\$NETC	Defines Services names.
SYM\$NETH	Defines a list of trusted hosts and users who are allowed to connect to the server.
SYM\$OPT	Defines the Options file. You can use the Options file to set the default behavior for certain SYMAPI commands.
SYSOUT	Contains IBM Language Environment runtime messages.
SYSPRINT	Contains output produced by the use of debugging controls.

The SYM\$OPT, SYM\$ENV, SYM\$AVD, SYM\$GAVD, and SYM\$INQ files can be either sequential datasets or members of a partitioned dataset. Their recommended file attributes are:

- ◆ LRECL=80
- ◆ RECFM=FB
- ◆ BLKSIZE=Multiple of the LRECL

These DD statements must not be allocated to JES SYSIN (DD*) files.

CA TCPAccess Support

If you are using the Unicenter:TCPAccess Communications Server stack from Computer Associates, you may want to add a SYSTCPD DD statement to the JCL to identify site specific configuration information. If the parameters pointed to by this DD are not correct when a client tries to connect, you may receive error messages. The dataset must have the following attributes:

- ◆ LRECL=80
- ◆ RECFM=FB
- ◆ BLKSIZE=Multiple of the LRECL
- ◆ DSORG=PS

When configuring TCPAccess to connect to the USS kernel, special system configuration steps are required. For more information on the SYSTCPD statement and USS configuration for Unicenter:TCPAccess, refer to the appropriate Computer Associates documentation.

The following statements are required for the SYMAPI Server:

```
TCPIPJOBNAME <TCPAccess Jobname>
DOMAINORIGIN <companyname.COM>
NSINTERADDR <DNS IP Address>
DNRSSID <TCPAccess Subsystem ID>
```

SYSTCPD statements must not be allocated to JES SYSIN (DD*) files.

SYMAPI Database Support

Solutions Enabler for z/OS supports the SYMAPI database and all the associated access modes, but it does not support the dynamic creation of a new database.

However, a SYMAPI application can specify the database by providing a DD name associated with the database using the following formats:

```
DD : <DDName>  
DDN : <DDName>
```

where <DDName> specifies a valid z/OS DD Name.

Solutions Enabler uses the following conventions to identify the database that it will associate with a particular session. The SYMAPI application specifies the database name in the SymInit() function call:

- ◆ with an explicit database name.
- ◆ as the database default name (by specifying NULL in the database argument).

SYMAPI Database Attributes

These SYMAPI database file attributes **must** be:

- ◆ LRECL=11500
- ◆ RECFM=FBS
- ◆ BLKSIZE=23000
- ◆ DSORG=PS
- ◆ SPACE=(CYL, (20,20))

SYMAPI Database Allocation Guidelines and Restrictions

Both the sample SYMAPI database and your production SYMAPI database can be allocated subject to SMS management active on your system. However, some characteristics of the SMS DATACLAS under which your databases are allocated are not compatible with the usage.

For this reason, you must take care in choosing a dataset prefix which will use a DATACLAS with compatible attributes. The following list shows the incompatible DATACLAS attributes:

- ◆ Compaction must be NO
- ◆ Extended Addressability must be NO
- ◆ Space Constraint Relife must be NO
- ◆ Volume count must be 1

Server Default Database Locking

If you are allowed to override the DATACLAS= operand in JCL DD allocations, and you specify a Data set Prefix that uses a DATACLAS with incompatible attributes, specify a substituted DATACLAS operand on the SYM\$DB DD statement in the #10ECCIN job.

The default database is described by the DD statement for SYM\$DB. When a session requests the default database, SYMAPI attempts to use SYM\$DB, handling locking for read-only and read/write sessions appropriately. If the session obtains database locks successfully, SYMAPI loads the database for the session in the mode (read-only, or read/write) desired.

Multiple users can share a database file in a read-only and read/write mode. Write integrity to the database is guaranteed by internal locking mechanisms. No two sessions can request read/write mode concurrently.

Once a read/write session has been started, the SYMAPI Server will prevent multiple read/write sessions by failing to initialize subsequent SymInit requests, or by blocking them until the first read/write session releases the database.

Note that the locking behavior applies to the actual dataset name associated with the DD name. Thus, two DD statements that reference the same physical can be coded, but the physical dataset will be protected in the manner described above.

Database Compression and Compatibility

The SYMAPI database is stored in compressed format to save disk space, and to reduce I/O in storing and loading it.

In addition to compression, three images of the database are stored in the physical dataset described by the SYM\$DB DD name. When the database is saved to disk by a read/write session, rather than rewriting the database image from the beginning, SYMAPI will write the new image in the first available block of the dataset following the previous image. The latest written image is always used in subsequent load operations, but the previous copies are available for rudimentary recovery scenarios when the latest image cannot be loaded.

Because of the multiple database images now stored in the physical dataset, you may notice an increase in disk space required for the dataset, in spite of the space savings that result from compression of each image.

For most configurations, three compressed images of the database will fit in the same space as one un-compressed image used in prior versions of SYMAPI. If you choose not to use compression, you may have to allocate a larger dataset.

Although the default is to store the database in compressed format, this behavior can be changed using the SYMAPI option `SYMAPI_DB_FILE_COMPRESSION`. For more information on specifying this option, refer to *Options File* on page 3-21. You may also refer to the RIMLIB member SYMOPT00 for directions on changing the option.

Compression will increase CPU utilization of database load and store operations, affecting processing of SymInit call during session initialization, and SymExit, SaveDbSave, SymCommit, and SymSync. If CPU utilization increases to an unacceptable level due to the use of compression, you can disable compression by setting `SYMAPI_DB_FILE_COMPRESSION` to the value `DISABLE`.

There is no option to suppress the storing of multiple images of the database. SYMAPI can read databases created by earlier versions, but a database created by Version 6.0.0 cannot be used by any version prior to Version 5.4.0.

IMPORTANT: This restriction does not apply to client/server connections where the client library version may be lower than the server version. Such connections are generally supported, except where otherwise noted.

Server Log

Solutions Enabler writes a log of events to the file described by the DD statement ANRLOG. ANRLOG enables you to keep the server log separate from the debugging output produced by the `SYMAPI_DEBUG` environment variable when its value is non-zero, or any other output to the SYSPRINT DD statement.

The server log is also distinct from the SYMAPI log file (DD `SYM$LOG`), recorded to by the API library.

The ANRLOG DD statement is optional. If you do not define it, Solutions Enabler will write the server log messages to the SYSPRINT DD statement. EMC recommends using the ANRLOG DD statement

instead of SYSPRINT for the server log, but there are some situations where interleaved log and debug messages may be useful. In such cases, you may want to comment out the ANRLOG DD statement.

For more information on the messages reported by Solutions Enabler, refer to Appendix A.

If errors occur while writing records to the log, the server will write a message to the operator console detailing the error, and explaining that log recording will be turned off.

The `-log` operand controls the volume of output to the log. For each type specified, the corresponding category of log messages will be issued.

Gatekeeper Devices

The use of *gatekeeper*-defined devices in a Symmetrix configuration does not apply to the z/OS type platforms. However, z/OS servers do communicate to the Symmetrix system using a UCB on the first device found in the Symmetrix storage array. The SYMAPI protocol selects the first on-line device as its gatekeeper. It is possible that this auto-select mechanism may not always be appropriate. For example, you may not want to have the system paging device or a JES SPOOL volume selected as the Symmetrix communication portal. The high I/O rate produced from the SYMAPI may adversely affect system performance. To control gatekeeper use by the Solutions Enabler server tasks, you can define specific devices to be used as gatekeepers, and you can also specify specific devices to be avoided as gatekeepers. For detailed information on how to specify gatekeeper device configuration, refer to *Solutions Enabler Configuration* on page 3-7.

Solutions Enabler Files

Solutions Enabler (SYMAPI) can reference various files during processing. These files and the following text are for experienced SYMAPI Server users and are not a prerequisite for normal use.

For simplicity's sake, the examples in the following tables show the use of instream data for input files. Generally, EMC does not recommend instream datasets for SYM\$NETC, SYM\$OPT, SYM\$AVD, SYM\$GAVD, SYM\$GSEL or SYM\$INQ. If the SYMAPI Server has a high connection rate, use of instream datasets may cause 013-C0 abends. Generally, the startup JCL should reference a PARMLIB member which contains the appropriate statements for the desired file. Prepare a PARMLIB member with the statements for one of the specified files above, and code a JCL statement using `DISP=SHR,DSN=DS-PREFIX.PARMLIB(member)` for each. For an example, see the specification of the SYM\$OPT DD statement in the #SEMAGNT member of the PARMLIB.

SYMAPI Files Table 3-2 lists and maps the SYMAPI files to corresponding DD statements.

Table 3-2 SYMAPI Files

SYMAPI File Identifier	DD Name	Description
SYMLOGPATH	SYM\$LOG	An output file that contains SYMAPI error and debug messages.
SYMDEFAULTDB	SYM\$DB	The default database file for the local SYMAPI application.
SYMNETSRVCDB	SYM\$NETC	An input file for the network services information, which contains the network service name.
SYMLICENSEDB	SYM\$LIC	An input file for the Solutions Enabler license information.
SYMOPTIONSFILE	SYM\$OPT	An optional input file that contains default constants, which can be set to change defaults to certain options within various SYMAPI calls.

Avoidance and Selection Files

Table 3-3 lists SYMAPI files or DDs that can exist in the Solutions Enabler startup JCL, which limit the scope or change the performance of Solutions Enabler during the discovery process.

These files can be used to customize and streamline command line coding for your specific environment.

These are editable files with device names or Symmetrix IDs that you use to limit the affect of commands to include or exclude the specified devices, gatekeepers, or Symmetrix units. The files hold either volume serial names (*volser*) or Symmetrix IDs (*Symmids*) with line entries having only one device name or ID per line. Lines beginning with a # (comment) are ignored.

Table 3-3 Solutions Enabler Avoidance and Selection Files

SYMAPI File Identifier	DD Name	Description
SYMAVOIDFILE	SYM\$AVD	This file affects the operation of the discovery process so that it skips devices that belong to the Symmetrix systems identified in this file. This may be useful if there are multiple Symmetrix systems connected to the host that you wish the discovery to avoid. The Symmetrix avoidance file is formatted with 12-character Symmetrix IDs, with one ID per line. For example, to avoid discover of the Symmetrix with a serial number of 0000183600186, code the following statements in your JCL: <pre>//SYM\$AVD DD * 0000183600186 /*</pre>
SYMINQFILE	SYM\$INQ	This file affects the inquiry and discovery processes so that they find only the volume serial name (volser) specified in this file. This maybe useful if you want to limit the command(s) to affect only certain Symmetrix devices from your host. The inquiry file is formatted with volume serial names (volser), with one volser per line. For example, to include information on volume ABC123 (only) and the Symmetrix to which it is attached, use a SYM\$INQ file that looks like this: <pre>//SYM\$INQ DD * ABC123 /*</pre>

Table 3-3 Solutions Enabler Avoidance and Selection Files (continued)

SYMAPI File Identifier	DD Name	Description
GKAVOIDFILE	SYM\$GAVD	This file affects calls to commands that use a gatekeeper to communicate to a Symmetrix unit. A gatekeeper whose volser matches any of the entries specified in the gkavoid file will not be chosen as a gatekeeper to communicate with the Symmetrix. This could be useful to designate certain Symmetrix devices that should not be used as gatekeepers. The gatekeeper avoidance file is formatted with volume serial names (volser), with one per line. For example, to instruct Solutions Enabler for z/OS to avoid using volume DEF456 as a gatekeeper device, use the following SYM\$GAVD statement: <pre>//SYM\$GAVD DD * DEF456 /*</pre>
GKSELECTFILE	SYM\$GSEL	In SYM\$GSEL, you will specify volume serials which are to be specifically used as gatekeepers. If a SYM\$GSEL list is defined, only the volumes specified will be candidates for use as gatekeepers. Specify one volume serial per line, with no other text on the line. If you specify a volume serial in both the SYM\$GAVD and the SYM\$GSEL, the entry in SYM\$GAVD takes precedence. Thus, SYM\$GSEL creates a limited list of candidate gatekeepers, and SYM\$GAVD further restricts the list by removing volumes from the candidate list. If you specify a gatekeeper selection list in SYM\$GSEL, be sure to specify at least one volume on each Symmetrix system you want to access through Solutions Enabler. For example, to instruct Solutions Enabler to use ONLY volumes GHI123, JKL123 and MNO123, use the following SYM\$GSEL DD statement: <pre>//SYM\$GSEL DD * GHI123 JKL123 MNO123 /*</pre> Note: If you specify a volume in BOTH the SYM\$GSEL and SYM\$GAVD, the entry in SYM\$GAVD takes precedence, effectively removing the volume from the list of potential gatekeepers. Thus, if the volume DEF456 also appeared in SYM\$GSEL, its entry in SYM\$GAVD (see example above) cancels its participation in gatekeeper selection.

Network Services File

Clients normally access the SYMAPI for z/OS Server by providing a host name and a port number, but service names can be set up that will predefine the host and port. If this file is to be used, it must be defined in the Server's startup JCL as ddname SYM\$NETC. The format of the file is:

Service name	host	IP	host	ip-address	server port
SERVICE01	HOST01	IP	HOST01	111.222.333.1	2707
SERVICE02	HOST01	IP	HOST01	111.222.333.1	3707

Trusted Host File

You can restrict access to the server by creating a Trusted Host file. The server DD statement SYM\$NETH references a dataset which contains a list of hosts and users who are allowed to connect to the server. The syntax for the trusted host file is shown below. Each line specifies a combination of host and user. A third token specifies a password, but its use is reserved for the future. In this release, the password must be coded as an asterisk (*).

```
host_name      user_name      *
```

Where:

host_name can be an IP address or hostname. Specify IP addresses if you are not using DNS name resolution services. Specify this for all hosts from which connection requests should be honored. You can specify an asterisk to indicate all hosts.

user_name is the name of a user who is allowed to connect to the server from the host indicated on the same line. You can specify an asterisk to indicate that all users from the specified host will be allowed to connect. You can also specify a list of users separated by commas. Do not use embedded blanks. The user names specified should be the effective names of the open system users connecting to the server.

For example:

xyz123.test.abccorp.com	joe	*	Allows joe to connect from the host xyz123.test.abccorp.com
*	bob	*	Allows bob to connect from any host
xyz456	*	*	Allows all users to connect from host xyz456
172.10.9.8	linda,mary	*	Allows linda and mary to connect from the host at IP address 172.10.9.8.

Note the following:

- ◆ If the SYM\$NETH DD statement is specified, the server will read it and apply the rules to all connection attempts. If the host/user combination of the connecting host does not match one of the rules, the connection is rejected.
- ◆ Do not specify SYM\$NETH DD DUMMY. The server will think the file exists but is empty. This will cause rejection of all connection requests. If you want to disable the use of the file, remove it from the JCL or change the DD name to a name that has no meaning to SYMAPI.
- ◆ In order to completely secure access to the server, you must:
 - Place the trusted host file in a SAF-protected dataset, and give read access to it to the user id under which the server will run. Grant read and write access to the maintainer of the file.
 - Protect the Solutions Enabler load library from all unauthorized access, such that only specific users can execute the SYMAPISV program. Unauthorized users must not be able to execute SYMAPISV or to copy it to another load library.
 - By protecting the SYM\$NETH file and the SYMAPISV program execution and copying, you can protect against unauthorized access to the server.

Environment Variable Definition File

The C runtime environment file can be used to turn on various debug levels for Solutions Enabler for z/OS. See the sample SYMENV00 member of the RIMLIB for more information on the various levels of debug information available to Solutions Enabler.

This debug facility produces a large amount of output which can impact your host. Use this facility only when directed by an EMC representative.

Configuring for Local Time Zone

The Solutions Enabler server software uses IBM Language Environment runtime library, and must execute with the LE option POSIX(ON). One of the side effects of running with POSIX(ON) is that the local time displays are influenced by the POSIX time semantic definitions. The default behavior defined by POSIX for local time interpretation may not fit your operation.

You can use the TZ environment variable to cause LE to display local time properly. There are several places where time stamps are displayed -- the ANRLOG and SYM\$LOG are the most important places. Use the TZ environment variable to establish your local offset from UTC (Coordinated Universal Time). The valid settings for TZ are standardized by the POSIX standard and are described in many publications, include the IBM Language Environment books.

In the PARMLIB member SYMENV00, you can set TZ. The sample setting in the distributed member causes the local time zone to be set to United States Eastern Standard Time, offset five hours from UTC (also known as Greenwich Mean Time or GMT), and EDT time may apply. The following example shows the same specification using an Instream dataset set for SYM\$ENV:

```
//SYM$ENV DD *
TZ=EST5EDT
/*
```

Options File

The `options` file or `SYM$OPT DD` in the SYMAPI startup JCL contains default settings that can be set to change defaults to certain options within associated SYMAPI calls.

The PARMLIB member `SYMOPT00` contains sample options that can be set. Each sample statement is commented, and can be enabled by removing the `#` in the first column.

Editing and File Format

This is an editable file to change the default behavior of certain SYMAPI command options.

The file contains editable default constants set to certain optional defaults in the line entries. Lines beginning with a `#` (comment) are ignored by SYMAPI.

Instant Split Option for `symmir`

The `instant (-instant)` option for the `symmir` command (or the `instant` flag of the SYMAPI BCV Control functions) improves the performance of a typical split operation by performing a quick foreground BCV split.

This option can be made the continual default split mode by setting the following variable in the options file:

```
SYMAPI_DEFAULT_BCV_SPLIT_TYPE = INSTANT
```

Wait BG Split Option

The `wait for background split` option can be used only when the previous option is set to `INSTANT` and you also want the SYMAPI calls to wait for the background split operation to complete before returning to the caller.

This option can be made to continually modify the default behavior of the instant split mode by setting the following in the options file:

```
SYMAPI_WAIT_FOR_BCV_BG_SPLIT = TRUE
```

Removing Default Options

To remove any default option, remove the line entry, rename the file, or comment the line by adding a pound (`#`) sign at the beginning of the line entry.

Maximum Connections Option

Using the option `SYMAPI_MAX_CLIENTS`, you can specify a maximum number of concurrent sessions active on the server. All subsequent connection attempts will be rejected when the number of sessions reaches the value specified for this option. As sessions end, new connections to the server will be accepted. The default value is 100.

**Database
Compression**

Using the option SYMAPI_DB_FILE_COMPRESSION, you can enable or disable the use of database image compression (refer to *Database Compression and Compatibility* on page 3-12). Compression is enabled by default. To disable compression, edit the options file you are using, uncomment the statement SYMAPI_DB_FILE_COMPRESSION and change the value to DISABLE.

Control Functions

By default, control functionality on the SYMAPI server is enabled. To disable it, set the SYMAPI_CTRL_VIA_SERVER option to DISABLE.

For the SYMAPI Server to execute control operations on behalf of a remote client running SYMCLI or EMC ControlCenter, you must set the SYMAPI_CTRL_VIA_SERVER option to ENABLE, and execute the #12CNTRL member provided in the RIMLIB dataset. For more information, refer to *Authorizing Control Operations* on page 3-23.

Authorizing Control Operations

Symmetrix control operations can be executed by the SYMAPI Server on behalf of a remote client running SYMCLI or EMC ControlCenter. Since such actions can impact user data or system operation, they are by default not allowed to be executed by the z/OS version of the server.

The following table lists the control operations that are disabled in the z/OS server:

Function	Action
SymAccessSessionStart	Starts an access control session
SymCgControl	Controls Consistency Groups
SymCgBcvControl	Invokes a BCV control operation affecting all standard devices in a composite group
SymCgRdfControl	Invokes an RDF control operation affecting all remotely mirrored RDF standard and R1 BCV devices in a composite group
SymConfigChangeSessionStart	Starts a configuration change session
SymDevBcvControl	Invokes a BCV control operation on the specified standard device and the specified BCV device
SymDevControl	Invokes a basic operation on one or all Symmetrix devices that meet a specified selection criteria
SymDevListBcvControl	Invokes a BCV control operation on a specified list of standard and BCV devices
SymDevListControl	Invokes a basic operation on a list of Symmetrix devices that meet a specified selection criteria
SymDevListRdfControl	Invokes an RDF control action on a list of devices
SymDgBcvControl	Invokes a BCV control operation affecting all standard devices in a device group, which has one or more associated BCV device
SymDgControl	Invokes a basic control operation affecting all standard, or optionally all BCV, devices in a device group
SymDgRdfControl	Invokes an RDF control operation affecting all remotely mirrored standard or RDF R1 BCV devices in a device group
SymDirControl	Invokes a director control operation on one or all SRDF RA directors
SymDirPortControl	Invokes a port control operation on a front-end director

Function	Action
SymLdevBcvControl	Invokes a BCV control operation affecting one standard device in a device group, which has one or more associated BCV devices
SymLdevControl	Invokes a basic control operation on a Symmetrix device in a device group
SymLdevListBcvControl	Performs a BCV control operation affecting a list of standard devices in a device group
SymLdevListControl	Executes a basic operation affecting the specified list of standard devices or BCV devices of a group
SymLdevListRdfControl	Invokes an RDF control operation affecting one remotely mirrored standard device, or one or more RDF R1 BCV devices in a device group
SymListDevListBcvControl	Invokes a single BCV or Snap control operation on a structure or array
SymNewCgControl	Invokes a basic control operation affecting devices of a specified type within a specific composite group
SymNewOptmzrControl	Invokes control operations on the Symmetrix Optimizer

The control operations can be enabled by executing the #12CNTRL member provided in the RIMLIB dataset. The job executes the AMASPZAP utility to change entries in a control table. Each entry in the table corresponds to one of the operations listed above, and comments in the AMASPZAP input clearly indicate the correspondence of zap text to operations.

This job is customized during the SEMJCL process, but requires a manual edit by the submitter before it can be used to enable the control operations. The extra edit provides an additional protection against accidental enabling of the control operations. Each REP statement has been commented out, so that no change will take place if you submit the job without making a conscious choice about it.

In addition to executing the #12CNTRL member, the SYMAPI_CTRL_VIA_SERVER option must be set to its default setting, ENABLE. For more information, refer to *Control Functions* on page 3-22.

If you want to enable control operations, you must:

1. Verify that the SYMAPI_CTRL_VIA_SERVER option is set to ENABLE.
2. Edit the #12CNTRL member in the RIMLIB.
3. Change each occurrence of *REP to REP (note that a blank replaces the asterisk character) that corresponds to a control operation you want to enable.

4. Submit the job. The job return code must be zero to indicate success.



CAUTION

No SAF security checks are made during the execution of control operations. By enabling them, you make it possible for open systems users to make changes to the Symmetrix configuration on your mainframe system.

You may undo the changes you made using #12CNTRL by reversing the VER and REP statements, and resubmitting the job.

This chapter discusses how to start the SYMAPI Server, how to examine its running status and adjust its behaviors, and how to stop it.

- ◆ Starting the Server 4-2
- ◆ Controlling the Server 4-5
- ◆ Stopping the Server 4-11

Starting the Server

The SYMAPI Server can be executed as a batch job step or as a started task (STC). If you choose to run the server as a batch job step, you may use the RIMLIB member #SEMAGNT as is after it has been customized for your site, or you can use #SEMAGNT as a model to create a different JCL stream. If you want to run the server as a started task, you can modify a copy of #SEMAGNT to create the catalogued procedure you will need. There is no sample procedure provided to start the server as a started task.

Running the Server as a Batch Job

Beyond the customization of #SEMAGNT, there are no unique considerations for running the server in a batch job.

Running the Server as a Started Task

To run the SYMAPI Server as a started task, you must:

1. Make a copy of #SEMAGNT and add a // PROC statement with symbolic parameters as you see fit.
2. Store the resulting JCL procedure in a catalogued procedure library.
3. Use the z/OS START command to start the procedure.

When running as a started task, the SYMAPISV program must have a userid associated with it. For details on how to set up an association between SYMAPISV and a userid, refer to *Started Task User Identity* on page 3-5.

SYMAPISV Command Operands

The SYMAPISV program recognizes several command operands. The command arguments established in the #SEMAGNT member should suffice for most operational situations. You may have additional specific needs for your site's operation, so you may have to change the samples or create new JCL members.

The SYMAPISV command operands may be preceded by runtime options recognized by IBM Language Environment. The default command options for Language environment are specified in

SEMAGNT as an empty string, resolved using the symbol &LEPARM. The &LEPARM symbol is followed by a forward slash character, which is in turn followed by the command operands that SYMAPISV will recognize. The example JCL below demonstrates this syntax, showing how you would specify a non-empty string for LEPARM. In the subsequent examples, the LEPARM is omitted for simplicity.

```
// SET LEPARM='RPTOPT(ON) '
//SEMAGNT EXEC PGM=SYMAPISV,
// PARM='&LEPARM/-port 2707 start'
```

The command operands for SYMAPISV are specified in the PARM= operand of the EXEC PGM=SYMAPISV statement in the startup JCL. The specified operands you must be concerned with are:

- ◆ `-port`
Specifies the TCP/IP port on which the server will listen for new connections. This operand is optional, and the default value is 2707.
- ◆ `-log`
Specifies a list of types of events which should be recorded in the server log file. This operand is optional, and the default value is NONE. The value of the log settings can be changed while the server is running using the SETVAR operator command.
- ◆ `start`
Indicates that the server is to be started. This operand is required.
- ◆ `-service`
Names a service name defined in the Network Services File. This file is defined in the SYM\$NETC DD statement. This operand is supported, but its use is not recommended, since its benefit accrues to open system users more than z/OS users.

Specifying the Port

The `-port` operand specifies the port on which the server will listen. If you take the default, you may omit this operand.

Usage Notes:

- ◆ The value specified must be a decimal number greater than 1000 and less than 65536.
- ◆ The port number must not be in use by any other application. The server will terminate immediately, issuing message ANR0204E if the port is in use by some other program.

Specifying the Logging Options

Example:

```
//SEMAGNT EXEC PGM=SYMAPI SV,  
// PARM='/-port 10005 start'
```

The `-log` operand specifies the types of events to record on the server log. The syntax of the `log_settings` is:

```
-log type+type+type . . .
```

Where *type* may be one of the following:

ALL	Write records for all log types to the log.
NONE	Write no records to the log.
SESSION	Show events that relate to session management – opening and closing of sessions, and other events.
APIREQ	show certain data about API request receipt, processing, and responses.
TCPIP	Show important TCP/IP events.
SERVER	Show events related to the server process as a whole.

The default value of the `-log` operand is `SERVER`; that is, server events are recorded in the log. The sample `#SEMAGNT JCL` specifies `NONE` to override the default. In initial tests using this version of the SYMAPI Server, it may be useful to specify `ALL`, so that you can become accustomed to the format and content of the log messages.

Specifying `ALL` is not recommended for long term use, as it produces a high volume of output. `APIREQ` also produces a high volume of output, since each API request writes two log records. However, it can be useful if you need to see more detail during a specific run, and it can be used in conjunction with output caused by setting the environment variable `SYMAPIDEBUG` to a non-zero number.

The `-log` operand is intended as a supplement for the environment variable `SYMAPI_NET_VERBOSE` in the context of the server. Its use is unrelated to `SYMAPI_NET_VERBOSE`, in that the verbose settings will cause output to the `SYSPRINT DD` statement based on the bit settings in the variable.

Controlling the Server

You can control the SYMAPI Server while it is running by issuing operator commands using the the z/OS system command `MODIFY` (abbreviated `F`):

```
F jobname,command
```

where:

jobname is the name of the batch job or started task under which SYMAPISV is running.

command is the text of the command passed to SYMAPISV.

The commands to control the server are described below.

Display Server

This command displays the current server settings. At a minimum, the following information will be shown:

- ◆ Complete version information of the server.
- ◆ Version information of the C runtime library in use.
- ◆ Processor type and number of CPUs.
- ◆ Number of current sessions.
- ◆ Hostname/IP address and port on which the server is listening.
- ◆ Total number of sessions handled since startup.
- ◆ Operating system name and version.
- ◆ z/OS address space ID and UNIX process ID.

SYNTAX

```
Display|Show [-]SErver
```

Display Sessions

This command displays the details of a specific session. At a minimum, the following information will be shown:

- ◆ Session number associated with this session.
- ◆ Thread number of the thread handling the session.
- ◆ Client IP address and hostname if known.
- ◆ API library version information of the client.
- ◆ Timestamp for the start of the session.
- ◆ Last request information:
 - Start and end timestamps of the last API request.
 - API request code and return code from the request. (If the request is in progress, the end timestamp and result are not displayed, and the start timestamp line displays in progress.)

SYNTAX

```
Display/Show [-]SESSion -num sess_num | -thread
    thread number
```

Sessions can be displayed individually or in a group. If the `-num` or the `-thread` operand is specified, the details about the specific session will display. If no qualifying number or thread option displays, details about all current sessions are displayed.

Display Variable

This command displays the value of a specific runtime option, or variable.

SYNTAX

```
Display/SHow [-]variable
              [-env]env variable name
```

OPTIONS

-variable

Valid values are:

$$-\log$$

The types of log entries written to the server log will be changed to the value specified. For more information, refer to *Specifying the Logging Options* on page 4-4.

```
-cond hdlr
```

Specifies whether the condition handler should be disabled for individual sessions. The valid values are ON and OFF. ON is the default and indicates that condition handling will be used for all sessions. If an abend occurs in a session, it will be intercepted and the session will be terminated. This protects the server as a whole from terminating when an abend occurs. The value OFF indicates that condition handling will be suppressed, allowing an abend to be handled by the operating system. This will cause the server address space to be terminated. `-con` is a valid abbreviation.

- env

Indicates a request to display an environment variable. The environment variable follows the keyword `-env`.

EXAMPLE

F SEMAGNT, SHOW -ENV TZ

Displays the standard TZ environment variable.

List Sessions

This command displays a list of the current sessions to the server. Minimum details about the session will be displayed. Complete information about the session can be shown using the `DISPLAY SESSION` command. The following information will be shown:

- ◆ Session number and thread number associated with this session
- ◆ Client username and hostname if known, or Client IP address if host is not known

SYNTAX

`List SESSions`

Setenv

This command will change the value of an environment variable while the server is running. This should be used for debugging purposes only, and only with the guidance of EMC Customer Support or engineering.

SYNTAX

`SETEnv variable value`

Where *variable* is the name of an environment variable to set, and *value* specifies the desired value. If the value contains imbedded blanks, and must be enclosed in quotes.

The behavior of specific environment variables will vary. Because the server and API software often inspect environment variables once, the setting may not have immediate effect.

Setvar

This command will change the value of a current execution option. The types of options which can be set are those for whom it makes sense to change the value. For example, it might be desirable to change the `-log` operand during server execution. But it does not make sense to change the value of the `-port` option.

The first argument to set is the name of the option to be changed, followed by the value that it is to be set to. The syntax of the value is the same as can be specified in the SYMAPISV command line.

SYNTAX

`SETVar option value`

OPTIONS*option*

Valid values are:

-log

The types of log entries written to the server log will be changed to the value specified. For more information, refer to *Specifying the Logging Options* on page 4-4.

-cond_hdlr

Specifies whether the condition handler should be disabled for individual sessions. The valid values are ON and OFF. ON is the default and indicates that condition handling will be used for all sessions. If an abend occurs in a session, it will be intercepted and the session will be terminated. This protects the server as a whole from terminating when an abend occurs. The value OFF indicates that condition handling will be suppressed, allowing an abend to be handled by the operating system. This will cause the server address space to be terminated. *-con* is a valid abbreviation.

Shutdown

This command shuts down the server address space and is an alternative to the z/OS STOP command.

SYNTAX

SHUTDOWN [NORmal | IMMEDIATE]

When no value is specified, the default is a normal shutdown. This command should always be used instead of the cancel command. No new sessions can be started and sessions in progress are allowed to continue. The shutdown will complete when the last active session has issued its SymExit.

OPTIONS

NORmal

Should be used when sessions are active and you need to shutdown the server.

IMMEDIATE

This command will terminate all sessions regardless of their state. Since it does not abort the tasks in a orderly manner, it is intended to be used when one or more tasks are hung.

Special Circumstances during Server Execution

Condition Handling

Normally, the server establishes a Language Environment condition handler for each session when the thread initializes. The condition handler is established to intercept abend situations, produce a system dump of the thread, and terminate the thread. The system tries to terminate the thread normally, although the client side will usually receive a SYMAPI return code that indicates the connection to the server was aborted. In this situation, the thread handling the session is terminated, but other threads are allowed to continue their work. The server continues to run, accept connections from new clients, and accept commands from the operator.

When a thread abend occurs, the condition handler receives control. It issues message ANR0222S to the system console, and writes the message to the server log, indicating it will produce a dump. The dump is written to a dynamically allocated DD statement and goes to the system spool. The format of the dump DDname is *DMPnnnnn* where *nnnnn* is the thread number on which the abend occurred. The thread number is always the same as the session number, so it will be fairly easy to associate the dump with a specific session's activity.

When the dump is complete, message ANR0223I is issued, indicating the thread will be terminated. The thread goes through a normal termination process, and the TCP connection to the client is broken.

Disabling Condition Handling

Condition handling is on by default for all sessions. Some circumstances may require condition handling to be disabled. Certain environmental conditions or unusual kinds of abends may be easier to diagnose with a standard system dump instead of a dump formatted by the Language Environment library.

You may disable condition handling by using the SYMAPI SV command `setvar -cond_hdlr off`. When you turn condition handling off, all new threads are started without the condition handling protection. When condition handling is disabled, message ANR0220I is written to the server log, indicating that the thread is running without condition handling.

If an abend occurs on any thread, the system terminates the entire address space with a system dump. All sessions are terminated, and new attempts to connect to the server will fail, since the server is no longer executing.

You must disable condition handling only when absolutely necessary, usually with the guidance of and request from EMC Customer Support. Sometimes it is easier leave condition handling on, and to establish a SLIP (Service Level Indication Processing) trap and let the system produce an SVCDUMP. SYMAPISV condition handling and SLIP traps can operate together, independently of each other. In this way, you can produce a dump for EMC Customer to analyze which may have better information than the LE dump, but you can also preserve the server operation for other threads.

In very rare cases, you may be asked to disable condition handling for the server address space. This may be necessary in order to produce a dump formatted by the system instead of an LE dump of the address space. This also should be done only at the request of EMC Customer Support.

If Language Environment condition handling is disabled at the job step level, thread condition handling is also disabled. In this case, message ANR0221E is written to the log and system console.

Stopping the Server

There are several methods for stopping the SYMAPI Server, depending on whether it is running as a batch job, or as a started task.

Stopping a Server Running as a Batch Job

If the server is running as a batch job, the following methods are available:

- ◆ Use the SYMAPI SV `SHUTDOWN` command to initiate a `NORMAL` or `IMMEDIATE` shutdown. For more information, refer to *Shutdown* on page 4-8.
- ◆ Use the z/OS `CANCEL` command to cancel the batch job. This is highly undesirable, and should be used only as a last resort (if the `SHUTDOWN` command does not have the desired result). Cancelling the job may result in a corrupt SYMAPI database.

Stopping a Server Running as a Started Task

If the server is running as a started task, the following methods are available:

- ◆ Use the SYMAPI SV `SHUTDOWN` command as in the batch job case.
- ◆ Use the z/OS `STOP` command (abbreviation `P`). The `STOP` command is treated as if `SHUTDOWN NORMAL` were entered. There is no way to initiate an immediate shutdown using the `STOP` command. You must use the SYMAPI SV `SHUTDOWN` command.

Considerations When Stopping the Server

Consider the following when determining a method for stopping the server:

- ◆ If there are sessions active, it may be undesirable to request an immediate shutdown of the server or to cancel the job. If any sessions are running Symmetrix control operations, causing those sessions to terminate prematurely may leave locks held on the Symmetrix or may leave the Symmetrix in an inconsistent state.

- ◆ Once a normal shutdown has begun, the server will wait for all sessions to terminate. There is no way to re-issue the `SHUTDOWN` command with `IMMEDIATE` operand. You must issue the system `CANCEL` command to stop a batch job, or the `STOP` command to stop a started task.
- ◆ If you are shutting the server down because you must schedule a system IPL, make sure the server has stopped before bringing down TCP/IP. If TCP/IP is brought down before the server, unpredictable results, including occasional client hangs may occur.

Solutions Enabler Messages

This appendix describes the EMC Solutions Enabler for mainframes messages that can be returned when running the Solutions Enabler (SYMAPI) Server. The messages are listed in numerical order within the following categories:

- ◆ Message Format A-2
- ◆ Messages..... A-4

Message Format

This section described messages that are written to the SYMAPI Server log (see *Server Log* on page 3-13) and to the system console. All messages begin with a message identifier, followed by text of the message.

When messages are written to the server log, the message identifier is prefixed by a date/time stamp, server address space identifier (ASID), thread number of the thread issuing the message, and a log type which corresponds to a value specified by the `-log` runtime option. When messages are written to the system console, the prefix area is not shown.

The message is in this format:

`yyyy/mm/dd hh:mm:ss Asid thread_id log_type msgid text`

where:

<code>yyyy/mm/dd</code>	Is the date the message was issued.
<code>hh:mm:ss.xxx</code>	Is the time the message was issued in hours, minutes, seconds, and milliseconds.
<code>Asid</code>	Is the Address space ID (ASID) of the issuing process. In future releases this may be changed to the USS PID of the process, depending on whether the server is running as a traditional batch address apace or as a USS process.
<code>thread_id</code>	Is the thread number of the issuing thread. This will be zero on non-threaded platforms.
<code>log_type</code>	Is the type specified in the <code>-log</code> operand which caused this message to be generated. The valid types are: SERVER, SESSION, TCPIP, and APIREQ. The type may appear as "*****." In this case, the server requested that a message be printed regardless of the selected log types specified by the <code>-log</code> operand.
<code>msgid</code>	Is made up of the following: ANR — Indicates the server issued the message. nnnn — A numeric identifier for the message. X — A one byte severity indicator. Valid values are: I indicates an Informational message E indicates an Error message S indicates a severe condition requiring a message
<code>text</code>	Is the message text.

In this section, each message shows the text of the message with indicators where substitutions are made into the text at runtime. Following the text are four paragraphs giving more information:

- ◆ *Set Step Return Code* — If this appears, it means that the issuing of the message can cause the job step return code to be set. See the table below for the correlation of message severity to job step return code. Some messages are issued by multiple locations in the code. Not all uses of the message will cause the step return code to be set.
- ◆ The *Destination* of the message — **Log** and/or **Console** is shown. Most messages are written to the server log file. Some messages are written to both the log and console, but not in all cases where the message is generated. Some messages are written to the system console only, particularly those related to operator command processing.
- ◆ The *Description* paragraph explains the circumstances that cause the message to be issued, and explains each substituted value. This section also describes any action that the Solutions Enabler software will take.
- ◆ The *Operator Action* paragraph suggests operator intervention actions where needed.

Some messages will cause the SYMAPI Server job step return code to be set to a non-zero value. This behavior is indicated in the *Set Step Return Code* paragraph for each message. The list below shows the mapping of message identifiers to job step return codes:

Message Identifier	Return Codes
I	0
W	4
E	8
S	12

If multiple messages are issued that cause the step return code to be set, the highest value will be remembered by the server, and returned to the system at job termination.

Messages

ANR0000I*text***Destination:** Log and Console.**Description:** This message is a general purpose message to be used for any arbitrary *text*.**Operator Action:** None.**ANR0001I**

SYMAPI Server for z/OS ready to accept connections

Destination: Log and console.**Description:** This message is issued when initialization is complete and the server is prepared to field connection requests from remote clients.**Operator Action:** None.**ANR0002I***shutdown_type* Shutdown requested by *source***Destination:** Log and console.**Description:** This message indicates that a shutdown request was made. See message ANR0002I for the description of *shutdown_type*. The *source* is always shown as 'system operator'.**Operator Action:** None.**ANR0003I***shutdown_type* Shutdown *progress*. Number of sessions remaining = *number***Destination:** Log and console.**Description:** This message is issued at the start of the shutdown process. *shutdown_type* indicates NORMAL, IMMEDIATE, or STOPPED-NORMAL. The system operator will request a NORMAL shutdown using the z/OS STOP command or the Solutions Enabler SHUTDOWN command. If STOPPED-NORMAL is indicated, a specific connection to the server requested that the server be shut down.The number of currently active sessions is shown in *number*. If this value is not 0, the following rules apply:

- ◆ If the *shutdown_type* is NORMAL, the server will wait for the active sessions to end. In this case, *progress* indicates *starting* or *in progress*. Each time a session ends, the *in progress* status will be reported.
- ◆ If the *shutdown_type* is IMMEDIATE, the server terminates without waiting for active sessions to end. See the description of the SHUTDOWN command for more details on when to use IMMEDIATE shutdown.

Operator Action: None

ANR0004I SYMAPI Server running as a started task

Destination: Log.

Description: The server knows when it is running as a started task (running in *STC mode*). This message serves as a visual confirmation that STC mode is active.

Operator Action: None, unless this is not what is intended.

ANR0005E Normal shutdown failed, attempting immediate shutdown

Set Step Return Code

Destination: Log and console.

Description: The server attempted to perform a normal shutdown, waiting for active sessions to complete. The normal shutdown process failed, and no recovery was possible. An immediate shutdown was attempted, because there is no other possible recovery action to take.

Operator Action: Be aware that the list of connections noted in message ANR0013I will be terminated before they are able to disconnect.

ANR0006E Wait returned without connection or console command ready, console ECB contents *value*

Destination: Log.

Description: The server waits for incoming connection requests and instructions from the operator concurrently. If the wait is somehow satisfied but neither of these events occurred, it is considered an error. The server will continue to wait for new events.

Operator Action: This is an abnormal situation and may indicate some error in TCP communications or management of the operator

console. If this happens repeatedly, shut the server down. Try restarting the server. If the problem persists, examine your system for evidence of other problems in the TCP or console management components of your system.

ANR0008I Server socket *socket*: *event* occurred

Destination: Log.

Description: This message is issued to confirm that connection request has arrived, or that some error condition has been reflected to the TCP socket on which the server is listening. The value of *socket_event* will be *connection request* or *exception condition*.

Operator Action: If the *socket_event* is *connection request* no action is necessary since this is a documentation message, and may aid in problem diagnosis. See the description of message ANR0009E, if the *socket_event* is *exception condition*.

ANR0009E Exceeded maximum exceptions on server socket, indicating PORT_EXCEPTION

Set Step Return Code

Destination: Log and console.

Description: This is issued after an exception condition has been raised (which may cause the issuing of ANR0008I). Currently, the maximum exception count is 1, meaning that there is no retry strategy when an exception occurs on the socket on which the server is listening. The server will stop listening and start a NORMAL shutdown when it notices this condition.

Operator Action: If *exception condition* in message ANR0008I is indicated, there will be other evidence in your system log showing TCP/IP problems. Use documentation from your TCP software provider to resolve the problems you find. When the problems are resolved, you can restart the server.

ANR0010I SYMAPI Server Shutdown complete

Destination: Log and console.

Description: The server has completed its shutdown process and will return to the operating system.

Operator Action: None. This should serve as a visual confirmation that the server is finished.

ANR0011E SYMAPI Server not executing from an APF-authorized library, cannot continue

Set Step Return Code

Destination: Log and console.

Description: The SYMAPISV program must execute from a library authorized by the z/OS Authorized Program Facility. The server checks to make sure that this condition is met and terminates immediately if it is not.

Operator Action: The Solutions Enabler load library can be authorized through APF in several ways. You can use the SETPROG APF command to authorize the library temporarily. In order to make the library authorized at subsequent IPLs, you must edit the PROGxx member of SYS1.PARMLIB. Refer to the IBM documentation for your level of OS/390 or z/OS to for exact syntax and editing instructions.

ANR0012I Accepted session *session_number* from *IP_address* on thread *thread_number* (Tid *thread_identifier*)

Destination: Log.

Description: The server successfully handled a connection request for a session, and started a thread to process API requests for the session. The session number is shown in *session_number* and it is being processed on a thread with the number *thread_number* and physical thread identifier *thread_identifier*. The session is running from a client program executing on the host at address *IP_address*.

Operator Action: None necessary. This message is documenting the start of a session. You should also see ANR0017I at the end of the session.

ANR0013I Shutdown will wait for client session *session_number* to *IP_address* on Tid *thread_identifier*}

Destination: Log and console.

Description: During a normal shutdown, the server will wait for all active sessions to terminate on their own. For each session still active, the server issues this message and will wait for the sessions to end. The substitution variables are the same as those in message ANR0012I.

Operator Action: None usually. If sessions are taking an excessive amount of time to complete, you may cancel the server address space. Once shutdown has begun, there is no way to reissue the shutdown

command with the IMMEDIATE operand. If the shutdown process waits for a very long time because a connection has not exited, you must cancel the job or encourage users of the active connections to terminate them.

ANR0014W Terminating client session *session_number* to *IP_address* on *Tidthread_identifier*}

Destination: Log and console.

Description: During an immediate shutdown, the server will report on all active sessions at the time the shutdown process begins. For each session still active, the server issues this message as a note to the operator to indicate which sessions will be terminated. The substitution variables are the same as those in message ANR0012I.

Operator Action: None.

ANR0015I Session *session_number* on thread *thread_identifier* ended with return value *return_value*

Destination: Log.

Description: This message is issued during a normal shutdown when the server is waiting for a session to end. The substitution variables are the same as those in message ANR0012I.

Operator Action: None.

ANR0016I Main thread is number *thread_number* (Tid *thread_identifier*)

Destination: Log.

Description: This message is issued during startup simply to report the thread number (*thread_number*) and thread identifier (*thread_identifier*) of the main thread (the server thread which listens for new connection requests and operator instruction).

Operator Action: None.

ANR0017I Ending session *session_number* from *IP_address* on thread *thread_number* (Tid *thread_identifier*)

Destination: Log.

Description: This message documents the end of a session. The substitution variables are the same as those in message ANR0012I.

Operator Action: None.

ANR0018E Rejecting session *session_number* for user *user_name* from *IP_address*
reason: *reason*

Destination: Log.

Description: A remote client attempted to connect to the server, and the server is running with the SYM\$NETH (the nethost file) allocated to the job. The combination of the user identification (*user_name*) and the host address (*IP_address*) were not specified in the nethost file. The session is counted in spite of being rejected by the server. The remote client SymInit calls will return to the caller with SYMAPI_C_HOST_FILE_REJECTION.

If the syntax of the statements in the nethost file is incorrect, unformatted, unnumbered messages are also written to the SYSPRINT DD statement. In this case, the statement number and the reason for the syntax error will be reported, aiding in correction of the statement.

Operator Action: The remote client user will probably ask for authorization to use the SYMAPI Server, or the nethost file syntax is incorrect. In either case, refer to *Trusted Host File* on page 3-18 for instructions on the syntax.

ANR0100I Command: *command_text*

Destination: Log and console.

Description: This message is issued when an operator command is directed to the server. The text of the command entered should be identical to the text *command_text*. This serves as a visual confirmation of the operator's intent.

Operator Action: None. If *command_text* is not identical to the command entered, a programming error occurred.

ANR0101I STOP command entered

Destination: Log and console.

Description: The operator entered the z/OS STOP command to initiate a NORMAL shutdown. This message serves the same purpose as ANR0100I, except that there is no additional command text to show.

Operator Action: None.

ANR0102E Ignoring invalid console command indicator *command_number*

Destination: Log and console.

Description: The server found an invalid value in the command indication provided by the system console interfaces. The invalid value was ignored.

Operator Action: None.

ANR0103E Console interface error function *console_interface_name*, return code *return_code* system return code, running without operator console active

Destination: Log and console.

Description: The server attempted to use a system interface function to communicate with the operator console. The call to the function (*console_interface_name*) failed with the return code (*return_code*). The server attempts to run without the operator console interface active. The operator will not be able to send commands to the server, but the server will be able to process connections from remote clients.

Operator Action: This probably indicates an error in the system. The routine (*console_interface_name*) is a Solutions Enabler function name, and it failed because of some system condition. Other messages written to the operator console by the system will provide other evidence of system failures, particularly related to console services. Use the other messages to diagnose the system condition causing this failure.

ANR0104E Command syntax error: *explanation*

Destination: Console.

Description: The operator entered a command with invalid syntax explained by *explanation*.

Operator Action: Examine the syntax description for the command you want to enter, and re-enter it with the proper operands.

- ANR0105E** Ambiguous or invalid command token entered *token_text*
Destination: Console.
Description: The operator entered a command but either the command verb or a keyword name in *token_text* was misspelled or its abbreviation was too short to uniquely identify the intent.
Operator Action: Examine the syntax description for the command you want to enter, and re-enter it with the proper operands.
- ANR0106I** Environment variable *name* has been set to *value*
Destination: Console.
Description: The operator entered the SETENV command, and the environment variable was successfully set.
Operator Action: None. This message provides confirmation that the variable was set as intended.
- ANR0107E** *option* is not a valid runtime option
Destination: Log and console.
Description: The operator entered the SETVAR command, but the name of the runtime option (*option*) was not recognized as a valid option.
Operator Action: Examine the description of the SETVAR command for the supported options. Re-enter the command with the desired option. SETVAR accepts the runtime option names with or without the dash prefix.
- ANR0108E** *value* is not a valid value for runtime option *option*
Destination: Log and console.
Description: The operator entered the SETVAR command with the name of a valid run runtime option (*option*), but the value (*value*) specified for *option* was not valid.
Operator Action: Examine the description of the SETVAR command for the proper values corresponding to each supported option. Re-enter the command with the corrected value for the desired option.

- ANR0109E** *value* is not a valid -log type
- Destination:** Log and console.
- Description:** This message serves a purpose similar to ANR0108E, but is more specific to the -log runtime option. The value (*value*) specified in the list of log types was not valid.
- Operator Action:** Examine the description of the SETVAR -log command for the proper values corresponding to the -log option. Re-enter the command with the corrected value. There are several abbreviations accepted for each log type supported.
- ANR0110E** Invalid *option* command option name found following successful parse: decimal value is *code_value*
- Destination:** Console.
- Description:** This message indicates a programming or environmental error in command parsing and execution. The parsing of the command was successful, but the secondary scan performed by the execution phase found an invalid token.
- Operator Action:** Collect and provide documentation as directed by EMC Customer Support.
- ANR0111I** *option* runtime option has been set to *value*
- Destination:** Log and console.
- Description:** The operator entered the SETVAR command to change the value of the runtime option *option*. The command text was successfully parsed, and the command was executed successfully. The new value of the variable is *value*.
- Operator Action:** None.
- ANR0112I** *command_name* command requires additional operands: *explanation*
- Destination:** Console.
- Description:** The operator issued command (*command_name*) without sufficient operands. Default processing could not be established. *explanation* gives details on what is required.
- Operator Action:** Re-enter the command with desired operands.

ANR0113I *option* current value: *value*

Destination: Console.

Description: This message is issued by the DISPLAY or SHOW command for a runtime option. The *option* is the runtime option specified in the SHOW command, and its current setting is *value*.

Operator Action: None. The operator may issue this command before changing the value of a runtime option, or may want to confirm its value after setting it (although message ANR0111I can be used for the latter purpose).

ANR0114I *environment_variable* is currently not set

Destination: Console.

Description: The operator entered the SHOW -ENV command to display the value of an environment variable. The variable has not been set.

Operator Action: None.

ANR0115I *environment_variable* is set to an empty value

Destination: Console.

Description: The operator entered the SHOW -ENV command to display the value of an environment variable. The variable is set in the environment of the server, but the value is the empty string.

Operator Action: None.

ANR0120I Symapi Active Session List:

Destination: Console.

Description: The operator issued the LIST SESSIONS command. There are active sessions to list. This messages is the heading for the list of sessions which follows.

Operator Action: None.

ANR0121I No active sessions found.

Destination: Console.

Description: The operator issued the LIST SESSIONS command or the SHOW SESSION command. No sessions were active to list/show.

Operator Action: None.

- ANR0122I** Session/Thread *number* is not active
- Destination:** Console.
- Description:** The operator issued the SHOW SESSION command with the -NUM or -THREAD option to display a specific session. If SHOW SESSION -NUM was entered, then *number* is a session number, and that session was not active. If SHOW SESSION -THREAD was entered, then the thread indicated by *number* was no longer active.
- Operator Action:** None.
- ANR0123I** Show Server Details:
- Destination:** Console
- Description:** The operator issued the SHOW -SERVER command to display the details for the server. This line is written to mark the beginning of the server details output.
- Operator Action:** None.
- ANR0124I** Show Session details for Session *session_number* on Thread *thread_number*:
- Destination:** Console.
- Description:** The operator issued the SHOW SESSION command to display details of one or more currently active sessions. This line is written at the beginning of the details for each session to be displayed.
- Operator Action:** None.
- ANR0140E** Secure sessions are not supported. The options or netcnfg file specified security level security_level
- Set Step Return Code**
- Destination:** Log and Console.
- Description:** Secure sessions are not supported in the z/OS environment. This message is issued when either the SYMAPI options file (DD SYM\$OPT) or the server network configuration file (DD SYM\$NETC) requests a security level which is not supported. In the case of the options file, the SYMAPI_SERVER_SECURITY_LEVEL statement specified ANY or SECURE. In the case of the network configuration file, the service name definition specified ANY; or

SECURE in the security level token (the seventh token in the definition).

Operator Action: If SYMAPI_SERVER_SECURITY_LEVEL statement is used, it must specify NONSECURE, although it is better to omit the statement altogether. If the service definition in the network configuration file is used, the security level token must be omitted, or must be set to a hyphen (-) or NONSECURE.

ANR0200E

service_name error return_code: explanation; from calling_routine, line line_number

Destination: Log.

Description: Server logic called the routine named by *service_name* and received a failure indicated by *return_code*, where *explanation* is text that corresponds to the *return_code*. The failure was detected at line *line_number* in the routine *calling_routine*. The routine *calling_routine* was not able to continue due to the failure of *service_name*

Operator Action: None, generally. This message may occur in very rare circumstances during handling of an operator command, and may indicate a syntax error that was not handled properly by parsing logic. Examine the command and reissue it if it was specified incorrectly.

ANR0201E

Unable to allocate *count* bytes for *object_name*

Destination: Log.

Description: The server attempted to allocate *count* number of bytes. *object_name* is a description of what the server was trying to allocate. This is an indication that the REGION= specification was not sufficient to support the profile of work you expect the server to handle. This message may also be a symptom of a memory leak in the system.

Operator Action: Increase the REGION= or specify REGION=0M. If this does not solve the problem, a memory leak may be indicated by other messages about failures. Collect and forward error documentation to EMC Customer Support for analysis.

ANR0202E Unable to *operation_name* port *port*, error *error_number* indicates *explanation*

Set Step Return Code

Destination: Log and console.

Description: An error occurred operating on the socket on which the server listens for new connections. *operation_name* will indicate an error during bind, listen, initialize, accept, or start new thread. The *error_number* is the decimal value of the system variable *errno*, and the *explanation* is the text that explains the meaning of *error_number*. *port* is the TCP/IP port which clients use to connect to the Solutions Enabler server. The server shuts down after issuing this message.

Operator Action: In most cases, other messages will also be issued giving other details about an error situation. Follow your normal procedures for detecting and correcting problems in your TCP/IP network. Correct the TCP/IP problem and restart the server.

ANR0203E TCPIP select error occurred: *error_number* (*explanation*)

Set Step Return Code

Destination: Log.

Description: The server uses the **select()** system call to handle connection requests and operator commands concurrently. The call to **select()** returned with an error condition.

Operator Action: This may be an indication of an error in the TCP/IP software on your system. Follow your normal procedures for detecting and correcting problems in your TCP/IP network. If the problem persists, the Solutions Enabler server will shut itself down. If this happens other messages will be issued to the console. Correct the TCP/IP problem and restart the server.

ANR0204E Unable to decode return value *return_value* from *process*

Set Step Return Code

Destination: Log and console.

Description: The *return_value* from a call to a routine or other logic could not be interpreted. *process* may be the name of the function or may be a general description of processing that resulted in a return value which could not be interpreted.

Operator Action: None. This message will be preceded by other error messages that explain more detail. If your normal processing is

unaffected, no action is necessary. Otherwise, you may need to collect and provide documentation as directed by EMC Customer Support.

ANR0205E *action* is not currently supported

Destination: Log and console.

Description: An action or feature was requested that is either not yet supported or is no longer supported. The name of the action or feature not supported is *action*.

Operator Action: None. The feature you requested is not available for use in this release. If you receive this message in error, examine the job log for other evidence of a failure which may be related to the action or feature you attempted to use.

ANR0220I Thread *thread_number* will execute without condition handling protection

Destination: Log.

Description: The thread *thread_number* will be executed without the protection of a condition handler. The **setvar -cond_hdlr OFF** command had been previously issued, causing condition handling suppression. This message is a confirmation that the session will be run without protection. An abend on the thread will cause the operating system to terminate the server address space.

You can associate the thread number with a session number by using the LIST SESSIONS command. The second column of the list sessions output is the thread number of the session.

Operator Action: None.

ANR0221E Unable to set condition handling for thread *thread_number*, msgno=*LE_message_num*, sev=*LE_severity*

Destination: Log and console.

Description: The thread (*thread_number*) handling a session attempted to set condition handling by calling the Language Environment routine CEEHDLR but received a non-zero return value from the call. The Language Environment feedback message number is shown in *LE_message_number* and the severity of the return is shown in *LE_severity*.

Operator Action: None.

- ANR0222S** Condition on thread *thread_number*; writing dump to DD *dump_location*
- Destination:** Log and console.
- Description:** An abnormal condition was raised during the session running on thread number *thread_number*. A dump will be written to the DD name *dump_location*. The general format of the DD name is 'DMPnnnnn' where *nnnnn* is the *thread_number*.
- Operator Action:** Consult EMC Customer Support for directions on complete documentation to provide for analysis and correction.
- ANR0223I** Dump to *dump_location* is complete; thread *thread_number* will be terminated
- Destination:** Log and console.
- Description:** Condition handling processing detected a recursive (or simply a second) entry into the condition handling routine. This may indicate an abend while attempting to handle an earlier abend.
- Operator Action:** None.
- ANR0224S** Recursive entry to condition handler on thread *thread_number*
- Destination:** Log and console.
- Description:** Condition handling processing detected a recursive (or simply a second) entry into the condition handling routine. This may indicate an abend while attempting to handle an earlier abend.
- Operator Action:** None.
- ANR0300E** API Request code *SYMAPI_request_code* rejected because it is restricted and disabled
- Destination:** Log.
- Description:** A SYMAPI request code that describes a control operation was received. The server checked the *SYMAPI_request_code* to determine whether execution has been disabled. The API request was found to be disabled.
- Operator Action:** None. Control operations can be selectively enabled but using the installation job #12CNTRL in the Solutions Enabler RIMLIB dataset.

- ANR0301I** API Request code *SYMAPI_request_code* to be dispatched
Destination: Log.
Description: The server received a SYMAPI Request code described by the decimal code *SYMAPI_request_code*. This message is issued when the server begins the execution of the API request.
Operator Action: None.
- ANR0302I** API Request code *SYMAPI_request_code* complete, processing status *SYMAPI_return_code (explanation)*
Destination: Log.
Description: The API request named in message ANR0301I completed executing. The decimal code of *SYMAPI_request_code* corresponds to the API request code. The return value of the API request was *SYMAPI_return_code*, and the corresponding text is *explanation*.
Operator Action: None.
- ANR0303I** Executing SymExit to clean up (client exited without calling SymExit)
Destination: Log.
Description: The client application exited its process before calling SymExit to end the remote session with the SYMAPI Server. The server calls SymExit on behalf of the client to free up resources which are still held.
Operator Action: None.
- ANR0304I** Cleanup SymExit return: *return_value (explanation)*
Destination: Log.
Description: The cleanup call to SymExit completed, and the return value was *return_value*. The *explanation* is the text associated with *return_value*.
Operator Action: None.

Symbols

#01ALLOC 2-10
 #02DFZON 2-10
 #03REPRO 2-10
 #04DDDEF 2-10
 #05RECEV 2-10
 #06APPLY 2-10
 #08SLMF 2-10
 #09IVP 2-12
 #10ECCIN 2-12
 #11ACCPY 2-13
 #12CNTRL 2-13

A

ANR0000I A-4
 ANR0001I A-4
 ANR0002I A-4
 ANR0003I A-4
 ANR0004I A-5
 ANR0005E A-5
 ANR0006E A-5
 ANR0008I A-6
 ANR0009E A-6
 ANR0010I A-6
 ANR0011E A-7
 ANR0012I A-7
 ANR0013I A-7
 ANR0014W A-8
 ANR0015I A-8
 ANR0016I A-8
 ANR0017I A-8
 ANR0018E A-9
 ANR0100I A-9

ANR0101I A-9
 ANR0102E A-10
 ANR0103E A-10
 ANR0104E A-10
 ANR0105E A-11
 ANR0106I A-11
 ANR0107E A-11
 ANR0108E A-11
 ANR0109E A-12
 ANR0110E A-12
 ANR0111I A-12
 ANR0112I A-12
 ANR0113I A-13
 ANR0114I A-13
 ANR0115I A-13
 ANR0120I A-13
 ANR0121I A-13
 ANR0122I A-14
 ANR0123I A-14
 ANR0140E A-14
 ANR0200E A-15
 ANR0201E A-15
 ANR0202E A-16
 ANR0203E A-16
 ANR0204E A-16
 ANR0205E A-17
 ANR0220I A-17
 ANR0221E A-17
 ANR0222S A-18
 ANR0223I A-18
 ANR0224S A-18
 ANR0300E A-18
 ANR0301I A-19
 ANR0302I A-19

ANR0303I A-19
 ANR0304I A-19
 ANRLOG DD statement 3-9
 APARS 2-3
 avoid file 2-13, 3-9, 3-16, 3-17

C

CA TCP access support 3-10
 comments xii
 Computer Associates TCP access support 2-4
 concurrent sessions, specifying the maximum
 number of 3-21
 condition handling 4-9
 disabling 4-9

D

dataset names 2-3
 DD statements 3-9
 Display Server command 4-5
 Display Sessions command 4-5
 Display Variable command 4-6
 distribution libraries 2-3

E

environment file 3-20
 environment variable, changing the value of 4-7
 environment variable, environment file 3-20
 execution option, changing the value of 4-7

F

FMID functions 2-3
 function management ID 2-2

G

gatekeeper devices 3-14

H

HOLDDATA 2-10

I

installation tape 2-2
 installation verification program 2-12
 installation, starting over 2-14

J

JCL 2-5
 sample startup 2-4

L

license keys 2-10
 License Management Facility 2-3
 using 2-10
 List Sessions command 4-7
 logging options, specifying 4-4

M

messages A-1

N

network services file 3-15, 3-18
 sample 2-4

O

options file 3-21
 removing a default option 3-21

R

RIMLIB dataset 2-3, 2-5
 runtime option, displaying the value of 4-6

S

security preparation 3-5
 selection file 3-9
 SEMAGNT job 2-13
 server address space, shutting down 4-8
 session details, displaying 4-5
 session number 3-3
 sessions, displaying 4-7
 Setenv command 4-7
 Setvar command 4-7
 Shutdown console command 4-8
 SRDF-TimeFinder Manager operations 1-2, 2-3,
 2-13
 STEPLIB DD statement 3-9
 STOP command 4-8
 SYM\$AVD 3-10
 SYM\$AVD file 2-13, 3-9, 3-16

SYM\$DB database 3-15
 SYM\$ENV 3-10
 SYM\$ENV DD statement 3-9
 SYM\$GAVD 3-10
 SYM\$GAVD file 3-9, 3-17
 SYM\$GSEL DD statement 3-17
 SYM\$GSEL file 3-9
 SYM\$INQ 3-10
 SYM\$INQ file 3-9, 3-16
 SYM\$LIC DD statement 3-9, 3-15
 SYM\$LOG DD statement 3-9, 3-15
 SYM\$NETC DD statement 3-9, 3-15
 SYM\$NETC file 3-18
 SYM\$NETH 3-18
 SYM\$NETH DD statement 3-9
 SYM\$OPT 3-10
 SYM\$OPT DD statement 3-9
 SYM\$OPT file 3-15, 3-21
 SYMAPI database initialization 2-3
 SYMAPI database support 3-11
 SYMAPI environment variable definitions,
 samples 2-4
 SYMAPI files 3-15
 SYMAPI Server
 condition handling 4-9
 controlling 4-5
 displaying current settings 4-5
 restricting access to 3-18
 running as a batch job 4-2
 running as a started task 4-2
 security preparation 3-5

 starting 4-2
 SYMAPIInn database 3-9
 SYMAPI\$V command operands 4-2
 SYMDB 3-11
 SYMENV00 file 3-20
 SYMLMF 2-3
 SYMLMF, using 2-10
 symmir command options
 instant 3-21
 wait for background split 3-21
 SYSMODS 2-10
 SYSOUT DD statement 3-9
 SYSPRINT DD statement 3-9

T

target libraries 2-3
 TCP/IP 3-10
 thread number 3-3
 time zone, configuring for local time 3-20
 TSO edit macro
 sample 2-4

U

user identity, associating with the SYMAP Server
 3-5

V

VOLSER format 2-2

