EMC® Solutions Enabler

Version 7.1.1

Installation Guide

P/N 300-008-918
REV A03

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

Published April, 2010

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.

For the most up-to-date regulatory document for your product line, go to the Technical Documentation and Advisories section on EMC Powerlink.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Contents

Appendix B Asynchronous Events

Appendix C Using Daemons

Appendix D EMC Solutions Enabler ConfigChecker

Appendix E UNIX Native Installation Support

Figures

Tables

Preface

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.

This guide provides installation procedures for installing the EMC Solutions Enabler software for your specific platform. The EMC Solutions Enabler software provides your host system with an API shared library and a special command set that comprises the Symmetrix Command Line Interface (SYMCLI). (For the z/OS platform, only the SYMAPI server is available.)

Related documentation

Related documents include:

- *EMC TimeFinder/Integration Modules Product Guide*
- *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*
- *Solutions Enabler SYMCLI Command Reference HTML Help*
- EMC host connectivity guides

Conventions used in this manual

EMC uses the following conventions for special notices.

Every use of the word SYMCLI means Solutions Enabler.

Note: A note presents information that is important, but not hazard-related.

CAUTION

A caution contains information essential to avoid data loss or damage to the system or equipment. The caution may apply to hardware or software.

IMPORTANT

An important notice contains information essential to operation of the software. The important notice applies only to software.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, filenames, functions, utilities • URLs, pathnames, filenames, directory names, computer names, links, groups, service keys, file systems, notifications
Bold:	Used in running (nonprocedural) text for: • Names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system call, man pages Used in procedures for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • What user specifically selects, clicks, presses, or types
Italic:	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis (for example a new term) • Variables
`Courier:`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text.

Courier bold:	Used for: • Specific user input (such as commands)
Courier italic:	Used in procedures for: • Variables on command line • User input variables
< >	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections - the bar means “or”
{ }	Braces indicate content that you must specify (that is, x or y or z)
...	Ellipses indicate nonessential information omitted from the example

Where to get help

Product information — For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Powerlink website (registration required) at:

`http://Powerlink.EMC.com`

Technical support — For technical support, go to EMC Customer Service on Powerlink. To open a service request through Powerlink, you must have a valid support agreement. Please contact your EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this document to:

`techpub_comments@EMC.com`

1

Pre-install Considerations

This chapter explains the tasks that you should perform before installing Solutions Enabler:

Introduction

An EMC® Solutions Enabler install provides your host with SYMAPI, CLARAPI, and STORAPI shared libraries for use by Solutions Enabler applications, and the Symmetrix® Command Line Interface (SYMCLI) for use by Storage Administrators and Systems Engineers.

SYMCLI is a specialized library of UNIX-formatted commands that can be invoked one at a time. It supports single command line entries and scripts to map and perform control operations on devices and data objects toward the management of your storage complex. It also monitors device configuration and status of devices that make up the storage environment. The target storage environments are typically Symmetrix, but can be CLARiiON® when you have a license and work with the mapping SRM component.

Solutions Enabler includes a monitoring option for situations on certain hosts where you need to limit the SYMCLI actions to monitor mode only and block all SYMCLI activity from executing any control actions or ones affecting any changes to the Symmetrix array.

Before you begin

Before you begin to install Solutions Enabler, be sure to complete the tasks listed in this section.

General tasks

The following tasks apply to all supported platforms:

- ❑ Obtain the software. Solutions Enabler is distributed in the following forms:
  - On the Solutions Enabler installation disc, which includes kits for all supported platforms.
  - As a platform-specific file download from the Powerlink® website at http://Powerlink.EMC.com
- ❑ Review the interoperability information in the E-Lab™ Interoperability Navigator which can be reached at http://elabnavigator.EMC.com
- ❑ Review the *EMC Solutions Enabler Release Notes*.
- ❑ If you are upgrading from a previous version, verify that all application processes that use the Solutions Enabler libraries and binaries are stopped. "Stopping the application processes" on page 168 provides instructions.
- ❑ If you are upgrading from a previous version, create copies of the host database and configuration directories. These copies will be useful should you want to *rollback* to the previous version of Solutions Enabler. The location of these directories vary according to the operating system. Appendix G provides more information.

UNIX-specific tasks

The following task is specific to UNIX environments:

- ❑ AIX and OSF1 do not allow changes to the destination path during installation. All binaries and libraries are installed under `/opt/emc`.

  If there is insufficient disk space under `/opt`, create a soft link to `/opt/emc/` as shown below and then run the installer:

  ```
  ln -s NewInstallationDir /opt/emc
  ```

  The root must have write permission on the *NewInstallationDir*.

Windows-specific tasks

During the installation process, the Windows **Services** dialog will open so you can select the daemons to start. You can prepare for this by reading the section "Setting up daemons for distributed application support" on page 91.

z/OS-specific tasks

The following tasks are specific to z/OS mainframe environments:

- ❑ Verify that you have a Windows host running a version of PKZIP or WinZip that supports 2.04 G compression.

  You will need the Windows host to FTP the installation files to the z/OS host.

- ❑ Install ResourcePak® Base.

  Solutions Enabler requires the use of EMC ResourcePak Base Version 5.6.0 at a minimum.

  If you have already installed ResourcePak Base Version 5.6.0 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.

- ❑ Choose an installation/configuration user account.

  In order to run the installation jobs, you must choose a TSO account in your system that has 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. Before running any Solutions Enabler jobs, ensure that you have a correctly defined OMVS segment.

  You should use this user's high-level qualifier when uploading the Solutions Enabler distribution file from the installation to the host.

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

❑ Gather the following customization 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.SSEM711 as the prefix, you will allocate EMC.SSEM711.LOADLIB, EMC.SSEM711.PARMLIB, and so on.

  Note: This should generally be the same prefix as the one you choose when you upload the distribution file from the installation CD.

- 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 that use its services. Solutions Enabler must have the same SCF SSID as the ResourcePak Base started task that you require it to use. 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 5.6.0 or higher. The default value is `EMC.SSCF560`, which is the default for the ResourcePak Base product, Version 5.6.0. 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; there is no default for the volume serial.

- SYMAPI base directory

  Specify a USS directory under which SYMAPI runtime sub directories will be created.

By default, the SYMAPI base directory is `/var/symapi`. However, during the execution of the Solutions Enabler SEMJCL installation procedure, you can change the default to any directory you want, provided that the security settings for the userids that run the Solutions Enabler jobs have read/write/execute permissions for the entire SYMAPI base directory tree.

- SYMAPI base directory space requirements

  The space requirements for the SYMAPI base directory vary according to the activities requested by clients (such as the EMC Symmetrix Management Console) of the Solutions Enabler tasks. In addition, the logging options (type, detail, rentention period) you select will also affect the space requirements for the SYMAPI base directory. In most cases, 50 to 100 MB should be sufficient.

- Time Zone

  The time stamp on messages written by Solutions Enabler to its internal logs will use the Portable Operating System Interface (POSIX) default—normally Coordinated Universal Time (UTC). If you prefer a local time stamp, you will need to provide a POSIX-compliant time zone value.

  "Configuring for local time zone" on page 153 provides more information.

❑ Define the UNIX system services requirements.

The following requirements apply to the userid of the installer, the userid assigned to the started tasks, or batch jobs used to run Solutions Enabler tasks (e.g., the SYMAPI server and event daemon). All userids running Solutions Enabler tasks must have an OMVS segment and full read/write/execute permissions to the SYMAPI base directory (by default `/var/symapi`) and all the sub-directories.

Note: Throughout the rest of this manual, this directory will be referred to as the *`symapi_installation_directory`*.

- Define the 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. This is also true for the userid assigned to the batch job running the server (if you choose to run it that way).

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

In addition, the userids must have full read/write permissions for the entire directory tree (specified during the install) of the *symapi_installation_directory*.

If these permissions are not granted to the installer or the SYMAPI tasks, then various security error messages may be issued during the the install or server setup.

For example:

```
ICH408I USER(user) Group(group) Name(username) 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)
```

Linux on System z-specific tasks

The following tasks are specific to Linux for IBM System z environments:

Note: Once you have completed the tasks in this section, continue with the UNIX installation procedure in Chapter 2, followed by the procedure "Install the Linux I/O module for CKD devices" on page 52.

- ❑ Verify that you have a supported version of Linux for System z.
- ❑ Verify that the installer is using root during both pre and post installation phases.
- ❑ Do the following, depending on whether SLES 10 is running as a guest under IBM's z/VM:
  - If SLES 10 is running as a guest under z/VM:

    Verify that all CKD Symmetrix devices are defined as VM unsupported DASD and attached to the Linux guest. The devices must be defined to VM (by way of `SET RDEV`) as:

    ```
    TYpe UNSUPported DEVCLass DASD DPS Yes
    RESERVE_RELease Yes
    ```

    For example:

    ```
    Set RDEVice 1300 TYpe UNSUPported DEVClass DASD DPS
    Yes RESERVE_RELease Yes
    ```

    By default, these devices will all function as gatekeepers. However, you can individually manage them by way of the gatekeeper include/exclude configuration files, as required.

    MVS formatted devices (regular MVS volumes) accessible by Linux on System z will appear in the Linux device tree. However, Solutions Enabler will not *discover* them, nor will it allow you to manage them by device name (e.g., /dev/dasdf). In certain cases, you will be able to manage these devices by Symmetrix device number (for example, on the `symld` command).
  - If SLES 10 is not running under z/VM, there are no special requirements for CKD devices.

Environment and system requirements

Consider the following when working with Solutions Enabler V7.1.

Host systems and Enginuity support

Solutions Enabler runs on a wide range of operating systems and works with certain Symmetrix Enginuity™ versions. For detailed interoperability information, please refer to E-Lab Interoperability Navigator, which can be reached at:

`http://elabnavigator.EMC.com.`

Disk space requirements

The disk space requirements (shown in MBs) are listed in four tables:

Note: A value of 0 MBs means the component is not supported on that platform.

Table 1 Disk space requirements for AIX, Solaris, Tru64 UNIX (1 of 2)

Install components (in MBs)	AIX	Solaris x86	Solaris	Tru64 UNIX
Persistent Data Files	2	1	1	1
Core Components	60	12	19	13
Base Storage Libraries	28	11	21	13
Base Mapping Libraries	0	0	0	0
Control Storage Libraries	17	6	7	7
Command Line Tools (optional component)	72	47	49	70
Database Mappings (optional component)	3	0	14	1

Table 1 Disk space requirements for AIX, Solaris, Tru64 UNIX (2 of 2)

Install components (in MBs)	AIX	Solaris x86	Solaris	Tru64 UNIX
Enable 64-bit Component install (optional component)	73	0	24	0
Base Storage 64-bit Libraries (optional component)	-	-	-	-
Base Mapping 64-bit Libraries (optional component)	-	-	-	-
Base SRM (optional component)	4	1	2	2
SMI-S Provider (optional component)	0	0	0	0
Java Native Interface (optional component)	68	0	102	0
Solutions Enabler Config Checker (optional component)	0	0	16	0
PERL 5.8 for STAR and symrecover (optional component)	58	48	50	58

Table 2 Disk space requirements for HP-UX, Linux, and Linux IA64 (1 of 2)

Install components (in MBs)	HP-UX	HP-UX (IA64)	Linux	Linux (IA64)
Persistent Data Files	1	1	1	1
Core Components	40	42	11	41
Base Storage Libraries	40	29	70	28
Base Mapping Libraries	0	0	1	0
Control Storage Libraries	16	16	5	16
Command Line Tools (optional component)	75	160	48	154
Database Mappings (optional component)	33	1	1	1
Enable 64-bit Component install (optional component)	1	0	0	0
Base Storage 64-bit Libraries (optional component)	-	-	-	-
Base Mapping 64-bit Libraries (optional component)	-	-	-	-

Table 2 Disk space requirements for HP-UX, Linux, and Linux IA64 (2 of 2)

Install components (in MBs)	HP-UX	HP-UX (IA64)	Linux	Linux (IA64)
SRM Base (optional component)	4	4	1	4
SMI-S Provider (optional component)	0	0	0	0
Java Native Interface (optional component)	102	0	94	0
PERL 5.8 for STAR and symrecover (optional component)	100	80	53	66
Symm Software Config Checker (optional component)	10	0	11	0

Table 3 Disk space requirements for LinuxPPC, Linux on System z, and Celerral (1 of 2)

Install components (in MBs)	Linux PPC	Linux on System z	Celerral
Persistent Data Files	1	1	2
Core Components	99	12	90
Base Storage Libraries	12	13	3
Base Mapping Libraries	0	0	0
Control Storage Libraries	6	6	6
Command Line Tools (optional component)	50	49	49
Database Mappings (optional component)	0	0	0
Enable 64-bit Component install (optional component)	0	0	0
Base Storage 64-bit Libraries (optional component)	-	-	-
Base Mapping 64-bit Libraries (optional component)	-	-	-
Base SRM (optional component)	2	2	0

Table 3 Disk space requirements for LinuxPPC, Linux on System z, and Celerral (2 of 2)

Install components (in MBs)	Linux PPC	Linux on System z	Celerral
SMI-S Provider (optional component)	0	0	0
Java Native Interface (optional component)	0	0	0
PERL 5.8 for STAR and symrecover (optional component)	54	73	0

Table 4 Disk space requirements for Windows (1 of 2)

Install components (in MBs)	Windows (x64)	Windows (IA64)	Windows (x86)
Persistent Data Files	1	1	1
Core Components	9	18	9
Base Storage Libraries	72	51	52
Base Mapping Libraries	0	0	0
Control Storage Libraries	6	19	5
Command Line Tools (optional component)	11	162	10
Database Mappings (optional component)	1	1	1
Enable 64-bit Component install (optional component)	0	0	0
Base Storage 64-bit Libraries (optional component)	0	0	0
Base Mapping 64-bit Libraries (optional component)	0	0	0
SRM Base (optional component)	2	5	2
SMI-S Provider (optional component)	47	0	45
Java Native Interface (optional component)	0	0	44

Table 4 Disk space requirements for Windows (2 of 2)

Install components (in MBs)	Windows (x64)	Windows (IA64)	Windows (x86)
Solutions Enabler ConfigChecker (optional component)	3	3	3
ECOM	36	36	36
PERL 5.8 for STAR and symrecover (optional component)	20	30	19

Client/server interoperability

The server component of Solutions Enabler V7.1.x SYMAPI is compatible with the client component of older SYMAPI versions from V6.0 and up. When planning to upgrade from V6.x to V7.1.x, it is possible to do so in a staged fashion, upgrading the server(s) first, and then the client(s). If access to V7.1.x enhanced features is required only from the server systems, then there is no requirement to upgrade client systems. For clients to gain access toV7.1.x enhanced features, they must be upgraded.

Secured sessions using SSL are only available when both the client and server are running Solutions Enabler V6.0 or later on platforms that support secure communication.

Non-secured sessions between SSL-capable clients/servers and a remote peer on a non SSL-capable platform are possible as long as you configure the security level of the SSL-capable clients/servers to ANY. For more information, refer to "Client or server installation" on page 38 and "Securing remote transmissions using SSL" on page 120.

Security settings

The following section details some Solutions Enabler security information.

The following sections assume that <SYMAPI_HOME> is:

- Windows: `c:\Program Files\emc\symapi...`
- UNIX: `/var/symapi/...`

Port usage

Solutions Enabler makes use of the following TCP/IP ports. In client/server mode, the Solutions Enabler server (`storsrvd`) listens by default at TCP/IP port 2707 for client connections.

You can configure a port by adding an entry to the *SYMAPI_HOME*/config/daemon_options file as follows:

```
storsrvd:port = nnnn
```

At client hosts, the *SYMAPI_HOME*/config/netcnfg configuration file can be changed to reflect the use of this non-default port.

When using the asynchronous events mechanism in client/server mode, the event daemon at the client host listens at a TCP/IP port for events being forwarded from the event daemon at the server. By default, the client event daemon asks the operating system to pick an unused port for it to use.

You can configure a specific port for use by adding an entry to the *SYMAPI_HOME*/config/daemon_options file on the client host as follows:

```
storevntd:event_listen_port = nnnn
```

While performing CLARiiON management operations, Solutions Enabler running on a host sometimes needs to open a TCP/IP connection to either port 443 or 2163 on the CLARiiON array. A configuration setting on the array determines which of these ports is the correct one—Solutions Enabler tries both of them.

Client/Server security settings

In client/server mode, network communications are protected using SSL when possible.

On platforms where Solutions Enabler supports SSL, servers default to securing all connections using SSL. For more information about hosts that support SSL, refer to Table 5 on page 39.

If a server needs to accept connections from clients for which SSL support is not present (for example, iSeries), you can enable it by adding the following line to the *SYMAPI_HOME*/config/daemon_options configuration file on the server host:

```
storsrvd:security_level = ANY
```

If you instead want to disable the use of SSL and not protect client/server communications, add the following line to *SYMAPI_HOME*/config/daemon_options configuration file on the server host:

```
storsrvd:security_level = NONSECURE
```

Likewise, you can also direct Solutions Enabler clients to only communicate with servers if the connection can be secured by SSL by setting the security level in their *SYMAPI_HOME*/config/netcnfg configuration file to SECURE.

Solutions Enabler provides support for client SSL certificate verification. By default, a server will verify a client's certificates if it is able to provide one. Servers can be directed to only accept connections for which client verification can be performed by adding the following line to the *SYMAPI_HOME*/config/daemon_options configuration file at that host:

```
storsrvd:security_clt_secure_lvl = MUSTVERIFY
```

A default set of self-signed certificates are created during the installation process. For information on replacing these certificates, refer to "Client/server security" on page 120.

For additional steps that can be taken to protect client/server operations, refer to "Remote Operations" on page 111.

Log files

Solutions Enabler log files are maintained at:

```
SYMAPI_HOME/log/symapi_yyyymmdd.log
```

Where *yyyymmdd* is the year, month and day.

You can configure the number of log files that Solutions Enabler maintains by adding an option to the SYMAPI_HOME/config/options file:

```
SYMAPI_LOGFILE_RENTENTION = Number_of_days
```

For more information on this option, refer to the default options file (README.options) installed with Solutions Enabler.

Individual log files are maintained for each of the background Solutions Enabler daemons at:

```
SYMAPI_HOME/log/storXXXX.log0
SYMAPI_HOME/log/storXXXX.log1
```

Where *storXXXX* is the name of the daemon (for example, storapid).

A secure audit log is maintained for operations on Symmetrix arrays on the storage array itself. Information from this log can be retrieved using the symaudit SYMCLI command. For more information on the audit log, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

Daemon security settings

Solutions Enabler uses a number of background daemons that provide services to it. For more information, refer to "Using Daemons" on page 235.

On Unix platforms, these daemons run as root by default. However, you can configure some of these daemons to instead run under a different user identity by invoking the following command:

```
stordaemon setuser Name -user Username [-v]
```

Where:

Name: specifies a daemon name, or `all`.

Username: specifies the user name that the daemon(s) should run as.

For more information, including the daemons for which this supported, refer to the installed stordaemon man page.

In addition, you can also refer to the `SYMAPI_HOME/config/README.daemon_users` that is installed with Solutions Enabler.

Securing access to Symmetrix arrays

Device masking allows you to mask the access privileges of Symmetrix devices to host bus adapters (HBAs). For more information, refer to the *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide*.

Symmetrix Access Control allows you to restrict the management operations that can be performed from individual hosts. For more information, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

Symmetrix User Authorization allows you to assign individual users to roles that limit the types of management operations that they can perform. For more information, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

z/OS-specific requirements

The following are the z/OS-specific requirements.

Note: The following Solutions Enabler features are not supported on z/OS: RDF daemon, GNS, SRM, and STAR.

Platform requirements

EMC Solutions Enabler for z/OS runs on all IBM supported releases of z/OS.

Some of the z/OS components that Solutions Enabler for z/OS uses are:

- Language Environment services.
- UNIX System Services socket support.

  Note: Solutions Enabler does not support older HPNS or IUCV sockets (non-integrated sockets).

- TCP/IP protocol stack.

  Note: Only IBM TCP/IP has been qualified by EMC. Support for other TCP/IP protocol stacks must be requested through the EMC Request for Price Quotation (RPQ) process.

  There are no special requirements to enable IBM TCP/IP support.

z/OS-specific directory structure requirements

With the introduction of SSL-protected client/server sessions in Solutions Enabler V6.2, the installation process looks for the installer's instructions about where to place the SYMAPI base directory. The base directory specifies a high level location where the standard SYMAPI directory will reside. Since use of SSL was optional, the USS directories were not required to be created.

The SYMAPI directory structure is required on any host running Solutions Enabler V6.3 or later. Configuration files must reside in the `config` directory under the base directory, and log files will be stored in the `log` directory.

USS file system requirements

The following are z/OS USS file system requirements.

Logging

The server, base, and event daemon write data to log files in the USS file system. Summary log data is written to `SYSPRINT DD`, but the comprehensive detail is written to USS files.

SYMAPI log file

Solutions Enabler writes all SYMAPI log data to a standard dated log file in the SYMAPI log directory.

USS file system options

The following USS file system options can be configured to meet your environment.

SYMAPI database

Starting with Solutions Enabler V7.0, an MVS dataset (via DD SYM$DB) is not supported. The USS file system will always be used to store the database.

Symmetrix Avoid, Gatekeeper Avoid and Select, and INQ files

If the base daemon is running and any of the select or avoid files are in use, the SYMAPI server should be configured to look for these files in USS. This configuration will eliminate the need to duplicate this information in the PARMLIB dataset, where it was previously stored. By removing the relevant DD statements (`SYM$AVD`, `SYM$GAVD`, `SYM$GSEL`, and/or `SYM$INQ`), SYMAPI will automatically look in USS for these files.

Running z/OS as a guest

When running z/OS as a guest under the VM operating system, the TimeFinder and SRDF utilities require special consideration. Devices must be defined to `VM (SET RDEV)` as:

```
TYpe UNSUPported DEVCLass DASD DPS Yes RESERVE_RELease Yes
```

These devices must be attached to the z/OS guest.

Note: VM does not allow volumes defined as unsupported to be attached to `SYSTEM`, or used to IPL a virtual machine.

Virtual memory requirements

EMC Solutions Enabler software always uses allocated memory above the 16 MB line. The actual region required depends on many factors such as the number of active tasks and connections, the number of managed Symmetrix arrays, and devices. EMC recommends specifying `REGION=0M` on the JOB card or EXEC card for the following jobs:

- `#01ECCIN`
- `#SEMAGNT` and any other JCL which uses `#STORSRV` as a model
- `#STORAPI` and any other JCL which uses `#STORAPI` as a model
- `#STOREVT` and any other JCL which uses `#STOREVT` as a model

These members are distributed with `REGION=0M` already specified on the EXEC cards. Your site may have SMF or JES exits or security rules established which restrict the use of `REGION=0M`. Check with your system programmer to verify that the submitting user has the authority to use `REGION=0M`.

Backward/forward compatibility for applications

An application built to run against any Solutions Enabler V5.*x* or V6.*x* version will execute with V7.1 SYMAPI runtime libraries without modification. Applications built to run a newer version of SYMAPI can successfully access any configuration database created by an earlier SYMAPI application.

Note: SYMAPI database access is not forward compatible because a SYMAPI library cannot access a database created by a newer version of a SYMAPI application. If, for example, the version of the local library becomes out of sync with the version of the local SYMAPI database (as a V5.5, or V6.x SYMAPI library call within EMC ControlCenter attempting to access a V7.1 database) it will return error: `SYMAPI_C_DB_FILE_TOO_NEW`.
This restriction relates only to local databases. In client/server environments, accesses to a server database of a later version are automatically resolved by the SYMAPI, which performs all necessary translation of information between the client and the server.

Client or server installation

If your computer is locally connected to a Symmetrix array, go to Chapter 2. If your computer is a client or the SYMAPI server, read the following sections.

Remote connection

You can run SYMCLI as a client to a remote SYMAPI server to manage a remotely-controlled Symmetrix array. Figure 1 shows a Symmetrix array in the client/server system.

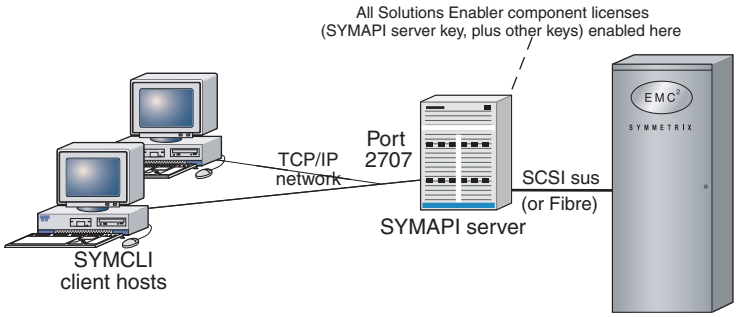

Figure 1 Client/server connection to Symmetrix array

Client/server IP communication

The SYMAPI client and server are both capable of negotiating sessions over the traditional Internet Protocol Version 4 (IPv4) and the newer Internet Protocol Version 6 (IPv6).

All hosts that use TCP/IP for communications use at least IPv4, a protocol well known to many applications. Newer versions of host operating systems will also support configuration of IPv6 local addresses, routing, and Domain Name Services as well. For the foreseeable future, many networks are likely to be running with dual protocol stacks activated, where communications will take place over IPv4 most of the time. Applications such as Solutions Enabler can also detect the presence of IPv6 configuration and use it whenever possible.

In UNIX, Linux, and Microsoft Windows Server environments, the SYMAPI server and client will interoperate with both IPv6 and IPv4 protocols on hosts that are configured to run both. The protocol

actually selected by the server and the client depends on the exact configuration of the host, router and DNS servers in your network, and on the settings in the Solutions Enabler network services configuration file.

Client/server security

Solutions Enabler uses Secure Socket Layer (SSL) protocol to enable secure communication in a client/server system. Using open source SSL (OpenSSL) technology, the client and server communicate over an authenticated, encrypted connection.

When a client attempts to connect with a server, the two machines exchange a handshake in which they both identify their security expectations and capabilities. If their security capabilities are the same, the two will negotiate the appropriate type of session (secure or non-secure). If their security capabilities are different, either the client or the server will reject the session.

The SYMAPI client and server are initially configured to communicate via secure sessions. You must modify this behavior if a platform in the environment does not support secure communications. "Securing remote transmissions using SSL" on page 120 provides instructions on modifying this default behavior.

Table 5 lists the host operating systems that support SSL.

Table 5 Host operating system support for SSL (1 of 2)

Supported operating system
AIX (32- and 64-bit)
HP-UX (32- and 64-bit) HP-UX Itanium (64-bit)
Tru64 UNIX (64-bit)
Linux (32-bit) Linux Itanium (64-bit) Linux AMD (64-bit) Linux/390 (32-bit)

Table 5 Host operating system support for SSL (2 of 2)

Supported operating system
Solaris (32- and 64-bit)
Windows (32-bit) Window Itanium (64-bit) Windows AMD (64-bit)
z/OS

Client/server system installation

The following information outlines procedures for installing Solutions Enabler in a client/server system:

1. Install Solutions Enabler software in the machine designated as the client, according to the procedures in Chapter 2.
2. Install the same Solutions Enabler software in the machine designated as the server, according to the procedures in Chapter 2. You need to invoke `symlmf` and apply the SYMAPI server license key.
3. Edit the `netcnfg` file in the client machine to include the host name or IP address of the server. "SYMCLI through a remote server" on page 112 provides instructions.
4. Issue a `stordaemon start storsrvd` command on the server machine. "SYMCLI through a remote server" on page 112 provides instructions.
5. Set environment variables `SYMCLI_CONNECT` and `SYMCLI_CONNECT_TYPE` on the client. "SYMCLI through a remote server" on page 112 provides instructions.

2

Installation

This chapter explains how to install/upgrade Solutions Enabler:

Installing Solutions Enabler on UNIX and Linux

This section describes how to install/upgrade Solutions Enabler on UNIX and Linux hosts.

Note: Solutions Enabler V7.1.1 is fully upgradeable. That is, you do not have to remove the previous version before installing V7.1.1.

Note: Before starting this procedure, be sure to review pre-install considerations in Chapter 1 and the *EMC Solutions Enabler Release Notes*.

Note: The default responses to the prompts in this section are in brackets [].

Step 1: Mount the installation DVD

To mount the installation DVD:

1. Log onto the host system as **`root`**.
2. Insert the Solutions Enabler installation DVD into the host's drive.
3. Mount the DVD to a subdirectory (for example, `/dvd`) by entering the appropriate platform-specific `mount` command from Table 6.

Table 6 UNIX mount commands

For	Enter
AIX[a]	**`mount -r -v cdrfs /dev/cd0 /dvd`**
Compaq Tru64 UNIX (OSF1)	**`mount -t cdfs -o [noversion] rrip /dev/disk/cdromxc /dvd`**
HP-UX Versions 11.0 and above	**`mount -F cdfs /dev/dsk/cxtxdx /dvd`**
Linux[b]	**`mount -t iso9660 -o ro /dev/dvd /dvd`**
Solaris	If automounter is running, the disc mounts unattended. To mount the disc manually, enter: **`mount -F hsfs -o ro /dev/dsk/cxtxdxs0 /dvd`**

a. With AIX, you may get a warning if the device and the directory do not have the same permissions. You can usually ignore these warnings.

b. You should not load Solutions Enabler on the Celerra® File Server Control station, as it is not a Linux client.

Step 2: Run the install script

To run the installation script:

1. Change directory to the location of the Solutions Enabler kit by entering the following:

   `cd /Install_disc_mount_point/Unix/operating_system`

2. Select an installation method from Table 7, and then run the appropriate command. For descriptions of the command options, refer to Table 8 on page 44.

Table 7 Installation method (1 of 2)

Method	Command	Comments
Interactive	`./se7110_install.sh -install`	Starts the interactive script documented in the remainder of this chapter. When using this method, continue with "Step 3: Select the installation directories" on page 46.
Silent (all components)	`./se7110_install.sh -install -silent -all`	Silently installs *all* Solutions Enabler components. When using this method, continue with "Step 5: Complete the installation" on page 51.
Silent (specific components)	`./se7110_install.sh -install -silent [-all] [-jni] [-db] [-64bit] [-star] [-symrec] [-ora] [-udb] [-syb] [-force] [-daemonuid] [-permission] [-homedir] [-datadir] [-nodeps] [-cfgchk] [-copy_lic]`	Silently installs only the specified components. When using this method, continue with "Step 5: Complete the installation" on page 51.
Incremental (specific components)	`./se7110_install.sh -increment [-jni] [-db] [-64bit] [-star] [-symrec] [-cfgchk]`	Incrementally adds the specified component to an existing installation. When using this method, continue with "Step 5: Complete the installation" on page 51. To use this method, you must have already installed the DATACORE, DATASTORBASE, CORE, SRMBASE, STOREBASE, and SYMCLI components.
Response file	`./se7110_install.sh -file Response_File_Name`	Runs the installation script according to the contents of your response file. To use this method, create a text file containing the relevant command line options (refer to the examples on the next page), and then run the command, specifying the name of your text file. Response file entries can be separated by a space or on separate lines and options must not have leading hyphens. (Continued on the next page)

Table 7 Installation method (2 of 2)

Method	Command	Comments

Response file (continued from previous page)		Using this method, you can specify the argument INCREMENT to perform an incremental installation or SILENT to perform a silent installation.
		For example, to incrementally install the SYMRECOVER and 64-bit components:
		1. Create the following response file:
		`spea30# cat responsefile.txt`
		`increment`
		`symrec 64bit`
		`spea30#`
		2. Run the command:
		`./se7110_install.sh -file responsefile.txt`
		For example, to silently install Solutions Enabler with the Java Interface and Star components:
		1. Create the following response file:
		`spea30# cat responsefile.txt`
		`install silent`
		`jni db`
		`star`
		`spea30#`
		2. Run the command:
		`./se7110_install.sh -file responsefile.txt`
		When using this method, continue with "Step 5: Complete the installation" on page 51.

Table 8 defines the various options used when running the installation commands detailed in Table 7 on page 43.

Table 8 UNIX installation options (1 of 2)

Option	Description
`-64bit`	Installs the 64 bit libraries.
`-all`	Installs all of the optional Solutions Enabler components, including the Java Interface; the Oracle, UDB, and Sybase daemons; the Star component; the SYMRECOVER component; and the Solutions Enabler ConfigChecker utility. Used with the `-silent` option.

Table 8 UNIX installation options (2 of 2)

Option	Description
`-cfgchk`	Installs the Solutions Enabler ConfigChecker utility. Used with the `-silent` option. **Note:** For more information on Solutions Enabler ConfigChecker, refer to Appendix D.
`-copy_lic=` *directory*	Copies the user-supplied `symapi_licenses.dat` file to `/var/symapi/config` during installation. Used with the `-silent` option. For example, the following command will copy the `symapi_licenses.dat` file from `/tmp` to `/var/symapi/config`: `bash-3.00# ./se7110_install.sh -install -copy_lic=/tmp -silent`
`-daemonuid=` *Name*	Changes ownership of the daemon to non root user. Used with the `-silent` option.
`-datadir=` *directory*	Sets the working root directory [/usr/emc]. Used with the `-silent` option.
`-db`	Installs all of the optional database components, including the Oracle, UDB, and Sybase daemons.
`-file`	Specifies to install Solutions Enabler with a response file.
`-force`	Kills all processes using the SYMAPI libraries. Used with the `-silent` option.
`-homedir=` *directory*	Sets the install root directory [/opt/emc]. Used with the `-silent` option.
`-jni`	Installs the Solutions Enabler Java Interface component.
`-ora`	Installs the optional Oracle daemon. Used with the `-silent` option.
`-permission =`*level*	Sets permission to /var/symapi directory. Used with the `-silent` option.
`-silent`	Specifies to perform a silent installation.
`-star`	Installs the Star component.
`-syb`	Installs the optional Sybase daemon. Used with the `-silent` option.
`-symrec`	Installs the SYMRECOVER component.
`-udb`	Installs the optional UDB daemon. Do not use with the `-db` or `-all` options.

Note: For help running the installation script, run the following:
`./se7110_install.sh -help`

Note: The installation script creates log files in the directory `/opt/emc/logs`. For more information, refer to Appendix H.

Step 3: Select the installation directories

To select the installation directories, do one of the following:

- If you are installing Solutions Enabler on a host for the first time, complete "Step 3A: Installing for the first time."
- If you are upgrading or reinstalling Solutions Enabler, complete "Step 3B: Upgrading /reinstalling" on page 47.

Note: It is recommended that you install EMC Solutions Enabler on your host's internal disks and not on a Symmetrix device.

Step 3A: Installing for the first time

If you are installing Solutions Enabler on a host for the first time, the following prompt displays:

```
Install root directory [/opt/emc]:
```

1. Press **Enter** to accept the default installation directory `/opt/emc`, or enter another root directory.

   If you enter a root directory (absolute directory) other than the default, you will be prompted to confirm the directory.

2. At the following prompt, press **Enter** to accept the default working directory `/usr/emc`, or enter another working directory. This directory is where the data and log files will be written:

   ```
   Working root directory [/usr/emc]:
   ```

   If you enter a working directory (absolute path) other than the default, you will be prompted to confirm the directory.

3. At the following prompt, specify whether to run the event daemon, Group Name Service daemon, and Watchdog daemon without root privileges. A `[Y]es` response will enable you to specify a non-root user to run the daemons:

```
Following daemons can be set to run as a non-root user:
storsrvd, storevntd, storgnsd, storwatchd
Do you want to run these daemons as a non-root user?
[N]:
```

4. Continue with "Step 4: Select installation options" on page 48.

Step 3B: Upgrading /reinstalling

If you are upgrading or reinstalling Solutions Enabler, the following prompt displays:

```
Install root directory of previous installation: opt/emc
Do you want to change Install root Directory ? [N]:
```

1. Respond `[N]o` to install Solutions Enabler into the same root directories (install and working) as the previous installation, or respond `[Y]es` to display the following prompts in which you can enter other root directories:

```
Install root directory [/opt/emc]:
Working root directory [/usr/emc]:
```

   If you enter a root directory (absolute directory) other than the default, you will be prompted to confirm the directory.

2. At the following prompt, specify whether to run the SYMAPI Server daemon, event daemon, Group Name Service daemon, and Watchdog daemon without root privileges. A `[Y]es` response will enable you to specify a non-root user to run the daemons:

```
Following daemons can be set to run as a non-root user:
storsrvd, storevntd, storgnsd, storwatchd
Do you want to run these daemons as a non-root user?
   [N]:
```

3. If the installation program detects that there are daemons currently running, the following prompt displays asking whether to shut them down or exit the installation. A `[Y]es` response shuts down the daemons. A **x** response exits the installation:

```
Do you want to shutdown SYMCLI daemons [Y] or Exit setup
   [X]? [Y]:
```

4. If you are upgrading, the following prompt displays asking whether to backup the previous installation. A `[Y]es` response backs up the SYMCLI binaries in the install root directory under `SYMCLI/symcli_old`:

   ```
   Do you want to save /opt/emc/SYMCLI/Vx.x.x ? [N]:
   ```

5. At the following prompt, specify whether to restore the previous installation's *persistent* data to the new working root directory. A `[Y]es` response backs up the data files as a tar file (`symapi_old.tar`) in the working root directory:

   ```
   Do you want to restore the SYMAPI (persistent) data of
      the previous installation ? [Y]:
   ```

6. Continue with "Step 4: Select installation options" on page 48.

Step 4: Select installation options

To select your installation options:

1. At the following prompt, specify whether to install *all* of the Solutions Enabler libraries:

   ```
   Install All Solutions Enabler Shared Libraries and Run
      Time Environment? [Y]:
   ```

   - A `[Y]es` response installs *all* the libraries.
   - A `[N]o` response produces the following series of prompts, which allow you to select the libraries to install. If you only want to install Solutions Enabler's core functionality, specify `[N]o` for each of the prompts. Solutions Enabler's core functionality includes `symapi[mt]`, `symlvm[mt]`, `storapi[mt]`, `storsrvd[mt]`, `symapisrv[mt]`, `storapid[mt]`, `storcore[mt]`, `stordaemon[mt]`, and `storpds[mt]`.

     – `BaseStor Library Component ? [Y]:`
     A `[Y]es` response installs `StorSil` and `Storbase`. This option provides base storage and host specific functionality, and an interface to storage arrays for features like I/O scan, device listings, statistics, and showings.

– `CtrlStor Library Component ? [Y]:`

  A `[Y]es` response installs `StorSil`, `Storbase`, and `Storctrl`. This option provides the same functionality as the `BaseStor` option (explained earlier), but includes storage control functionality for features like Snap, device masking, and device mirroring.

– `BaseMapping Library Component ? [Y]:`

  A `[Y]es` response installs `stormap`, which provides Storage Resource Management (SRM) functionality.

2. Solutions Enabler is available in both 32 and 64 bit support on some operating systems. The following prompt only displays if Solutions Enabler can support both 32 and 64 bit versions of the libraries and executables on the host:

   ```
   Install Solutions Enabler 64-bit Shared libraries ?
      [N]:
   ```

   A `[Y]es` response installs the 32 *and* 64 bit libraries. A `[N]o` response *only* installs the 32 bit libraries.

3. At the following prompt, specify whether to install the collection of binaries known as SYMCLI. A `[Y]es` response installs the SYMCLI binaries:

   ```
   Install Symmetrix Command Line Interface (SYMCLI) ?
      [Y]:
   ```

4. If you are installing Solutions Enabler on a host with a Linux, HP-UX, SunOS, or AIX operating system, the following prompt displays, asking whether to install one or more *optional* database components:

   ```
   Install Solutions Enabler SRM Database Run Time
      Components ? [N]:
   ```

   A `[Y]es` response displays the following prompts, depending on your operating system:

   - `SRM Oracle Database Component ? [N]:`

     This prompt only displays on operating systems where Solutions Enabler supports Oracle. A `[Y]es` response installs the *optional* Oracle daemon.

- `SRM Sybase Database Component ? [N]:`

  This prompt only displays on operating systems where Solutions Enabler supports Sybase. A `[Y]es` response installs the *optional* Sybase daemon.

- `IBM UDB Database Component ? [N]:`

  This prompt only displays on operating systems where Solutions Enabler supports UDB. A `[Y]es` response installs the *optional* UDB daemon.

5. At the following prompt, specify whether to install the Solutions Enabler Java interface component. You should install this component if your Solutions Enabler application uses a Java interface. A `[Y]es` response installs the JNI component:

   ```
   Install Option to Enable JNI Interface for EMC
      Solutions Enabler APIs ? [N]:
   ```

6. At the following prompt, specify whether to install the Solutions Enabler Star component. A `[Y]es` response installs the Star component:

   ```
   Install option to Enable EMC Solutions Enabler Star
      component ? [Y]:
   ```

7. At the following prompt, specify whether to install the Solutions Enabler SRDF® session recovery component. A `[Y]es` response installs the SYMRECOVER component:

   ```
   Install option to Enable EMC Solutions Enabler
      SYMRECOVER component ? [Y]:
   ```

   Note: The SYMRECOVER component depends on the Star component. Therefore, installing SYMRECOVER will also install the Star component.

8. At the following prompt, specify whether to install the Solutions Enabler ConfigChecker utility. A `[Y]es` response installs the component:

   ```
   Install EMC Solutions Enabler ConfigChecker
      Component ? [N]:
   ```

   Note: For more information on Solutions Enabler ConfigChecker, refer to Appendix D.

9. At the following prompt, specify whether to change the default UNIX file permissions. A `[Y]es` response displays another prompt in which you can specify a new value:

```
Do you want to change the default permission on
   /var/symapi directory [755] ? [N]:
```

10. If you are upgrading, the following prompt displays, asking whether to **move** the previous installation's data files to the `symapi_old` directory. A `[Y]es` response **moves** your persistent data from the `/usr/emc/API/symapi` directory to `/usr/emc/API/symapi_old`. A `[N]o` response retains your persistent data:

```
Do you want to move this data to
   /usr/emc/API/symapi_old ? [N]:
```

11. If you are upgrading, the following prompt displays, asking whether to remove the `symcli_old` directory. A `[Y]es` response removes the directory:

```
Do you want to remove the symcli_old directory ? [Y]:
```

Step 5: Complete the installation

This section explains how to complete your Solutions Enabler installation.

Verifying your installation

To verify your installation, run the following command:

```
./se7110_install.sh -check
```

For example, the command:

```
spea30# ./se7110_install.sh -check
```

produces the following output in a Solaris environment:

```
SI No   Package       Version
  1     SYMcore       7.1.1
  2     SYMdcore      7.1.1
  3     SYMdsbase     7.1.1
  4     SYMsrmbse     7.1.1
  5     SYMstrbse     7.1.1
  6     SYMstrful     7.1.1
  7     SYMsymcli     7.1.1
```

Removing temporary files

During installation, the install script creates the temporary file `/tmp/emc_app_data_path`. This file holds the value that was entered for the install root directory from the previous installation. This value is used as the default install root directory in subsequent installations.

For example:

```
EMC_APPLICATION_PATH:/OPT/EMC
```

In some cases this file will be removed when you reboot your system. If not, you may want to manually remove it to conserve disk space.

Unmounting the installation disc

To unmount the installation disc, enter:

```
umount mount_point
```

Install the Linux I/O module for CKD devices

In a Linux on System z installation on Novell SLES 10, you must load a kernel object file in order to issue I/O to Symmetrix arrays by way of CKD devices. In addition, failing to load the object file in an environment where a guest can only see CKD devices will prevent Solutions Enabler from discovering Symmetrix arrays.

To load the kernel object file, locate the operating system-specific object in the directory `/usr/symapi/ioctl/SUSE_Version`, and then use the `insmode s390ioctl.ko` command to load it.

For example, to load the kernel object in a SLES 10 Service Pack 1 environment, enter:

```
cd /usr/symapi/ioctl/
cd suse10sp1
insmod s390ioctl.ko
```

Enabling the Solutions Enabler components

You must now enable your Solutions Enabler features by entering the appropriate license keys.

Note: For instructions, refer to "Enabling your software" on page 72.

Installing Solutions Enabler on Windows

You can install/upgrade Solutions Enabler on a Windows host using either an InstallShield wizard (described below), or command line options (refer to "Using the command line" on page 55).

Note: Solutions Enabler V7.1.1 is fully upgradeable. That is, you do not have to remove the previous version before installing V7.1.1.

Note: Before starting this procedure, review the pre-install considerations in Chapter 1 and the *EMC Solutions Enabler Release Notes*.

Using the InstallShield wizard

To install/upgrade Solutions Enabler using the InstallShield wizard:

1. Save all files and exit all Windows applications.
2. Insert the Solutions Enabler installation disc into the host's disk drive.
3. Change directory to the location of the Solutions Enabler kit by entering the following:

   `cd /Install_disc_mount_point/Windows`

4. Start the installation program by running the following:

`cd /Install_disc_mount_point/Windows/se7110-Windows-Processor_type.exe`

   Where `Processor_type` can be x86, x64, or IA64.

5. In the **InstallShield Wizard for Solutions Enabler Welcome** dialog box, click **Next**.
6. In the **Destination Folder** dialog box, select an installation directory and click **Next**.
7. In the **Setup Type** dialog, select **Typical** to install the default components, select **Complete** to install the full Solutions Enabler product set, or select **Custom** to install a subset of the options. Click **Next** when done.

8. If you selected **Custom**, the **Custom Setup** dialog box opens. Select the options (Table 9) to install, where to install them, and then click **Next**.

Note: In Table 9, components in shaded rows are required and therefore selected by default.

Table 9 Windows installation options

Option	Description
CORE	Installs Solutions Enabler core functionality, including symapi, symlvm, storapi, symapisrv, storapid, storcore, stordaemon, and storpds. This option is part of the shared library and runtime environment. It is a corequisite for other options, and is therefore mandatory for a successful installation.
ConfigChecker	Installs the Solutions Enabler ConfigChecker. This is a utility for validating your host environment configuration settings against those recommended for Solutions Enabler. For more information on Solutions Enabler ConfigChecker, refer to Appendix D.
JNI	Installs the Solutions Enabler Java Interface component.You should install this component if your Solutions Enabler application uses a Java interface.
ORACLE	Installs the Oracle daemon.
SQL	Installs the SQL daemon.
SRMBASE	Installs Storage Resource Management base mapping library. This option is part of the shared library and runtime environment.
STAR_PERL	Installs the Solutions Enabler Star component.
STORBASE	Installs `StorSil` and `Storbase`. This option provides base storage and host specific functionality, and an interface to storage arrays for features like I/O scan, device listings, statistics, and showings. This option is part of the shared library and runtime environment.
STORFULL	Installs control storage libraries, which include features like Snap, device masking, and device monitoring. This option is part of the shared library and runtime environment.
SYMCLI	Installs the collection of binaries known as SYMCLI.
SYMRECOVER	Installs the SRDF session recovery component.
UDB	Installs the UDB daemon.

9. In the **Select Services** dialog, select the services to install/start. The services available in this dialog are based on the installation options you selected. "Setting up daemons for distributed application support" on page 91 includes descriptions of the Solutions Enabler daemons.

10. In the **Ready to Install the Program** dialog, click **Install**.
11. In the **InstallShield Wizard Completed** dialog box, click **Finish** to complete the setup, and then go to "Enabling your software" on page 72.

Using the command line

To install/upgrade Solutions Enabler using the command line:

1. Save all files and exit all Windows applications.
2. Insert the installation disc into the host's disk drive.
3. Change directory to the location of the Solutions Enabler kit by entering the following:

   ```
   cd /Install_disc_mount_point/Windows
   ```

4. Run one of the following commands to start the installation program:

   Note: In the following command examples, *Processor_type* can be x86, x64, or IA64.

   - To perform an interactive install (with prompts) use the following command:

     ```
     start /wait se7110-Windows-Processor_type.exe
     ```

   - To perform a silent install, use the following command:

     ```
     start /wait se7110-Windows-Processor_type.exe
        /s /v/qn
     ```

   - To perform an install and generate a verbose log of the script's actions in a temporary directory, use either of the following commands:

     ```
     start /wait se7110-Windows-Processor_type.exe /s
     /v"/log "C:\install.log" /qn
     ```

     or

     ```
     start /wait se7110-Windows-Processor_type.exe /s
     /v"/l*v "C:\install.log" /qn
     ```

   - To install from a response file:

     ```
     start /wait se7110-WINDOWS-Processor_type.exe /s
     /v"WSC_CONFIG_FILE=
     path_to_the_response_file_with_the_filename /qn"
     ```

Installing Solutions Enabler on z/OS

This section describes how to install Solutions Enabler on a z/OS mainframe to operate as a SYMAPI server.

The following procedure can be used for either a new installation, or to upgrade an existing installation.

Note: Before starting this procedure, be sure to review the pre-install considerations in Chapter 1 and the *EMC Solutions Enabler Release Notes*.

Step 1: Copy the files from installation disc

To copy files from the installation disc:

1. Insert the Solutions Enabler installation disc into the disk drive of the Windows host from which you will be running the FTP upload.
2. Copy the file `emc.ssem711.zip` from the `zOS` directory of the install disc to a temporary directory on the Windows desktop.
3. In the temporary directory you just created, extract the files from the `.zip` file, and then execute the command `uploadSE`.
4. When prompted, provide the following information:
   - The name or IP address of the z/OS host on which you are installing
   - The userid and password to login to the FTP server on the z/OS host, and other optional FTP information
   - The high-level qualifier of the dataset name to use during allocation of the distribution file
   - The name of a volume and esoteric unit name on which to allocate the distribution file

   Once the upload completes, the distribution file will be ready for remaining installation steps.
5. Once the files are uploaded, login to the z/OS host and continue the installation.

Note: If you plan on running the Solutions Enabler server using secure (SSL) communications, you must create and install the certificates for z/OS before starting the server. To do this, you must run the Windows batch file `zoscert.bat` from the same location you ran the `upload.bat` batch file.You cannot do this until after you have run job `#07DFLTS`, as this job creates some requisite directories in the UNIX System Services filesystem. For more information, refer to "Installing the SSL certificates" on page 144.

Step 2: Receive the transmit file

The file that you transferred to the host was created using the TSO `TRANSMIT` command. Therefore, you must use the TSO `RECEIVE` command to convert the file to a library of materials that you will use to complete the installation.

To receive the transmit file:

1. Do one of the following:
   - From the TSO READY prompt, enter the following command:

```
RECEIVE INDS('high_level_qualifier.EMC.ssem711.XMITFILE')
```

     Where *high_level_qualifier* is the same qualifier used during the CD-based batch upload procedure.

   - In the **Utilities.DSList (3.4)** of the main ISPF menu, type **RECEIVE INDS(/)** on the line where the uploaded transmit file is shown in the list.

   In either case, the following displays:

```
INMR901I Dataset EMC.ssem711.XMITLIB from
emcdist on NODENAME
INMR906A Enter restore parameters or 'DELETE' or
'END'
```

2. Press ENTER to accept the allocation of the XMITLIB under your high-level qualifier, or respond with the following to change the allocated dataset name:

```
DSN('ds_prefix.xmitlb')
```

Note: The dataset name you specify must end in the XMITLIB extension.

Step 3: Extract the additional files from the XMITLIB

Edit the job `$EXTRACT` member of the XMITLIB and make the following changes:

1. Add a JOB card to comply with your site's batch JCL standards.
2. Change all occurrences of *`ds-prefix`* to the desired prefix for your Solutions Enabler libraries.
3. Change all occurrences of `DVOL` to the volume on which you want to allocate the libraries.
4. Change all occurrences of `DISK-UNIT` to the disk unit name that includes the volume you specified in the `DVOL` change above.
5. Submit the job, and look for a zero return code. The `$EXTRACT` job creates some temporary data sets which will be deleted by the `#99ECLN` job after the installation is complete. It also creates some data sets for permanent use with Solutions Enabler.

Step 4: 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 ISPF edit macro in your CLIST library and then running it against every member of the RIMLIB whose name starts with a pound sign (#).

Note: If you prefer to manually customize the JCL, customize the # prefixed members as necessary, and then continue with "Step 5: Run the jobs" on page 61.

To use SEMJCL:

1. In the **Utilities.DSList (3.4)** of the main ISPF menu type the first few qualifiers of your RIMLIB dataset name, and then press ENTER.

   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 displays.
3. Scroll to the SEMJCL member in the RIMLIB, and then type **`exec`** (or **`ex`**) in the input area to the left of the member name.

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

```
.----- Customize EMC Solutions Enabler 7.1.1 Electronic Kit Install JCL ------.
| Command ===> ____________________________________________                    |
| Press PF3 to Cancel or PF1 for Help                                          |
| Press ENTER to run edit macro SEMX711 which                                  |
| will customize the installation JCL                                          |
|                                                                              |
|     Data Set Name Prefix:  EMC.SSEM711                                       |
|    SMP/E Data Set prefix:  EMC.SMPE                                          |
|         SCF Subsystem Id:  EMC                                               |
|       SCF Linklib Prefix:  EMC.SSCF560                                       |
|           Disk Unit Name:  SYSDA      Disk Volume Serial:  SYM001            |
|                Time Zone:  EST5                                              |
|    SYMAPI Base Directory:  /var/symapi                                       |
|                                                                              |
| Enter JOB card below ('%MEMBER%' is replaced by the member name):            |
| //USERID1A JOB ACCT,'EMC SEM 7.1.1',                                         |
| // CLASS=A,             <-- CHANGE IF NEEDED                                 |
| // MSGCLASS=A,          <-- CHANGE IF NEEDED                                 |
| // NOTIFY=&SYSUID       <-- CHANGE IF NEEDED                                 |
|                                                                              |
|                                                                              |
|                                                                              |
'------------------------------------------------------------------------------'
```

Figure 2 SEMWIN1 pop-up window

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

Tip To cancel the SEMJCL, press **PF3** (that is, the **END** key).

   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.

   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 **Time Zone** field, enter the appropriate setting for your time zone location. This setting must be a POSIX-compliant time zone value. This value is used to set the TZ environment variable of the Solutions Enabler task. If you do not supply a value, the time stamps of the Solutions Enabler internal messages written to the log files will default to UTC time.

   For example, entering a value of `EST5` will set the time stamp to the United States Eastern Standard Time, 5 hours earlier than UTC.

CAUTION

The default time zone value is UTC time.

h. In the **SYMAPI Base Directory** field, specify the location of the USS directory under which the SYMAPI runtime directories will be created.

Note: The userid used in the Solutions Enabler batch jobs must have write access to the entire SYMAPI base directory.

i. In the **Job Card Information** field, specify up to four statements for your job card.

   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.

Note: Statement syntax is not validated until jobs are submitted.

j. Press **Enter**.

SEMJCL generates an edit macro and uses the ISPF editor to apply the specified values to all 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 5: Run the jobs

Run each of the following jobs:

- `#01ALLOC`

  Creates all the datasets not allocated by the `$EXTRACT` job for installing the product, and copies sample configuration members from the RIMLIB into the Solutions Enabler PARMLIB.

- `#04DDDEF`

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

- `#05RECEV`

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

  Note: 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`

  Selectively applies 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.

- `#07DFLTS`

  This job assembles and links the assembler source in member `#SYMDFLT`. `#SYMDFLT` will have been updated when the exec SEMJCL was run. This job also creates the SYMAPI directory structure, based on your specification of the SYMAPI Base directory on the **SEMJCL Customization** panel.

- #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 that 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
000046 //STEPLIB   DD  DSN=EMC.SSEM711.LOADLIB,DISP=SHR
000047 //SYM$LIC   DD  DSN=EMC.SSEM711.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  /*
```

Note: 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 authorize 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 21.

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.

- `#10ECCIN`

  This job creates and loads the database that is supplied to EMC ControlCenter (or SYMCLI) 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 arrays that you do not want remote clients (such as ControlCenter) to view, you may exclude them from the discover process as follows.

  Note: If the configuration of any Symmetrix array attached to a host is changed, then you must re-run job `#10ECCIN` to correctly discover the changed Symmetrix array.

  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
/*
```

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

- `#16CFGCP`

  Copies the sample configuration files to the SYMAPI configuration directory.

Step 6: Complete the installation

Do the following to complete the installation:

1. Perform all other customizing and any testing as required. A sample startup job (`#STORSRV`) for the SYMAPI server daemon can be customized and run on a z/OS system. Note that you can either run `STORSRV` 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 `SSEM`*`vrm`* into the distribution zone.

3. If configuration change, authorization, SRDF or 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 155.

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 5.

Note: If you plan on using the optional Secure Socket Layer (SSL) encrypted communications between the SYMAPI server and its connecting clients, and you plan on running the server in SECURE or ANY modes, you must create and install the SSL certificates before starting the server. For more information, refer to "Installing the SSL certificates" on page 144.

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 up until you run job `#11ACCPT`.

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.

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

Restoring the RIMLIB

In the event that customization of the RIMLIB has rendered it difficult to work with, you can use job `#RIMREST` in the RIMLIB to re-create the RIMLIB. This job will create a new RIMLIB with the suffix `.REST` and will not alter the original RIMLIB. However, you should verify that the JCL in `#RIMREST` is appropriate before running the job.

Installing Solutions Enabler on OpenVMS

This section describes how to install/upgrade Solutions Enabler on an OpenVMS host.

Note: Before starting this procedure, be sure to review the pre-install considerations in Chapter 1 and the *EMC Solutions Enabler Release Notes*.

Step 1: Access the software

Solutions Enabler is distributed in the following forms:

- On the Solutions Enabler installation disc, which includes kits for all supported platforms.
- As a platform-specific file download from the Powerlink® website at:

  http://Powerlink.EMC.com

  Possible filenames are:

`SE711RT.SAV`	HP Alpha hardware platform.
`SE711RIA.SAV`	HP Integrity hardware platform. **Note:** This filename has changed from the previous version of Solutions Enabler when it was SE652RTIA.SAV.

Note: Throughout the remainder of this installation procedure, substitute the appropriate filename for any occurrence of the variable *`InstallKit`*.

From an install disc

To access the software from an installation disc:

1. Insert the Solutions Enabler disc into the host's disk drive.
2. Mount the disc by entering:

```
mount/media=cdrom/undefined=(fixed:cr:32256)/over=id dkd600
```

3. Copy the kit from the Solutions Enabler directory `[other.ovms]` to a temporary directory on your machine by entering:

```
copy /log dkd600:[other.ovms]InstallKit sys$sysdevice:[EMC.KITS]*.*
```

From Powerlink

To access the software from Powerlink:

1. On Powerlink, select **Support** > **Software Downloads and Licensing** > **Downloads S** > **Solutions Enabler** and click the platform-specific installation kit.
2. Save the installation kit to the host's disk drive and run the following command against it:

   ```
   set file/attr=(RFM:FIX,LRL:32256) InstallKit
   ```

Step 2: Install the software

To install the software:

1. Extract the command procedure after setting `[set DEF SYS$SYSDEVICE:[EMC.KITS]` by entering:

   ```
   backup/select=instcli.com InstallKit/sav instcli.com;
   ```

2. With both files (`instcli.com` and *`InstallKit`*) in the same temporary directory, run the installation procedure by entering:

   ```
   @instcli.com
   ```

3. At the following prompt, specify whether to allow lower privileged users to execute `sym*` commands.

   ```
   Do you want to enable lower privilege user capability?
   ```

   A `[Y]es` response will enable lower privileged users to execute commands. Step 5 on page 68 describes the privileges these users require.

   Note: For these users to execute `sym*` commands, the base daemon (**storapid**) must be running. Currently, `sym*` commands executed through **storapid** can run slower than without the **storapid**.

   The installation produces the following DCL command procedures:

   - `emc_cli.com` should be called by the system `login.com` or by each user's login procedure.
   - `emc_start_storsrvd.com` can be placed in SYSTARTUP_VMS.COM file to start storsrvd on system startup. In a cluster environment, this should be done for only one machine in the cluster since this release only supports one storsrvd process running in the cluster.

- `emc_install_sys_specific.com` is generated to provide a way to install the data directories in the `sys$specific` directory on each node in a cluster. This allows you to run separate daemons on each of the machines in a cluster. At this point in the installation, this DCL procedure has already been executed on the machine where Solutions Enabler was installed.

  Note: After the installation, all the data files from the installation will be located in the `sys$specific:[emc.symapi]` directories. If there were data files located in a previous installation area, the following files will be copied from the previous installation area to the `sys$specific:[emc.symapi]` directories:

  - The `config` directory files are copied from the previous installation area to the `sys$specific:[emc.symapi.config]` directory.

  - The database file for the machine on which Solutions Enabler is being installed is copied from the previous installation area to the `sys$specific:[emc.symapi.db]` directory.

  The log directory files are copied from the previous installation area to the sys$specific:[emc.symapi.log] directory.

  The previous installation area data files and directories will remain intact until all the nodes in a cluster have executed the emc_install_sys_specific.com at which time they are deleted. Even though they remain intact they are not used by the just installed software.

  ---

4. Ensure that each SYMCLI user's login procedure calls the `emc_cli.com` procedure to establish their proper SYMCLI environment.

5. Each user must have the following privileges for the SYMCLI to properly function. Take care when granting these privileges.

   NETMBX — Can create network device.
   SYSLCK — Can lock system wide resources.
   SYSNAM — Can insert in the system logical name table.
   CMKRNL — Can change mode to kernel.

   In addition to the above privileges, users who will be installing and controlling the daemons, require the following privileges:

   DIAGNOSE — Can diagnose devices.
   PHY_IO — Can perform physical I/O.

SHMEM — Can create/delete objects in shared memory.
SYSPRV — Can access objects by way of system protection.
WORLD — Can affect other processes in the world.
Users with lower privileges require the EMCSERVERS right so they can run the `sym*` commands.

6. Set the following minimum process quotas for each user account:

   FILLM:1000
   BIOLM:300
   DIOLM:300
   ASTLM:500
   ENQLM:4000
   BYTLM:500000
   WSEXTENT:32768

7. You can use the following formulas to calculate an approximation of the WSdef and Pgflquo quotas you should use. Depending on the configuration, you may need to set these values higher. You should revaluate these values if the configuration changes significantly.

   - For the WSdef quota, use the following formula:

     (B + ((S * SN) + (D * DN) + (V * VN) + (P * PN) + (H * HN) + (G * GN))

   - For the Pgflquo quota, use the following formula:

     (B + (S *SN) + (S * RN) + (D * DN) + (V * VN) + (P * PN) + (H * HN) + (G * GN))

   Where:

   B = Minimum base of 10000 pagelets.

   S = 14900 pagelets per Symmetrix array.

   SN = Number of locally attached Symmetrix arrays.

   RN = Number of remotely attached Symmetrix arrays.

   D = Two pagelets per disk.

   DN = Number of disks. This is the total number of devices when adding up single devices, RAID members, meta members, etc. that Solution Enabler will see in *all* arrays attached to the host.

   V = One pagelet per volume.

VN = Number of volumes. This is the number of OpenVMS volumes (1DGAxxxx as well as shadow volumes) that this host will see on all arrays visible to this host.

G = 12 pagelets per group.

GN = Number of groups. This is the total number of Solution Enabler disk groups that Solutions Enabler will be able to see on all arrays connected to this host.

P = One pagelet per physical disk.

PN = Number of physical disks. This the total number of all devices on all the arrays attached to this host which Solutions Enabler will see.

H = One pagelet per hyper volume.

HN = Number of hyper volumes. This is the total number of hypers visible to Solutions Enabler on all arrays connected to this host.

8. Once you have calculated the changes to the quotas for your configuration, edit the file **daemon_start_template.com** to reflect these new values. This file is located in `emc$root:[emc.symcli.bin]`.

9. The installation is complete. Go to "Enabling your software" on page 72.

3

Post-Install for UNIX, Windows, and OpenVMS

Once you have installed Solutions Enabler, you need to perform certain follow-up procedures to enable your software's features and to establish your command environment. This chapter provides the follow-up procedures for a Solutions Enabler installation in UNIX, Windows, and OpenVMS environments:

Enabling your software

Before you can use your installed Solutions Enabler components, you must first enable each component's functionality in the API by entering the appropriate license keys.

License keys

Table 10 lists the possible Solutions Enabler keys.

Table 10 License matrix (1 of 3)

Required component license	Command available
SYMAPI Server	storsrvd
Base	symacl
	symaudit
	symauth
	symbcv
	symcfg
	symcg
	symcli
	symdev
	symdg
	symdisk
	symdrv
	symevent
	symgate
	syminq
	symlabel
	symld
	sympd

Table 10 License matrix (2 of 3)

Required component license	Command available
Base (con't)	symqos
	symreturn
	symstat
Configuration Manager	symconfigure
DeltaMark (Change Tracker)	symchg
Device Masking Auto-provisioning Groups	symaccess
	symconnect
	symmask
	symmaskdb
	symsg (for Auto-provisioning Groups)
IPsec	symipsec
Double Checksum (Oracle PAK)	symchksum
Dynamic Cache Partitioning	symqos -cp
FAST for V-Max	symfast
	symsg (for FAST operations)
	symtier
Mapping Solution (SRM)	symhost
	symhostfs
	symioctl
	symlv
	sympart
	symrdb
	symrslv
	symvg

Table 10 License matrix (3 of 3)

Required component license	Command available
Open Replicator/LM[a] Open Replicator/DM[b]	symrcopy
Optimization (Control)	symmigrate
	symoptmz
SRDF®/Synchronous	symrdf
	symioctl
SRDF/A	symrdf set mode async (*Asynchronous mode support*)
SRDF/Automated Replication	symrecover symreplicate[c]
SRDF/Consistency Groups	symcg -cg *CgName* enable
SRDF/Star	symstar[d]
Symmetrix® Priority Control	symqos -pst
TimeFinder® or TimeFinder/Mirror	symioctl
	symmir
	symreturn
TimeFinder/Clone	symclone symmir[e]
TimeFinder/Consistency Groups	symmir split -consistent symreplicate start -consistent symsnap activate -consistent symclone activate -consistent
TimeFinder/Snap	symsnap

a. Online Pull only.

b. All functions except Online Pull.

c. Also requires SRDF and TimeFinder licenses.

d. Also requires SRDF/A, SRDF/S, and SRDF/CG licenses.

e. For Emulation mode.

Enabling components

To enable the components:

1. Ensure you are still at the install path by entering the following according to your operating system.

Operating system	Directory
UNIX	`/usr/symcli/bin`
Windows	`C:\Program Files\EMC\SYMCLI\bin`
OpenVMS	`SYMCLI$BIN`

2. Invoke the Solutions Enabler License Management Facility (LMF) by entering the following:

   `symlmf`

   The LMF displays the following :

```
      E M C   S O L U T I O N S   E N A B L E R
SOLUTIONS ENABLER LICENSE MANAGEMENT FACILITY
```

3. At the following prompt, enter **`y`** to continue the registration:

```
Register Solutions Enabler LICENSE Key (y/[n]) ? y
```

4. At the following prompt, enter the license key of the component you want to enable:

```
Enter Solutions Enabler License Key:
```

   LMF displays a message indicating success. For example, the following display indicates that the Base component was registered successfully:

```
The Solutions Enabler License Key for the BASE feature
   was successfully registered.
```

   Note: This message does not display in an OpenVMS environment.

5. At the following prompt, enter **`y`** to register another component, or **`n`** to quit:

```
Register Solutions Enabler License Key (y/[n]) ? y
```

6. Verify that the registered components are in the `symapi_licenses.dat` license file in the SYMAPI configuration directory.[1]

1. The location of this directory varies according to the operating system. For more information, refer to Appendix G.

Initial steps for post-install of Solutions Enabler

This section describes the initial steps you must consider before you begin using Solutions Enabler SYMCLI commands.

Building the SYMAPI database

Before using the SYMCLI commands, you need to run the `symcfg discover` command to build your configuration (SYMAPI) database. This needs to be done once after installation, and after any changes are made to your Symmetrix configuration.

Note: To include information on CLARiiON arrays in the SYMAPI database, you must perform an assisted discovery. For more information on building the SYMAPI database, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

Setting environment variables

After installing Solutions Enabler, you should set the environment variables or paths so you can directly access both the SYMCLI commands and the online help (man pages). The online help path allows you direct access to descriptions of the command set.

Note: For information on setting these variables, refer to "Setting the CLI path" on page 79 and "Setting the online help path" on page 80.

SYMCLI also provides additional environment variables that you can preset to streamline your command line session. These variables can be set to common argument values for a series of associated commands, which eliminates repeated key strokes for your session.

To view a list of environment variables that can be set for a given SYMCLI session, enter:

```
symcli -env
```

To view the environment variables that you currently have set, enter:

```
symcli -def
```

Note: For a complete list of the SYMCLI environment variables, refer to the *Solutions Enabler SYMCLI Command Reference HTML Help*.

Setting access permissions to directories

By default, the completed Solutions Enabler installation disables write access to other users beyond the owner. If you desire a different permission scheme, you can change it now.

Starting the SCSI generic driver

Linux Kernel 2.4 requires that the SCSI generic driver be running. You can either compile it into the kernel or compile it as a loadable kernel module.

Note: For instructions, refer to the `README` file in the top level directory of your Linux source package.

Note: The SCSI generic driver is not required in Linux Kernel 2.6 or higher.

Setting the CLI path

Before using SYMCLI, append the SYMCLI binary directories to your PATH environment variable according to your operating system.

UNIX

For UNIX C shell, ensure the following SYMCLI directories are appended to variable PATH:

```
set path = ($path /usr/symcli/bin)
```

For UNIX Korn or Bourne shell, ensure the following SYMCLI directories are appended to variable PATH:

```
PATH=$PATH:/usr/symcli/bin
export PATH
```

Windows

For Windows, ensure the following SYMCLI directories are appended to the MS-DOS variable PATH:

```
C:\Program Files\EMC\SYMCLI\bin
```

OpenVMS

For OpenVMS, ensure the following SYMCLI directory has been defined for all users (use `emc_cli.com` in the system `login.com`):

```
SHOW LOGICAL SYMCLI$BIN
```

Setting the online help path

A complete set of online help (man pages) is provided for SYMCLI. To access these man pages in your environment, do the following according to your operating system.

UNIX For UNIX C shell, ensure the following man page directories are added to variable `MANPATH`:

```
setenv MANPATH ${MANPATH}:/usr/storapi/man;/usr/storapi/storman
```

For UNIX Korn or Bourne shell, ensure the following man page directories are added to variable `MANPATH`:

```
export MANPATH=$MANPATH:/usr/storapi/man;/usr/storapi/storman
```

Windows For Windows, the manual pages are located, by default, in the following directory:

```
C:\Program Files\EMC\SYMCLI\man
```

To open a file, double-click it and select **NotePad** from the **Open With** dialog box.

OpenVMS For OpenVMS, you can view help pages with the DCL utility SYMHELP.

Managing Symmetrix gatekeeper devices

Low level I/O commands executed using SYMCLI are routed to the Symmetrix array by way of a Symmetrix storage device that is specified as a *gatekeeper*. The gatekeeper device allows SYMCLI commands to retrieve configuration and status information from the Symmetrix array without interfering with normal Symmetrix operations. A gatekeeper is not intended to store data and is usually configured as a small device (under 10 MB). The gatekeeper must be accessible from the host where the commands are being executed.

Note: CLARiiON storage systems do not use gatekeepers.

Using the gkavoid and gkselect files

The `gkavoid` file affects calls to various online type SYMCLI commands which use a gatekeeper to communicate to a Symmetrix array. A gatekeeper whose PdevName 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 physical device names with one PdevName (`/dev/rdsk/c2t0d1s2`) per line.

The gatekeeper selection file (`gkselect`) is formatted with physical device names, with one PdevName (for example, `/dev/rdsk/c2t0d1s2`) per line. Those devices whose PdevNames match any of the entries specified in the `gkselect` file will be the preferred devices used as gatekeepers. This can be useful if you want to use only specific devices as gatekeepers.

Note: If there are no devices listed in the `gkselect` file for a particular Symmetrix array or if the devices listed in the file do not exist at the time the file is read (every time a CLI command is run), then normal gatekeeper selection rules apply.

Creation of the `gkselect` file is optional, however, doing so will override any gatekeepers defined by the `symgate` command.

Note: If a device is listed in both the `gkavoid` file and the `gkselect` file, the device will be avoided.

Using a dedicated gatekeeper

A gatekeeper device can be *dedicated* or not. If a gatekeeper is dedicated (defined), its Symmetrix device should not be used by the host system for normal data processing.

Sizing a gatekeeper

When a Symmetrix array is installed, the EMC Customer Engineer selects and configures Symmetrix devices with less than 10 cylinders (less than 5 MB) for use as gatekeeper devices.

However, the gatekeeper device must be at least as large as the minimum volume size accessible by your host, which is usually, 6 cylinders, 2.8 MB. Consult your host documentation for the minimum device size accessible by your particular host to determine the minimum gatekeeper device size for your environment.

You can determine the storage size of a Symmetrix device using:

- The `sympd` command using the `list` and `show` arguments as follows:

  `list` — Displays a list of physical device names and storage size (in MBs) for a specific Symmetrix.

  `show` — Displays the parameters of a specified physical device that includes the device capacity or size in blocks and megabytes.

- The `syminq` command and specifying the physical device name.

Note: Usually the EMC Customer Service Engineer configures a few Symmetrix devices for use as gatekeepers. You can distinguish these devices in a list executed by `syminq [PdevName]` by locating a symbol `GK` next to the PdevName (physical device name). Otherwise, they are not easily distinguished from other devices in other configuration lists, except perhaps by their size, which tends to be smaller than other devices.

Note: For more information about gatekeepers, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

Managing database and gatekeeper locking

Within a SYMCLI session, gatekeeper and database locks are used to avoid conflicts in accessing a Symmetrix array by way of gatekeepers or the configuration database.

Note: CLARiiON storage systems do not use gatekeepers.

Semaphore requirements on UNIX

On a UNIX[1] system, SYMCLI allocates a system semaphore for each accessed Symmetrix gatekeeper device. These semaphores are not deallocated from the system, but are re-used whenever a resource is accessed again.

An adequate number of semaphores should be configured into the UNIX kernel to meet the SYMCLI semaphore requirements as follows:

- One semaphore ID for each Symmetrix gatekeeper device.

  The number of system-wide semaphores is specified by the UNIX kernel parameter `semmns`, or its equivalent.

- A minimum of three semaphores per semaphore set.

  The maximum number of semaphores per semaphore set is specified by the UNIX kernel parameter `semmsl`, or its equivalent.

- A minimum of three operations per `semop` call.

  The maximum number of operations per `semop` call is specified by the parameter `semopn`, or its equivalent.

These requirements are usually within the bounds of the default semaphore parameter settings on a UNIX system. However, for information about maximizing these parameters on your specific platform, refer to Appendix F.

1. Solaris 10 does not use semaphores.

Meeting semaphore requirements

If the requirements are not within the bounds of the default semaphore parameter settings on a UNIX system, the UNIX kernel must be reconfigured. If the UNIX kernel is not reconfigured, the SYMCLI gatekeeper and database locking will fail. For more information about adjusting semaphore parameters for your operating system, refer to Appendix F.

Refreshing the semaphores

After you have reconfigured the UNIX kernel, you may need to reboot the UNIX system to refresh the kernel semaphore structures.

You can use the following UNIX command to view the currently allocated system semaphores:

```
ipcs -s
```

De-allocating semaphores

If you exceed the maximum number of semaphores allocated, you may need to de-allocate system semaphores in order to obtain more semaphores.

To de-allocate a system semaphore, use the following UNIX command:

```
ipcrm -s IpcID
```

Semaphore identifier

SYMCLI uses the UNIX function `ftok()` to derive a semaphore identifier from the gatekeeper or database pathname. The `ftok()` function generates a unique identifier based on the i-node number of the gatekeeper raw device name or database pathname.

The SYMCLI semaphore lock functions depend on the i-node remaining constant during the course of an operation in order to acquire and release a specific lock.

OpenVMS locking

On OpenVMS, SYMCLI uses the Distributed Lock Manager to accomplish locking. These locks are automatically de-allocated from the system when the last process, which has opened the lock, finishes or is terminated. There is no kernel configuration requirement. The lock name is derived from the gatekeeper or database pathname.

Windows locking

On Windows, SYMCLI allocates named mutexes to accomplish locking. These mutexes are automatically de-allocated from the system when the last thread which has opened the mutex finishes accessing the mutex, or is terminated. There is no mutex kernel configuration requirement. The mutex name is derived from the gatekeeper or database pathname.

Avoidance and selection files

The following optional files can exist in the SYMAPI configuration directory[1], and limit the scope or change the performance of SYMCLI online commands, particularly, `symcfg discover` and `symingq`:

- `gkavoid`
- `gkselect`
- `inqfile`
- `symavoid`

Note: These files and the following text are for experienced SYMCLI or SYMAPI users and are not a prerequisite for normal use.

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

CAUTION

Be sure to delete these files when they are no longer needed as they can cause unexpected behavior and command limitations.

Editing and file format

These are editable files with device names or Symmetrix IDs you can use to limit SYMCLI or SYMAPI from seeing certain Symmetrix arrays, devices, or gatekeepers which would otherwise be affected by various commands.

The files hold either physical device names (*PdevNames*) or Symmetrix IDs (*Symmids*) with line entries having only one device name or ID per line. Lines beginning with a "#" (comment) are ignored by SYMCLI.

1. The location of this directory varies according to the operating system. For more information, refer to Appendix G.

gkavoid and gkselect

The `gkavoid` and `gkselect` files affect calls to various online SYMCLI commands that use a gatekeeper to communicate with a Symmetrix array.

Note: For more information on using these files, refer to "Managing Symmetrix gatekeeper devices" on page 81.

inqfile

The `inqfile` file configures calls to `syminq` and `symcfg discover` to find only the PdevNames specified in this file. This can be useful if you want to limit the command(s) to view only certain Symmetrix devices from your host. The inquiry file is formatted with physical (host) device names with one PdevName per line.

Table 11 provides platform specific PdevName examples.

Table 11 PdevName examples

Operating system	Example Pdevname
UNIX	`/dev/rdsk/c2t0d2s2`
Windows	`\\.\PHYSICALDRIVE1`
OpenVMS	`$1$DGA6401:`
z/OS	`VOL001`

Note: For more information on PdevNames, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

symavoid

The `symavoid` file affects the operation of `symcfg discover` so that it does not look for devices that belong to the Symmetrix arrays specified in this file. This may be useful if there are multiple Symmetrix arrays connected to the host that you want SYMCLI to avoid. The Symmetrix avoidance file is formatted with 12-character Symmetrix IDs with one ID per line.

To obtain a list of Symmetrix IDs, enter:

```
syminq -symmids
```

Changing the default behavior of SYMCLI

The `options` file (initially installed as `README.options`) in the SYMAPI configuration directory contains behavior parameters that can be set to critically change the default behavior of SYMCLI operations, SYMAPI calls, and their control actions. It can be used to impart certain global restrictions as well as customize and streamline command line coding to your specific environment.

CAUTION

This file and the text in this chapter are for experienced SYMCLI or SYMAPI users and are not a prerequisite for normal use. Improper adjustment of these parameters can impose unwanted restriction of features or possibly render your Symmetrix environment inoperative.

The `options` file must be created and placed in the SYMAPI configuration directory.[1]

Editing the options file

Once this file is created, you can edit it to change the default behavior of certain SYMCLI or SYMAPI command options. The file contains editable parameters to set certain optional defaults in the line entries. SYMAPI ignores lines beginning with a "#" (comment).

Removing default options

To remove a 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.

Options file parameters

For `options` file parameter descriptions, refer to *Solutions Enabler SYMCLI Command Reference HTML Help.*

1. The location of this directory varies according to the operating system. For more information, refer to Appendix G.

Oracle multiple instances through a remote server

If you have the Storage Resource Management (SRM) license and intend to perform database mapping calls from your host to a remote server that has more than one Oracle instance, you must complete the following procedure:

1. With the remote SYMAPI service stopped, set the remote server UNIX environment variables `ORACLE_HOME` and `ORACLE_SID` for the system requirements. When set, re-start `storsrvd`.
2. Configure Oracle SQL*Net (V7) or Net8 to include other instance names (TNS names) in a network service. The TNS names are located in the `$ORACLE_HOME/network/admin/tnsnames.ora` file. The Oracle instance to which your `ORACLE_HOME` points is the only instance that must have the TNS names registered.
3. Configure the Oracle listener service for the other Oracle instances with which you need to work.
4. Test your Oracle environment for a valid configuration by running `$ORACLE_HOME/bin/sqlplus` as follows:

   `sqlplus` *`user/passwd@service`*

   where:

   `user/passwd` describes your Oracle username and password

   `service` is the TNS name you registered for the Oracle instance.

   Note: For more information about configuring SQL*Net or Net8, refer to the appropriate Oracle documentation.

5. Set the EMC environment variable `SYMCLI_RDB_CONNECT` to describe your user name, password, and service name with the format `usr/`*`passwd@service`* to the instance of choice.

Client/server RDBMS environment variable behavior

The commands `symioctl` and `symrdb` scan the client's current environment variables and apply them across the client/server connection. For example, when the following is invoked from the client:

```
symrdb -type oracle list
```

`symrdb` will search for ORACLE_HOME and ORACLE_SID on the client side. If found, the variables are passed to the SYMAPI server and used with subsequent database mapping calls.

Set the LD_LIBRARY_PATH environment variable for all databases except Oracle and SQL Server.

Setting up daemons for distributed application support

To improve performance on a number of applications or scripts running at once, you can employ Solutions Enabler daemons (services) that run in the background with root privileges to a local Symmetrix storage resource. Applications do not have to run as a privileged user.

The base daemon (**storapid**) coordinates all Symmetrix locks and parallel application syscalls to your operating system kernel, which optimizes their operations (such as TimeFinder-type actions).

For storage resource management (SRM) applications, there are a number of vendor-specific database daemons available to improve the speed of database access or mapping operation. SRM database performance is improved by using a persistent database connection, a fast communication mechanism, and parallel operations. For SRM, a single database daemon can support connections to multiple instances/databases. In addition, there is also an SRM daemon (**storsrmd** and **storsrmd64**) that allows non-root users and non-administrators to perform certain SRM operations.

When your host is locally-connected to the Symmetrix array, applications and daemons must reside in that host. However, for client/server systems, the storage management applications reside in the client, and most of the daemons must reside in the SYMAPI server. The one exception to this is the event daemon, which runs on both the client and server.

Table 12 lists the available daemons. Additional information is contained in the specific documentation for each. Note that on certain platforms, only some of these daemons are supported.

Table 12 Daemon support matrix (1 of 2)

Daemon name	Platforms supported	Description	Daemon-specific parameter documentation
storapid	UNIX[a], Win32, z/OS, AS400, BS2000, Open VMS	Base API daemon	Refer to "Managing the base daemon" on page 98 in this guide.
storgnsd	UNIX, Win32	Group Name Services daemon	*EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*
storrdfd	UNIX, Win32	RDF daemon	*EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*

Table 12 Daemon support matrix (2 of 2)

Daemon name	Platforms supported	Description	Daemon-specific parameter documentation
storevntd	UNIX, Win32, z/OS	Event daemon	Refer to "Setting up the event daemon for monitoring" on page 101 in this guide.
storsrvd	UNIX, Win32, z/OS, AS400, BS2000, OpenVMS	SYMAPI Server daemon	Refer to Chapter 4 in this guide.
storwatchd	UNIX, BS2000, Open VMS	UNIX only: Watchdog daemon	*EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*
storsrmd storsrmd64	Solaris, AIX, HP-UX	SRM daemon	*EMC Solutions Enabler Storage Resource Management CLI Product Guide*
storstpd	UNIX, Windows	Statistics daemon	
stororad		SRM daemon for Oracle DB	
storora64d		SRM daemon for Oracle DB (64-bit)	
storudbd		SRM daemon for UDB DB	
storsqld		SRM daemon for SQL DB	
storsybs12d		SRM daemon for Sybase DB - version 12	
storsybs12.5d		SRM daemon for Sybase DB - version 12.5	
storsybs12.5_64d		SRM daemon for Sybase DB - version 12.5 (64-bit)	

a. UNIX represents Sun, AIX, HP-UX, Linux, and OSF1/Tru64 systems.

For information on using daemons, refer to the remainder of this chapter.

Starting daemons

Most daemons are automatically started as their services are required. For example, **storgnsd** is automatically started the first time a group operation is performed.

However, in situations where you need to manually start a daemon, you can use the following command:

```
stordaemon start DaemonName [-wait Seconds]
```

By default, the `stordaemon` command waits 30 seconds to verify that the daemon is running. To override this, use the `-wait` option. For example, to start an SRM daemon for an Oracle database and wait five seconds for it to come up, enter:

```
stordaemon start stororad -wait 5
```

In an OVMS cluster, a daemon can be started on any member of the cluster as long as the DCL command procedure `emc_install_sys_specific.com` has been executed on the member machine. For example, this allows a base daemon to be started on each member of the cluster.

Stopping daemons

To stop a daemon, apply the following command:

```
stordaemon shutdown DaemonName|all [-wait Seconds]
           [-immediate] [-abort]
```

By default, stopping a daemon causes it to no longer accept commands from client processes using it services; it does not actually exit until all client programs using its services exit first.

The immediate flag causes the daemon to exit regardless of whether there are still client programs connected to it.

The `-abort` option sends a KILL signal, instead of asking the specified daemon to shut itself down. Only privileged users (root) can use this option. [Supported on UNIX only.]

Viewing daemons

To view what daemons are present, enter either of the following:

```
stordaemon list [-running] [-all] [-v]
```

or

```
stordaemon show DaemonName
```

For the database daemons, an instance identifier is appended to the daemon name. For example, a **stororad** daemon started with the instance name `ords` would display as `stororadords`.

Setting daemons to auto-start on boot

To set a daemon to automatically start upon reboot of your system, enter the following:

```
stordaemon install DaemonName -autostart
```

To undo this, enter the following:

```
stordaemon uninstall DaemonName
```

Authorizing daemon connections

Typically, daemons run with root/administrator privileges,[1] which enable them to handle the tasks required by SYMCLI commands (and any SYMAPI call) that require privileged access. This enables non-privileged users to run the SYMAPI application.

For example, when a SYMAPI call attempts to open a gatekeeper (which requires a privileged user), the request is actually passed to the base daemon process, which will open the gatekeeper device. If you were to run `adb` and check the per-process file table, the open files would appear in the base daemon process, not in the user process. From this point on, the transfer CDB requests are passed to the base daemon since it is the process that opened the gatekeeper.

By default, the daemons only accept connection requests from users running with root or administrator privileges. For non_root users to use this feature, you need to create a `daemon_users` file (initially installed as `README.daemon_users`) with a list of allowed usernames.

The `daemon_users` file is an editable template file installed in the SYMAPI configuration directory.[2]

1. Starting with Solutions Enaber V7.1, some daemons can run as non-root on UNIX systems.
2. The location of this directory varies according to the operating system. For more information, refer to Appendix G.

Using a text editor, a System Administrator can add entries to this file using the following formats:

`smith storapid`	Local user smith is authorized to use the **storapid** daemon.
`ENG/smith storapid`	Windows local user smith in the ENG domain is authorized to use the **storapid** daemon.
`smith storora*`	The `*` is a wildcard. Local user smith is authorized to use any daemon whose name begins with `storora`. For example, the SRM Oracle DB daemons.
`smith stororad freeze,...`	Local user smith is authorized to perform freeze and thaw operations via the **stororad** daemon. The third column consists of a coma separated list of operations that the user is authorized to perform. Valid values are: • freeze: The user is authorized to to perform DB freeze and thaw operations. • startup_instance: The user is authorized to start a DB instance. • shutdown_instance: The user is authorized to shutdown a DB instance.

Note: There is no reason to add privileged users to this file, as they are automatically authorized.

Note: For more information, refer to the `daemon_users` file.

Controlling daemon behavior

The `daemon_options` file (initially installed as `README.daemon_options`) contains parameters to control the behavior of the various Solutions Enabler daemons. As each daemon starts, it reads this file and applies all applicable settings.

CAUTION

These parameters are intended for experienced Solutions Enabler users. In most cases, the daemon default settings will be sufficient.

The `daemon_options` file is an editable template file located in the SYMAPI configuration directory.[1]

1. The location of this directory varies according to the operating system. For more information, refer to Appendix G.

Using a text editor, a System Administrator can add lines to this file using either of the following formats:

`NAME = VALUE`	Sets the parameter NAME for all daemons that understand this parameter.
`stororad:NAME = VALUE`	Sets the parameter NAME for only the **stororad** daemon.
`storora*:NAME = VALUE`	Sets the parameter NAME for all daemons whose name begins with `storora`. The `*` is a wildcard that can be used to match the remainder of a daemon's name.

Note: For more information, refer to the `daemon_options` file.

Controlling daemon logging

All Solutions Enabler daemons use a consistent infrastructure for logging events, which you can customize using the general logging options in the `daemon_options` file (Table 13). In addition, the `daemon_options` file also includes daemon-specific options that allow you to further customize logging for a particular daemon (for example, **storevntd** and **storsrvd**).

By default, each daemon records its log data in a pair of files (*`daemon_name`*`.log0` and *`daemon_name`*`.log1`) in the Solutions Enabler logging directory. Using this method, the daemons will alternate logging from one file to the other as they become full.

Optionally, you can configure each daemon to record its logs to a dated log file in the form *`daemon_name-yyyymmdd`*`.log`. Using this method, each daemon will begin recording to a newly dated log file on the first write after 12 A.M.

Table 13 shows the general logging configuration options you can use to customize the Solutions Enabler daemon log files. For details on the syntax and values, refer to the sample `daemon_options` file installed in the configuration directory.

Table 13 General logging configuration options in the daemon_options file

Option	Description
logfile_type	Controls file switching strategy. Possible values are WRAP or DATED.
logfile_size	Used for wrapping log files, this option specifies the maximum number of KBs to write before a switch to the other file of the pair.
logfile_retention	Used for dated log files, this option indicates how many days to retain old log files.
logfile_perms	Specifies the permissions on any newly created log files.

For logging configuration options specific to the event daemon, refer to "Setting up the event daemon for monitoring" on page 101, and for options specific to the SYMAPI server daemon, refer to "Specifying server behavior" on page 134.

Managing the base daemon

The base daemon (**storapid**) (Figure 3) provides centralized gatekeeper device management for all Solutions Enabler applications requiring access to Symmetrix arrays, along with the GNS and RDF daemons. This alleviates contention when there are limited gatekeeper resources available and also eliminates the need for every client to constantly select, open, lock, and ping for an available gatekeeper device for every online function.

Additionally, the base daemon monitors Symmetrix External Locks (SEL) and Device External Locks (DEL), and will automatically release any lock(s) held by a crashed application. The base daemon also eliminates the need for Solutions Enabler applications to run as root.

Each host running an instance of the RDF daemon (**storrdfd**) must also run the base daemon, as it requires the use of the gatekeeper management services.

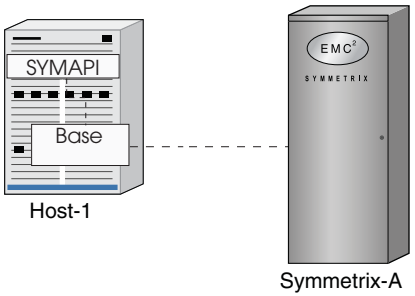

Figure 3 Base daemon

Starting the base daemon

By default, the base daemon will automatically start the first time a Solutions Enabler application attempts to access a Symmetrix array. In addition, you can use either of the following methods to start the base daemon:

- Manually start the daemon via the `stordaemon` command line utility as follows:

  ```
  stordaemon start storapid [-wait Seconds]
  ```

  Note: For more information on this command, refer to "Starting daemons" on page 93.

- Set the base daemon to automatically start every time the local host is booted using the following command:

  ```
  stordaemon install storapid -autostart
  ```

  Note: Starting with Solutions Enabler V7.1, storapid is installed with the -autostart option set by default.

Manually pre-starting the daemon will eliminate any performance delay incurred when the base daemon needs to be started by an application the first time it tries to connect.

If the base daemon abnormally terminates, the Solutions Enabler watchdog daemon (storwatchd) will automatically restart it. This ensures that the base daemon is always running.

Stopping the base daemon

To stop the base daemon, use the following command:

```
stordaemon shutdown storapid | all [-wait Seconds]
   [-immediate] [-abort]
```

Applying the `-all` option will stop all of the daemons currently running.

If there are applications with connections to the base daemon, you can use the `-immediate` option to shut it down immediately; otherwise, it will not shutdown until the applications are done using it.

The `-abort` option sends a KILL signal, instead of asking the base daemon to shut itself down. Only privileged users (root) can use this option. [Supported on UNIX only.]

Setting the optional base daemon behavior parameters

The `daemon_options` file contains a set of parameters that can be modified to affect base daemon behavior. The file contains editable behavior parameters set to certain optional defaults in the line entries. Commented lines beginning with a pound sign (#) are ignored.

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

Table 14 lists the possible optional base daemon parameters.

Table 14 Base daemon optional behavior parameters[a]

Parameter	= <OptValue | defaultvalue>	Description
storapid:inquiry_timeout	0 - *nn*, -1 | 900	Specifies how long (in seconds) inquiry results are to remain in cache before expiring, and new data retrieved from the host and array. A value of `-1` indicates the data *never* expires. A value of zero indicates the data *always* expires.

a. For more information on the available parameters, refer to the `daemon_options` file.

Setting up the event daemon for monitoring

In UNIX, Linux, and Windows environments,[1] the event daemon (**storevntd**) enables you to monitor Symmetrix operations by detecting and reporting events as they happen. The event daemon continually collects Symmetrix event information in real-time, filters the events by severity and type, and responds by doing the following:

- Alerting client event applications and/or:
- Logging events to specified targets

When using the daemon with a client event application (for example, the Symmetrix Management Console), the application registers with the event daemon, specifying the events in which it is interested. When used in this manner, the daemon will automatically start when the client application requests its services.

When configuring the daemon to log events, you can specify to log the events to the UNIX Syslog, the Windows Event log, SNMP, and/or a file on disk. When used in this manner, you should configure the daemon to automatically start at system boot.

Starting the event daemon

By default, the event daemon will automatically start the first time a Solutions Enabler application requires its services. In addition, you can use either of the following methods to start the event daemon:

- Manually start the event daemon via the `stordaemon` command line utility as follows:

  `stordaemon start storevntd [-wait` *`Seconds`*`]`

  Note: For more information on this command, refer to "Starting daemons" on page 93.

- Set the daemon to automatically start every time the local host is booted using the following command:

  `stordaemon install storevntd -autostart`

1. The event daemon is supported in the z/OS environment per Symmetrix Management Console requirements. For more information, refer to "Running the event daemon on z/OS" on page 165.

Note: You should configure the daemon to automatically start at system boot when you will be using it to log events to a Syslog, Event log, SNMP, or file on disk.

Reloading the event daemon

To reload the event daemon, run the following command:

```
stordaemon action storevntd -cmd reload
```

Issuing the `reload` command causes the daemon to re-read the contents of the `daemon_options` file.

Listing supported event categories

To view a list of event categories currently supported by a running event daemon:

1. Run the following command to load the Symmetrix event module:

   ```
   stordaemon action storevntd -cmd load_plugin Symmetrix
   ```

2. Run the following command to list the supported event categories:

   ```
   stordaemon action storevntd -cmd list -categories
   ```

Stopping the event daemon

To stop the event daemon, run the following command:

```
stordaemon shutdown storevntd | all [-wait Seconds]
```

Note: For more information on using the `shutdown` command, refer to "Stopping daemons" on page 93.

Enabling event logging

Enabling event logging involves the following steps:

1. Specify a logging mechanism.
2. Configure an event target.
3. Build an event category list.

The remainder of this section explains each of these steps in detail.

Note: Changes made to the `daemon_options` file while the daemon is running will not take effect until you issue a `stordaemon reload` command, as described in "Reloading the event daemon" on page 102.

Step 1: Specify a logging mechanism

To specify a logging mechanism, define the following parameter in the `daemon_options` file:

```
storevntd:log_event_targets = snmp
                              syslog
                              system
                              file
```

where:

`snmp` specifies to log events by way of SNMP traps.

`syslog` (supported on all platforms) specifies to log events to a Syslog server across the network, bypassing (if on UNIX) the local host's Syslog service and its configuration settings.

`system` does the following depending on the operating system:

- In UNIX, it specifies to log events to local host's Syslog services. The Syslog's configuration settings control where it directs the message.
- In Windows, it specifies to log events to the Windows Event Log.

`file` specifies to log events to a file on disk.

For example:

```
storevntd:log_event_targets = snmp system
```

Step 2: Configure an event target

To configure an event target, do the following based on the logging mechanism you specified in "Step 1: Specify a logging mechanism" above:

- If you specified to log events by way of SNMP (`snmp` option), complete "Step 2A: Configure an SNMP event target" on page 104.
- If you specified to log events in a log file (`file` option), continue with "Step 2B: Configure a log file" on page 106.
- If you specified to log events to the Syslog server across the network (`syslog` option), continue with "Step 2C: Configure a Syslog target" on page 107.
- If you specified to log events to Syslog or the Windows Event Log, (`system` option), you do not have to configure an event target. In this case, you should continue with "Step 3: Build an event category list" on page 107.

Step 2A: Configure an SNMP event target

The event daemon provides the necessary SNMP MIB support and trap generation services required to monitor the status of Symmetrix storage environments from third-party enterprise management frameworks.

The event daemon includes a loadable SNMP library which, once enabled and configured in the `daemon_options` file, acts as a self contained SNMP agent. It is responsible for maintaining internal Fibre Alliance MIB (V3.0) tables, responding to SNMP browse requests, and generating traps in response to events.

For an application to receive SNMP trap information from the event daemon, you must specify it as a trap target by defining the following parameter in the `daemon_options` file:

```
storevntd:snmp_trap_client_registration =
   IP,Port,Filter,State
```

where:

`IP` is the application's IP address.

`Port` is the port on which the application will be listening for the trap.

Filter is the trap filtering severity level as defined in the fcmgmt MIB. The application will only receive traps of the specified severity level (or lesser). The possible values range from 1 through 10, where:

1 = Unknown	6 = Warning
2 = Emergency	7 = Notify
3 = Alert	8 = Info
4 = Critical	9 = Debug
5 = Error	10 = Mark (all messages logged)

Note: For information on the events associated with each of these severity levels, refer to Appendix B.

State is the start up row state in the trap_client_registration table in the fcmgmt MIB. Possible values are ACTIVE and INACTIVE.

Multiple entries should be on their own line, delineated with a backslash (\) character on the preceding line.

For example, the following registration file specifies that the daemon will only send SNMP traps to the indicated clients when it detects an event of a severity level less than or equal to 5 (that is, Error, Critical, Alert, Emergency, Unknown). The daemon will ignore events with a severity level greater than 5:

```
storevntd:snmp_trap_client_registration = 111.222.333.444,162,5,ACTIVE \
                                          555.666.777.888,162,5,ACTIVE
```

Step 2B: Configure a log file

The `daemon_options` file contains parameters (Table 15) that allow you to configure the log file.

Table 15 Event log file configuration options

Parameter	= <OptValue | defaultvalue>	Description
storevntd:log_event_file_name	*LogEventFileName* | events	Specifies the base name of the event log files. This file is created in the standard Solutions Enabler log directory. For UNIX, the directory is: `var/symapi/log` For Windows, the directory is: `c:\Program Files\EMC\SYMAPI\log`
storevntd:log_event_file_type	dated | wrap	Specifies the type of file to use. `dated` specifies that a new event log file should be created each day, with the name *xxxx*-*YYYMMDD*.log. `wrap` specifies that event logging will alternate between two files (*xxxx*.log1 and *xxxx*-log) - switching from one to the other when it reaches its maximum size, as specified in the log_event_file_size parameter.
storevntd:log_event_file_size	> 0 - *nn* | 1 MB	When used with the log_event_file_type parameter set to wrap, this parameter specifies the maximum file size (in KB) allowed before wrapping to the alternate file. This value should be a decimal number greater than zero.
storevntd:log_event_file_retention	> 0 - *nn* | 3 days	When used with the log_event_file_type parameter set to dated, this parameter specifies the number of days to retain the log files. This value should be a decimal number greater than zero.
storevntd:log_event_file_perms	rw, n | r	Specifies the permissions for the event log files. `rw` specifies that anyone can read or write to the files. `r` specifies that anyone can read the files, but only the root/administrator (or whatever identity the event daemon is running as) can write to the files. `n` specifies that only the root/administrator (or whatever identity the event daemon is running as) can read and write to the files.

Step 2C: Configure a Syslog target

The `daemon_options` file contains parameters (Table 16) that allow you configure a Syslog target,.

Table 16 Event log file configuration options

Parameter	= <OptValue | defaultvalue>	Description
storevntd:log_event_syslog_host	*SyslogHostName*	Specifies the name of the host on which the Syslog server is running. This value must be supplied.
storevntd:log_event_syslog_port	*nnn* | 514	Specifies the port on which the server is listening.

Step 3: Build an event category list

All Symmetrix events are organized into categories. These categories are hierarchical in that a category can contain individual events, as well as other categories. An event category list is a mechanism for specifying the types of events for which to generate traps.

To build a event category list, define the following parameter in the `daemon_options` file:

```
storevntd:log_symmetrix_events = [sid=SymmID,]
   UID|Category ... [,ignore] [,tgt=TGT]
```

where:

SymmID is the 12-digit ID of the Symmetrix array to which the record applies.

UID is the numerical event UID value.

Category can be one or more of the following event categories, separated with a comma:

status
events (*includes the following sub-categories*)

- array subsystem
- checksum
- device
- device pool
- diagnostic
- director
- disk
- environmental
- service processor
- srdf consistency group
- srdf link
- srdf system
- srdfa session

Note: Each of the event categories may contain numerous individual events, as shown in Appendix B.

`ignore` specifies to not log information on the specified event, or if used with a sub-category, specifies to not log information on each event in the sub-category.

When using `ignore` to disable logging for events that have otherwise been enabled, the order of the fields (ignore and enable) does not matter. For example, both of the following log all events in the *status* category, except for UID 1200:

```
storevntd:log_symmetrix_events = \
        status ;\
        1200, ignore

storevntd:log_symmetrix_events = \
        1200, ignore ;\
        status
```

TGT is the specified target to which the daemon should log the event(s). Possible values are: snmp, syslog, system, and file.

The value you specify for *TGT* must match one of the values you specified in the `log_events_target` parameter; otherwise, the daemon will not log events for this record.

By default, events are logged to all targets specified in the `log_event_targets` parameter.

Multiple entries should be on their own line, delineated with a semicolon (;) and a backslash (\) character on the preceding line.

For example, the following event category list :

```
storevntd:log_symmetrix_events = \
          sid=000011112222, srdf link, srdf system ; \
          status ; \
          1003, 1004, ignore ; \
          checksum, tgt=file
```

- Logs `srdf link` and `srdf system` events from Symmetrix 000011112222.
- Logs `status` events from any Symmetrix array.
- Ignores event UID 1003 and 1004 from any Symmetrix array.
- Logs `checksum` events from any Symmetrix array to the file target (if enabled through the `log_events_targets` parameter).

Event output examples

The following examples illustrate the format of the various event outputs. In these examples:

- `Symmetrix:000000006190` is the event entity; normally a storage array.
- `date=`*`xxx`* corresponds to the date/time that the event was originally generated. If the date field contains a `Z` suffix, the date is in UTC time, otherwise, it is local time. If the example contains a second date field, it indicates when the logging service (for example, Syslog) posted the event.

Log file

The following example illustrates the format of an event as reported in a log file (target = file):

```
[evtid=1200] [date=2008-12-22T09:08:17]
   [symid=000000006190] [Device:0010] [sev=normal]  =
   Device state has changed to Offline.
```

Syslog service (local UNIX host)

The following example illustrates the format of an event as reported by Syslog service on a local UNIX host (target = system).

Note that the italicized text was generated by local Syslog service. In this case, a Solaris host:

```
Dec 22 09:08:17 l82ab139 storevntd[14505]:
   [ID 989319 user.info] [evtid=1200]
   [date=2008-12-22T09:08:17] [symid=000000006190]
   [Device:0010] [sev=normal]  = Device state has changed
   to Offline.
```

Syslog service (different system)

The following example illustrates the format of an event as reported to a Syslog service on a different host (target = syslog):

```
Dec 22 09:03:01 EMCstorevntd: [evtid=1200]
   [date=2008-12-22T04:08:17Z] [symid=000000006190]
   [Device:0010] [sev=normal]  = Device state has changed
   to Offline.
```

Windows event log

The following example illustrates the format of an event as reported in a Windows event log (target = system):

```
[evtid=1200] [date=2008-12-22T09:08:17]
   [symid=000000006190] [Device:0010] [sev=normal]  =
   Device state has changed to Offline.
```

SNMP trap SNMP traps are formatted according to the Fibre Alliance MIB (V3.0). Messages contained in a trap are the same as used with the system and file logging.

4

Remote Operations

This chapter provides information on configuring and operating Solutions Enabler in a client/server environment:

SYMCLI through a remote server

In the UNIX, Linux, Windows, and OpenVMS environments, the SYMAPI server runs in a background process started by the `stordaemon start storsrvd` command. In the z/OS environment, it runs as a job step task specified on the EXEC PGM= statement in a job stream. The server reads its configuration from the `daemon_options` file, and records log information in its own log file set, which resides in the SYMAPI logging directory.

The server is a multi-threaded program that listens for SYMAPI sessions and management requests initiated by the `stordaemon` command. The server also listens for management requests from the system operator console.

While session threads come and go, the server continues to accept connection requests until an 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, in which case the server will simply terminate all current session threads without giving them a chance to end on their own. The former method is preferred, when there is time to let sessions continue until they are done. The latter method can be used in an emergency, especially when a catastrophic condition occurs that requires a restart of the entire system.

Each session has a sequentially assigned session number, and an associated thread number. The operator can use the session number when referring to a session in a command. For example:

```
stordaemon action storsrvd –cmd show -sessions –num
   session_number
```

You can use the thread name (`SESS` *nnnn*, where *nnnn* is the session number) to identify log message issued by session threads.

Client configuration

This section explains how to configure a Solutions Enabler client.

Editing the netcnfg file

At this point in the install, the `netcnfg` file is a template and an editable file located in the SYMAPI configuration directory.[1]

Using a text editor, a System Administrator must add the network services to the file in the following format:

```
service_name domain_name network_protocol server_node_name server_network_address
   port_number security_level
```

where:

`service_name` is the name of the service.

`domain_name` should be unspecified and substituted with a hyphen (-).

`network_protocol` must be TCPIP.

`server_node_name` is the name of the server host.

`server_network_address` is the network address of the server.

Note: You can substitute a hyphen (-) for an unspecified *`server_node_name`* or *`server_network_address`*, but at least one must be specified. For more information, refer to "Considerations for specifying server_node_ name and server_network_address" on page 114.

`port_number` is the server port number.

`security_level` is the type of connection the server is expecting to negotiate. Possible values are SECURE, ANY, and NONSECURE. In addition, you can specify a hyphen (-) to use the platform's default setting. For more information, refer to "Securing remote transmissions using SSL" on page 120.

1. The location of this directory varies according to the operating system. For more information, refer to Appendix G.

Example: In the following example, three site specific service names (SYMAPI_SERVER, BACKUP_SERVER and SERVER_IP6) are specified as available by the administrator:

```
SYMAPI_SERVER -  TCPIP node001 12.345.67.89                                7777 ANY
BACKUP_SERVER -  TCPIP node002 -                                            6666 SECURE
SERVER_IP6    -  TCPIP node003 3FFE:80C0:22C:18:250:88FF:FEAD:F92F          6666 SECURE
```

Comment text can be entered by placing a pound sign (#) in the first character space of the comment line.

Considerations for specifying server_node_ name and server_network_address

Although the syntax of each service definition allows you to specify both the node name and the network address, only one is in fact required. Specifying both can serve as documentation for your expectation of the mapping between node and address, but it has no real effect on connections established between the client and the server.

Any unspecified tokens in the service definition must be replaced with a hyphen, so if either the `server_node_name` or `server_network_address` are to be omitted, be sure to place a hyphen character in its position.

Use the following general rules to decide whether to specify a real value for `server_node_name` or `server_network_address`:

- If you do not want to have to remember or look up IP addresses, or if your network administrator discourages routing by address, then specify a real value for `server_node_name` and place a hyphen in the `server_network_address` field. The SYMAPI client library will look up the node name in DNS, and will attempt to connect to the server using the list of known addresses for the node. If you specify `server_node_name`, however, you cannot predict the address that will be used to successfully connect.

  Note that the value specified in the `server_node_name` can generally be a local node without qualifying domain, or it can be a fully-qualified domain name (FQDN). Your results depend on the configuration of name resolution in your network.

Another key reason for using node name is that the client will try all eligible network addresses for a given node to complete the connection. Even though you have no specific control over the protocol or address used, the server availability may be improved using node name.

- If you want more control over the network address chosen (including the protocol) for the connection, specify a real value for `server_network_address` and place a hyphen in the `server_node_name` field. In fact, if any value is specified in the address field, it will be used, regardless of the value specified in the `server_node_name` field.

  Note that specifying the address implies that you know the protocols that will be in use on the server host. For example, if you specify an IPv4 address for a server which is no longer using IPv4 (not likely for years to come), the connection will fail. If you specify an IPv6 address for a server host whose IPv6 link is inoperative, the connection will fail. A host in this state might still be reachable over IPv4; by using the node name instead, the connection might succeed.

You can specify an IPv4 address or an IPv6 address. You may be able to use an IPv4-mapped, but a successful connection using the mapped address will depend on the whether the operating system of the server host is one that uses V4-mapping. In general, using IPv4-mapped addresses is discouraged.

Setting environment variables for remote access

To use SYMCLI through a remote SYMAPI service, you should set environment variable SYMCLI_CONNECT to an available service name of the server connection (defined in netcnfg). For example, for service name SYMAPI_SERVER, set the environment variable as follows:

```
setenv SYMCLI_CONNECT SYMAPI_SERVER        for UNIX C shell
define SYMCLI_CONNECT SYMAPI_SERVER        for OVMS
set SYMCLI_CONNECT=SYMAPI_SERVER           for Windows
```

To determine what network services are available, enter:

```
symcfg list -service
```

Connection variable SYMCLI_CONNECT_TYPE should define the local/remote mode of the local host (client). Possible values for the client are:

REMOTE

Defines a client operation in which all the remote SYMCLI commands are strictly executed on the server, and the Symmetrix database is strictly read and updated remotely.

LOCAL

Defines a local connection to the Symmetrix array. (Not used for a client-server connection.)

Example: To set the connection environment variables for a locally-cached remote operation, enter:

```
setenv SYMCLI_CONNECT_TYPE  REMOTE
```

Client/server IP interoperability

In a UNIX, LINUX, or Windows environment, the SYMAPI client and server are both capable of negotiating sessions over the traditional Internet Protocol Version 4 (IPv4) and the newer Internet Protocol Version 6 (IPv6).

The IPv6 designers expected migration from the old protocol to the new protocol to take years. They designed the new protocol for interoperation in networks where both are present. A network administrator can introduce the IPv6 protocol as a supplement to IPv4, where IPv4 hosts and IPv6-capable hosts can interoperate with minimal disruption. Over time, as network configuration is improved and problems are reduced and eliminated, IPv4 protocols can be dropped in favor of IPv6. Such a transition scheme is essential in environments where continual operation is a key business success factor.

In the UNIX, Linux, and Microsoft Windows Server environments, Solutions Enabler also supports the transition from IPv4 to IPv6 in a seamless fashion. With proper configuration of host operating systems, routers, and DNS servers, Solutions Enabler supports concurrent connections from clients using both IPv4 and IPv6. The client and server software will choose either IPv4 or IPv6 to communicate, depending on specification in configuration files of the host operating system and Solutions Enabler.

IPv6 addresses

The IPv4 address is familiar to most computer users: a 32-bit unsigned integer is displayed in a dotted-decimal string. For example, 172.23.191.20 (0xAC17BF14).

The IPv6 address supports many addressing features, but the most obvious attribute is its much wider addressing space: a 128-bit code is displayed as a series 16-bit groupings (represented in hexadecimal) separated by colons. Shorthand notation rules improve the usability of the IPv6 display address; nonetheless, an IPv6 address is not a human-friendly object. For example, one machine might be represented with this address:

3ffe:80c0:22c:18:250:8bff:fead:f92f
(0x3FFE80C0022C001802508BFFFEADF92F)

IPv4 address mapping

The interoperation of IPv4 and IPv6 varies from one operating system to another, according to the specification of IPv6. On some host operating systems, IPv4 connections are made through the native IPv4 protocol, and IPv4 addresses are represented as the dotted-decimal addresses which are familiar.

Other OS vendors have chosen to complete client connections from an IPv4 machine over IPv6, where the IPv4 address is represented as an IPv4-mapped address. An IPv4-mapped address appears in colonated-hexadecimal form, where the last 32-bits of the address are shown as the dotted-decimal IPv4 address (they may also be shown as two pairs of hexadecimal bytes). Immediately preceding the IPv4 address is the string ::FFFF:. For example, a host whose IPv4 address is 172.23.191.20 can be represented as a IPv4-mapped address as follows:

::FFFF:AC17:BF14 or
::FFFF:172.23.191.20
(0x00000000000000000000FFFFAC17BF14)

IPv4-mappped addresses are used by operating systems that do not support concurrent binding to the same port over both IPv6 and IPv4. AIX, Tru64 and Linux generally use IPv4-mapped addresses.

SunOS, HP-UX, and Microsoft Windows 2003 allow concurrent binding on both IPv6 and IPv4 protocols.

Server operation

The SYMAPI server listens for arrival of client connections on either IPv6 or IPv4 protocols, or on both where possible. The server begins by attempting to bind to the *unspecified address* using the IPv6 protocol. It then attempts to bind the unspecified address using the IPv4 protocol, as well.

The *unspecified address* is a special-purpose internet address used primarily by server applications. It indicates that an application is ready to receive a connection on any internet address configured on the host with a matching protocol. For hosts that have multiple network interfaces, it increases the availability of the server application by not limiting connections to arrive by way of a specific address.

The server insists on at least one successful bind on either IPv6 or IPv4 protocols, and will use both if available to continue initializing.

If both bind attempts fail, the server will terminate immediately, since no network is accessible or the port is in use.

When the server has finished initializing for network communication, it will write the following message to its SYMAPI log file and to the terminal device, if one is available:

```
ANR0020I SYMAPI server listening on port port over
protocols
```

Where *port* is the decimal port number to which client connections should be directed, and *protocols* are the protocols the server is using to listen for client connections. Possible values are:

- **IPv6 and IPv4** — Indicates that the server will accept connections from clients running either IPv6 or IPv4.
- **IPv6 with IPv4 mapping** — Also indicates that the server will accept connections from clients running either IPv6 or IPv4. Connections from IPv4 clients will be represented on the server side as an IPv4-mapped address (refer to "IPv4 address mapping" on page 118).
- **IPv4 only** — Indicates that IPv6 bind failed. Connections can only be accepted from IPv4 clients.

Client operation

The SYMAPI client library will attempt to connect to the server either by node name or by internet address, depending on how the service name is specified in the `netcnfg` file.

If the internet address of the server is specified, the client makes a single attempt to connect to the server. The client chooses the protocol based on the nature of the address: if it is an IPv4 address, it will specify IPv4 as the protocol. Similarly, specifying an IPv6 address (including an IPv4-mapped address) will result in the client using the IPv6 protocol to connect to the server.

If the node name of the server is specified, the client will lookup the server host by name. Such a lookup operation can return a list of candidate addresses, potentially including both IPv4 and IPv6 addresses. The client library will try to connect to all eligible addresses until either a connection attempt succeeds, or the list is exhausted with no successes. The list of eligible server addresses depends on the static and dynamic name resolution configuration of the host on which the client is running.

Client/server security

This section explains how to configure Solutions Enabler to operate in a secure environment.

Securing remote transmissions using SSL

By default, the SYMAPI client and server, on platforms that will support it, are initially configured to negotiate only secure sessions. To modify this default behavior, you can configure the security level at which the client and server are operating.

When configuring the security level, it is important to know that the security level specifies the capability of the local side and the local side's expectation of the remote side. In addition, it is important to know whether the host is SSL-capable or SSL-incapable.

The possible security levels are:

- **Level 3 (SECURE)** — (Default) Indicates that only secure sessions will be negotiated between the client and server. This is the highest level of security and should only be used when there is no chance of an SSL-incapable client attempting to connect with the server, or a new client connecting to an SSL-incapable server.
- **Level 2 (ANY)** — Indicates that either secure or non-secure sessions will be negotiated between the client and server on SSL-capable platforms.
- **Level 1 (NONSECURE)** — Indicates that only non-secure sessions will be negotiated between the client and server. This level is intended as a last resort in situations where SSL cannot be used for some reason, or is undesirable. In addition, this level can also be useful in matters of performance and availability.

Note: The default security level is SECURE on platforms that support secure communications and NONSECURE on platforms that do not support secure communications.

The messages ANR0141E through ANR0145E, ANR0147E, ANR0148E, and ANR0150E through ANR0153E may be issued should SSL related problems occur.

Configuring the client

To configure the client's security level, use either of the following methods:

Note: The following methods are listed in order of precedence to the client.

- Define the client's security level in the `netcnfg` file. For instructions on editing the `netcnfg` file, refer to "Editing the netcnfg file" on page 113.
- Define the client's security level by setting the SYMAPI_SERVER_SECURITY_LEVEL option in the `options` file. This file provides the global security level when no other source (`netcnfg` file) specifies a more specific level. For instructions on editing the `options` file, refer to "Changing the default behavior of SYMCLI" on page 88.

Configuring the server

To configure the server's security level, use one of the following methods:

Note: The following methods are listed in order of precedence to the server.

- Define the server's security level by setting the `security_level` option in the `daemon_options` file. For instructions on editing the `daemon_options` file, refer to "Controlling daemon behavior" on page 95.
- Define the server's security level by setting the SYMAPI_SERVER_SECURITY_LEVEL option in the `options` file. This file provides the global security level when it has not been specified in the `daemon_options` file. For instructions on editing the `options` file, refer to "Changing the default behavior of SYMCLI" on page 88.

Note: The `stordaemon reload storsrvd` command does not check the SYMAPI `options` file for the security level. The reload process only checks the `daemon_options` file.

Client/server behavior

Table 17 details the type of session negotiated if a client and server are at the same security level, implied or configured.

Table 17 Client/server behavior when security levels are the same

Client security level	Server security level	Negotiated session type
SECURE	SECURE	SECURE
NONSECURE	NONSECURE	NONSECURE
ANY	ANY	SECURE

If the client and server security levels are different (Table 18), either the client or server must be configured as ANY to negotiate the session.

Table 18 Client/server behavior when security level are different

Client security level	Server security level	Negotiated session type
SECURE	ANY	SECURE
ANY	SECURE	SECURE
NONSECURE	ANY	NONSECURE
ANY	NONSECURE	NONSECURE

Client/server compatibility

To secure a session, both the client and server must be running on SSL-capable platforms. Table 19 details what happens when a session is negotiated between mismatched client and server versions.

Table 19 Client/server behavior between mismatched versions

Client security level	Server[a] security level	Session results
NONSECURE	NONSECURE or ANY	Accepted
SECURE	SECURE or ANY	Accepted
NONSECURE	SECURE	Rejected by the server with return code SYMAPI_C_SECLEVEL_REMOTE_REFUSAL
SECURE	NONSECURE	Rejected by the server with return code SYMAPI_C_SECLEVEL_REMOTE_REFUSAL

a. In the z/OS environment, the applicable server version is V6.2.0 since SSL support for z/OS was introduced in V6.2.0. In Windows IA-64 and Windows x-64 environments, the applicable version is V6.4.0 since SSL support was introduced for these platforms in V6.4.0.

Certificate files

The following certificate files enable a client to verify a server's identity and a server to verify a client's identity:

- `symapisrv_cert.pem` is the SYMAPI server's certificate file. It is created specifically for its particular host during installation. It is signed by the EMC SPEA Root certificate. This file must be in the `cert` directory on the SYMAPI client and server for client/server security to work.
- `symapisrv_trust.pem` is the EMC SPEA Root certificate used to sign the SYMAPI server certificate file. It must be in the `cert` directory on every client and server.
- `symapisrv_key.pem` is the SYMAPI key file. It is created specifically for its particular host during installation. It is generated during the certificate creation process. This file must be in the `cert` directory on the SYMAPI client and server for client/server security to work.

For information on the location of `cert` directory, refer to Appendix G.

Note: The following sections describe how to use the `manage_server_cert.sh` and `manage_server_cert.bat` scripts to recreate the certificate files for special circumstances. While recreating the certificates, it is important to note that host names, server names, and cluster names cannot exceed 55 characters in length. This number includes the spaces that are used to separate arguments. To workaround this limitation, you can use the simple wildcarding feature. An asterisk (*) will match zero or more characters in a host name. For example, the string `emc*` in a certificate will match any host names starting with `emc` (e.g., `emc001`, `emc002`, etc.). For backwards compatibility, changing from fully qualified name to simple names is still supported but maybe discontinued in a future release.

In addition, when recreating certificate files in a Windows environment, the number of arguments is limited to eight, not including the create argument.

Changing a host's name

If you change the name of the host on which the client or server is running, you must also change its name in its certificate. To do this, you must run the `manage_server_cert.sh create` command in the host's `cert` directory. This command reads the host name from the environment and recreates the certificate. The format for this command is different for each operating system:

- In UNIX and Linux environments, run the following:

  `/usr/storapi/bin/manage_server_cert.sh create`

- In Windows environments, you must specify the full path to the `bin` directory along with the command. For example:

  `\Program Files\EMC\SYMCLI\bin\`
  `manage_server_cert.bat create`

The following is example illustrates the use of the `manage_server_cert.sh create` command:

```
/usr/storapi/bin/manage_server_cert.sh create
Generating a 1024 bit RSA private key
.........++++++
.....................................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd foo.emc.com/ST=MA/C=US/L=Hopkinton/
   emailAddress=support@emc.com/O=EMC/OU=SE
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the directory
   /var/symapi/config/cert.
```

Replacing SYMAPI-generated certificates

You can replace a SYMAPI-generated certificate with one generated by your certificate authority or an external certificate authority (for example, Verisign).

When creating a replacement certificate, you must adhere to the following rules in order for the certificate to work with the client/server:

- The common name of the certificate must include `storsrvd` and the fully qualified name of the host on which the certificate will be installed separated by a space. For example: `storsrvd foo.emc.com`
- The certificate must be created in pem format.

Once you have created the certificate, do the following on the host:

1. Move the certificate file and the associated key file to the `cert` directory.
2. Move the trusted certificate file(s) that signed the created certificate to the same `cert` directory.
3. Run the `manage_server_cert update` command in the `cert` directory. The format for this command differs for each operating system:
   - In UNIX and Linux environments, run the following:

     `/usr/storapi/bin/manage_server_cert.sh update`
   - In Windows environments, you must specify the full path to the `bin` directory along with the command. For example:

     `\Program Files\EMC\SYMCLI\bin\`
     `manage_server_cert.bat update`
4. Edit the following options in the `config` directory:

   Note: If this is a client certificate, edit these options in the `options` file. If this is a server certificate, edit these options in the `daemon_options` file.

   - Change the SYMAPI_SECURITY_ALT_CERT_FILE option to equal the name of the new certificate file.
   - Change the SYMAPI_SECURITY_ALT_KEY_FILE option to equal the name of the new key file.

Making the server more secure

To allow you to recreate the SYMAPI-generated certificate on a host, certain files are left on the host. If there is no need to recreate the SYMAPI-generated certificate, it is recommended that you remove these files. Note that once you remove these files, you will need to perform a full Solutions Enabler install should you need to recreate the SYMAPI-generated certificate files.

To remove these files, run the `manage_server_cert secure` command in the `cert` directory. The format for this command differs for each operating system:

- In UNIX and Linux environments, run the following:

  `/usr/storapi/bin/manage_server_cert.sh secure`

- In Windows environments, you must specify the full path to the `bin` directory along with the command. For example:

  `\Program Files\EMC\SYMCLI\bin\`
  `manage_server_cert.bat secure`

CAUTION

It is recommended that you only run this command on systems that will not perform client/server operations.

Working with a host in multiple domains

When client/server is running on a host that is in multiple domains (for example, foo.emc.com and foo.example.com), a client or server may not be able to verify the certificate being sent by the other host. This occurs because only one fully qualified name is written to the certificate during installation.

To correct this problem, you can create a certificate with a hostname wildcard in it by running the following command from the `cert` directory:

Note: For more on wildcards, refer to "Certificate files" on page 123.

- In UNIX and Linux environments, run the following:

  `/usr/storapi/bin/manage_server_cert.sh create`
  `*host_wildcard* ...`

- In Windows environments, you must specify the full path to the `bin` directory along with the command. For example:

  `\Program Files\EMC\SYMCLI\bin\manage_server_cert.bat`
  `create *host_wildcard* ...`

For example, running the following command in a UNIX environment:

```
/usr/storapi/bin/manage_server_cert.sh create
   foo.*.com
```

produces output similar to the following:

```
Generating a 1024 bit RSA private key
.........++++++
.....................................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd foo.*.com
/ST=MA/C=US/L=Hopkinton/emailAddress=support@emc.com/O=EMC/OU=SE
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the
   directory /var/symapi/config/cert.
```

Working with a Windows cluster

This section describes procedures to perform when working with Windows clusters.

Creating a certificate that includes both the host name and cluster name

During client/server operations, if either the client and/or server is in a Windows cluster, the host name sent to the other system maybe a cluster name and not a host name. When the host tries to verify the certificate, it will not find the cluster name since it is not included during installation on the server.

To correct this problem, you can create a certificate with the host name and the cluster name in it by running the following command from the `cert` directory:

```
full_path_to_bin_directory manage_server_cert.bat
   create fully_qualified_host_name fully_qualified
   _cluster_name
```

Where *`full_path_to_bin_directory`* is the location of the `bin` directory.

For example, running the following command:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\Program
   Files\EMC\SYMCLI\bin\
   manage_server_cert.bat" create foo.emc.com
   foocluster.emc.com
```

produces output similar to the following:

```
Generating a 1024 bit RSA private key
.........++++++
.....................................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd foo.emc.com
   foocluster.emc.com/ST=MA/C=US/L=Hopkinton/emailAddress=support@emc
   .com/O=EMC/OU=SE
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the
   directory C:\Program Files\EMC\SYMAPI\Config\Cert.
```

Adding virtual server names to a certificate

If virtual server names are defined in the Windows cluster, you should also add these names to the certificate by running the following command from the `cert` directory:

```
full path to bin directory manage_server_cert.bat
   create servername1 servername2 servername3
   servername4 servername5 servername6 servername7
   clustername
```

Where *`full path to bin directory`* is the location of the `bin` directory.

Note: The `manage_server_cert.bat` command allows a total of eight arguments, which allows you to add up to seven virtual server names and a cluster name to the certificate.

For example, running the following command from the `cert` directory:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\Program
   Files\EMC\SYMCLI\bin\
   manage_server_cert.bat" create svr1 svr2 svr3 svr4
   svr5 svr6 svr7 cluster
```

produces the following output:

```
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....................................................++++++
.....................++++++
writing new private key to 'symapisrv_key.pem'
-----
Loading 'screen' into random state - done
Signature ok
subject=/CN=storsrvd svr1 svr2 svr3 svr4 svr5 svr6 svr7
   cluster/ST=Massachusetts/
C=US/L=Hopkinton/emailAddress=support@emc.com/O=EMC
   Corporation/OU=Storage Platf
orm Enablers and Applications
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the
   directory
 C:\Program Files\EMC\SYMAPI\config\cert.
```

You can also use the following command to perform the same operation:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\Program
   Files\EMC\SYMCLI\bin\manage_server_cert.bat"
   create svr* cluster
```

Working with a multi-homed host

When the client/server is running on a multi-homed host (that is, a host with multiple host names), a client or server may not be able to verify the certificate being sent by the other host. This occurs because only one fully qualified name, which is obtained from the hostname command, is written to the certificate during installation.

To correct this problem, you can create a certificate for the multi-homed host containing all of its host names by running the following command from the `cert` directory:

- In UNIX and Linux environments, run the following:

  ```
  /usr/storapi/bin/manage_server_cert.sh create
     multi-homed-name1 multi-homed-name2 [...]
  ```

- In Windows environments, you must specify the full path to the `bin` directory along with the command. For example:

  ```
  \Program Files\EMC\SYMCLI\bin\manage_server_cert.bat
     create multi-homed-name1 multi-homed-name2 [...]
  ```

For example, running the following command in a UNIX environment:

```
/usr/storapi/bin/manage_server_cert.sh create
   foo1.emc.com foo2.emc.com foo3.emc.com
```

produces output similar to the following:

```
/usr/storapi/bin/manage_server_cert.sh using
   /usr/storapi/bin/storssl64 to create keys
Generating a 1024 bit RSA private key
.................++++++
.....................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd foo1.emc.com foo2.emc.com
   foo3.emc.com/ST=Massachusetts/C=US/L=Hopkinton/emailAddress=suppor
   t@emc.com/O=EMC Corporation/OU=Storage Platform Enablers and
   Applications
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the
   directory /var/symapi/config/cert.
```

Listing the host names in the certificate

To list the host names contained in the certificate, run the following operating-specific command:

- In UNIX and Linux environments, run the following:

  /usr/storapi/bin/manage_server_cert.sh list

- In Windows environments, you must specify the full path to the `bin` directory along with the command. For example:

  ```
  \Program Files\EMC\SYMCLI\bin\manage_server_cert.bat
     list
  ```

For example, running the following command in a UNIX environment:

/usr/storapi/bin/manage_server_cert.sh list

produces output similar to the following:

```
The host names in this machine's certificate:
host.emc.com
*.emc.com
```

Authorizing SYMAPI sessions

An optional file (`nethost`) for trusted-user host access can also be present in the server configuration directory.[1] When this file exists (maintained by a System Administrator), only the nodes/users listed in this file are allowed to connect to the server to execute remote SYMAPI functions. The trusted host file uses the following format:

```
node                    user-1 [, …, user-n]
address                 user-1 [, …, user-n]
*                       user-1 [, …, user-n]
node                    *
address                 *
*                       *
```

Note: * denotes a wild card for any host or any user.

1. The location of this directory varies according to the operating system. For more information, refer to Appendix G.

Note: The server considers the contents of this file *before* deciding whether it will negotiate a secure session with the client. If the client host and user are not defined in the `nethost` file, a secure session will not be negotiated. For more information on the SSL security settings, refer to "Securing remote transmissions using SSL" on page 120.

Considerations for specifying node and address

The identity of a client host may vary from the perspective of the server, since the server can accept connections from IPv4 and IPv6 clients. Thus the network address of the client could be an IPv4 or IPv6 address. If you have decided to specify the network address in the `nethost` file instead of the node name, then the exact syntax of the address is important. If you incorrectly specify an address, connections from some clients may be denied.

In general, specifying the node name (or the FQDN) is advised, since proper DNS configuration will usually ensure that the name of the host will be consistent, regardless of the network address of the client.

If you have to specify the address, keep these factors in mind:

- The rules for specifying an IPv4 address are unchanged and are simple: specify the complete address in its dotted-decimal form, without leading zeros in each octet. For example:

```
172.23.191.20          user1
10.243.142.82          user1
```

- If you want to specify an IPv6 address, you can generally expect to follow the shorthand rules which are a part of the IPv6 standard:
  - Leading zeros in each quartet can be omitted.
  - Contiguous sets of zeros can be replaced by two adjacent colons, but only once in an address. If there are multiple non-adjacent sets of contiguous sets of zeros, only one set of double colons can be used. The other set of zeros must be specified.

  For example:

```
3FFE:80C0:22C:18:250:88FF:FEAD:F92F
```

If you are uncertain about the address syntax, ask your network administrator to determine the exact syntax. For most UNIX and Linux hosts, the `ifconfig -a` command can be used to display the IPv6 address of a machine. In the Microsoft Windows environment, use the `ipconfig /all` command to display the IPv6 address.

- If you have IPv4 client hosts that will connect to IPv6-capable servers on AIX, Linux, or Tru64, the client network address will appear as IPv4-mapped addresses. The server host file validation logic takes this into account, and treats IPv4-mapped addresses as though they are native IPv4 addresses. Thus you can specify the regular IPv4 address as described in the first point above.
- You may have to experiment to find the right address.

Specifying server behavior

Table 20 describes the `daemon_options` file parameters that you can use to control the behavior of the SYMAPI server daemon (**storsrvd**).

For information on editing these parameters, refer to "Controlling daemon behavior" on page 95.

Table 20 storsrvd options for the daemon_options file (1 of 3)

Parameter	=<*optionalvalue* | *defaultvalue*>	Set with SETVAR	Reloadable
port Specifies the decimal port number.	= *nnnn* | 2707	No	No
security_level Specifies the session security level. For more information, refer to "Securing remote transmissions using SSL" on page 120.	= NONSECURE ANY SECURE | based on platform	Yes	Yes
log_show_category Specifies whether the specific storsrvd log category value should be displayed when a log message is written.	= ENABLE | DISABLE ENABLE: The category associated with the log event is shown as part of the text message. DISABLE: The category is not shown as part of the message.	Yes	Yes
log_show_msgid Specifies whether the specific storsrvd message identifier should be displayed when a log message is written.	= ENABLE | DISABLE ENABLE: The message ID of a storsrvd application log message is shown as part of the text message. DISABLE: The message ID is not shown as part of the message.	Yes	Yes

Table 20 storsrvd options for the daemon_options file (2 of 3)

Parameter	=<*optionalvalue* | *defaultvalue*>	Set with SETVAR	Reloadable
log_filter Specifies the types of events to log.	= SERVER SESSION APIREQ CONTROLS SERVER: Log high level events related to initialization, termination, and main thread. SESSION: Log logical session events (arrival, termination, security level, authorization rejections). APIREQ: Log SYMAPI activity (request start and stop (with completion status)). CONTROLS: Log control session handling information (command parsing, execution). **Note:** Leaving this parameter commented out will result in the SYMAPI server application-level messages not being logged.	Yes	Yes
security_alt_cert_file Specifies an alternate certificate file to the certificate file provided at installation. The specified file should have a matching security_alt_key_file option set for the match key file. A full path name should not be specified. Only a simple file name.	= *Any valid simple file name* | symapisrv_cert.pem	No	No
security_alt_key_file Specifies an alternate key file to the key file provided at installation. The file specified should have a matching security_alt_cert_file option set for the matching certificate file. A full path name should not be specified. Only a simple file name.	= *Any valid simple file name* | symapisrv_key.pem	No	No

Table 20 storsrvd options for the daemon_options file (3 of 3)

Parameter	=<*optionalvalue* | *defaultvalue*>	Set with SETVAR	Reloadable
security_clt_secure_lvl Controls the verification of the client certificate by the server. This parameter is not supported in z/OS. This value is ignored if secure communications are not established.	= NOVERIFY MUSTVERIFY | VERIFY NOVERIFY: Indicates that the server will not verify the client certificate. MUSTVERIFY: Indicates that the server will only accept communications from a version of the client that can send a certificate to be verified. VERIFY: Indicates that the server will verify a client certificate if the version of the client can send a certificate.	No	Yes

Controlling the server

This section explains the commands used to control the SYMAPI server.

Starting the server

If you have not already configured your host to start the server automatically, then you must start the SYMAPI service using the following command executed from the server side:

```
stordaemon start storsrvd
```

Note: For OpenVMS, you can automate this process by placing a call to `emc_start_storsrvd.com` in `SYSSTARTUP_VMS.COM`. For more information, refer to "Installing Solutions Enabler on OpenVMS" on page 66.

Stopping the server

To stop the remote SYMAPI service from the client or server side, use the following command:

```
stordaemon shutdown storsrvd
```

Showing server details

The `stordaemon show storsrvd` command displays the following information regarding the SYMAPI server:

- SYMAPI version
- Total number of sessions since startup
- Current active sessions
- log_show_msgid setting
- log_show_category setting
- Enhanced authentication setting

In the z/OS environment:

- cond_hdlr (condition handler)
- Version of the language environment library

The `stordaemon action storsrvd -cmd show server` command displays the same information as the `stordaemon show storsrvd` command with the addition of operating system information.

The following example displays the output of a `stordaemon show storsrvd` command:

```
stordaemon show  storsrvd

 Daemon State                    : Running
 Daemon Start Time               : Fri Feb 7 16:38:19 2009
 Version                         : 7.1.1
 Auto-Restart by Watchdog        : Disabled

 Total Number of Connections     : 1
 Number of Active Connections    : 0
 Total Number of Requests        : 0

 ANR0123I Show Server Details    :

 API Version                     : 7.1.1
 SYMAPI Session Total/Active     : 0/0
 SYMAPI Session Port             : 4000
 Security Level                  : NONSECURE
 Show ANR Category               : Disabled
 Show ANR Message Id             : Enabled
 Enhanced Authentication         : Disabled
```

In the above example:

- The first seven lines of the display are generated by common logic. All daemons display lines similar to these, with information that reflects the state of the daemon.
- The lines following the message ANR0123I are generated by storsrvd, and will not display for any other daemon.
- `Total Number of Connections` is the total connections handled during the life of the daemon process. For most daemons, this includes control sessions (those that execute commands to control the daemon) and application sessions (those that need application services provided by the daemon). This number does not include the dedicated session managed by the z/OS Console thread.
- `Number of Active Connections` is the number of currently executing control sessions and application sessions. See the note below about storsrvd connection counts.
- `Total number of Requests` is the number of control commands and application requests (SYMAPI function calls received at the server).
- SYMAPI Session Total/Active is the number of SYMAPI sessions only; it does not include the number of control sessions.

The following example displays the output of a `stordaemon action storsrvd -cmd show server` command:

```
./stordaemon action storsrvd -cmd show server
 storsrvd

ANR0123I Show Server Details:

SYMAPI Version                : 7.1.1
SYMAPI Session Total/Active   : 1/0
SYMAPI Session Port           : 2799
Security Level                : ANY
Show ANR Category             : Disabled
Show ANR Message Id           : Enabled
Enhanced Authentication       : Disabled

ANR0123I Show OS Information Details:

Process ID                    : 26677
Host OS Name/Version          : Linux/2.6.9-11.ELsmp
Processor Model/CPUs          : x86_64/4
```

Displaying networking information

The `show -netinfo` command displays information about the storsrvd networking interfaces. For example:

```
stordaemon action storsrvd -cmd show -netinfo

 ANR0123I Show Network Details:

 SYMAPI Session Port           : 5000
 IP Protocols                  : IPv6 with IPv4 mapping
 Host Name                     : Host1051
 IP address                    : 172.23.193.51
```

The above example includes information on the following:

- The port on which the server is listening is shown since it is an important networking object.
- The IP protocols accepted by the server are shown.
- The node name without the domain is shown.
- The IP address line will be repeated for as many IP addresses as are known by the resolver configuration (local host files or DNS) on the host. Multi-homed hosts may show multiple lines, and hosts known by both IPv4 and IPv6 addresses may show multiple lines.

Reloading the daemon_options file

The `reload` command re-reads the `daemon_options` file, and adjusts its behavior according to the specified options. For example:

```
stordaemon action storsrvd -cmd reload
```

Summarize active SYMAPI sessions

The `list -sessions` command shows a one line summary of each currently active SYMAPI session thread. The list includes the session number (ordered by connection arrival), the thread number processing the session, the client host userid, and the host name where the session originated. For example:

```
stordaemon action storsrvd -cmd list -sessions
```

Show session details

The `show -session` command displays details about active sessions. This command uses the following form:

```
stordaemon action storsrvd -cmd show -session
   [-num session_num] [-hostinfo]
```

Where:

`-num` *`session_number`* shows details on a particular session. If this option is not specified, the command will show details for all active sessions. If this option is used and the session number does not exist, an error message will display. You can view a list of session numbers using the `list -sessions` command.

`-hostinfo` shows details about the client host.

The following example displays the output of a `show -session` command:

```
./stordaemon action storsrvd -cmd show -session

 storsrvd
ANR0124I ==== Show Session Details for Session 1 on Thread 2:
User/Host:       Joe/Host127.aaa.bbb.com
SYMAPI Version:  7.1.1
Session Started: 2008/04/07 17:25:53   Seclevel: NONSECURE
Last Req Start:  2008/04/07 17:25:53   Code:     2187
Last Req End:    2008/04/07 17:25:53   Result:      0 (SYMAPI_C_SUCCESS)
```

The previous example includes information on the following:

- Remote client user name and host name (if it can be resolved, IP address if it cannot be resolved)
- API library version in use by the client, and architecture (32-bit, 64-bit)
- Session start time and security level
- Start time of the last API request, and the numeric code of the API
- End time of the last API request and the completion code, as well as the SYMAPI return code name (as defined in efbcore.h)
- Process ID of the client

Controlling and using the storsrvd log files

The server writes data to its log files provided by the common daemon infrastructure. These log files are named and handled in a manner consistent with other daemon log files. For example, under the default log management behavior, the files `storsrvd.log0` and `storsrvd.log1` are created in `/var/symapi/log`.

The behavior of the log files is subject to the standard daemon options: `logfile_type`, `logfile_size`, `logfile_perms` and `logfile_retention`. Thus, you can configure the logs as dated files with retention controls instead of the common wrapping duet of log0 and log1. The same rules apply to storsrvd as to all other daemons.

You can control the volume of data written to the log files with the `daemon_options` file parameters `log_filter` and `log_level`. For a description of these options, refer to "Specifying server behavior" on page 134.

Numbered messages issued by storsrvd

The SYMAPI server application-level messages are distinguished from messages issued by the Solutions Enabler common daemon support by the use of a messages identifier. The complete set of storsrvd messages is documented in Appendix A.

The following `daemon_options` file keywords affect the appearance of the storsrvd messages:

- `log_show_category` displays or suppresses the category (also known as the filter) that applies to a message.
- `log_show_msgid` displays or suppresses the display of the message identifier in the message.

For a description of these options, refer to "Specifying server behavior" on page 134.

5

Post-Install for z/OS

Once you have installed Solutions Enabler, you need to perform certain follow-up procedures to enable your software's features and to establish your command environment. This chapter provides the follow-up procedures for a Solutions Enabler installation in a z/OS mainframe environment:

SYMAPI server security preparation

This section explains how to control access to the SYMAPI server.

Started task user identity

The SYMAPI server is installed to be run as a batch job, but you can also customize it to run as a started task.

If you choose to run the server as a started task, you must associate a user identity 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 details on preparing this table.

Installing the SSL certificates

Solutions Enabler optionally allows the use of SSL encrypted communications between the SYMAPI server and the clients connecting to it. You can enable or disable SSL support from either the client or server side.

Note: If you plan on using the optional SSL encrypted communications and you plan on running the server in SECURE or ANY modes, you must create and install the SSL certificates before starting the server.

Note: For information on configuring the security level on the server side, refer to "Configuring the server" on page 121.

If SSL is requested on startup (through settings in the `options` file or `daemon_options` file), the SYMAPI server will attempt to read the SSL certificates from the USS file system. Therefore, you must create and install the certificates before starting the server.

To install the certificates into the USS directory:

1. Run the batch file `zoscert.bat` with the `create` parameter in the temporary directory you created on the Windows host in "Step 1: Copy the files from installation disc" on page 56.

   For example:

   ```
   zoscert.bat create
   ```

   Note: When running the `zoscert.bat` batch file on a Windows host, you may receive a Windows error message regarding missing dlls, for example `msvcr80.dll`. This is most likely due to runtime files for Visual Studio 2005 (also known as Visual Studio 8) not being installed. To fix this problem, install the Visual Studio redistribution kit that is provided on the DVD in the Solutions Enabler Version 7.1.1 kit. The OS-specific files `vcredist_x86.exe` and `vcredist_x64.exe` are located in the folder `Other\zOS\VC8`.

2. When prompted, provide the following information:

   - The fully qualified name of the z/OS host (hostname including the domain name). To get this name, ping the z/OS host from the Windows host.
   - The FTP port number (default 21) of the z/OS host.
   - The z/OS userid used to sign in. The userid must have all the requisite permissions to write to the SYMAPI base directory.
   - The SYMAPI base directory (specified when running the SEMJCL exec on z/OS).
   - The password for the z/OS userid.

Once completed, the certificates will be generated and uploaded to the correct location inside the USS file system on the z/OS host. For example, if you specified the SYMAPI base directory as `/var/symapi`, the certificates will be uploaded to the directory `/var/symapi/config/cert`.

The certificate configuration is now complete and the server is capable of running in a secure mode.

Note: For more information on certificate management, refer to "Certificate files" on page 123. The `zoscert.bat` file accepts the same parameters as the batch file `manage_server_cert.bat`, namely `create`, `update`, and `secure`. The file `manage_server_cert.bat` must not be run directly. It will be invoked by `zoscert.bat` with the correct options.

Configuring Solutions Enabler

This section explains how to configure Solutions Enabler in a z/OS environment.

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
```

Note: 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. Solutions Enabler will refer to (or create) it in the *symapi_installation_directory*/db directory in USS.

Note: Beginning with Solutions Enabler V7.0, the SYMAPI database (`symapi_db.bin`) uses a new format. Solution Enabler V7.0 and later can use a database written by an earlier version of Solutions Enabler as long as it is in the *`symapi_installation_directory/db`* directory in USS. However, versions of Solutions Enabler prior toV7.0.cannot read/use a database written to by V7.0 or later.

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

`/path/to/``db.file`

where:

`/path/to` is a valid, existing, writable USS path and `db.file` is the name of the SYMAPI database.

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:

- As the database default name (by specifying NULL in the database argument)
- With an explicit database name

Note: If an explicit location is specified for the database, SYMAPI will use it; otherwise, specifying just a filename will result in the file being stored in the *`symapi_installation_directory`*`/db` directory.

Server default database locking

The default database is described in the fully qualified USS path of the database. When a session requests the default database, SYMAPI attempts to use the fully qualified USS path, 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 fully qualified path.

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 SYMAPI dataset. 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 "Modifying default behavior with the options file" on page 153. 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. If an existing database with compression enabled has already been saved to disk and you want to disable compression, you will need to

re-discover the database again and save it to disk. The easiest way to do this is to use the `#10ECCIN` job with the option to disable compression set.

There is no option to suppress the storing of multiple images of the database. SYMAPI can read databases created by earlier versions.

Note: 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.

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 SYMAPI server tasks, you can define specific devices to be used as gatekeepers, and you can also specify devices to be avoided as gatekeepers. For detailed information on how to specify gatekeeper device configuration, refer to "Configuring Solutions Enabler" on page 146.

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 sake, the examples in the following tables show the use of instream data for input files. Generally, EMC does not recommend instream datasets for SYMOPT, SYMAVD, SYM$GAVD, SYM$GSEL or SYM$INQ. If the SYMAPI server has a high connection arrival 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 #STORSRV member of the PARMLIB.

SYMAPI files

Table 21 lists and maps the SYMAPI files to corresponding DD statements. It also shows which files can be defined in PARMLIB members or in datasets, and which files can optionally be defined in USS files.

Note: For USS supported files, SYMAPI will only use a USS location if the corresponding DD name is not specified in the SYMAPI server JCL (comment it out or delete it).

Table 21 SYMAPI files

DD name	File type	Description
SYM$LIC	Dataset, USS	An input file for the Solutions Enabler license information. Dataset: *ds-prefix*.License USS: *symapi_installation_directory*/config/symapi_licenses.dat
SYM$OPT	PARMLIB, USS	The SYMAPI options file. For more information, refer to "Changing the default behavior of SYMCLI" on page 88. PARMLIB: *ds-prefix*.PARMLIB (symopt00) USS: *symapi_installation_directory*/config/options
SYM$ENV	PARMLIB, Dataset	Contains the C runtime environment variables. This file must be either a sequential dataset or a member of a partitioned dataset. This file must only be used with the direction of the EMC Customer Support Center. PARMLIB: *ds-prefix*.PARMLIB (symenv00)
SYM$NETH	PARMLIB, Dataset, USS	Defines a list of trusted hosts and users who are allowed to connect to the server. For more information, refer to "Authorizing SYMAPI sessions" on page 131. USS: *symapi_installation_directory*/config/nethost
SYSOUT	Spool	Contains IBM Language Environment runtime messages.
SYSPRINT	Spool	Contains summary log output and output produced by the use of debugging controls.

Avoidance and selection files

Table 22 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 effect of commands to include or exclude the specified devices, gatekeepers, or Symmetrix arrays. 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 22 Solutions Enabler avoidance and selection files (page 1 of 2)

DD name	File type	Description

SYM$AVD[a]	PARMLIB
	USS
		This file affects the operation of the discovery process so that it skips devices that belong to the Symmetrix arrays identified in this file. This may be useful if there are multiple Symmetrix arrays connected to the host that you wish the discovery to avoid. The Symmetrix avoidance file is formatted with 12-character Symmetrix IDs, with one ID per line.
		For example, to avoid discover of the Symmetrix with a serial number of 0000183600186, code the following statements in your JCL:
		`//SYM$AVD DD *`
		`0000183600186`
		`/*`
		
		PARMLIB: *ds-prefix*.PARMLIB (*member*)
		USS: *symapi_installation_directory*/config/symavoid

SYM$INQ[a]	PARMLIB
	USS
		This file affects the inquiry and discovery processes so that they find only the volume serial name (volser) specified in this file. This maybe useful if you want to limit the command(s) to affect only certain Symmetrix devices from your host. The inquiry file is formatted with volume serial names (volser), with one volser per line.
		For example, to include information on volume ABC123 (only) and the Symmetrix to which it is attached, use a SYM$INQ file that looks like this:
		`//SYM$INQ DD *`
		`ABC123`
		`/*`
		
		PARMLIB: *ds-prefix*.PARMLIB (*member*)
		USS: *symapi_installation_directory*/config/inqfile

Table 22 Solutions Enabler avoidance and selection files (page 2 of 2)

DD name	File type	Description

SYM$GAVD[a]	PARMLIB
	USS
		This file affects calls to commands that use a gatekeeper to communicate to a Symmetrix array. A gatekeeper whose volser matches any of the entries specified in the gkavoid file will not be chosen as a gatekeeper to communicate with the Symmetrix array. This could be useful to designate certain Symmetrix devices that should not be used as gatekeepers. The gatekeeper avoidance file is formatted with volume serial names (volser), with one per line.
		For example, to instruct Solutions Enabler for z/OS to avoid using volume DEF456 as a gatekeeper device, use the following SYM$GAVD statement:
		`//SYM$GAVD DD *`
		`DEF456`
		`/*`
		
		PARMLIB: *ds-prefix*.PARMLIB (*member*)
		USS: *symapi_installation_directory*/config/gkavoid

SYM$GSEL[a]	PARMLIB
	USS
		In SYM$GSEL, specify serials for the volumes you prefer to be gatekeepers. Specify one volume serial per line, with no other text on the line.
		Note: If a SYM$GSEL list is not defined for a particular Symmetrix array or if the specified volumes to do not exist at the time the file is read (every time a CLI command is run), then normal gatekeeper selection rules will apply for that Symmetrix array.
		
		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 give preference to volumes GHI123, JKL123 and MNO123, use the following SYM$GSEL DD statement:
		`//SYM$GSEL DD *`
		`GHI123`
		`JKL123`
		`MNO123`
		`/*`
		
		PARMLIB: *ds-prefix*.PARMLIB (*member*)
		USS: *symapi_installation_directory*/config/gselect
		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.

a. This file can also be referred to by way of a PATH statement. However, if you want to store it in USS, it is recommended that you remove or comment out this DD statement and let it default to the correct USS location.

Configuring for local time zone

The SYMAPI 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 **storsrvd** log files and SYMAPI log file are the most important places. Use the TZ environment variable to establish your local offset from Coordinated Universal Time (UTC). The valid settings for TZ are standardized by the POSIX standard and are described in many publications, including 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
/*
```

In the **Time Zone** field of the SEMJCL panel (Figure 2 on page 59), you can enter the appropriate setting for your time zone. For more information, refer to "Installing Solutions Enabler on z/OS" on page 56.

Modifying default behavior with the options file

The `options` file contains statements that can be modified to change the default behavior of SYMCLI operations, SYMAPI calls, and their control actions. It can be used to impart certain global restrictions as well as customize and streamline command line coding to your specific environment. Each sample statement is commented, and can be enabled by removing the `#` in the first column.

Note: For descriptions of the `options` file parameters, refer to *Solutions Enabler SYMCLI Command Reference HTML Help*.

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.

Table 23 lists some of the control operations that are disabled in the z/OS server.

Table 23 Disabled z/OS control operations (page 1 of 2)

Function	Action
SymAccessSessionStart	Starts an access control session.
SymAuthzRuleDelete	Maintains internal authorization rules.
SymAuthzRuleUpdate	Updates internal authorization rules.
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.

Table 23 Disabled z/OS control operations (page 2 of 2)

Function	Action
SymDirControl	Invokes a director control operation on one or all SRDF RA directors.
SymDirPortControl	Invokes a port control operation on a front-end director.
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`.

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.

Controlling the server

You can inspect and control the behavior of the server using the `stordaemon` command or the system console. For information on the commands accepted by the SYMAPI server, refer to "Controlling the server" on page 137.

This section describes specific methods of entering the commands.

Starting the server

To start the SYMAPI server, you can submit the job stream contained in the the `#STORSRV` member of the Solutions Enabler RIMLIB for batch execution.

Note: `#STORSRV` was customized when you used SEMJCL to specify configuration information appropriate for your site during the installation procedure.

You can execute the SYMAPI server program `storsrvd` as a started task. You can prepare a catalogued procedure for use as a started task. No such procedure is provided with the installation kit.

You cannot use `stordaemon start` in the z/OS environment to start the server.

Stopping the server

To stop the SYMAPI server, you can use the `stordaemon shutdown` command, or the equivalent command from the z/OS system console.

You can also use the z/OS `STOP` command regardless of whether the server is running as a started task or as a batch job. Using the `STOP` command (for example, `"P STORSRVD"`) starts a normal shutdown, waiting for all SYMAPI sessions to terminate normally.

Using the console

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.

Usage notes

When issuing commands from the system console, you should be aware of the following:

- While `stordaemon` commands are sent to the daemons without upper case conversion, text entered on the system console (and all virtualized consoles) is normally folded to uppercase by the operating system. Enclosing the text in apostrophes (not quotes) alters the behavior, resulting in the command text being sent as is to the application.
- Commands issued using the `stordaemon action` verb must be entered with apostrophes to preserve the case. Complete enclosure in apostrophes is not necessary; a leading apostrophe is sufficient to preserve case. A closing apostrophe will be accepted and ignored.
- Dashed options are not required. The SYMAPI server allows the specification or omission of the dash on the command options. The console command parsing logic will accept a dash if specified, but ignore it for the purposes of option identification.
- Commands entered from the console are directed to a specific running daemon. Thus the multi-daemon commands and operands are not supported when entered from the console. The `list` command and the `all` option of the `shutdown`, `setvar`, `getvar` commands are not supported when entered at the console.
- The daemon name must be omitted in the command text, since the `MODIFY` system command specifies the jobname which directs the command to the correct daemon. Thus, the command text will begin with the verb.

- The `action` verb can be omitted only if the `-cmd` verb and/or operands can unambiguously distinguish the command from all general commands. For example, in the case of storsrvd, the general `show` command will show basic status information. The action `-cmd show` command will show other detailed information specific to storsrvd.
- The `-cmd` option can be omitted also. If either `action` or `-cmd` are specified, the command text will be passed to the running daemon for execution. If the daemon application log parses the command text successfully, it may execute the command and produce the appropriate output. If the application logic does not recognize the command, an error message will be generated and written to the console.
- Commands that change the environment outside of the daemon will not be accepted from the console. These are `start`, `install`, and `uninstall`.
- The `-wait` option of the `stordaemon shutdown` command is not supported and will be ignored if entered from the console.
- The `showlog` command is not supported from the console.

Examples

Table 24 compares the syntax of the `stordaemon` commands issued from a USS shell to the syntax of the same commands entered on the z/OS console. Assume that the jobname of the server is `STORSRVD`, and the daemon name is also `storsrvd`. Note that the z/OS system command `MODIFY` alias is `'F'`.

Table 24 stordaemon command syntax for the z/OS system console

Command	stordaemon syntax	Console syntax
Show daemon status long. Show daemon status (state).	`stordaemon show storsrvd` `stordaemon show storsrvd -brief`	`F STORSRVD,SHOW` `F STORSRVD,SHOW [-]BRIef`

Stop the daemon.
 	
Stop the daemon immediately.
	`stordaemon shutdown storsrvd`
	
	`stordaemon shutdown storsrvd -immediate`
		`F STORSRVD,SHUTDOWN`
		
		`F STORSRVD,SHUTDOWN [-]IMMediate`

Show the current value of an operational variable (port in this example).	`stordaemon getvar storsrvd -name port`	`F STORSRVD,'getvar [-]name port'`
Change the current value of an daemon option (takes effect immediately.	`stordaemon setvar storsrvd -name log_filter=SESSION,APIREQ`	`F STORSRVD, 'setvar [-[name log_filter=SESSION, APIREQ'` **Note:** The `-name` option can be abbreviated to 3 chars and the dash can be omitted.
Store a new value of a daemon option for reload or subsequent execution. In this example, change the port to 2708.	`stordaemon setoption storsrvd -name port=2708`	`setoption` is not supported from the console in this release.
Issue a `storsrvd` extending action. In this example, show details for SYMAPI session number 4.	`stordaemon action storsrvd -cmd show -session -num 4`	`F STORSRVD,'action show -ses -num 4` **Note:** In this example the `-cmd` keyword is omitted, and a closing quote is also omitted.
Show network information.	`stordaemon action storsrvd -cmd show -netinfo`	`F STORSRVD, 'action show -netinfo'`

In general, command-generated output shown on the z/OS console will suppress blank lines for the sake of brevity and to reduce messages rolling off the console screen.

Using stordaemon TSO commands

In the TSO command shell, the `stordaemon` command operates as it does on all platforms. If the Solutions Enabler load library is in the TSO STEPLIB or CMDLIB, you can issue the `stordaemon` command as shown in the following example:

```
IKJ56455I DMCLEL3 LOGON IN PROGRESS AT 13:33:01 ON NOVEMBER 1, 2007,
IKJ56951I NO BROADCAST MESSAGES,
REXX/SOCKETS z/OS V1R6 January 5, 2007,
Network Info - IP:10.243.142.82,
          Domain:LSS.EMC.COM,
        Hostname:MFX02,
READY

STORDEMN show storsrvd
<output will show here>

CALL 'EMC.SSEM711.LOADLIB(STORDEMN)' 'show storsrvd'
<output will show here>
```

Optionally, you can trap all output of the `stordaemon` command with the REXX language function outtrap(). In which case, all output will be saved in a REXX variable array, where it can be processed programmatically.

Using stordaemon in a USS shell

The following example illustrates how you can configure `stordaemon` to run from USS. For the sake of this example, assume that the user has already logged in to the z/OS USS shell either via rlogin or the TSO `OMVS` command:

```
$ cd /var/symapi
$ mkdir bin
$ cd bin
$ ln -e STORDEMN stordaemon
$ export STEPLIB=EMC.ssem711.LOADLIB
$ stordaemon show storsrvd
$ stordaemon shutdown storsrvd
$
```

In the example, the user makes an external link from a USS file to the Solutions Enabler load library module. By setting the STEPLIB environment variable, the shell follows the link from the USS file to the load library, finding the member stored there. The load library member executes the `stordaemon` application. Any z/OS supported `stordaemon` functions can be used in this environment.

Running the base daemon on z/OS

The base daemon (`storapid`) is optional for the z/OS SYMAPI server. The base daemon provides numerous benefits for the z/OS environment, including improved performance (via caching of syscall results) and enhanced Symmetrix lock management.

Most of the information in this section is similar to the daemon information described in Chapter 3; however, this section describes it from the z/OS point of view.

Installing or uninstalling the base daemon

The base daemon is automatically installed during Solutions Enabler installation, as its load modules reside in the same loadlib.

In z/OS, there is no support for uninstalling the base daemon.

Starting the base daemon

Once the server is running, start the base daemon by submitting the job `#STORAPI` in the RIMLIB. This job will have been correctly configured when the SEMJCL process was run. If necessary, you can modify this job and convert it to run as a started task.You cannot use the `stordaemon` command to start the base daemon.

Note: As there is no watchdog daemon in z/OS, the base daemon will not automatically start/restart.

Stopping the base daemon

Table 25 lists the commands for stopping the base daemon.

Table 25 Commands for stopping the base daemon

From	Use the command
Console	`F STORAPID,SHUTDOWN`
TSO	`stordemn shutdown storapid`
USS shell	`stordaemon shutdown storapid`

For more information on using these methods, refer to "Controlling the server" on page 158.

Using and configuring the base daemon

The base daemon behavior is determined by parameters set in the configuration file `daemon_options`. This file is found in the *`symapi_installation_directory`*`/config` folder. It is a standard text file that you can edit by way of oedit or any other text editor. For detailed information on editing the parameters in this file, refer to "Controlling daemon behavior" on page 95.

By default, if the base daemon is running, then the z/OS SYMAPI server will connect to it and use its features. If it is not running, then the server will not attempt to start or use it.

Base daemon logging

Solutions Enabler daemons all use a common infrastructure mechanism for logging messages and events. For information on the options available to manage the way the base daemon uses its log files, refer to "Controlling daemon logging" on page 96.

Avoidance and selection files and the base daemon

The base daemon will not recognize or use JCL specified selection and avoidance files. It will only use the appropriate files in the *`symapi_installation_directory`*`/config` folder in USS.

You should not use both MVS datasets (for the server) and USS files (base daemon) for these selection and avoidance files. Doing so will likely result in inconsistent definitions and confusion. If you use the base daemon, you should place the avoidance and selection files for both the SYMAPI server and the base daemon in the relevant USS location. For the SYMAPI server, the relevant DDnames in the job should be removed or commented out, so that the server will refer to the correct files in USS.

For more information on the avoidance and selection files, refer to "Avoidance and selection files" on page 151.

Running the event daemon on z/OS

The use of the event daemon (`storevtd`) is optional for the z/OS SYMAPI server. For information regarding the event daemon, refer to "Setting up the event daemon for monitoring" on page 101.

In the z/OS context, the event daemon is primarily used to enable monitoring capabilities on behalf of other clients. For this release, the only client expected to use the event daemon is the EMC Symmetrix Management Console (SMC).

Installing or uninstalling the event daemon

The event daemon is automatically installed during Solutions Enabler installation, as its load modules reside in the same loadlib.

In z/OS, there is no support for uninstalling the event daemon.

Starting the event daemon

Once the server is running, start the event daemon by submitting the job `#STOREVT` in the RIMLIB. This job will have been correctly configured when you ran the SEMJCL process. If necessary, you can modify this job and convert it to run as a started task.You cannot use the `stordaemon` command to start the event daemon.

Note: As there is no watchdog daemon in z/OS, the event daemon will not automatically start/restart.

Stopping the event daemon

Table 26 lists the commands for stopping the event daemon.

Table 26 Commands for stopping the event daemon

From	Use the command
Console	`F STOREVTD,SHUTDOWN`
TSO	`stordemn shutdown storevntd`
USS shell	`stordaemon shutdown storevntd`

For more information on using these methods, refer to "Controlling the server" on page 158.

Using and configuring the event daemon

The event daemon behavior is determined by parameters set in the configuration file `daemon_options`. This file is found in the *`symapi_installation_directory`*`/config folder`. It is a standard text file that you can edit by way of oedit or any other text editor. For detailed information on editing the parameters in this file, refer to "Controlling daemon behavior" on page 95.

Event daemon logging

Solutions Enabler daemons all use a common infrastructure mechanism for logging messages and events. For information on the options available to manage the way the event daemon uses its log files, refer to "Controlling daemon logging" on page 96.

6

Uninstalling Solutions Enabler

This chapter explains how to uninstall Solutions Enabler:

Overview

To uninstall Solutions Enabler, you must first shutdown the application processes that use the Solutions Enabler libraries and binaries, and then uninstall the software.

Stopping the application processes

To stop the application processes:

1. Issue the following command to stop the Solutions Enabler daemons:

   ```
   stordaemon shutdown all
   ```

   Note: For more information on this command, refer to "Stopping daemons" on page 93.

2. Issue the following command to verify that the daemon(s) have stopped:

   ```
   stordaemon list -all
   ```

   Note: For more information on this command, refer to "Viewing daemons" on page 93.

3. For UNIX, you can also issue the following command to identify any other applications using the Solutions Enabler libraries:

   ```
   fuser /usr/lib/libsym* /usr/lib/libstor*
   ```

   For AIX, issue:

   ```
   fuser -x -f /usr/symcli/shlib/library_name
   ```

Uninstalling the software

To uninstall the Solutions Enabler software, refer to the following:

- For UNIX, refer to "Uninstalling Solutions Enabler from UNIX" on page 169.
- For Windows, refer to "Uninstalling Solutions Enabler from Windows" on page 173.
- For OpenVMS, refer to "Uninstalling Solutions Enabler from OpenVMS" on page 177.

Uninstalling Solutions Enabler from UNIX

You can uninstall Solutions Enabler from a UNIX host using either the Solutions Enabler uninstall script or your native install tools (e.g., `rpm -erase` on Linux).

CAUTION

Take care when removing Solutions Enabler, as it may be a prerequisite for other installed products.

Using the script

To use the script to uninstall Solutions Enabler from a SunOS or Linux host, change directory to `/usr/symcli/install` and run the following script:

```
./se7110_install.sh -uninstall
```

To use the script to uninstall Solutions Enabler from all other supported UNIX hosts, run the same command from the Solutions Enabler 7.1.1 installation media image.

For help running the uninstall script, run the following script:

```
./se7110_install.sh -help
```

The uninstall script creates log files in the format `SE_NI_`*`KitVersrion_TimeStamp`*`.log`, where *`TimeStamp`* is in the form *`YYMMDD_HHmmSS`*.

Persistent data

The persistent data will remain under `/usr/emc/API/symapi` or in the data directory selected during installation.

The persistent data will remain accessible from the softlink `/var/symapi`.

Decremental method

To uninstall a single Solutions Enabler component you can use the `-decrement` option:

```
./se7110_install.sh -decrement [-jni] [-db] [-64bit]
   [-star] [-symrec] [-cfgchk]
```

For example, to uninstall the Solutions Enabler Star component, enter:

```
./se7110_install.sh -decrement -star
```

Using native tools

When using your native tools to uninstall Solutions Enabler, you *must* uninstall the Solutions Enabler packages in the following order:

1. SMI
2. CFGCHK
3. 64BIT
4. UDB
5. SYBASE
6. ORACLE
7. JNI
8. SYMCLI
9. SYMRECOVER
10. STAR_PERL
11. SRMFULL
12. STORFUL
13. SRMBASE
14. STORBASE
15. CORE
16. DATASTORBASE
17. DATACORE

In addition, you must also verify that all application processes using the Solutions Enabler libraries and binaries are stopped. For instructions, refer to "Stopping the application processes" on page 168.

Uninstalling from Linux

Use the following commands when uninstalling Solutions Enabler from a Linux host:

`rpm -qa|grep symcli`

Lists all of the installed RPMs.

`rpm -ql` *`RPMName`*

Lists all of the files in the specified RPM. For example, to list all of the files in the core component, enter:

```
rpm -ql symcli-core
```

`rpm -e` *`RPMName`*

Uninstalls the specified RPM. For example, to uninstall the core component, enter:

```
rpm -e symcli-core
```

Uninstalling from AIX

Use the following commands when uninstalling Solutions Enabler from an AIX host:

```
lslpp -L | grep SYMCLI
```

Lists all installed Solutions Enabler filesets.

```
installp –u FilesetName
```

Uninstalls a fileset. For example, to uninstall the core component, enter:

```
installp -u SYMCLI.CORE
```

Uninstalling from HPUX

Use the following commands when uninstalling Solutions Enabler from an HPUX host:

```
swlist –l  fileset | grep SYMCLI
```

Lists all of the installed Solutions Enabler filesets.

```
swremove FilesetName
```

Uninstalls a fileset. For example, to uninstall the Solutions Enabler core component, enter:

```
swremove SYMCLI.CORE
```

Uninstalling from HP Tru64

Use the following commands when uninstalling Solutions Enabler from an HP Tru64 host:

```
setld -i | grep -v "not installed" | grep SYMCLI
```

Lists all of the installed Solutions Enabler packages.

```
setld -d SubsetName
```

Uninstalls a subset. For example, to uninstall the Solutions Enabler core component, enter:

```
setld -d SYMCLICORE711
```

Uninstalling from Solaris

Use the following commands when uninstalling Solutions Enabler from a Solaris host:

```
pkginfo  | grep SYM
```

Lists all of the installed Solutions Enabler packages.

```
pkgrm PackageName
```

Uninstalls a package. For example, to uninstall the Solutions Enabler core component, enter:

```
pkgrm SYMcore
```

Uninstalling Solutions Enabler from Windows

This section describes the various methods available for uninstalling Solutions Enabler from a Windows host.

CAUTION

Take care when removing Solutions Enabler, as it may be a prerequisite for other installed products.

Using the InstallShield wizard

To uninstall Solutions Enabler using the InstallShield wizard:

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

   Note: For instructions, refer to "Stopping the application processes" on page 168.

2. Change directory to the location of the Solutions Enabler kit by entering the following:

   `cd /Install disk mount point/Windows`

3. Start the uninstall by running the following:

`cd /Install disk mount point/Windows/se7110-Windows-OS.exe`

   Where *OS* is the operating system. Possible values are x86, x64, and IA64.

4. In the **InstallShield Wizard for Solutions Enabler Welcome** dialog box, click **Next**.
5. In the **Program Maintenance** dialog box, select **Remove** and click **Next**.
6. In the **Remove the Program** dialog box, click **Remove**.
7. In the **InstallShield Wizard Completed** dialog box, click **Finish** to complete the removal process.

Using the command line

To uninstall Solutions Enabler from the command line using the msi installer options:

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

   Note: For instructions, refer to "Stopping the application processes" on page 168.

2. Run the following command:

   ```
   start /wait FullPathToInstallImage\
      se7110-Windows-Processor_type.exe /S /X /V/qn
   ```

   Where:

   Processor_type can be x86, x64, or IA64.

   FullPathToInstallImage is the path to the executable.

   S is the command to run silently.

   x is the command to uninstall.

   /V is the command gateway for msiexec.exe.

   /qn is the silent option.

Removing the msi image

You can use either of the following methods to uninstall the msi image:

- Enter the following command, specifying the GUID of the product to uninstall:

  ```
  start /wait msiexec.exe /x {GUID} /qn
  ```

  Possible values for *GUID* are:

{BF4B1AB4-E546-4C1F-BEDD-8F5AEE08E5C8}	Solutions Enabler
{D3C61295-9F1A-4DEA-9614-2BD9935F0541}	EMC ControlCenter
{B0BDFA88-C2F5-42A8-99AF-EE775D643121}	SP
{ECFAEC3E-5252-4ECA-9209-078266219EB3}	STORBLK
{FC73592D-80AE-4EF5-B3D5-085661E00C33}	SMI

{DC064490-B713-406E-BC97-D96593F60B93}	VSS
{F7DF5A59-EE48-4449-9605-F39ABF332B01}	VDS
{ABD6C01E-CF8B-4C0D-9E32-3372169C3243}	SDK
{CCF64999-3C86-4328-A644-54A43C6C95B6}	TCLIENT

- Use the Windows Installer Clean Up utility, msicuu2.exe:
  a. Download the msicuu2.exe from Microsoft and install it on the host.
  b. From the Windows **Start** menu, select **All Programs**.
  c. Select the application to remove and click **Remove**.
  d. Stop the following services in the order listed below. You can do this from either the cmd prompt or the **Services** dialog.

     Storsrvd
     Storgnsd
     Storrdfd
     Storevntd
     Storsrmd
     Storstpd
     Stororad
     Storsqld
     Storudbd
     Storemud
     Storapid

  e. Remove the list of files from System32. The list of files is the same as those in *InstallDir*\Symcli\shlib.
  f. Remove the Symcli directory and all its subdirectories.
  g. Remove the subdirectories from Symapi, except for the Config and db directories.
  h. Remove the following registry entries:

     HKEY_LOCAL_MACHINE\SOFTWARE\EMC\EMC Solutions Enabler

     HKEY_LOCAL_MACHINE\SOFTWARE\EMC\SYMCLI

     HKEY_LOCAL_MACHINE\SOFTWARE\EMC\TimeFinder Integration Modules

     HKEY_LOCAL_MACHINE\SOFTWARE\EMC\WideSky

i. From under the following registry key, remove the entries that only point to the SYMAPI or SYMCLI:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDlls

Using the Windows Add/Remove Programs dialog

To uninstall Solutions Enabler from the Windows **Add or Remove Programs** dialog:

1. From the Windows **Start** menu, select **Settings**, **Control Panel**, **Add or Remove Programs**.
2. In the **Add or Remove Programs** dialog, select **EMC Solutions Enabler** and click **Remove**.

Uninstalling Solutions Enabler from OpenVMS

To uninstall Solutions Enabler from an OpenVMS host:

CAUTION

Take care when removing Solutions Enabler, as it may be a prerequisite for other installed products.

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

   Note: For instructions, refer to "Stopping the application processes" on page 168.

2. Delete all the files in the `sys$specific:[emc]` and `sys$specific:[000000]emc.dir` directories. If the environment is a cluster, delete these files from every node in the cluster where Solutions Enabler was running.
3. Delete all the files from the installation directory.

Rolling back an upgrade

To roll back your upgrade, you must have created copies of the host database and config directories, as explained in "Before you begin" on page 21:

1. Verify that all application processes that use the Solutions Enabler libraries and binaries are stopped.

   Note: For instructions, refer to "Stopping the application processes" on page 168.

2. Export all device groups from the current SYMAPI database:
   a. Issue a `symdg list` command to list all the device groups.
   b. Issue a `symdg export` command to export the device groups.
   c. Issue a `symcg list` command to list all the composite groups.
   d. Issue a `symcg export` command to export the composite groups.

   Note: This export is necessary because older versions of Solutions Enabler may not be able to read a database once a newer version of Solutions Enabler has converted it.

   Note: For more information on these commands, refer to the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

3. Uninstall your software according to the platform-specific procedures earlier in this chapter.
4. Install the desired version of Solutions Enabler.
5. Once the installation is complete, issue a `symcfg list` command to verify that the SYMAPI database can be used by the older version:
   - If the database can be used, the rollback is done.
   - If the database cannot be used, issue a `symcfg discover` command to create a Symmetrix host database file, `symapi_db.bin`, and import all the exported device groups.

7

Deploying the Solutions Enabler Virtual Appliance

This chapter explains how to deploy the Solutions Enabler Virtual Appliance in a VMware infrastructure environment:

Introduction

The Solutions Enabler Virtual Appliance is a VMware ESX server virtual machine that provides all the components you need to manage your Symmetrix environment using the `storsrvd` daemon and Solutions Enabler network client access. These include:

- EMC Solutions Enabler V7.1.1 (solely intended as a SYMAPI server for Solutions Enabler client access)
- Linux OS (SUSE 10 SP2 JeOS)
- SMI- S Provider V4.1.1

In addition, the Solutions Enabler Virtual Appliance includes a browser-based configuration tool, called the Solutions Enabler Virtual Appliance Configuration Manager. This tool enables you to perform the following configuration tasks:

- Monitor the application status
- Start and Stop selected daemons
- Import and export persistent data
- Configure the `nethost` file (required for client access)
- Discover storage arrays
- Modify options and daemon options
- Add license keys
- Run a limited set of Solutions Enabler CLI commands

Note: For information on using the Configuration Manager, refer to its online help.

Note: Root login is not supported on the SUSE 10 Virtual Machine. Local login is restricted to the seconfig account with restricted access.

Before you begin

Before you begin to deploy the Solutions Enabler Virtual Appliance, be sure to complete the tasks listed in this section.

- ❑ Verify that you are installing the latest version of the appliance by checking Powerlink for updates.
- ❑ Verify that the client is running:
  - VMware Infrastructure Client V2.5 (or above) or vSphere Client
  - Either of the following browsers with cookies and javascript enabled:
    - Internet Explorer 6.0 through 8.0
    - Firefox 3.0
- ❑ Verify that the VMware ESX Server meets the following minimum requirements:
  - Version 3.5 or later
  - 4 GB of disk space
  - 512 MB of memory
  - 1 CPU

Deploying the Solutions Enabler Virtual Appliance

This section describes how to deploy the Solutions Enabler Virtual Appliance into a VMware infrastructure environment.

Step 1: Import the Virtual Appliance

To start deploying the Solutions Enabler Virtual Appliance:

1. Download the zip file containing the installation program from Powerlink and extract it to a temporary directory.
2. Start the vSphere Client and log in to either vCenter Infrastructure Server or the ESX Server on which you will be deploying the appliance.
3. Click **Ignore** in the security warning message.
4. If you are using vCenter Infrastructure Server, select the ESX Server on which you will be deploying the appliance from the navigation tree.
5. Depending on the ESX Sever version, do the following:
   - For ESX Server V3.5, do the following:
     a. On the **Getting Started** tab, click **Import a virtual machine** (vCenter Infrastructure Server) or click **Deploy OVF Template** (vSphere Client) to launch the Import Virtual Appliance Wizard.
     b. On the **Import Location** page, select **Import from file** and specify the path to the OVF file. This file is located at the top level of the temporary directory you created earlier.Click **Next**.
   - For ESX Server V4.0, do the following:
     a. From the **File** menu, select **Deploy OVF Template**.
     b. Browse to the OVF file. This file is located at the top level of the temporary directory you created earlier.
6. On the **Details** page, verify the details about the appliance and click **Next**.
7. On the **End User License Agreement** page, select **Accept all license agreements** and click **Next**.

8. On the **Name and Location** page, specify a name for the appliance and select the location for the virtual machine. Click **Next**.
9. On the **Ready to Complete** page, verify the information and click **Finish**.
10. In the Completed Successfully message, click **Close**.
11. Continue with "Step 2: Select gatekeepers" on page 183.

Step 2: Select gatekeepers

Present uniquely defined gatekeeper by way of raw device mappings (RDM). For instructions, refer to the appropriate VMware documentation.

Solution Enabler manages Symmetrix arrays through gatekeeper devices mapped to the virtual appliance as RDM pass-through devices. The management is done through EMC proprietary commands using SCSI 3B/3C write/read commands. For every call, a WRITE command is issued to send the request then a READ command to get the results.

Continue with "Step 3: Configure the Virtual Appliance.".

Step 3: Configure the Virtual Appliance

To configure the Virtual Appliance:

1. On the **Summary** page of the Virtual Infrastructure Client, click **Power On**.
2. Click the **Console** tab and watch as the appliance starts up.

   If this is a new installation, continue with step 3; otherwise, continue with step 7.
3. Read and accept the license by typing **`yes`** at the following prompt and pressing **Enter**.

   ```
   Do you agree with the terms of the end user license
      agreement? yes/no [no]:
   ```
4. At the following prompt, type a new password for the seconfig account and press **Enter**:

   ```
   Please change password for user seconfig
   New password:
   ```

5. At the following prompt, retype your new password and press **Enter**:

```
New Password:
```

6. At the following prompt, specify whether you want to set the time zone:

   ```
   Do you want to set the time zone? y/[n] :
   ```

   - A `[n]o` response continues the configuration. If you select this option, you can use the appliance console to specify the time zone at a later time.
   - A `[y]es` response produces the following series of prompts that will enable you to set the time zone:
     - `Please select a continent or ocean`
       Type the number that corresponds to the time zone location and press **Enter**.
     - `Please select a country`
       Type the number that corresponds to the country-specific time zone you want to set and press **Enter**.
     - `Please select one of the following time zone regions`
       Type the number that corresponds to regional time zone you want to set and press **Enter**.
     - `Is the above information OK?`
       Verify that the time zone information is correct. If it is, type **`Yes`** at the following prompt; otherwise, type **`No`** to step through it again.

7. At the following prompt, specify whether you want to enter the Solutions Enabler license keys.

   ```
   Do you want to register Solutions Enabler license keys
      yes/[no]? :
   ```

   - A **`no`** response continues the configuration. If you select this option, you must use the Configuration Manager to enter the license keys at a later time. For instructions, refer to the Configuration Manager's online help.
   - A **`yes`** response prompts you for a license key. In which case you should type the license key and press **Enter**. When prompted to enter another license key, type **`yes`** to enter another key, or **`no`** to continue with the installation.

   A Welcome dialog opens.

8. Continue with "Step 4: Configure the network settings for the appliance" on page 186.

Step 4: Configure the network settings for the appliance

To configure the network settings for the appliance:

1. Use the arrow keys to scroll to the **Configure Network** option and press **Enter**.
2. At the following prompt, specify whether you want to use a DHCP Server instead of a static IP address:

   ```
   Use a DHCP Server instead of a static IP Address? y/n
      [y]:
   ```

   - A [**y**]es response instructs the installation program to attempt to locate a DHCP server on your network. If successful, the configuration continues.
   - A [**n**]o response produces the following series of prompts that will enable you to configure your network:
     - `IP Address [ ]:`

       Type the address assigned to the appliance and press **Enter**.
     - `Netmask [ ]:`

       Type the mask of the network on which the appliance will be running and press **Enter**
     - `Gateway [ ]:`

       Type the gateway address to network on which the appliance will be running and press **Enter**
     - `Is a proxy server necessary to reach the internet? y/n [n]:`

       A [**y**]es response enables you to specify the IP address of the proxy server and the port.

       A [**n**]o response continues the configuration.
     - `Is this correct? y/n [y]:`

       Verify that the network information is correct. If it is, type **y** at the following prompt; otherwise, type **n** to step through it again.

   You have now finished installing the Solutions Enabler Virtual appliance.
3. Continue with "Step 5: Launch the Configuration Manager" on page 187.

Step 5: Launch the Configuration Manager

Once you have finished installing the appliance, you can launch the Configuration Manager.

To launch the Virtual Appliance Configuration Manager:

1. Type the following URL in a browser:

   `https://appliance_IP:5480`

2. On the log in panel, type **`seconfig`** for the **ID** and the password you created during installation for the **Password**, and then click **Sign In**.

   The Virtual Appliance Configuration Manager displays. For information on using the Configuration Manager, refer to its online help.

Updating the Solutions Enabler Virtual Appliance

Periodically, EMC will release security patches and hot-fixes for the Solutions Enabler Virtual Appliance on Powerlink.

To update an existing Virtual Appliance:

1. Login to the web console of the exiting appliance.
2. Click **Export Persistent Data** to download a zip file containing Solutions Enabler persistent data to your desktop.
3. Extract the zip file to your machine. Note the location of the file **encrypt_se_export_persistent_data_***time-stamp***.zip.gpg**. You will need this file later to complete this procedure.
4. Power off the old appliance.
5. Import and deploy the new appliance in your ESX server. For instructions, refer to "Deploying the Solutions Enabler Virtual Appliance" on page 182.
6. Login to the new appliance's web console.
7. Click **Import Persistent Data** and browse to the location of the gpg file you extracted earlier in this procedure.
8. Click **Import**.
9. When the message `Import successful` appears, close the dialog. The update is complete.

Deleting the Solutions Enabler Virtual Appliance

To delete the Solutions Enabler Virtual Appliance:

1. In the Configurator Manager interface, backup the persistent data.
2. In the VMware management interface, power down the appliance.
3. Right-click on the appliance and select **Delete from Disk**.
4. Click **Yes** in the confirmation message.

A

SYMAPI Server Daemon Messages

This appendix describes the log messages issued by the SYMAPI server daemon (`storsrvd`):

Message format

This section describes messages that are written to the SYMAPI server log (see "Controlling and using the storsrvd log files" on page 142) and to the system console in z/OS. All messages begin with a message identifier, followed by message text.

The message is in this format:

```
yyyy/mm/dd hh:mm:ss pid thread_name log_category msgid
   text
```

where:

yyyy/mm/dd	Is the date the message was issued.
hh:mm:ss.xxx	Is the time the message was issued in hours, minutes, seconds, and milliseconds.
pid	Is the process ID of the issuing process.
thread_name	Is the thread name of the issuing thread.
log_category	Is the category specified in the `storsrvd:log_filter` statement in the `daemon_options` file, which caused this message to be generated. The valid categories are: SERVER, SESSION, CONTRO, and APIREQ.
msgid	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 *W* indicates a Warning message *E* indicates an Error message S indicates a severe condition requiring a message
text	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* — In a z/OS environment, some messages will cause the SYMAPI server job step return code to be set to a non-zero value. The following table shows 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.

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.

- 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 Console destination applies only to z/OS.
- 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.

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 *security_level* 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. *security_level* indicates the types of sessions the server will accept. Possible values are:

- ONLY NONSECURE — Indicates that client must expect to negotiate non-SSL sessions with the server.
- ONLY SECURE — Indicates that the server will require clients to negotiate a secure session.
- Both SECURE and NONSECURE — Indicates that the server will accept sessions from clients that cannot negotiate secure and will negotiate secure sessions with clients who can.

Operator Action: None.

ANR0002I *shutdown_type* Shutdown requested

Destination: Log and console.

Description: This message indicates that a shutdown request was made. See message ANR0003I for the description of *shutdown_type.*

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.

In open systems environments, shutdown is requested by the `stordaemon` command.

In Microsoft Windows, you can use the Service Control Manager; in this case the shutdown process will always be IMMEDIATE.

In a z/OS environment, the system operator will request a NORMAL shutdown using the z/OS `STOP` command or the `SHUTDOWN` command.

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: In a z/OS environment, the server detects 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 and 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. Refer to 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.

ANR0011W SYMAPI Server not executing from an APF-authorized library, cannot continue

Set Step Return Code

Destination: Log and console.

Description: In a z/OS environment, the SYMAPI server program `storsrvd` must execute from a library authorized by the z/OS Authorized Program Facility, if the base daemon is not in use. The server checks to make sure that this condition is met. This message is issued as a warning , but an error condition may not be reflected until a SYMAPI session requests storage discovery services.

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 z/OS for exact syntax and editing instructions.

ANR0012I Accepted *seclevel* session *session_number* from *IP_address* on thread *thread_number*

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.* The session is running from a client program executing on the host at address *IP_address*. *seclevel* indicates the negotiated security level of the session. If *seclevel* is SECURE, transmission is protected using SSL; if *seclevel* is NONSECURE, SSL protection is not in use.

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* from *IP_address* to terminate itself

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 session(s) 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 can reissue the shutdown command with the IMMEDIATE operand to terminate the session immediately.

ANR0016I SYMAPI listener thread is running on thread *thread_number*

Destination: Log.

Description: This message is issued during startup simply to report the thread number (*thread_number*) of the SYMAPI listener thread (the server thread which listens for new connection requests).

Operator Action: None.

ANR0017I Ending session *session_number,* total requests executed *total_requests*

Destination: Log.

Description: See also message ANR0012I. This message documents the end of a session. The total number of API requests executed on the session is shown by *total_requests*.

Operator Action: None.

ANR0018E Rejecting session *session_number* for *user_name@node: reason*

Destination: Log.

Description: A remote client attempted to connect to the server, and the server found that the nethost file is present; the server validated the client host and username against data in the nethost file. The combination of the user identification (*user_nane)* 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 stdout stream, if the process has stdout mapped to a device such as a terminal or system spool file. 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 "Authorizing SYMAPI sessions" on page 131 for instructions on the syntax.

ANR0020I SYMAPI server listening on port *port_number* over *protocols*

Destination: Log and console.

Description: This message is issued in conjunction with message ANR0001I to inform the system operator about the port (*port_number*) and internet protocols over which the server is communicating. Possible values for *protocols* are:

- IPv4 ONLY — Indicates that the server is listening for connections only using IPv4. Clients that expect an IPv6 connection will fail connecting to the server. In Solutions Enabler Version 6.1.0, IPv4 is the only supported protocol.
- IPv6 and IPv4 — Indicates that the server is listening explicitly for connections using IPv6 and IPv4. In Solutions Enabler Version 6.1.0, this is not supported.
- IPv6 with IPv4 mapping — Indicates that the IPv6 protocol supports connections from clients who are running either IPv4 or IPv6. In Solutions Enabler Version 6.1.0, this is not supported.

Operator Action: None.

ANR0021I The current working directory is *directory*

Destination: Log.

Description: This message is issued early in server initialization after the server process attempts to make the SYMAPI database directory the current working directory.

Operator Action: None. This is an informational message.

ANR0022I SYMAPI server is running on a Symmetrix Service Processor, forcing port *port*

Destination: Log.

Description: This message is written when the server detects it is running on a Symmetrix service processor. In this case, the server forces the use of the default port.

Operator Action: None. This is an informational message.

ANR0023I SYMAPI server Symmwin Pipe Server is initialized

Destination: Log.

Description: This message is written when the special server thread to field requests from the SymmWin component has been started successfully. This will only happen if the server is running on a Symmetrix service processor.

Operator Action: None. This is an informational message.

ANR0024I SYMAPI server Enhanced Authentication is ENABLED | DISABLED

Destination: Log and console.

Description: This message is issued during server initialization to indicate Enhanced User Authentication is enabled or disabled.

- ENABLED indicates that if a client sends an authentication message it will be verified.
- DISABLED indicates that if a client sends an authentication message it will not be verified.

Operator Action: On non-Windows hosts, if the authentication mode indicated in the message is not the mode desired, verify that the `/etc/krb5.keytab` file exists, that its permissions indicate that `storsrvd` can access it, and verify that the `klist -k` value in the file shows the correct entry for the host. If the conditions are all correct, turn on high levels of diagnostic logging to look for additional information.

ANR0030E Failed to load configuration for *name*

Set Step Return Code

Destination: Log and console.

Description: This message is issued when an error is detected in the loading of the configuration settings for the SYMAPI server daemon. The instance name is the name of the daemon for which configuration was attempted.

Operator Action: Examine the messages that precede this message. A syntax error in the configuration file section for the daemon instance *name* is the most likely cause. For example, the port definition may have specified an invalid number for the port, or an invalid security level may have been specified for the security_level option.

ANR0031E The security_level (or -seclevel) keyword requires a security level to be specified

Set Step Return Code

Destination: Log.

Description: This `-seclevel` operand was specified without a value on the `stordaemon setvar` command line.

Operator Action: If you specify security level at all, you must specify a valid value for the security level through the `stordaemon setvar` command. The valid values are NONSECURE, ANY, and SECURE. No abbreviations are accepted.

ANR0032E The -log_filter keyword requires list of log filter types to be specified

Set Step Return Code

Destination: Log.

Description: This `-log_filter` operand was specified without a value on the `storsrvd` command line.

Operator Action: If you specify `-log_filter`, you must specify the desired list of filter types. Use the `stordaemon getvar storsrvd -name log_categories` for the list of appropriate filter types.

ANR0033E The '-port' or 'storsrvd:port' keyword requires a non-zero decimal number less than 65535

Set Step Return Code

Destination: Log.

Description: An invalid value was specified for the SYMAPI server port. If the `storsrvd` command operand `-port` or the `storsrvd:port` statement is used, the value specified for the port must be a non-zero decimal number less 65535. Many port numbers in the lower ranges must also be avoided since they are used by well known processes (for example, the `inetd` and `ftpd` daemons).

Operator Action: Correct the command line or `daemon_options` file specification, and restart the server.

ANR0034I The port is not reloaded while the server is running, bypassing any new port definition

Destination: Log.

Description: During execution of the `reload` command, a change to the port specification was detected. This message is issued to alert the administrator to the fact that the port definition cannot be changed during the reload operation.

Operator Action: In order to change the port, you must shut down the storsrvd process, make the port change, and restart `storsrvd`.

ANR0104E Command syntax error: *explanation*

Destination: Log and 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: Log and 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 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.

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

Destination: Console.

Description: The operator issued command *command_name* without sufficient operands. Default processing could not be established.

Operator Action: Re-enter the command with desired operands, according to the documentation. You can also use the `help` command to determine the required 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.

ANR0116I The *option* runtime option may not be changed while the server is running

Description: The operator or stordaemon user issued the `stordaemon setvar -name` command to change an option which cannot be changed while the server is running.

Operator Action: To change the desired option on the next run of `storsrvd`, you can use `stordaemon setoption` or edit the `daemon_options` file in the SYMAPI configuration directory. If you use the `setoption` command and then try to use `reload`, additional log messages may be issued indicating that some changed options will not be reloaded.

ANR0120I SYMAPI Active Session List:

Destination: Console.

Description: The operator issued the `LIST SESSIONS` command and there are active sessions to list. This message 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 and there are no active sessions to list/show.

Operator Action: None.

ANR0122I Session *number* is not active

Destination: Console.

Description: The operator issued the `SHOW SESSION` command with the `-NUM` option to display a specific session, and the specified session was not 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_nmber* 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 on this platform. The security level specified is *security_level*

Set Step Return Code

Destination: Log and Console.

Description: This message is issued when either the SYMAPI `options` file or `daemon_options` file specified a security level of ANY or SECURE on a platform where secure sessions are not supported. In the case of the `options` file, the SYMAPI_SERVER_SECURITY_LEVEL= statement specified this value. In the case of the `daemon_options` file, the storsrvd:security_level specified ANY or SECURE. The value may have been specified for the `-seclevel` operand of the `storsrvd` command.

Operator Action: If security level is specified through any configuration statement or `storsrvd` command operand, it must specify NONSECURE on platforms where secure sessions are not supported. It is safer to omit the specification altogether, or to specify the dash character '-'. Please refer to the *EMC Solutions Enabler Support Matrix* for a list of platforms where secure sessions are supported.

ANR0141E Could not extract server *file* filename, rc=*returncode*

Destination: Log.

Description: During initialization, the SYMAPI server was not able to determine the name of the file to be used in SSL initialization. The string *file* refers to the SSL type file that the server was about to reference. The failing return code is displayed in *returncode*.

Operator Action: In an Open Systems environment, the server certificate and private key files should have been installed by the normal installation procedure. In z/OS and Microsoft Windows, the location of the Solutions Enabler configuration directory can be adjusted to your configuration needs. Follow the platform specific installation instructions to install the default server certificate files.

ANR0142E *function* establishment failed with rc= *returncode (error_message)*

Destination: Log.

Description: During SSL initialization, the component referred to by *function* failed to be established. If *function* is CERTIFICATE or PRIVATE KEY, then the `symapisrv_cert.pem` file may be damaged or it may not have been successfully copied to the SYMAPI configuration directory.

Operator Action: If server certificate and key files are not installed by default on the platform where the server is running, additional installation steps are necessary. Refer to the platform specific installation instructions to install the files. You can specify NONSECURE for the security level if desired; in which case, the server will not attempt to load the certificate and key files.

ANR0143E Rejected session *address*: security level mismatch reason: *error_message*

Destination: Log.

Description: A mismatch of security levels occurred when an initiating client session requested a security mode that the server was not able to honor.

address is the IP address of the client and *error_message* contains the error message indicating the actual problem.

Operator Action: If possible, modify the security level of the client to match the security mode that the server is using. If that is not possible, then (unless other clients will be impacted), modify the security level of the server to match the security level the client is requesting.

ANR0144E Secure Library Init error: rc=*return_code (error_message)*

Destination: Log.

Description: Some component failed during SSL initialization. The *return_code* value corresponds to the message explained in the string *error_message*.

Operator Action: If you are unable to resolve the problem indicated in string *error_message*, contact EMC technical support for assistance with this error.

ANR0145E The value *value* specified for security level is invalid

Set Step Return Code

Destination: Log.

Description: This message is issued when an attempt is made to set the security level for the SYMAPI server daemon using one of the supported methods, and the value specified is invalid. The methods to set the security level are: the `storsrvd -seclevel` command line option, the `storsrvd:security_level` statement in the `daemon_options` file, or the `stordaemon setvar` command. The valid values are NONSECURE, ANY, or SECURE. Note that a separate message (ANR0148E) is issued if an invalid value is specified in the SYMAPI options file.

Operator Action: Correct the value specified on the command line or in the `daemon_options` file, and restart the server or re-execute the `stordaemon` command.

ANR0146I Security level has changed from *old_security_level*. New sessions will use *new_security_level*

Destination: Log.

Description: The security level to be used by the server was changed successfully using the `setvar` or `reload` command through the stordaemon CLI on the z/OS console. The level was changed from *old_security_level* to *new_security_level*. New sessions will negotiate based on the new security level set, but existing sessions are unaffected by the new level, and will continue to use the security level negotiated when they started.

Operator Action: Confirm that the *new_security_level* is the intended security level. If so, no further action is required. If not, you may want to refer to the server logs or other logs to determine why the security level was changed.

ANR0147I The SYMAPI options file specified an empty value for SYMAPI_SERVER_SECURITY_LEVEL, changing to platform internal default *security_level*

Destination: Log.

Description: A configuration file statement specified an empty value for the server security level. Such a specification is an error, but the server will substitute the default security level value for the platform on which the server is running. The default value is ANY for platforms that support secure mode and NONSECURE for those platforms that do not support secure mode.

Operator Action: The omission of the security level on an explicit configuration is most likely a mistake. Refer to the SYMAPI `options` file or the `daemon_options` file to correct the omission, if you want to suppress the appearance of ANR0147I.

ANR0148E The SYMAPI option SYMAPI_SERVER_SERCURITY_LEVEL specified an invalid value

Set Step Return Code

Destination: Log and Console.

Description: This message is issued during server initialization when an invalid value is specified in the SYMAPI `options` file statement SYMAPI_SERVER_SECURITY_LEVEL. The value for security level is extracted from the SYMAPI `options` file only if no other specification was made on the command line or in the `daemon_options` file. Note that a separate message (ANR0145E) is issued if an invalid value is specified in any of the other methods: `storsrvd` command line, `daemon_options` file, or the `stordaemon setvar` command.

Operator Action: Correct the value specified in the SYMAPI `options` file statement SYMAPI_SERVER_SECURITY_LEVEL . The valid values are NONSECURE, ANY, or SECURE.

ANR0149D Security level has been taken from the SYMAPI options file

Destination: Log.

Description: This message is issued when the value for the server security level is defined in the SYMAPI `options` file and has not been specified on the `storsrvd` command line, nor in the `daemon_options` file. This message is informational only.

ANR0150E The value *value* specified for client certificate verification is invalid

Destination: Log.

Description: This message is issued during server initialization when the value for the client certificate verification option, as defined in the `daemon_options` file, is invalid. It can also be issued when attempting to change this option with the `stordaemon` command to an invalid value.

Operator Action: Correct the value specified in the `daemon_options` file statement `security_clt_secure_lvl` or as specified on the command line. The valid values are NOVERIFY, VERIFY or MUSTVERIFY.

ANR0151E Common Name in client certificate not valid: expected *name*, received *common name*

Destination: Log.

Description: This message is issued during setup of secure mode between client/server. The common name in the client certificate does not match the name the server is expecting.

Operator Action: Check the client certificate to verify that the names contained in the certificate are known hostnames to the server. Either generate a client certificate with the hostname that the server is expecting or add the common name in the client certificate to the applicable `/etc/hosts` file on the server.

ANR0152E Issue detected with server certificate file *filename*

Destination: Log.

Description: This message is issued during initialization of the secure library. A problem with the certificate file has been detected.

Operator Action: Check for the existence of the certificate file on the server. If you have set the `security_alt_cert_file` parameter in the `daemon_options` file, verify that it points to a valid file.

ANR0153E Issue detected with server PrivateKey file *filename*

Destination: Log.

Description: This message is issued during initialization of the secure library. A problem with the PrivateKey file has been detected.

Operator Action: Check for the existence of the `Privatekey` file on the server. If you have set the `security_alt_key_file` parameter in the `daemon_options` file, verify that it points to a valid file.

ANR0154E Host Name pattern in certificate is not valid: pattern

Destination: Log.

Description: This message is issued during setup of secure mode between client/server. It indicates an illegal pattern has been put into the client certificate.

Operator Action: Generate a new client certificate without the illegal host name pattern. The only characters allowed for a host name pattern are letters, numbers, periods (.) or colons (:).

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 message may indicate that the server is over-committed with regard to the number of concurrent sessions, or that there may be a memory leak in the server.

Operator Action: Increase the amount of memory available to the server using the appropriate method for the platform the server is running on. If this does not solve the problem, a memory leak may be indicated by other failure messages. 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 error variable ***errno*** (in Windows, the value returned from the GetLastError() call), 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 SYMAPI 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.

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 provide 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.

ANR0207S Failed to start *name* thread, error = *code (explanation)*

Set Step Return Code

Destination: Log and console.

Description: This message is issued in two cases:

- During server initialization, the attempt to start the dedicated SYMAPI listener thread failed. In this case, *name* is *SYMAPI Listener*. The server will immediately abort initialization and will stop.
- During handling of the arrival of a SYMAPI session, the attempt to start a dedicated thread for the session failed. In this case, *name* is *SYMAPI session*. The server continues to listen for other sessions, although the ability to start new threads can be limited. Other messages may accompany this one with additional diagnostic detail. The return code and explanation from the thread-start service call are displayed in *code* and *explanation*.

Operator Action: Examine other messages in the log files and other system output. You may be able to determine the cause and corrective action from other messages. In the second case, system resources required to start threads may be exhausted due to the current SYMAPI session count. Your system may be configured to allow a maximum number of threads per process, and this limit may have been exceeded. Complete diagnosis may require assistance of EMC technical support.

ANR0208E Unable to verify SYMAPI Database directory *db_dir*

Set Step Return Code

Destination: Log and console.

Description: The server attempts to make the SYMAPI database directory the current directory during initialization in order to cause non-default database files to be placed in the database directory if the name is not a fully-qualified pathname. This message is issued during server initialization if the SYMAPI database directory does not exist or is inaccessible. The most common reason is that the database directory does not exist. The name of the directory the server attempted to verify is shown in *db_dir*.

Operator Action: The Solutions Enabler installation process creates the database directory normally. If this operation failed during installation, the installation process would have terminated with an

error. You can create the directory using the tool appropriate to your platform. Use the directory name shown in *db_dir* in the message text.

ANR0209I Authentication service name *service_name* exceeds maximum length

Destination: Log and console.

Description: The `storsrvd` process is attempting to copy *service_name* to an internal structure and is unable to because of its length.

Operator Action: If possible, shorten the name of the host shown in *service_name*. Otherwise, you may need to collect and provide documentation as directed by EMC Customer Support. The server will continue to operate in non-authenticated mode.

ANR0210E EMCSAI version does not meet minimum version requirement of *nn.nn.nn*

Destination: Log and console.

Description: The version of ResourcePak running on the host does not meet the minimum version required by Solutions Enabler.

Operator Action: Ensure that Solutions Enabler is configured to work with EMC ResourcePak Base at the indicated version or later.

ANR0211E Unable to obtain EMCSAI version, RC=%a (%b) EMCRC=%c, EMCRS=%d

Destination: Log and, in some cases, the console.

Description: This message is issued as a result of an interface error when Solutions Enabler checks the EMC ResourcePak Base version.

Where:

%a is the return code from the call to the ResourcePak Base EMCSAI interface

%b is a text description of the message.

%c is the EMCSAI Return Code (emcrc)

%d is the EMCSAI Reason Code (emcrs)

The most common cause of error is that Solutions Enabler is configured to work with a version of EMC ResourcePak Base which is not running or which does nor exist. Either one of these conditions will result in the following message being issued:

> ANR0211E Unable to obtain EMCSAI version, RC=28 (Symmetrix Control Facility is not available) EMCRC=0, EMCRS=0

Operator Action: In all other cases of the message, contact EMC for support.

ANR0220I Thread *thread_number* will execute without condition handling protection

Destination: Log.

Description: In a z/OS environment, the session on 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: In a z/OS environment, 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 handler involved on thread *thread_number*; writing dump to DD *dump_location*

Destination: Log and console.

Description: In a z/OS environment, an abnormal condition was raised during the session running on *thread_number.* A dump will be written to the DD name *dump_location.* The general format of the DD name is DMP*nnnnn* where *nnnnn* is the *thread_number.*

Operator Action: Consult EMC Customer Support for directions on completing 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: In a z/OS environment, condition handling processing detected a recursive (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: In a z/OS environment, condition handling processing detected a recursive (second) entry into the condition handling routine. This may indicate an abend while attempting to handle an earlier abend.

Operator Action: None.

ANR0225E Condition handling is not supported on this platform

Destination: Log.

Description: In a z/OS environment, language environment *condition handling* supports capturing abnormal termination of a thread without affecting other threads in the process (job). This message is issued when an attempt is made to set or display the current condition handling setting in a non-z/OS environment, using the `stordaemon getvar` or `setvar` command.

Operator Action: Correct the `setvar` or `getvar` command to specify an option which is supported in the environment where you are using the `stordaemon` command.

ANR0300E API Request code *SYMAPI_request_code API_name* rejected; 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* (function named in *API_name*) to determine whether execution has been disabled. The API request was found to be disabled. This message will only be issued in the z/OS environment.

Operator Action: None. In a z/OS environment, control operations can be selectively enabled by using the installation job `#12CNTRL` in the Solutions Enabler RIMLIB dataset.

ANR0301I API Request code *SYMAPI_request_code API_name* executing

Destination: Log.

Description: The server received a SYMAPI request described by the decimal code *SYMAPI_request_code.* This message is issued when the server begins executing the API request. The name of the SYMAPI function name is *API_name*.

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.

ANR0305E REMOTE_CACHED mode not supported for client node *Host_name* version *client_version_number* - connection rejected

Destination: Log.

Description: A client running a version of Solutions Enabler earlier than V6.4 attempted to connect to a SYMAPI server running V7.1, which is not allowed.

Operator Action: None.

B

Asynchronous Events

This appendix lists the possible asynchronous error and message events trapped by the event daemon:

Symmetrix event codes

Asynchronous Symmetrix event codes are required when creating an event exceptions list. Asynchronous event codes are also part of the SNMP trap created and sent in response to a Symmetrix event.

Table 27 lists the possible asynchronous Symmetrix event codes, which are shown in alphabetical order by the following event categories:

- Array subsystem
- Checksum
- Device
- Device pool
- Diagnostic
- Director
- Disk
- Environmental
- Optimizer
- Service processor
- SRDF consistency group
- SRDF link
- SRDF system
- SRDFA session
- Status

Table 27 Asynchronous events (Page 1 of 13)

Code	Category event description
1053	Array subsys Memory bank(s) automatically disabled due to cache error.
1066	Array subsys Bus Arbiter problem.
1083	Array subsys Subsystem unable to set a shared register.
1084	Array subsys Disabled memory bank error reported to a host.

Table 27 Asynchronous events (Page 2 of 13)

Code	Category event description
1155	Array subsys Entries are being written to the audit log at an unusually high rate.
1156	Array subsys Audit log has lost its redundancy due to SFS mirror being offline.
1157	Array subsys Audit log entries have been overwritten in an unusually short period of time.
1108	Checksum Database Double Checksum event triggered.
1127	Checksum Double Checksum (generic) event triggered.
1055	Device M2 mirror resynchronizing with its M1 mirror.
1056	Device M1 mirror resynchronizing with its M2 mirror.
1058	Device Data migration completed on all migration devices.
1059	Device Device resynchronization process started.
1070	Device Device mirror not ready.
1071	Device Device mirror write disabled.
1072	Device SRDF R2 device not ready.
1114	Device Volume (device) not ready.
1111	Device pool SAVE device pool is full.
1115	Device pool SAVE device pool is almost full.

Table 27 Asynchronous events (Page 3 of 13)

Code	Category event description
1116	Device pool Active device in SAVE device pool not ready.
1050	Diagnostic Diagnostic event trace triggered.
1051	Diagnostic Remote (SRDF) diagnostic event trace triggered.
1057	Director Disk director not responding.
1063	Director Fibre Channel front-end has failed or is inoperable.
1117	Director Director not responding.
1054	Disk Hot Spare invoked against a disk.
1060	Disk Hot Spare invoked against a remote R2 mirror disk.
1067	Environmental Internal temperature too high.
1068	Environmental Alarm signal was set, but no alarm found.
1069	Environmental Power subsystem alarm or fault has occurred.
1075	Environmental Abnormal DC voltage (perhaps 12 Volts) situation exists.
1076	Environmental Power subsystem environmental sense cable missing.
1077	Environmental Power system AC line interruption detected.
1078	Environmental Battery system not fully charged.

Table 27 Asynchronous events (Page 4 of 13)

Code	Category event description
1079	Environmental Latched alarms discovered for the power subsystem.
1085	Environmental Validity problem detected during environmental test.
1086	Environmental Environmental testing enabled in diagnostic mode.
1087	Environmental Communication board data does not match expected value.
1088	Environmental Communication board information mismatch.
1089	Environmental Failure detected during thermal test.
1090	Environmental Power-on-Time inconsistencies detected.
1097	Environmental Battery test detected a failure.
1103	Environmental Service processor found environmental readings to be out of limits.
1105	Environmental Service processor detected a smoke detector malfunction.
1106	Environmental Service processor detected a smoke detector alert.
1128	Environmental One of the Power Zones is down - shutdown will occur in 20 hours.
1129	Environmental One of the Power Zones is down - shutdown will occur in five hours.
1142	Environmental Power Supply A multiple fan fault.
1143	Environmental Power Supply A single fan fault.

Table 27 Asynchronous events (Page 5 of 13)

Code	Category event description
1144	Environmental Power Supply A faulted.
1145	Environmental Power Supply A shutdown.
1146	Environmental Power Supply B multiple fan fault.
1147	Environmental Power Supply B single fan fault.
1148	Environmental Power Supply B faulted.
1149	Environmental Power Supply B shutdown.
1150	Environmental Link Card Controller A temperature high.
1151	Environmental Link Card Controller B temperature high.
1152	Environmental Supplemental Power Supply internal fault.
1153	Environmental Supplemental Power Supply battery end of line.
1154	Environmental Supplemental Power Supply low input AC Voltage.
1289	Optimizer FAST tiers have changed.
1290	Optimizer FAST policies have changed.
1291	Optimizer FAST associations have changed.
1292	Optimizer Optimizer/FAST time windows have changed.

Table 27 Asynchronous events (Page 6 of 13)

Code	Category event description
1293	Optimizer Optimizer/FAST control parameters have changed.
1073	Service processor Service processor down or not communicating with array.
1074	Service processor Service processor could not complete a call for service.
1082	Service processor Service processor successfully completed a call for service.
1091	Service processor No records found for the last service processor connection time.
1092	Service processor Service processor communicating via a serial line.
1093	Service processor Remote session to the service processor connected.
1094	Service processor Remote session to the service processor denied access.
1095	Service processor Remote session to the service processor disconnected.
1096	Service processor Service processor detected excessive memory usage.
1098	Service processor Service processor could not communicate with a director.
1099	Service processor Service processor could not query a director.
1100	Service processor Service processor is communicating via local director.
1101	Service processor Service processor unable to read an environmental sensor.
1102	Service processor Service processor detected a failed or unrecognized communications card.

Table 27 Asynchronous events (Page 7 of 13)

Code	Category event description
1104	Service processor Service processor disk is full.
1107	Service processor Service processor triggered a call home for service.
1109	SRDF Cg RDF CG trip event triggered.
1110	Service processor Service processor successfully rebooted.
1064	SRDF link No SRDF links in an RDF group are operational.
1065	SRDF link All SRDF links in an RDF group are operational.
1080	SRDF link Single SRDF link in an RDF group is not operational.
1081	SRDF link Single SRDF link in an RDF group is now operational.
1052	SRDF system Too many suspend/halt chains encountered switching to Adaptive Copy Write Pending Mode.
1061	SRDF system SIM message initiated to a remote SRDF attached array.
1062	SRDF system SRDF error occurred.
1118	SRDF system Timeout writing to an R2 device; writes pending limit reached.
1112	SRDFA Session SRDF/A session inactive.
1113	SRDFA session SRDF/A session active.

Table 27 Asynchronous events (Page 8 of 13)

Code	Category event description
1119	SRDFA session SRDF/A session dropped: write pending limit reached. Host throttling is disabled.
1120	SRDFA session SRDF/A session dropped: write pending limit reached. Host throttling is enabled.
1121	SRDFA session SRDF/A session dropped: device not ready.Tolerance mode is off.
1122	SRDFA session SRDF/A session dropped: device not ready through consistency group.
1123	SRDFA session SRDF/A session dropped: no SRDF links operational.
1124	SRDFA session SRDF/A session dropped: timeout in MSC mode.
1125	SRDFA session SRDF/A session dropped: timeout on Host Adapter.
1126	SRDFA session SRDF/A session dropped: timeout on RA.
1130	SRDF/A session SRDF/A session drop requested [host software initiated].
1131	SRDF/A session SRDF/A session drop at cycle boundary requested [host software initiated].
1132	SRDF/A session SRDF/A session transition out of Asynchronous mode requested [host software initiated].
1133	SRDF/A session SRDF/A session transition from Asynchronous to Synchronous mode requested [host software initiated].
1134	SRDF/A session SRDF/A session drop requested.
1135	SRDF/A session SRDF/A session drop at cycle boundary requested.

Table 27 Asynchronous events (Page 9 of 13)

Code	Category event description
1136	SRDF/A session SRDF/A session transition out of Asynchronous mode requested.
1137	SRDF/A session SRDF/A session transition from Asynchronous to Synchronous mode requested.
1138	SRDF/A session SRDF/A session entering transmit idle state.
1139	SRDF/A session SRDF/A session recovered from transmit idle state.
1140	SRDF/A session SRDF/A session dropped, transmit idle state timeout.
1141	SRDF/A session SRDF/A session dropped, no online RAs
1158	SRDF/A session SRDF/A session dropped, write pending limit reached on a cache partition.
1200	Status Device state has changed to [Not Present | Unknown | Online | Write Disabled | Offline | Failed].
1201	Status Array state has changed to [Not Present | Unknown | Online | Write Disabled | Offline | Failed].
1202	Status Director state has changed to [Not Present | Unknown | Online | Write Disabled | Offline | Failed].
1203	Status Port state has changed to [Not Present | Unknown | Online | Write Disabled | Offline | Failed].
1204	Status Disk state has changed to [Not Present | Unknown | Online | Write Disabled | Offline | Failed].
1205	Status Device configuration has changed.

Table 27 Asynchronous events (Page 10 of 13)

Code	Category event description
1206	Status [Snap SAVEdev | SRDF/A Delta Set Extension] pool state has changed to [Not Present | Unknown | Online | Write Disabled | Offline | Failed].
1207	Status [Snap SAVEdev | SRDF/A Delta Set Extension] pool configuration has changed.
1208	Status [Snap SAVEdev | SRDF/A Delta Set Extension] pool utilization is now *nn*%.
1209	Status Symmetrix external lock has been [released | acquired].
1210	Status Disk hot spare [is no longer invoked against a failed disk | has been invoked against a failed disk].
1215	Status Port configuration has changed.
1216	Status %s Pool device state has changed.
1230	Status Array configuration has changed.
1231	Status Device Masking database has changed.
1232	Status Access Control definitions have changed.
1233	Status Dynamic RDF operation performed on device.
1234	Status Snap session created, activated or deleted.
1235	Status BCV device pairing has changed.
1236	Status HPUX device identifier has changed.

Table 27 Asynchronous events (Page 11 of 13)

Code	Category event description
1237	Status Device name has changed.
1238	Status Device nice name has changed.
1239	Status OpenVMS device identifier has changed.
1280	Status Cache partitioning configuration has changed.
1281	Status Dynamic mapping configuration for a device has changed.
1282	Status Meta configuration for a device has changed.
1211	N/A Number of available disk hot spares is *N*.
1212	N/A Virtual device is now *N* percent allocated.
1213	N/A Virtual device is now *N* percent used.
1214	N/A Director configuration has changed.
1240	N/A Device reservations data has changed.
1241	N/A Time since last SRDF/A cycle switch exceeds minimum cycle time by *n* seconds.
1242	N/A SRDF/A cycles now using *nn* percent of the cache available for it.
1243	N/A Write pending data is now using *nn* percent of the cache.

Table 27 Asynchronous events (Page 12 of 13)

Code	Category event description
1244	N/A Component state has changed to {Not Present | Unknown | Online | Write Disabled | Offline | Failed].
1283	N/A Initiator group has changed.
1284	N/A Storage group has changed.
1285	N/A Director Port group has changed.
1286	N/A Masking view has changed.
1287	N/A Feature registration database has changed.
1288	N/A Application registration database has changed.
1400	N/A User authorization rules have changed.
1401	N/A Audit log is at nn percent of capacity (before wrapping).
1402	N/A Security Alert from Audit Log.
1403	N/A Security Failure Alert from Audit Log.
1404	N/A Message from Audit Log.
1500	N/A Optimizer Swap activity (from Audit Log).
1501	N/A Optimizer Move activity (from Audit Log).
1502	N/A Optimizer configuration change (from Audit Log).

Table 27 Asynchronous events (Page 13 of 13)

Code	Category event description
1503	N/A FAST Controller Swap activity (from Audit Log).
1504	N/A FAST Controller Move activity (from Audit Log).
1505	N/A FAST Controller configuration change (from Audit Log).
1506	N/A Optimizer Rollback activity (from Audit Log).
1507	N/A User approval is required for a Config Change plan generated by the Optimizer/FAST Controller.
1508	N/A The FAST controller has switched to a different state.
1509	N/A The Optimizer has switched to a different mode.

C

Using Daemons

This chapter provides the man page descriptions of the Solutions Enabler daemons.

storapid

DESCRIPTION

If `storapid` is running, the Solutions Enabler runtime libraries (SYMAPI, STORAPI) forward all I/O operations on Symmetrix storage arrays to it for processing.

The `storapid` daemon only performs these I/O operations on behalf of local users who are authorized to use it.

- Privileged users (root on UNIX, in the Administrator's group on Windows) are implicitly authorized to make use of `storapid`.
- Otherwise, users listed in the `daemon_users` authorization file are permitted to make use of `storapid`. This file is located in the following directory:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

Within this file, user `smith` is authorized via the following line:

```
smith     storapid
```

For compatibility with prior releases, users can also be authorized to make use of `storapid` by using an entry in the older `bdmnusers` file located in the same directory.

Within this file, user `smith` is authorized via the following line:

```
smith
```

It is important that both of these authorization files (`daemon_users` and `bdmnusers`) be adequately protected - such that they can only be modified by privileged users.

If a user is not authorized to make use of `storapid`, the Solutions Enabler libraries will perform these I/O operations directly - in the context of the user/application.

OPTIONS

The `stordaemon` command can be used to start, stop and monitor the `storapid` daemon. See the `stordaemon` man page for details.

A number of configuration options can be set in the `daemon_options` file located in the following directory:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

DIAGNOSTICS

Diagnostic messages are written to the log file `storapid-yyyymmdd.log` if the `logfile_type` option is set to `dated`, or to the log files `storapid.log0` and `storapid.log1`. Log files are located in the following directory:

```
UNIX:     /var/symapi/log
Windows:  c:\\Program Files\\EMC\\SYMAPI\\log
```

stordaemon

Controls the EMC Solutions Enabler daemon processes. This command can be used to start, stop, or query on a daemon's status.

SYNTAX

```
stordaemon [-h]  [-scripting]

   list [-v] [-running] [-all] [-brief]
   [-exit_num_running]

   start DaemonName [-inst inst] [-wait numsecs] [-args
   ...]

   shutdown DaemonName |all [-immediate] [-abort] [-wait
   numsecs]

   install DaemonName [-autostart] [-inst inst]

   uninstall DaemonName

   show DaemonName

   showlog DaemonName [-lines numlines]

   getvar DaemonName -name var

   setvar DaemonName -name var=value

   action DaemonName -cmd ...

   setoption DaemonName -name option=value [-force]

   setuser DaemonName |all -user User [-v]

   disable all  [-duration numminutes]

   enable all
```

DESCRIPTION

The `stordaemon` is used to start, stop, query status from or otherwise control the Solutions Enabler daemons.

Note: Note that `stordaemon` can only manage daemons installed on the local machine. It cannot be used for the daemons that will be used by `storsrvd` on a remote host.

For the `start` and `shutdown` operations, `stordaemon` will wait for as many as 30 seconds for the specified daemon(s) to actually start or

exit. The `-wait N` option can be used to specify a different value or to indicate that `stordaemon` should not wait at all.

ARGUMENTS

`action`

Asks the daemon to perform some action. The action to be performed follows the `-cmd` option. The documentation for a daemon lists any operational actions supported by that daemon. Also, an action of 'help' for an individual daemon will list all supported actions.

`disable`

Disables subsequent daemon starts (such as, with `stordaemon start`). This operation is only supported with a daemon name of `all` -- applying to all daemons. Note that there is no effect on daemons that are already running. This operation is provided for use by the Solutions Enabler install scripts. Its use in other scenarios is strongly discouraged. The `-duration` option, if supplied, gives the number of minutes to hold this disabled state. This defaults to 60.

`enable`

Enables subsequent daemon starts—reversing the effect of an earlier `disable` operation. This operation is only supported with a daemon name of `all`—applying to all daemons.

`getvar`

Retrieves a configuration setting for the running specified daemon. The name of the setting is given by the *Var* parameter. The settings, if any, that can be changed for a daemon are detailed in the documentation for that daemon.

`install`

Installs the specified daemon. On Windows, this records the daemon within the Service Control Manager database. On UNIX, this has no effect unless the `-autostart` is supplied as well.

If the `-autostart` flag is provided as well, the operating system is told to automatically start the daemon at system boot time. On Windows, the Service Control Manager handles this startup. On UNIX, this is done by creating an appropriate OS initialization script (with Solaris, under `/etc/init.d and /etc/rc3.d`).

`list`

Lists daemons installed on the host - along with an indication ('*') of which are currently running.

`setoption`

Updates the `daemon_options` file with an options setting for the daemon. The name of the option and its value are given by the Option=*Value* parameter. By default, a line for this option (including the daemon name) must already be present in the `daemon_options` file -- (perhaps commented out). For example, for the `storapid` daemon and the XXX option: # storapid:XXX = ... storapid:XXX = ... The `-force` option can be supplied to override this behavior - adding the setting regardless of whether it is already present in the file or not.

`setuser`

This operation is only supported on UNIX platforms.

By default, daemons on UNIX run as root - since they are installed with root as their owner and their setuid flags turned on. This operation can be used to re-configure a daemon to run under a different user identity. It does this by changing the owner of the daemon executable and re-protecting data files used by that daemon so that it will continue to have access to them.

This is only supported for a subset of the daemons: `storgnsd`, storsrvd, `storevntd`, `storrdfd`, `storwatchd`, and `storstpd`. Specifying a daemon name of 'all' will cause all of these daemons to be processed.

A restriction is that all daemons configured to run as something other than root must run as the same user identity. You cannot, for example, have one daemon run as 'sys' and another as 'bin'.

The `-v` argument can be used to display details on the changes that are made.

`setvar`

Changes a configuration setting for the running specified daemon. The name of the setting and its value are given by the Var=*Value* parameter. The settings, if any, that can be changed for a daemon are detailed in the documentation for that daemon.

`show`

Displays a status message for the specified daemon.

`showlog`

Displays the location of the log files used by the specified daemon. If the `-lines` option is specified, the contents of the log file are displayed as well.

`shutdown`

Shuts down the specified daemon. If `all` is specified, all running daemons are stopped.

Unless the `-immediate` option is specified as well, daemons will not actually terminate until any clients currently connected to them have first disconnected (exited). While in this `shutdown pending` state, connections from new clients will not be accepted.

`start`

Starts the specified daemon. Only privileged users (root or Administrator) can perform this operation.

`uninstall`

Un-installs the specified daemon. On Windows, this removes the definition from the Service Control Manager database. On UNIX, the only effect is to undo any startup file changes that were made by a prior `install -autostart` operation for the daemon.

OPTIONS

`-abort`

Sends a KILL signal, instead of asking the specified daemon(s) to shut themselves down. Only privileged users (root) can use this option. [Supported on UNIX only.]

`-all`

Lists all daemons - even those not installed on the local host.

`-args ...`

Indicates that the remaining arguments are passed to the daemon being started for its use. Refer to the documentation for each of the daemons to see which, if any, additional arguments they accept. This option must be the last `stordaemon` option. The first argument following this must begin with a dash (-).

`-autostart`

Starts the daemon automatically at system boot time.

`-brief`

Displays only daemon names, 1 per line.

`-cmd`

Indicates that the remaining arguments are passed to the daemon as a command to be executed. Refer to the documentation for each of the daemons to see which, if any, commands they accept.

`-duration`

Indicates the number of minutes to disable the daemons. If omitted, this defaults to 60.

`-exit_num_running`

Sets `stordaemon's` exit status to the number of daemons that are currently running.

`-force`

Performs the operation even if the specified option is not already present within the `daemon_options` file -- or if the `daemon_options` file doesn't exist.

`-h`

Provides brief, online help.

`-immediate`

Asks the specified daemon(s) to shutdown immediately. By default, it will wait for the existing client connections to terminate before exiting.

`-inst`

Starts or installs one or more specific instances of a daemon. This option should rarely be needed. Refer to the documentation on each of the database daemons for information on when its use would be appropriate. This option can be used to start multiple instances of certain of the database daemons. One or more instance names can be specified.

`-lines`

Displays the end (tail) of the daemon's log file. If omitted, this defaults to 20.

`-name`

Indicates the name of the setting to be changed and its value. The Var=*Value* parameter is required.

`-running`

Lists only daemons that are actually running at this time.

`-scripting`

Omits any descriptive header messages. This may be useful in scripts when parsing output from `stordaemon`.

`-user`

Specifies the name of the user for which the daemon is to run. For the `setuser` argument.

`-v`

Displays a status message for each daemon that is running. For the setuser argument: a detailed message is displayed for each file whose protections (or owner) is changed.

`-wait`

Waits up to the specified number of seconds for the specified daemon(s) to actually start or terminate. By default, this value is 30 (seconds). If a value of 0 is supplied, `stordaemon` will return immediately instead of waiting for the daemon to start or terminate.

PARAMETERS *Daemon*

The daemon name - as displayed by the `list` argument. For the `show`, `shutdown` and `showlog` arguments, this daemon name must include any instance qualifier used to start the daemon. Refer to the examples section. For the `shutdown` argument, the value `all` causes the operation to be performed on all running daemons.

Inst

One or more daemon instance names, in a comma separated list. For example: `-inst 80` or `inst 80,90,100`. As noted above, this parameter should rarely be needed. For the `install` argument: only a single name may be supplied.

NumLines

The number of lines to display from the end of the daemon's log file.

NumMinutes

The number of minutes that daemons should be disabled for. If omitted, this defaults to 60.

NumSecs

The maximum number of seconds to wait for the specified daemons to terminate.

Option=Value

The daemon option to be set along with its value.

User

The name of the user for which the daemon is to run. For the `setuser` argument.

Var=Value

The name of the configuration setting to be changed and its value.

RETURN CODES

Refer to the *Solutions Enabler SYMCLI Command Reference HTML Help* for a complete list of return codes.

EXAMPLES

To list all daemons that have been installed, along with an indication of which are currently running, enter:

```
stordaemon list
```

To list all daemons that are currently running, along with status information on each, enter:

```
stordaemon -running -verbose list
```

To start the `stororad` daemon and wait up to 5 seconds for it to come up, enter:

```
stordaemon -wait 5 start stororad
```

To start the `stororad` daemon with a specific instance qualifier. The daemon will have an actual name of `stororad80`.

```
stordaemon -inst 80 start stororad
```

To start two instances of the `stororad` daemon named `stororad90` and `stororad100`, enter:

```
stordaemon -inst 90,100 start stororad
```

To shut down the `stororad` daemon started above, enter:

```
stordaemon shutdown stororad
```

To shut down the `80` instance of the `stororad` daemon started above, enter:

```
stordaemon shutdown stororad80
```

To shut down all running daemons, and wait up to 5 seconds for them to actually terminate, enter:

```
stordaemon -wait 5 shutdown all
```

To display a status message for the `stororad` daemon, enter:

```
stordaemon show stororad
```

To display the last 100 lines from the `stororad` daemon's log file, enter:

```
stordaemon showlog stororad -lines 100
```

To set the `gns_remote_mirror` option for the GNS daemon, enter:

```
stordaemon setoption storgnsd -name
   gns_remote_mirror=enable
```

To reconfigure the GNS daemon to run under the `daemon` user identity, enter:

```
storgnsd setuser storgnsd -user daemon
```

storevntd

The `storevntd` daemon acts as the clearing house for Solutions Enabler events on a host.

- In most cases, events are first delivered by a producer tothe Event Daemon... and from there are forwarded to any client applications that have registered an interest in them.
- In Client/Server mode, events are forwarded from the Event Daemon on the remote (server) host to the one on the local (client) host ... and from there to client applications.
- The Event Daemon itself generates (by polling in some cases) events related to Symmetrix and CLARiiON storage arrays.
- The Event Daemon can also be optionally configured to monitor certain events and automatically log them using one or more of the following mechanisms.
  - SNMP traps to an SNMP management application
  - a local log file
  - Syslog (UNIX) or the Windows Event Log

As described below, an options setting can be made to cause a watchdog mechanism to monitor `storevntd` - and automatically restart it if it crashes.

Only authorized users are permitted control to the `storevntd` daemon.

OPTIONS

The `stordaemon` command can be used to start, stop and monitor the `storevntd` daemon. Refer to "stordaemon" on page 238 for details.

A number of configuration options can be set in the `daemon_options` file located in the following directory:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

Some of the more commonly used options are the following:

```
storevntd:symm_poll_interval  = nn
```

How often to poll Symmetrix arrays for events that need to be delivered. This defaults to 60 seconds.

```
storevntd:autorestart  = enable|disable
```

Whether to use a watchdog mechanism to automatically restart storevntd if it crashes. The default setting is enable.

```
storevntd:log_event_targets = ...

storevntd:log_symmetrix_events = ...
```

Whether to automatically log events and, if so, where.

Refer to the `README.daemon_options` file that is installed.

DIAGNOSTICS

By default, diagnostic messages are written to the log files `storevntd.log0` and `storevntd.log1`, located in the following directory:

```
UNIX:     /var/symapi/log
Windows:  c:\\Program Files\\EMC\\SYMAPI\\log
```

storgnsd

GNS is an optional feature. If enabled, it provides a global, distributed repository for DG and CG group definitions. Changes made to a group from one host are automatically visible to other hosts on which GNS is enabled.

GNS stores group definitions on Symmetrix arrays. Specifically, a group's definition is distributed across the arrays for which it contains devices.

The `storgnsd` daemon runs on each host for which GNS is enabled - and is responsible for providing GNS group semantics to other Solutions Enabler components on that host.

Typically, GNS use is enabled via the SYMAPI_USE_GNS option within the options file:

```
SYMAPI_USE_GNS      = ENABLE | DISABLE
```

The ENABLE setting enables GNS for all applications running on the host that are using the default SYMAPI database file. By default, GNS use is disabled on a host.

If GNS is being used to maintain consistency groups within PowerPath or the RDF daemon, the use of this watchdog - along with the automatic start of `storgnsd` at system boot time - is important to ensure that `storgnsd` is always running.

Only authorized users are permitted to use GNS or control the `storgnsd` daemon.

OPTIONS

The `stordaemon` command can be used to start, stop and monitor the `storgnsd` daemon. Refer to "stordaemon" on page 238 for details.

A number of configuration options can be set in the `daemon_options` file located in the following directory:

```
UNIX:      /var/symapi/config
Windows:   c:\\Program Files\\EMC\\SYMAPI\\config
```

Some of the more commonly used options are the following:

```
storgnsd:autorestart  = enable|disable
```

Specifies whether to use a watchdog mechanism to automatically restart `storgnsd` if it crashes. The default setting is `enable`.

```
storgnsd:gns_device_poll_interval = nn
```

Specifies how often `storgnsd` polls for group changes made from other hosts. The smaller this value is, the more quickly a host will recognize group changes made from a other hosts.

```
storgnsd:gns_ppath_poll_interval = nn
```

Specifies how often `storgnsd` checks whether PowerPath or the RDF daemon need to be updated to reflect changes made to CGs stored within GNS.

```
storgnsd:gns_remote_mirror = enable|disable
```

Specifies whether `storgnsd` should automatically mirror RDF1 and RDF2 type DG and CG group definitions for use on remote (SRDF) Symmetrix arrays.

Limitations include:

- REGULAR type groups are not mirrored.
- Mirrors will not be created if the remote Symmetrix array is also directly attached to the local host.
- For CG groups, BCV type devices are not mirrored. The default setting is disable.

```
storgnsd:gns_rmtarr_update_interval = nn
```

If `gns_enb_remote_mirror` is set to enable, how often `storgnsd` attempts to propagate DG group changes over to remote (SRDF) Symmetrix arrays.

```
storgnsd:gns_shadow_file = enable|disable|needed
```

Specifies whether `storgnsd` should maintain backup copies of GNS data for attached Symmetrix arrays. Refer to the discussion under BACKUP SHADOW FILES below.

If set to `enable`, backups are maintained for all local Symmetrix arrays. If set to `needed`, backups are only maintained for Symmetrix arrays with Enginuity versions lower than 5*x*71. The default setting is `disable`.

```
storgnsd:gns_init_remote_symms = enable|disable
```

If enabled (the default), GNS will initialize the GNS group repository on remote Symmetrix arrays while remotely mirroring groups. If disabled, a GNS daemon directly connected to such remote arrays has to perform the initialization when it encounters the array. The default setting is `enable`.

This option should be disabled if (a) remote mirroring is enabled (`gns_remote_mirror`), (b) GNS daemons will be actively running on both sides of the SRDF link, (c) groups are being modified on both the local and remote Symmetrix arrays and (d) GNS backup/restore is enabled `gns_shadow_file`. Failing to disable this option in this scenario may result in data loss due to interaction between the backup/restore and remote mirroring mechanisms.

Refer to the `README.daemon_options` file for additional options, and default values used by `storgnsd`.

BACKUP SHADOW FILES

As previously mentioned, the `gns_shadow_file` option can be set to cause a GNS daemon to automatically maintain backups of GNS data from Symmetrix arrays. These files are written to the local disk as follows, where <*SID*> corresponds to a Symmetrix ID:

```
UNIX:    /var/symapi/gns/<SID>.shadow
Windows: c:\\Program
   Files\\EMC\\SYMAPI\\gns\\<SID>.shadow
```

These files are overwritten each time a GNS daemon notices a change to group data on the Symmetrix array in question. GNS will automatically restore from them if it notices that the GNS repository on a Symmetrix array has been erased. Note that these backups will only be maintained for local arrays; not remote (SRDF) arrays.

It is recommended that this option be enabled on a subset of hosts running GNS daemons. Because of its effect on performance, it usually does not make sense to have more than 2-4 GNS daemons managing these backup copies.

As an alternative to the automatic mechanism, backups to and restores from shadow files can be manually triggered by invoking the `storgnsd` program as follows:

```
storgnsd backup -sid SID
```

Generates a GNS backup file for the specified Symmetrix array.

```
storgnsd backup
```

Generates a GNS backup file for all locally attached Symmetrix arrays.

```
storgnsd restore -sid SID [-noprompt]
```

Restores GNS data to the specified Symmetrix array from a backup file previously created.

Since this operation is destructive in nature, a positive confirmation is required before it will proceed. The `-noprompt` argument will, if present, cause this confirmation step to be skipped.

NOTES

- The `storgnsd` executable is located within the daemons directory in the Solutions Enabler installation area.
- This manual mechanism will typically be most useful if the automatic mechanism is disabled, as it is by default.

  If the automatic mechanism is enabled, it makes sense to regularly backup the files that it generates -- to ensure that they will be available if needed for a future manual restore. This is due to the fact that the automatic mechanism regularly overwrites these backup files.
- These commands can be issued while the GNS daemon is already running normally. If so, a restore operation will be noticed by the daemon the next time it polls that Symmetrix array.

DIAGNOSTICS

By default, diagnostic messages are written to the log files `storgnsd.log0` and `storgnsd.log1`, located in the following directory:

```
UNIX:     /var/symapi/log
Windows:  c:\\Program Files\\EMC\\SYMAPI\\log
```

The style and maximum size of GNS log files can be controlled via options in the `daemon_options` file. Refer to `README.daemon_options` for additional information.

GNS space utilization on each Symmetrix can be displayed with the following command:

```
stordaemon action storgnsd -cmd list -freespace
```

Although very unlikely, it is possible for the GNS data on a Symmetrix to become corrupt if I/O errors occur in the middle of certain critical updates to it. While in such a state, GNS operations to that Symmetrix array will fail.

This can be recognized by the presence of CORRUPT errors in the previously mentioned log file, and will also be indicated by the output produced by the `-freespace` command. In such cases, a `fsck` like repair operation can be initiated via the following command.

This can be performed while the GNS daemon is already running normally.

```
storgnsd repair -sid SID
```

Alternatively, a restore operation can be initiated as described above to recover from the damage.

In extreme circumstances, the following command can be used to fully re-initialize the GNS data area on a given Symmetrix array. This command should be used with extreme care.

```
storgnsd reset -sid SID
```

SECURITY NOTES

Only privileged users are allowed to perform the administrative operations discussed above.

```
storgnsd backup
storgnsd restore
storgnsd reset
storgnsd repair
```

If User Authorization is enabled for a Symmetrix array, the user performing the command must be assigned a role of Admin or StorageAdmin. Refer to the `symauth` command in the *EMC Solutions Enabler Symmetrix CLI Command Reference*.

If User Authorization is not enabled for a Symmetrix array, the user must be the super-user (UNIX) or have Administrative privileges (Windows).

stororad

`stororad` is an SRM API facility that significantly improves communication between applications and Oracle database servers. It runs as a UNIX daemon or Windows service that forwards requests from the application's API function calls to the Oracle database server and returns the results back to the application.

Applications will use the daemon if it is running. The daemon is started automatically when an application issues the SymDbConnect() call to connect to a database instance.

The daemon only supports local client connections so the application must either run on the same host or run on a different host and connect to a SYMAPI server that is local to the daemon. The application can not communicate directly with a daemon on another host.

ARGUMENTS

The `stordaemon` command can be used to start, stop and monitor the stororad daemon. Refer to "stordaemon" on page 238 for details. On Windows, the service control manager can also be used to start the `stororad` service. The default behavior is to start the daemon automatically from the SymDbConnect() API, however if the user starts the daemon using the `stordaemon` command, the applications will use such a daemon.

FILES

If the `-log` option is used on the `stordaemon start` command, a log file will be created with the specified name and path. The daemon, when started without the `-log` option, logs messages in a file in the directory `/var/symapi/log`. The filename is constructed using the daemon name and the ORACLE_SID environment variable that is used to start the daemon. For example, if an Oracle daemon is associated with the ORACLE_SID `orcl8`, then the daemon named `stororadorcl8` will log to files named `stororadorcl8.log0` and `stororadorcl8.log1`.

DIAGNOSTICS

The daemon log file may contain information that may be helpful in diagnosing problems. To enable additional logging the daemon configuration file can be used and the variables `debug` and `pdsdebug` may be set to appropriate values depending on the extent of logging required. Additionally, the SYMAPI log file may also contain useful information for resolving problems.

NOTES

If the daemon cannot be started, applications will use the old mechanism to communicate with Oracle. If an application has already connected through the daemon and the daemon is stopped, the application will continue to run using the old mechanism. If an application has already connected with the old mechanism and the daemon is started, the application will automatically switch to using the daemon. This behavior is true only when the applications have calls that work with the old mechanism and the daemon and does not apply to calls like SymDbSqlExecute() which work only with the daemon.

For applications that need to connect to 64 bit oracle instances, the `storora64d` daemon will be used.

On the Windows platform, `stororad` will be run as a service. After the service is installed, it will appear in the service list dialog box as `stororad` and can be accessed by clicking on the services icon in the control panel (Windows NT) or the administrative tools icon (Windows 2000). The system administrator can modify the startup options in the service list dialog box to start the daemon automatically at boot time.

It is not necessary to set the ORACLE_HOME and ORACLE_SID environment variables when starting the daemon. When the application calls SymDbConnect() or StorDbConnect(), the values will be: (1) retrieved from the application's environment, (2) forwarded with the connection request to the daemon, (3) set by the daemon in its own environment, and (4) retrieved by Oracle's client connection code from the daemon's environment.

The database connection is maintained by a thread in the daemon until the application calls SymDbDisconnect() or StorDbDisconnect(). The autostart mechanism requires all the applications to set the appropriate environment variables required to communicate to the database.

The daemon started automatically by the API SymDbConnect() is associated with the database instance specified by ORACLE_HOME and ORACLE_SID. Applications which have the same setting for ORACLE_HOME and ORACLE_SID are serviced by the daemon so started.

For example: If the user has ORACLE_HOME set to `/db/oracle/8.0.6` and ORACLE_SID set to `orcl8` then a daemon named `stororadorcl8` is started. Applications that have the same settings for ORACLE_HOME and ORACLE_SID will use the daemon

`stororadorc18`. Applications that have different settings for the environment variables will start a different daemon. The ORACLE_HOME and ORACLE_SID together determine the daemon that would service the application.

storrdfd

DESCRIPTION

The RDF daemon, `storrdfd`, is an optional feature that provides consistency for SRDF/A MSC and RDF-ECA composite groups (CGs) in multi- Symmetrix environments.

RDF consistency protection is enabled when an application enables a CG for MSC or RDF-ECA protection. When providing MSC consistency protection, the RDF daemon performs MSC cycle switching and cache recovery for all Symmetrix arrays participating in the RDF consistency-enabled CG. When providing RDF-ECA protection the RDF daemon monitors ECA trip events and propagates any ECA trips to all devices on all Symmetrix arrays participating in the RDF consistency-enabled CG.

The `storrdfd` daemon runs on each host for which RDF consistency is required. If GNS is enabled the RDF daemon relies on GNS to propagate CG definitions to all pertinent hosts. If GNS is not enabled, the user must explicitly create the exact same CG definitions on each participating host. Refer to "storgnsd" on page 248 for details on GNS usage.

The watchdog daemon is required by the RDF daemon to automatically restart it if it crashes.

ARGUMENTS

The `stordaemon` command can be used to start, stop and monitor the `storrdfd` daemon. Refer to "stordaemon" on page 238 for details.

A number of configuration options can be set in the `daemon_options` file located in the following directory:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

Some of the more commonly used options are the following:

```
storrdfd:logfile_size = nn
```

The maximum size (in KB) that the RDF daemon's log file is permitted to grow to before wrapping. Refer to the notes under DIAGNOSTICS below.

Refer to the `README.daemon_options` file for additional options - along with default values that `storrdfd` uses.

DIAGNOSTICS

Diagnostic messages are written to the log files `storrdfd.log0` and `storrdfd.log1`, located in the following directory:

```
UNIX:     /var/symapi/log
Windows:  c:\\Program Files\\EMC\\SYMAPI\\log
```

storsqld

`storsqld` is an SRM API facility that significantly improves communication between applications and Microsoft SQL Server database servers. It runs as a Windows service that forwards requests from the application's API function calls to the Microsoft SQL Server database server and returns the results back to the application. Applications will use the service if it is running. The service is started automatically when an application issues the SymDbConnect() call to connect to a database instance.

The service only supports local client connections so the application must either run on the same host or run on a different host and connect to a SYMAPI server that is local to the service. The application can not communicate directly with a service on another host.

ARGUMENTS

The `stordaemon` command can be used to start, stop and monitor the `storsqld` service. Refer to "stordaemon" on page 238 for details. In addition, the service control manager can also be used to start the `storsqld` service. The default behavior is to start the service automatically from the SymDbConnect() API, however if the user starts the service using the `stordaemon` command, the applications will use such a service.

FILES

If the `-log` option is used on the `stordaemon start` command, a log file will be created with the specified name and path. The daemon when started without the `-log` option logs messages in a file which resides in the directory `C:\Program Files\Emc\symapi\log`. The log filename is constructed using the service name. The `storsqld` service will log to files named `storsqld.log0` and `storsqld.log1`.

DIAGNOSTICS

The service log file may contain information that may be helpful in diagnosing problems. To enable additional logging the daemon configuration file can be used and the variables `debug` and `pdsdebug` may be set to appropriate values depending on the extent of logging required. Additionally, the SYMAPI log file may also contain useful information for resolving problems.

NOTES

If the service cannot be started, applications will use the old mechanism to communicate with Microsoft SQL Server. If an application has already connected via the service and the service is stopped, the application will continue to run using the old

mechanism. If an application has already connected with the old mechanism and the service is started, the application will automatically switch to using the service.

`storsqld` runs as a service. After the service is installed, it will appear in the service list dialog box as EMC `storsqld` and can be accessed by clicking on the services icon in the control panel (Windows NT) or the administrative tools icon (Windows 2000). The system administrator should modify the startup options in the service list dialog box to start the service automatically at boot time.

The database connection is maintained by a thread in the service until the application calls SymDbDisconnect() or StorDbDisconnect().

storsrmd

storsrmd is an SRM API facility that allows non-root or non-administrator to do the SRM operations. The storsrmd64 daemon is for 64-bit operating system.

One of the primary benefits of the SRM daemon is that SRM applications no longer have to execute as a root or administrator user.

The SRM daemon only performs the operations on behalf of local users who are authorized to use it. Privileged users (root on UNIX, in the Administrator's group on Windows) are implicitly authorized without using storsrmd (or storsrmd64 in 64-bit).

If users are permitted to do certain SRM control operations, the following line needs to be added to daemon_users:

```
smith     storsrmd    <control operation>
```

There is a list of SRM control operations in the daemon_users file. There are currently 16 control operations for SRM daemons.

To use certain operations, non-root users also need to have operating system-level permission, for example:

To only permit root user access:

```
-rwx------  root   other   /usr/vxfs/root.bin
```

To permit user smith and root access:

```
-rwx------  smith   symapi   /usr/vxfs/smithroot.bin
```

To permit all users access:

```
-rwxrwxrwx  root   other   /usr/vxfs/allusers.bin
```

(Assume -rwxrwxrwx for /usr and /usr/vxfs.)

It is important that the authorization files (daemon_users) be adequately protected so that they can only be modified by privileged users.

The SRM daemon is supported on AIX, SOLARIS,HP-UX, Windows, OSF1, and Linux platforms.

The daemon only supports local client connections so the application must either run on the same host or run on a different host and connect to a SYMAPI server that is local to the daemon. The

application can not communicate directly with a daemon on another host.

OPTIONS

The `stordaemon` command can be used to start, stop and monitor the `storsrmd` daemon. Refer to "stordaemon" on page 238 for details.

A number of configuration options can be set in the `daemon_options` file located in the following directory:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

DIAGNOSTICS

The daemon log file may contain information that may be helpful in diagnosing problems. To enable additional logging the daemon configuration file can be used and the variables `debug` and `pdsdebug` may be set to appropriate values depending on the extent of logging required. Additionally, the SYMAPI log file may also contain useful information for resolving problems.

Diagnostic messages are written to the log file `storsrmd-yyyymmdd.log` if the `logfile_type` option is set to `dated`, or to the log files `storsrmd.log0` and `storsrmd.log1`.

```
Log files are located in the following directory:
        UNIX:     /var/symapi/log
        Windows:  c:\\Program Files\\EMC\\SYMAPI\\log
```

NOTES

For a 64 bit application, it will automatically use 64 bit SRM daemon (`storsrmd64`). User may also use `stordaemon` to start the 64 bit SRM daemon.

storsrvd

The `storsrvd` daemon replaces symapisrv as the provider of SYMAPI services to remote clients. `storsrvd` listens for SYMAPI sessions from remote clients and also responds to management requests from authorized users of the `stordaemon` command on the same host.

LOGGING

The `storsrvd` daemon uses the common logging infrastructure of all Solutions Enabler daemons. You can find log files under:

```
UNIX:      /var/symapi/log
Windows:   c:\\Program Files\\EMC\\SYMAPI\\log
```

The `storsrvd` log files by default operate as a pair of files which alternate, switching from one to the other as the size reaches an installation chosen limit. You will see: `storsrvd.log0` and `storsrvd.log1` by default. If you choose to use dated logs. `storsrvd` will write to a single file per day, switching on first write to the log after midnight. In this case you will see one or more files of this form:

```
storsrvd-yyyymmdd.log
```

See the `daemon_options` file `log_filetype` option for more information on using dated logs.

Logging volume is controlled using two options:

The `-log_level` option controls inclusion of log messages based on severity: debug, info, warning, error, critical, `log_filter` controls the inclusion of log messages based on category. The categories used by `storsrvd` are:

SERVER: high level events related to general server operation

SESSION: show events that track SYMAPI session arrival and termination.

APIREQ: show events that track SYMAPI API request arrival and termination. This setting can cause a high volume of log output.

CONTROLS: show events that relate to execution of management commands.

TCPIP: show network events.

For more information on logging volume controls, visit the `daemon_options` file. See also the description of the `log_show_category` option below.

All log messages issued by the core `storsrvd` logic include message identifiers. You will see `ANRnnnnX` which clearly identifies messages issued by `storsrvd` and distinguishes them from common daemon log messages. In the message identifier, *nnnn* is a decimal number, and *X* corresponds to the severities listed above in the `log_level` item. The `storsrvd` log messages include a description of the circumstances under which the message is issued, and what action must be taken, if any is necessary.

AUTHORIZATION

SYMAPI Sessions - NETHOST FILE

An optional file that lists trusted hosts and users can be placed in the Solutions Enabler configuration directory described above on the host where `storsrvd` runs. You create the file `nethost` in:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

The server uses the `nethost` file to authorize incoming client connections from applications that are executing remote SYMAPI functions. If this file exists, then only nodes and users entered in the file will be allowed to connect to the server to execute remote API functions. If this file does not exist, then all SYMAPI clients are allowed to connect to the SYMAPI server.

The valid formats of an entry are as follows:

```
node            user-1[,...,user-n]
address         user-1[,...,user-n]
*               user-1[,...,user-n]
node            *
address         *
*               *
```

(where * means wildcard for any host or any user)

Note: For readability and maintainability when specifying a long list of users, the node or address can be repeated with additional user names listed.

CONTROL Sessions - daemon_users Designated users can interact with the storsrvd daemon using the stordaemon command. All control commands are authorized in one of the following ways:

- Privileged users (root on UNIX, in the Administrator's group on Windows) are implicitly authorized to use stordaemon to manage storsrvd.
- Otherwise, users listed in the daemon_users authorization file are permitted to manage storsrvd.

This file is located in the Solutions Enabler configuration directory:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

Each user listed in daemon_users may be authorized for one or more daemons. For complete instructions on the use of daemon_users authorization, visit the file as described in the previous paragraph.

It is important that the daemon_users authorization file be adequately protected - such that it can only be changed by privileged users.

ACTIONS

The stordaemon command can be used to start, stop and monitor the storsrvd daemon. Refer to "stordaemon" on page 238 for details. storsrvd will respond to the base commands supported by all daemons through stordaemon, and supports a set of extended commands using the stordaemon action command.

```
stordaemon action storsrvd -cmd reload
```

Tells storsrvd to re-read the storsrvd section for changed options (see also stordaemon setvar).

```
stordaemon action storsrvd -cmd list -sessions
```

Tells storsrvd to display a list of currently active SYMAPI sessions.

```
stordaemon action storsrvd -cmd show -sessions  [-num
   nnn]
```

Tells storsrvd to display the details for a specific session (if -num *nnn* is specified) or for all active SYMAPI sessions.

OPTIONS

A number of configuration options can be set in the daemon_options file located in the following directory:

```
UNIX:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

`storsrvd` will support all the common daemon options for logging, auto-restart, and debugging data. In addition, the options specific to `storsrvd` are shown below:

```
storsrvd:port = nnnn
```

Specifies a positive decimal number less than 65536, to indicate which TCP/IP port `storsrvd` will listen for remote client connections. The default is 2707.

```
storsrvd:security_level  = NONSECURE | ANY | SECURE
```

Specifies the desired security level to apply to remote connections. The default depends on the platform. For interoperability information refer to EMC E-Lab Interoperability Navigator at: http://elabnavigator.EMC.com.

```
storsrvd:log_show_category  = enable | disable
```

Enables or disables the display of the category used for each `storsrvd` numbered log message. The default value is disable.

storstpd

The storstpd daemon is an optional feature and is used to collect raw performance counters for a Symmetrix array, directors, devices, disks, ports, LRU, SRDF/A, Open Replicator, device groups, and cache partitions. These counters are collected according to criteria defined in the daemon_options file, and stored in a TTP formatted ASCII file stored in the following location:

```
Unix:     /var/symapi/stp
Windows:  c:\Program Files\EMC\SYMAPI\stp
```

The storstpd daemon can be started in a number of ways,

- An administrator can explicitly start storstpd as follows:

  stordaemon start storstpd

- An administrator can arrange for the OS to automatically start storstpd at system boot time as follows:

  stordaemon install storstpd -autostart

As described below, an options setting can be set that causes a watchdog mechanism to monitor storstpd, and automatically restart it if it crashes.

Once running, the user (an administrator) can communicate with the storstpd daemon using the stordaemon CLI.

Starting with Solutions Enabler V6.5, storsptd also provides collection services to the EMC Symmetrix Management Console (SMC) clients.

DESCRIPTION

The storstpd daemon is the Solutions Enabler performance statistics collector which provides collection service to multiple applications, including TTP and SMC. Like all Solutions Enabler daemons, the storstpd daemon is manipulated using the stordaemon command line utility.

Common collector

Starting with Solutions Enabler 6.5, a common collector is introduced for efficiency and to reduce host and Symmetrix resource consumption. The common collector collects performance metric data at regular intervals and application packagers provide the collected data to the different applications supported (TTP, SMC).

OPTIONS

The `stordaemon` command can be used to start, stop, install, and show the `storstpd` daemon. See the "stordaemon" on page 238 for more details.

There are a number of configuration options that can be set using the `stordaemon setoption` command or editing the `daemon_options` file located in the following directory:

```
Unix:     /var/symapi/config
Windows:  c:\\Program Files\\EMC\\SYMAPI\\config
```

The `storstpd` daemon has options that can be modified using the `stordaemon setoption` command.

```
stordaemon setoption storstpd -name
   option_name=option_value
```

Options that are specific to the operation of the daemon itself include the following:

Option	Default value
`dmn_service_processor_delay`	9 minutes (SP only)
`dmn_start_spa`	Enabled (not on service processor)
`dmn_start_smc`	Enabled (not on service processor)
`dmn_start_ttp`	Enabled

`dmn_root_location`	Unix:
	`/var/symapi/stp`
	Windows:
	`c:\\Program Files\\EMC\\SYMAPI\\stp`

Symmetrix Management Console (SMC)

The `storstpd` daemon supports SMC's QOS monitor by sending specific performance statistics for system-level metrics, cache partition metrics, and device group metrics, through event postings to the event daemon to all registered clients. Currently, only the QOS monitor has this capability.

Options that are specific to the operation of the real-time collector include the following:

Option	Default value
smc_collect_interval	5 Minutes

The smc_collect_interval can be 2, 3, 4, 5, 10, or 15 minutes.

TTP

TTP provides performance collection data to external tools such as STP Navigator and SymmMerge.

Options that are specific to the operation of the TTP collection include the following:

ttp collection option	Default value
ttp_archive_interval	0
ttp_archive_time	00:00
ttp_audit_log	Enabled
ttp_auto_archive	Enabled
ttp_collect_iterations	0
ttp_collection_interval	15
ttp_config_db	Enabled
ttp_cpdev_metrics	Disabled
ttp_cpt_metrics	Disabled
ttp_dev_groups	N/A
ttp_dev_metrics	Enabled
ttp_dgdev_metrics	Disabled
ttp_dir_metrics	Enabled
ttp_disk_metrics	Enabled
ttp_disk_space_threshold	80%
ttp_lru_metrics	Disabled
ttp_ors_metrics	Enabled

ttp collection option	Default value
ttp_port_metrics	Enabled
ttp_rdfa_metrics	Enabled
ttp_rdfdev_metrics	Disabled
ttp_rdfdir_metrics	Disabled
ttp_rdfgrp_metrics	Disabled
ttp_rdflnk_metrics	Disabled
ttp_rdfsys_metrics	Disabled
ttp_re_nwc_metrics	Disabled
ttp_re_sg_metrics	Disabled
ttp_retention_days	7
ttp_retention_policy	Enabled
ttp_se_nw_metrics	Disabled
ttp_se_nwi_metrics	Disabled
ttp_se_tcp_metrics	Disabled
ttp_symmids	N/A
ttp_sys_metrics	Enabled

If SMC is enabled, the `ttp_collection_interval` can be 2, 3, 4, 5, 10, or 15 minutes.

If SMC is disabled, the `ttp_collection_interval` can be 1, 2, 3, 4, 5, 10, or 15 minutes.

If running on the service processor, the `ttp_collection_interval` can be 1, 2, 3, 4, 5, 10, or 15 minutes.

Note: A `ttp_collection_interval` of less than 5 minutes on the service processor is not recommended.

ACTIONS (TTP only)

The `storstpd` daemon supports the following action commands for controlling TTP collection. Action commands, invoked using the `stordaemon action` command, are used as follows:

```
stordaemon action storstpd -cmd start
```

Tells `storstpd` to start a TTP collection based on the current defined options if a collection is not currently running.

```
stordaemon action storstpd -cmd stop
```

Tells `storstpd` to stop all TTP collections currently running.

```
stordaemon action storstpd -cmd archive
```

Tells `storstpd` to close all open (running) TTP collection files, open new TTP collection files, and continue with TTP collection.

```
stordaemon action storstpd -cmd restart
```

Tells `storstpd` to stop the current TTP collections, close all TTP data files, refresh options to current `daemon_options`, and start a new TTP collection.

The `storstpd` daemon is supported on Windows and UNIX platforms.

DIAGNOSTICS

By default, diagnostic messages are written to the log files `storstpd.log0` and `storstpd.log1`, located in the following directory:

```
Unix:     /var/symapi/log
Windows:  c:\\Program Files\\EMC\\SYMAPI\\log
```

The style and maximum size of `storstpd` log files can be controlled using options in the `daemon_options` file. Refer to `daemon_options.README` for additional information.

storsybs11d

`storsybs11d` is an SRM API facility that significantly improves communication between applications and Sybase Adaptive Server database servers. It runs as a UNIX daemon that forwards requests from the application's API function calls to the Sybase Adaptive Server database server and returns the results back to the application. Applications will use the daemon if it is running. The daemon is started automatically when an application issues the SymDbConnect() call to connect to a database instance.

The daemon only supports local client connections so the application must either run on the same host or run on a different host and connect to a SYMAPI server that is local to the daemon. The application can not communicate directly with a daemon on another host.

ARGUMENTS

The `stordaemon` command can be used to start, stop and monitor the `storsybs11d` daemon. Refer to "stordaemon" on page 238 for details. The default behavior is to start the daemon automatically from the SymDbConnect() API, however if the user starts the daemon using the `stordaemon` command, the applications will use such a daemon.

FILES

If the `-log` option is used on the `stordaemon start` command, a log file will be created with the specified name and path. The daemon when started without the `-log` option logs messages in a file which resides in the directory `/var/symapi/log`. The filename is constructed using the daemon name and the DSQUERY environment variable that is used to start the daemon. For example if an Sybase daemon is associated with the DSQUERY `syb119`, then the daemon named `storsybs11dsyb119` will log to files named `storsybs11dsyb119.log0` and `storsybs11dsyb119.log1`.

DIAGNOSTICS

The daemon log file may contain information that may be helpful in diagnosing problems. To enable additional logging, the daemon configuration file can be used and the variables `debug` and `pdsdebug` may be set to appropriate values depending on the extent of logging required. Additionally, the SYMAPI log file may also contain useful information for resolving problems.

NOTES

If the daemon cannot be started, applications will use the old mechanism to communicate with Sybase. If an application has already connected using the daemon, and the daemon is stopped, the application will continue to run using the old mechanism. If an

application has already connected with the old mechanism and the daemon is started, the application will automatically switch to using the daemon. This behavior is true only when the applications have calls that work with the old mechanism and the daemon and does not apply to calls like SymDbSqlExecute() which work only with the daemon.

The `storsybs11d` daemon can only connect to Sybase ASE 11.9 servers. It can not connect to Sybase ASE 12.0 or 12.5 servers.

It is necessary to set the LD_LIBRARY_PATH environment variable to the Sybase Adaptive Server library directory when starting the daemon. It is not necessary to set the SYBASE or DSQUERY environment variables when starting the daemon. When the application calls SymDbConnect() or StorDbConnect(), these values will be: (1) retrieved from the application's environment, (2) forwarded with the connection request to the daemon, (3) set by the daemon in its own environment, and (4) retrieved by Sybase Adaptive Server's client connection code from the daemon's environment. The database connection is maintained by a thread in the daemon until the application calls SymDbDisconnect() or StorDbDisconnect(). The autostart mechanism requires all the applications to set the appropriate environment variables required to communicate to the database.

The daemon started automatically by the API SymDbConnect() is associated with the database instance specified by SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS.

Applications which have the same setting for the environment variables mentioned above are serviced by the daemon so started. For example, if the user has SYBASE set to `/db/sybase/11.9` and DSQUERY set to `syb119` then a daemon named `storsybs11dsyb119` is started. Applications that have the same settings for SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS will use the daemon `storsybs11dsyb119`.

Applications that have different settings for the environment variables will start a different daemon. The variables SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS together determine the daemon that would service the application.

storsybs12.5d

`storsybs12.5d` is an SRM API facility that significantly improves communication between applications and Sybase Adaptive Server database servers. It runs as a UNIX daemon that forwards requests from the application's API function calls to the Sybase Adaptive Server database server and returns the results back to the application. Applications will use the daemon if it is running. The daemon is started automatically when an application issues the SymDbConnect() call to connect to a database instance.

The daemon only supports local client connections so the application must either run on the same host or run on a different host and connect to a symapi server that is local to the daemon. The application can not communicate directly with a daemon on another host.

ARGUMENTS

The `stordaemon` command can be used to start, stop and monitor the `storsybs12.5d` daemon. Refer to "stordaemon" on page 238 for details. The default behavior is to start the daemon automatically from the SymDbConnect() API, however if the user starts the daemon using the `stordaemon` command, the applications will use such a daemon.

FILES

If the `-log` option is used on the `stordaemon start` command, a log file will be created with the specified name and path. The daemon when started without the `-log` option logs messages in a file which resides in the directory `/var/symapi/log`. The filename is constructed using the daemon name and the DSQUERY environment variable that is used to start the daemon. For example if an sybase daemon is associated with the DSQUERY `syb125`, then the daemon named `storsybs12.5dsyb125` will log to files named `storsybs12.5dsyb125.log0` and `storsybs12.5dsyb125.log1`.

DIAGNOSTICS

The daemon log file may contain information that may be helpful in diagnosing problems. To enable additional logging the daemon configuration file can be used and the variables `debug` and `pdsdebug` may be set to appropriate values depending on the extent of logging required. Additionally, the SYMAPI log file may also contain useful information for resolving problems.

NOTES

If the daemon cannot be started, applications will use the old mechanism to communicate with Sybase. If an application has already connected via the daemon and the daemon is stopped, the

application will continue to run using the old mechanism. If an application has already connected with the old mechanism and the daemon is started, the application will automatically switch to using the daemon. This behavior is true only when the applications have calls that work with the old mechanism and the daemon and does not apply to calls like SymDbSqlExecute() which work only with the daemon.

The `storsybs12.5d` daemon can only connect to Sybase ASE 12.5 servers. It can not connect to Sybase ASE 11.9 or 12.0 servers.

For applications that need to connect to 64 bit Sybase instances, the `storsybs12.5_64d` daemon will be used.

It is necessary to set the LD_LIBRARY_PATH environment variable to the Sybase Adaptive Server library directory when starting the daemon. It is not necessary to set the SYBASE or DSQUERY environment variables when starting the daemon. When the application calls SymDbConnect() or StorDbConnect(), these values will be: (1) retrieved from the application's environment, (2) forwarded with the connection request to the daemon, (3) set by the daemon in its own environment, and (4) retrieved by Sybase Adaptive Server's client connection code from the daemon's environment. The database connection is maintained by a thread in the daemon until the application calls SymDbDisconnect() or StorDbDisconnect(). The autostart mechanism requires all the applications to set the appropriate environment variables required to communicate to the database.

The daemon started automatically by the API SymDbConnect() is associated with the database instance specified by SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS.

Applications which have the same setting for the environment variables mentioned above are serviced by the daemon so started. For example, if the user has SYBASE set to `/db/sybase/12.5` and DSQUERY set to `syb125` then a daemon named `storsybs12.5dsyb125` is started. Applications that have the same settings for SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS will use the daemon `storsybs12.5dsyb125`.

Applications that have different settings for the environment variables will start a different daemon. The variables SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS together determine the daemon that would service the application.

storsybs12d

`storsybs12d` is an SRM API facility that significantly improves communication between applications and Sybase Adaptive Server database servers. It runs as a UNIX daemon that forwards requests from the application's API function calls to the Sybase Adaptive Server database server and returns the results back to the application. Applications will use the daemon if it is running. The daemon is started automatically when an application issues the SymDbConnect() call to connect to a database instance.

The daemon only supports local client connections so the application must either run on the same host or run on a different host and connect to a SYMAPI server that is local to the daemon. The application can not communicate directly with a daemon on another host.

ARGUMENTS

The `stordaemon` command can be used to start, stop and monitor the `storsybs12d` daemon. Refer to "stordaemon" on page 238 for details. The default behavior is to start the daemon automatically from the SymDbConnect() API, however if the user starts the daemon using the `stordaemon` command, the applications will use such a daemon.

FILES

If the `-log` option is used on the `stordaemon start` command, a log file will be created with the specified name and path. The daemon when started without the `-log` option logs messages in a file which resides in the directory `/var/symapi/log`. The filename is constructed using the daemon name and the DSQUERY environment variable that is used to start the daemon. For example if a Sybase daemon is associated with the DSQUERY `syb120`, then the daemon named `storsybs12dsyb120` will log to files named `storsybs12dsyb120.log0` and `storsybs12dsyb120.log1`.

DIAGNOSTICS

The daemon log file may contain information that may be helpful in diagnosing problems. To enable additional logging the daemon configuration file can be used and the variables `debug` and `pdsdebug` may be set to appropriate values depending on the extent of logging required. Additionally, the SYMAPI log file may also contain useful information for resolving problems.

NOTES

If the daemon cannot be started, applications will use the old mechanism to communicate with Sybase. If an application has already connected via the daemon and the daemon is stopped, the application will continue to run using the old mechanism. If an

application has already connected with the old mechanism and the daemon is started, the application will automatically switch to using the daemon. This behavior is true only when the applications have calls that work with the old mechanism and the daemon and does not apply to calls like SymDbSqlExecute() which work only with the daemon.

The `storsybs12d` daemon can only connect to Sybase ASE 12.0 servers. It can not connect to Sybase ASE 11.9 or 12.5 servers.

It is necessary to set the LD_LIBRARY_PATH environment variable to the Sybase Adaptive Server library directory when starting the daemon. It is not necessary to set the SYBASE or DSQUERY environment variables when starting the daemon. When the application calls SymDbConnect() or StorDbConnect(), these values will be: (1) retrieved from the application's environment, (2) forwarded with the connection request to the daemon, (3) set by the daemon in its own environment, and (4) retrieved by Sybase Adaptive Server's client connection code from the daemon's environment. The database connection is maintained by a thread in the daemon until the application calls SymDbDisconnect() or StorDbDisconnect(). The autostart mechanism requires all the applications to set the appropriate environment variables required to communicate to the database.

The daemon started automatically by the API SymDbConnect() is associated with the database instance specified by SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS.

Applications which have the same setting for the environment variables mentioned above are serviced by the daemon so started. For Example: If the user has SYBASE set to `/db/sybase/12.0` and DSQUERY set to `syb120` then a daemon named `storsybs12dsyb120` is started. Applications that have the same settings for SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS will use the daemon `storsybs12dsyb120`.

Applications that have different settings for the environment variables will start a different daemon. The variables SYBASE, DSQUERY, SYBASE_ASE and SYBASE_OCS together determine the daemon that would service the application.

storudbd

`storudbd` is an SRM API facility that significantly improves communication between applications and IBM DB2/UDB database servers. It runs as a UNIX daemon or Windows service that forwards requests from the application's API function calls to the IBM DB2/UDB database server and returns the results back to the application. Applications will use the daemon if it is running. The daemon is started automatically when an application issues the SymDbConnect() call to connect to a database instance.

The daemon only supports local client connections so the application must either run on the same host or run on a different host and connect to a symapi server that is local to the daemon. The application can not communicate directly with a daemon on another host.

ARGUMENTS

The `stordaemon` command can be used to start, stop and monitor the `storudbd` daemon. Refer to "stordaemon" on page 238 for details. On Windows, the service control manager can also be used to start the storudbd service. The default behavior is to start the daemon automatically from the SymDbConnect() API, however if the user starts the daemon using the `stordaemon` command, the applications will use such a daemon.

FILES

If the `-log` option is used on the `stordaemon start` command, a log file will be created with the specified name and path. The daemon when started without the `-log` option logs messages in a file which resides in the directory `/var/symapi/log`. The filename is constructed using the daemon name and the DB2INSTANCE environment variable that is used to start the daemon. For example if an DB2/UDB daemon is associated with the DB2INSTANCE `db2v7in1`, then the daemon named `storudbddb2v7in1` will log to files named `storudbddb2v7in1.log0` and `storudbddb2v7in1.log1`.

DIAGNOSTICS

The daemon log file may contain information that may be helpful in diagnosing problems. To enable additional logging the daemon configuration file can be used and the variables `debug` and `pdsdebug` may be set to appropriate values depending on the extent of logging required. Additionally, the SYMAPI log file may also contain useful information for resolving problems.

NOTES If the daemon cannot be started, applications will use the old mechanism to communicate with IBM DB2/UDB. If an application has already connected using the daemon, and the daemon is stopped, the application will continue to run using the old mechanism. If an application has already connected with the old mechanism and the daemon is started, the application will automatically switch to using the daemon. This behavior is true only when the applications have calls that work with the old mechanism and the daemon and does not apply to calls like SymDbSqlExecute() which work only with the daemon.

On the Windows platform, `stoudbd` will be run as a service. After the service is installed, it will appear in the service list dialog box as `storudb` and can be accessed by clicking on the services icon in the control panel (Windows NT) or the administrative tools icon (Windows 2000). The system administrator can modify the startup options in the service list dialog box to start the daemon automatically at boot time.

It is necessary to set the LD_LIBRARY_PATH environment variable to the IBM DB2/UDB library directory when starting the daemon. It is not necessary to set the DB2INSTANCE environment variable when starting the daemon. When the application calls SymDbConnect() or StorDbConnect(), this value will be: (1) retrieved from the application's environment, (2) forwarded with the connection request to the daemon, (3) set by the daemon in its own environment, and (4) retrieved by IBM DB2/ UDB's client connection code from the daemon's environment. The database connection is maintained by a thread in the daemon until the application calls SymDbDisconnect() or StorDbDisconnect(). The autostart mechanism requires all the applications to set the appropriate environment variables required to communicate to the database.

The daemon started automatically by the API SymDbConnect() is associated with the database instance specified by DB2INSTANCE. Applications which have the same setting for DB2INSTANCE are serviced by the daemon so started.

For example, if the user has DB2INSTANCE set to `db2v7in1` then a daemon named `storudbddb2v7in1` is started.

Applications that have the same settings for DB2INSTANCE will use the daemon `storudbddb2v7in1`. Applications that have different settings for the environment variables will start a different daemon. The DB2INSTANCE variable determines the daemon that would service the application.

storwatchd

The `storwatchd` daemon is used to monitor the core Solutions Enabler daemons (`storapid`, `storrdfd`, `storgnsd`, `storevntd`) and automatically restart them if they crash.

This watchdog mechanism is enabled by default. Detailed instructions on how to disable it can be found in the installed `README.daemon_options` file.

The `storwatchd` daemon is only used on UNIX. On Windows, this functionality is provided by the Service Control Manager instead.

Normally, users should have no reason to interact directly with `storwatchd`. It is automatically started when its services are first required -- for example, when a daemon starts that has been configured to make use of it.

Only authorized users are permitted to control (explicitly shutdown) the `storwatchd` daemon.

ARGUMENTS

Users should normally never need to interact directly with `storwatchd`.

The `stordaemon` command can be used to start, stop and monitor the `storwatchd` daemon. Refer to "stordaemon" on page 238 for details.

DIAGNOSTICS

Diagnostic messages are written to the log files `storwatchd.log0` and `storwatchd.log1`, located in the following directory:

```
/var/symapi/log/
```

D

EMC Solutions Enabler ConfigChecker

This appendix describes how to install and use the EMC Solutions Enabler ConfigChecker utility:

Introduction

The EMC Solutions Enabler ConfigChecker utility is a component included with Solutions Enabler that provides the infrastructure and engine for environmental tests and reports. This infrastructure is used by Solutions Enabler, and other applications that require Solutions Enabler, for validating host configuration settings against those recommended for Solutions Enabler. Applications that use the Solutions Enabler ConfigChecker infrastructure do so by providing a configuration checklist that can be executed with the base engine.

For Solutions Enabler, ConfigChecker is used for validating host configuration settings against those recommended or supported by Solutions Enabler.

The utility retrieves information on the following settings, and then generates a report that you can save in XML, HTML, or text:

- Database versions:
  - Oracle
  - UDB
  - Sybase
  - MSSQL_Server
- Symmetrix Enginuity version
- Stordaemon status
- SRM installation
- Veritas Volume Manager version
- Common_Serial_Number (C) flag
- SCSI_3 (SC3) flag
- SCSI_Support1 (OS2007) flag
- Clar Flare version
- Solutions Enabler log grab
- Operating system parameters, including semaphore count

Installing Solutions Enabler ConfigChecker

This section describes how to install Solutions Enabler ConfigChecker.

Solutions Enabler ConfigChecker is an component of the Solutions Enabler runtime kit for the following operating systems:

- Windows 32 and 64 bit
- Windows IA
- Sun Solaris - SPARC
- Sun Solaris-x86
- Linux 32 and 64 bit
- HP-UX/PA-RISC
- HP-UX Itanium

Table 28 lists the locations of the Solutions Enabler ConfigChecker binary and files.

Table 28 Solutions Enabler ConfigChecker binary location

Operating system	Directory
UNIX	`/usr/symcli/cfgchk/`
Windows	`C:\Program Files\EMC\SYMCLI\ConfigChecker`

Installing on Windows

You can install Solutions Enabler ConfigChecker on a Windows host using either an InstallShield wizard, the command line, or a response file.

Note: For more detailed installation procedures, refer to Chapter 2.

Using the InstallShield wizard

To install Solutions Enabler ConfigChecker using the InstallShield wizard:

1. Double click the SE711 RT kit exe image.
2. In the **Setup Type** dialog, select **Custom**.
3. In the **Custom Setup** dialog box, select **ConfigChecker**, and then click **Next**.

4. In the **Ready to Install the Program** dialog, click **Install**.
5. In the **InstallShield Wizard Completed** dialog box, click **Finish**.

Using the command line

To install Solutions Enabler ConfigChecker using the command line (in silent mode):

1. Change directory to the location of the Solutions Enabler kit by entering the following:

   ```
   cd /Install_disc_mount_point/Windows
   ```

2. Run the following command:

   ```
   start /wait SE.exe /S
        /V"ADDLOCAL=CFGCHK_COMPONENT /qn"
   ```

   Where *SE* is the name of the solutions Enabler kit. For example:

   ```
   start /wait se7110-Windows-Processor_type.exe /S
      /V"ADDLOCAL=CFGCHK_COMPONENT /qn"
   ```

Using the response file

To install Solutions Enabler ConfigChecker using a response file:

1. Create a file similar to the following:

   Note: In the following example, the SRMBASE, STORFULL, SYMCLI, STAR, SYMRECOVER and STORBLK components are shown set to their default settings (TRUE). The default settings for these components are modifiable.

   ```
   [COMPONENTSELECTION]
   STAR_PERL_COMPONENT:TRUE
   ORACLE_COMPONENT:TRUE
   SQL_COMPONENT:FALSE
   UDB_COMPONENT:FALSE
   JNI_COMPONENT:FALSE
   STORBLK_COMPONENT:TRUE
   SYMRECOVER_COMPONENT:FALSE
   CFGCHK_COMPONENT:TRUE
   SYMCLI_COMPONENT:TRUE
   STORFULL_COMPONENT:TRUE
   SRMBASE_COMPONENT:TRUE

   [PATHSELECTION]
   EMC_ROOT_PATH="C:\Program Files\EMC\"
   EMC_DATA_ROOT_PATH="C:\Program Files\EMC\SYMAPI\"
   WIDESKY_SDK_KEY="<SDKKEY>"
   ```

2. Run the following command:

```
start /wait SEsetup.exe /S /V"WSC_CONFIG_FILE=
   path_to_the_repsonse_file_with_the_filename /qn"
```

Installing on UNIX

You can install Solutions Enabler ConfigChecker on a UNIX host using either an interactive mode or an increment CLI option.

Note: For more detailed installation procedures, refer to Chapter 2.

Using the interactive mode

To install Solutions Enabler ConfigChecker using the interactive mode:

1. Change directory to the location of the Solutions Enabler kit by entering the following:

```
cd /Install_disc_mount_point/Unix/operating_system
```

2. Run the following command:

```
./se711*_install.sh -install
```

3. At the following prompt, enter [**Y**]es:

```
Install EMC Solutions Enabler Config Checker
   Component ? [N]:
```

Using the increment command line options

To install Solutions Enabler ConfigChecker using the increment CLI, run the following:

```
/se711*_install.sh -increment -cfgchk
```

Using Solutions Enabler ConfigChecker

This section explains how to use the Solutions Enabler ConfigChecker utility.

Before you begin

Before using Solutions Enabler ConfigChecker, you must either automatically or manually discover Symmetrix arrays, or manually configure the SPa or SPb address of each array on which you want to generate a report.

Modifying the checklist file

The default checklist file `checklist_SE.txt` in the installation directory contains all supported tests. To modify this file, enter an input value for *xxx* in the following format:

```
#ifonly platform
SUBSYSTEM="SubsystemName" REQUIRED="yes" DISPLAY=
"message to display about the test"
   cfgchk_template_tests:
Testmethodname Test'sinputValue="xxx"
#endifonly
```

Note: You cannot add new tests in the checklist file; you can only modify the input value *xxx* per your requirements.

Using the optional checklist file

You can use an optional checklist file to perform a subset of the tests included in the full checklist file (described earlier on this page). To do this, copy and modify the template file `checklist_SE.txt`, and then enter:

```
cfgchk -configfile optionalchecklist.txt
```

Enabling/disabling logging

Solutions Enabler ConfigChecker maintains logs in `var/symapi/log` directory.

To enable or disable logging, use the following format:

```
cfgchk -logging enable|disable
```

Examples

The following examples demonstrate some of the more common commands. For full command syntax, refer to the man page `cfgchk.1`.

To view a list of all the command line options, enter:

```
cfgchk -help
```

To execute all testcases per the checklist_SE.txt file, enter

```
cfgchk
```

To generate an XML report named cfgchk.xml, enter:

```
cfgchk -suppress -xml
```

To execute only the semaphore testcase, enter:

```
cfgchk -subsystem SESEMAPHORE
```

To execute Oracle testcases, and set the environment variable ORACLE_HOME and ORACLE_SID for this testcase, enter:

```
cfgchk -orauser username -orapassword password
```

E

UNIX Native Installation Support

This appendix describes how to install Solution Enabler using UNIX PureNative installation kits:

Before you begin

Before you begin to install Solutions Enabler, be sure to complete the tasks listed in this section.

❑ Review the following best practices:

- Backup persistent data and uninstall previous versions of Solutions Enabler before performing major upgrades.
- Use the response file method for mass deployments.
- The automated installers: Kickstart, Jumpstart, and Ignite are recommended.
- To achieve full installation functionality, use the Solutions Enabler installation wrapper script.

❑ For AIX, Linux, and Solaris hosts with GPG installed, import the public key and verify the digital signature:

a. Locate the public key (public_key) and the signature. For example, the digital signature for Linux is:

```
se7110-Linux-i386-ni.rpm.sig
```

b. Import the key, by entering:

```
gpg --import public_key
```

c. Verify the imported key using, by entering:

```
-bash-3.00# gpg --list keys
```

d. Edit the imported key and trust it ultimately, by entering:

```
-bash-3.00# gpg --edit key C4E34013
```

e. Verify the digital signatures, by entering:

```
gpg --verify SigFile
```

Where *SigFile* is the name of the digital signature.

For example, to verify the digital signature for Linux, enter:

```
gpg --verify se7110-Linux-i386-ni.rpm.sig
```

PureNative installation kits

Solutions Enabler PureNative kits are available for the following UNIX platforms:

- AIX
- HP-UX (PA/RISC and IA64)
- HP Tru64 (OSF1)
- Linux (x86, IA64, PPC64, and 390)
- Solaris (SunOS Sparc and SunOS x86)

The kits use the following naming convention:

`se`*`MmPp-OS-ARCH`*`-ni.tar.z`

Where:

`M` = Major version

`m` = Minor version

`P` = Point

`p` = Patch

`OS` = Operating System

`ARCH` = Processor architecture

For example:

```
se7110-SunOS-sparc-ni.tar.Z
```

Table 29 on page 292 lists the kit components by operating system.

Note: In Table 29 on page 292:
N/A indicates that the component is not supported in the corresponding operating system.

Components within shaded rows are required.

Table 29 Solutions Enabler PureNative kit contents

OS-specific component names					Description
AIX	HP-UX	Linux	HP Tru64	SunOS	
SYMCLI.CORE.rte	SYMCLI.CORE	symcli-core	SYMCLICORE711	SYMcore	Installs Solutions Enabler core functionality, including symapi, symlvm, storapi, symapisrv, storapid, storcore, stordaemon, and storpds. This option is part of the shared library and runtime environment. It ia a corerequiste for other options, and is therefore mandatory for a successful installation.
SYMCLI. DATACORE.rte	SYMCLI.DATACORE	symcli-datacore	SYMCLIDATACORE711	SYMdcore	Installs persistent data files.
SYMCLI. DATASTORBASE.rte	SYMCLI. DATASTORBASE	symcli-datastorbase	SYMCLIDATASTORBASE 711	SYMdsbase	Installs the base storage library.
N/A	N/A	symcli-smi	N/A	N/A	Installs the SMI Provider.
SYMCLI. SRMBASE.rte	SYMCLI.SRMBASE	symcli-srmbase	SYMCLISRMBASE711	SYMsrmbse	Installs Storage Resource Management base mapping library. This option is part of the shared library and runtime environment.
SYMCLI. STAR_PERL.rte	SYMCLI.STAR_PERL	symcli-star_perl	SYMCLISTARPERL711	SYMstarp	Installs the Solutions Enabler Star component.
SYMCLI. STORBASE.rte	SYMCLI.STORBASE	symcli-storbase	SYMCLISTORBASE711	SYMstrbse	Installs shared libraries and runtime environment - Base Storage Library component.
SYMCLI. STORFULL.rte	SYMCLI.STORFULL	symcli-storfull	SYMCLISTORFULL711	SYMstrful	Installs shared libraries and runtime environment - Control Storage Library component.
SYMCLI. SYMCLI.rte	SYMCLI.SYMCLI	symcli-symcli	SYMCLISYMCLI711	SYMsymcli	Installs the collection of binaries known as Symmetrix Command Line Interface (SYMCLI).
SYMCLI. SYMRECOVER.rte	SYMCLI. SYMRECOVER	symcli-symrecover	SYMCLISYMRECOVER 711	SYMsymrec	Installs the SRDF session recovery component.

Table 29 Solutions Enabler PureNative kit contents

OS-specific component names					Description
AIX	HP-UX	Linux	HP Tru64	SunOS	
N/A	SYMCLI.CFGCHK	symcli-cfgchk	N/A	SYMcfgchk	Installs the Solutions Enabler ConfigChecker. This is a utility for validating your host environment configuration settings against those recommended for Solutions Enabler. For more information on Solutions Enabler ConfigChecker, refer to Appendix D.
SYMCLI.JNI.rte	SYMCLI.JNI	symcli-jni	N/A	SYMjni	Installs the Solutions Enabler Java Interface component.You should install this component if your Solutions Enabler application uses a Java interface.
SYMCLI. ORACLE.rte	SYMCLI.ORACLE	symcli-oracle	SYMCLIORACLE711	SYMoracle	Installs the Oracle daemon.
N/A	N/A	symcli-srmfull	N/A	N/A	Installs the shared libraries and runtime environment - base mapping component.
SYMCLI. SYBASE.rte	SYMCLI.SYBASE	symcli-sybase	N/A	SYMsybase	Installs the SRM SYBASE database runtime component.
SYMCLI.UDB.rte	SYMCLI.UDB	symcli-udb	N/A	SYMudb	Installs the SRM IBM UDB database runtime component.
SYMCLI.64BIT.rte	SYMCLI.64BIT	symcli-64bit[a]	N/A	SYM64Bit	Installs the 64-bit libraries.

a. Only for Linux X64.

Installing Solutions Enabler

This section describes how to install Solutions Enabler using native installer commands.

Installing on AIX

To install on an AIX host:

1. Uncompress and untar the installation kit.
2. Do either of the following depending on whether you want to perform a full or customized installation:
   - To perform a full installation, run the following command:

     ```
     installp -ac -d absolute_path_to_SYMCLI*.bff_file
        all
     ```

   - To perform a custom installation and install only specific components, run the following command:

     ```
     installp -a -d absolute_path_to_SYMCLI*.bff_file
        FileSetName
     ```

     Where *FileSetName* is a component name from Table 29 on page 292.
3. Run the following command to verify the component installation:

   ```
   lppchk -f FileSetName
   ```

   A `0` value is returned for a successful installation.
4. Repeat steps 2 and 3 for each component to install.

Installing on HP-UX

You can install Solutions Enabler on a HP-UX host using either a command line option or a response file.

Using the command line

To install on an HP-UX host using the command line:

1. Uncompress and untar the installation kit.
2. From the local file system, run the following commands to start the installation:

```
swreg -l depot AbsolutePathtoSYMCLI.depot

swinstall -s AbsolutePathtoSYMCLI.depot
   FileSetName:InstallPath
```

   Where *FileSetName* is a component name from Table 29 on page 292.

3. Repeat step 2 for each component to install.

Using a response file

To install on an HP-UX host using a response file:

1. Create a response file similar to the following:

```
#cat response_file_bin
SYMCLI.CORE:/opt/emc
SYMCLI.STORBASE:/opt/emc
SYMCLI.SRMBASE:/opt/emc
SYMCLI.STORFULL:/opt/emc
SYMCLI.STAR_PERL:/opt/emc
SYMCLI.SYMRECOVER:/opt/emc
SYMCLI.SYMCLI:/opt/emc
SYMCLI.JNI:/opt/emc
SYMCLI.ORACLE:/opt/emc
SYMCLI.SYBASE:/opt/emc
SYMCLI.UDB:/opt/emc
SYMCLI.64BIT:/opt/emc

#cat response_file_data
SYMCLI.DATACORE:/usr/emc
SYMCLI.DATASTORBASE:/usr/emc
```

2. Run the following command, specifying the location of the installation package and the name of your response file:

```
swinstall -s AbsolutePathtoSYMCLI.depot
   -f ResponseFile
```

Installing on Linux

You can install Solutions Enabler on a Linux host using either RPM, a response file, or Yum.

Using RPM

To install on a Linux host using the command line:

1. Uncompress and untar the installation kit.
2. Run the following command to start the installation:

   ```
   rpm -i se711*-Linux-*.rpm
   ```

3. Run the following command to verify the component installation:

   ```
   rpm -qa | grep symcli
   ```

Using a response file

To install on a Linux host using a response file:

1. Create a response file similar to the following in `/usr/temp/emc_se_linux_response_file`:

   ```
   -bash-2.05b# # cat emc_se_linux_response_file
   EMC_APPLICATION_PATH:/opt/emc
   EMC_VAR_PATH:/usr/emc
   ADDITIONAL_COMPONENTS:jni oracle sybase udb cfgchk
   ```

2. Run the following command to start the installation:

   ```
   rpm -i se711*-Linux-*.rpm
   ```

3. Run the following command to verify the installation:

   ```
   rpm -qa | grep symcli
   ```

Using Yum

To install on a Linux host using Yum:

1. Run the following command to create a directory for the Solutions Enabler repository:

   ```
   mkdir /symapi.repo
   ```

2. Run the following command to extract the Solutions Enabler rpms into the repository directory:

   ```
   cd /symapi.repo
   ```

3. Depending on whether the kit is in the form of a tar ball or an RPM, run the following command to extract all files into the Solutions Enabler repository:

   - If in a tar ball, run:

     ```
     tar -xvf se711*-Linux-*.tar
     ```

- If in an RPM, run:

```
rpm2cpio se711*-Linux-*.rpm | cpio -id
   mv kit_arch_dir/*.rpm current_working_dir
   rm -rf kit_arch_dir
```

4. Verify that the rpm files (components) and an XML file are extracted into the `/symapi.repo` directory. For file names and descriptions, refer to Table 29 on page 292.
5. Run the following command to create Yum Solutions Enabler repository:

```
createrepo -g symapi.xml /symapi.repo
```

6. Run the following command to add the Solutions Enabler repository into the Yum repositories:

```
cat > /etc/yum.repos.d/symapi.repo << EOF
[symapi]
baseurl=file:///symapi.repo
enabled=1
gpgcheck=0
EOF
```

7. Run the following command to start the installation:

```
yum groupinstall SYMAPI -y
```

Installing on HP Tru64 (OSF1)

To install on a HP Tru64 host:

1. Uncompress and untar the installation kit.
2. From the local file system, run the following command to start the installation:

```
setld -l absolute_path_to_OSF1*NIKIT SubsetName
```

   Where *SubsetName* is a component name from Table 29 on page 292.
3. Run the following command to verify the component installation:

```
setld -v SubsetName
```

4. Repeat steps 2 and 3 for each component to install.

Installing on Solaris

You can install Solutions Enabler on a Solaris host using either a command line option or a response file.

Using the command line

To install on a Solaris host using the command line:

1. Uncompress and untar the installation kit.
2. Run the following command to view a list of packages:

   ```
   pkgadd -d .
   ```

3. Run the following, depending on whether you want to start an interactive or silent installation:

   Interactive: `pkgadd -d . PkgName`

   Silent: `pkgadd -n -d. -a Full_path_to_ADMINFile -r ResponseFile PkgName`

   Where *ResponseFile* is the name of your response file and *PkgName* is a component name from Table 29 on page 292.

   Install the components in the following order:

   SYMdcore
   SYMdsbase
   SYMcore
   SYMstrbse
   SYMsrmbse
   SYMstrful
   SYMstarp
   SYMsymrec
   SYMsymcli
   SYMjni
   SYMoracle
   SYMsybase
   SYMudb
   SYM64bit

4. Run the following command to verify the installation:

   `pkgchk -f PkgName`

   A `0` value is returned for a successful installation.

5. Repeat steps 3 and 4 for each component to install.

Using a response file

To install on Solaris host using a response file:

1. Uncompress and untar the installation kit.
2. Create a response file similar to the following:

```
-bash-2.05b# cat response_file_bin
BASEDIR=/opt/emc
-bash-2.05b# cat response_file_data
BASEDIR=/usr/emc
```

3. Customize the following admin file:

```
#cat admin_file
mail=
basedir=default
runlevel=quit
conflict=nocheck
setuid=nocheck
action=nocheck
partial=nocheck
instance=overwrite
idepend=quit
rdepend=quit
space=quit
```

4. Run the following command to start the installation:

```
pkgadd -n -d . -a Full_path_to_ADMINFile -r
   ResponseFile PkgName
```

   Where *ResponseFile* is the name of your response file and *PkgName* is a component name from Table 29 on page 292.

5. Install the components in the following order:

   SYMdcore
   SYMdsbase
   SYMcore
   SYMstrbse
   SYMsrmbse
   SYMstrful
   SYMstarp
   SYMsymrec
   SYMsymcli
   SYMjni
   SYMoracle
   SYMsybase
   SYMudb
   SYM64bit

Note: For component descriptions, refer to Table 29 on page 292.

6. Run the following command to verify the installation:

   `pkginfo`

7. Repeat steps 2 through 6 for each component to install.

Uninstalling Solutions Enabler

This section describes how to uninstall Solutions Enabler using native installer commands.

Uninstalling from AIX

To uninstall from an AIX host, run the following command:

```
installp -u FileSetName
```

Where *FileSetName* is a component name from Table 29 on page 292.

Uninstalling from HP-UX

To uninstall from an HP-UX host, run the following command:

```
swremove FileSetName
```

Where *FileSetName* is a component name from Table 29 on page 292.

Uninstalling from Linux

To uninstall from an Linux host, run the following command:

```
rpm -e `rpm -qa |grep -i symcli`
```

Uninstalling from HP Tru64 (OSF1)

To uninstall from an HP Tru64 host, run the following command:

```
setld -d SubsetName
```

Where *SubsetName* is a component name from Table 29 on page 292.

Uninstalling from Solaris

To uninstall from an Solaris host, run the following, depending on whether you want to start a interactive or silent uninstall:

Interactive: `pkgrm PkgName`

Silent: `pkgrm -n -a Full_path_to_ADMINFile PkgName`

Where *PkgName* is a component name from Table 29 on page 292.

F

Host Issues

This section describes the issues in running Solutions Enabler on various hardware platforms. You will find additional information in the Release Notes, which are distributed in hard copy with the Solutions Enabler kits.

The information in this section is organized by hardware platform and operating system:

General issues

This section describes issues that apply to all supported platforms.

Host system semaphores

In UNIX[1] and Linux environments, SYMAPI uses semaphores to serialize access to the gatekeeper devices, to its host configuration database file, to the SYMAPI server, and to allow multiple sessions to concurrently run within one process. You or the System Administrator may need to optimize the host system semaphore parameter settings. When optimizing the semaphore parameters, the following values are recommended:

- `semmni` — Specifies the number of semaphore identifiers for the host. Solutions Enabler requires one identifier for each gatekeeper, and one for each SYMAPI database. The minimum recommended value for this parameter is 256.
- `semmns` — Specifies the number of semaphores for the host. Solutions Enabler requires one semaphore for each gatekeeper, and one for each SYMAPI database. The minimum recommended value for this parameter is 256.
- `semmnu` — Specifies the number of undo structures for the host. Solutions Enabler requires one undo structure for each gatekeeper, and one for each SYMAPI database. The minimum recommended value for this parameter is 256.
- `semume` — Specifies the number of undo structures per process. The minimum recommended value for this parameter is 256.

RDF daemon thread requirements

The RDF daemon allocates threads based on the number of locally attached Symmextrix arrays visible to its host. On some host operating system configurations the default number of threads allowed per process may not be enough to accommodate the RDF daemon's requirements. Although the exact number of threads needed for a given daemon cannot be exactly predicted, a rule of thumb is to allow 16 threads per locally attached Symmextrix array.

1. Solaris 10 does not use semaphores.

HP-UX-specific issues

This section describes the HP-UX system issues concerned with compatibility with the SYMCLI/SYMAPI database file, gatekeeper, and BCV device requirements.

Creating pseudo-devices for gatekeepers and BCVs

If the device you want to use as a gatekeeper or BCV device is accessed through the HP-PB (NIO) SCSI bus controller and you want the device to be visible to your host, you must create a pseudo-device for that device. (A pseudo-device is necessary for every device you want visible to the host.)

Note: Your HP-UX operating system may require a patch to support the HP-PB (NIO) SCSI board. Patches for the HP-PB SCSI Pass-Thru driver (spt0) are available for HP-UX V11.0 and higher from HP on an Extension Media CD. Consult your HP representative about spt drivers for your specific system.

Note: If your HP system is configured with an HSC fast-wide differential SCSI interface board and a device accessed through the HSC SCSI bus is available, you can specify the gatekeeper devices through the procedure outlined in the *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*.

To create pseudo-devices and specify devices as gatekeepers and BCV devices:

1. Execute the `ioscan` command and find the full pathnames of the gatekeeper and BCV devices.

   For example, the full pathname of the Symmetrix volume designated to be the gatekeeper is `/dev/rdsk/c1t2d1`.

2. Enter the `lsdev` command and note the output. For example:

```
lsdev -d spt0
Character     Block   Driver     Class
    75          -1     spt0       spt
```

Note: The wide SCSI Pass-Thru driver is identified as `spt0`**.** If there is no output in response to this command, the `spt0` driver is missing. Install the proper driver before proceeding.

Note: There is also an `spt` driver. The `spt` driver will not work in this environment.

3. Create the device node for the gatekeeper device.

Note: This step creates a pseudo-device that is incapable of functioning like a normal device. It can only be used as a gatekeeper device or to process TimeFinder control functions directed to a BCV device.

For example, to create the device node:

```
mknod /dev/rdsk/pseudo_c1t2d1 c 75 0x012100
```

where:

`/dev/rdsk/pseudo_c1t2d1` is the full pathname of the pseudo-device associated with `/dev/rdsk/c1t2d1`.

`c` specifies character (raw) device node creation.

`75` is the character value from the output of the `lsdev` command. This is the major number of the device file.

`0x012100` is the minor number of the device file. The individual values of the minor number are:

`0x` indicates that the number is hexadecimal.

`01` is the hexadecimal number of the controller referenced by `/dev/rdsk/c1t2d1`

`2` is the hexadecimal number of the target ID referenced by `/dev/rdsk/c1t2d1`

`1` is the hexadecimal number of the LUN referenced by `/dev/rdsk/c1t2d1`

`00` must be the last two digits of the minor number.

4. Repeat step 3 for all BCV devices and alternate gatekeeper devices.

CAUTION

Do not perform I/O through the device (`/dev/rdsk/cxtxdx`) associated with the pseudo-device, nor use the pseudo-device as a normal device. If you do, you have two paths to the same device from two different device drivers. Unknown results may occur.

5. To create the mapping information of standard devices to pseudo-devices, create the file:

   `/var/symapi/config/pseudo_devices`

   For each gatekeeper and BCV device, add a mapping to a pseudo-device. For example, in the `pseudo_devices` file, add the following line to map the pseudo-device filename (in **bold**), to the Symmetrix device file:

   `/dev/rdsk/c1t0d0` **`/dev/rdsk/pseudo_c1t0d0`**

   SYMAPI will then use this pseudo-device instead of the physical device file name.

   When the SymDiscover() function is used, the pseudo-device mappings get posted in the log file (`/var/symapi/log/symapi*.log`).

swverify command not supported

The native UNIX command `swverify` is not supported in this release of Solutions Enabler and will fail with the following error:

```
#swverify SYMCLI:/opt/emc

=======  03/02/09 20:34:54 IST  BEGIN verify AGENT SESSION (pid=4279)
         (jobid=spea20-4824)

       * Agent session started for user "root@spea20.lss.emc.com".
         (pid=4279)

       * Beginning Analysis Phase.
       * Target:           spea20:/
       * Target logfile:   spea20:/var/adm/sw/swagent.log
       * Reading source for file information.
       *     Configured    SYMCLI.64BIT,l=/opt/emc,r=T7.1.1.365
       *     Configured    SYMCLI.CFGCHK,l=/opt/emc,r=T7.1.1.365
```

```
          *     Configured    SYMCLI.CORE,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.INFORMIX,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.ORACLE,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.SRMBASE,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.STAR_PERL,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.STORBASE,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.STORFULL,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.SYBASE,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.SYMCLI,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.SYMRECOVER,l=/opt/emc,r=T7.1.1.365
          *     Configured    SYMCLI.UDB,l=/opt/emc,r=T7.1.1.365
ERROR:    File "/opt/emc/usr/lib/liboslevtd64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libstorapi64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libstorbase64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libstorcore64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libstorctrl64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libstormap64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libstorpds64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libstorsil64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libsymapi64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libsymevtd64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libsymlvm64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/liboslevtd64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libstorapi64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libstorbase64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libstorcore64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libstorctrl64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libstormap64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libstorpds64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libstorsil64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libsymapi64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libsymevtd64mt.sl" missing.
ERROR:    File "/opt/emc/usr/lib/pa20_64/libsymlvm64mt.sl" missing.
ERROR:    Fileset "SYMCLI.64BIT,l=/opt/emc,r=T7.1.1.365" had file
          errors.
ERROR:    File "/opt/emc/usr/lib/libcfgchk.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libcfgchk_template_tests.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libcfgchk_tests.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libext_stl_cfgchk.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libext_thread_cfgchk.sl" missing.
ERROR:    File "/opt/emc/usr/lib/liblocale_cfgchk.sl" missing.
ERROR:    File "/opt/emc/usr/lib/liblogging_cfgchk.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libprocess_cfgchk.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libutil_cfgchk.sl" missing.
ERROR:    Fileset "SYMCLI.CFGCHK,l=/opt/emc,r=T7.1.1.365" had file
          errors.
ERROR:    File "/opt/emc/usr/lib/libbtp.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libbtptodata.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libemc_crypto.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libemc_crypto.sl.0.9.8" missing.
ERROR:    File "/opt/emc/usr/lib/libemc_crypto64.sl" missing.
ERROR:    File "/opt/emc/usr/lib/libemc_crypto64.sl.0.9.8" missing.
```

```
ERROR:   File "/opt/emc/usr/lib/libemc_ssl.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libemc_ssl.sl.0.9.8" missing.
ERROR:   File "/opt/emc/usr/lib/libemc_ssl64.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libemc_ssl64.sl.0.9.8" missing.
ERROR:   File "/opt/emc/usr/lib/liboslevtdmt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstorapimt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstorcoremt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstorpdsmt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstptobtp.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libsymapimt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libsymevtdmt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libsymlvmmt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/pa20_64/libemc_crypto64.sl" missing.
ERROR:   File "/opt/emc/usr/lib/pa20_64/libemc_crypto64.sl.0.9.8"
         missing.
ERROR:   File "/opt/emc/usr/lib/pa20_64/libemc_ssl64.sl" missing.
ERROR:   File "/opt/emc/usr/lib/pa20_64/libemc_ssl64.sl.0.9.8" missing.
ERROR:   Fileset "SYMCLI.CORE,l=/opt/emc,r=T7.1.1.365" had file errors.
ERROR:   File "/opt/emc/usr/lib/libsapacosprep_emc.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstormapmt.sl" missing.
ERROR:   Fileset "SYMCLI.SRMBASE,l=/opt/emc,r=T7.1.1.365" had file
         errors.
ERROR:   File "/opt/emc/SYMCLI/T7.1.1/PERL/perl.zip" missing.
ERROR:   File "/opt/emc/SYMCLI/T7.1.1/PERL/unzip" missing.
ERROR:   Fileset "SYMCLI.STAR_PERL,l=/opt/emc,r=T7.1.1.365" had file
         errors.
ERROR:   File "/opt/emc/usr/lib/libEmcpegclient.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libEmcpegcommon.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libEmcpegexportclient.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libEmcpegexportserver.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libEmcpeglistener.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libEmcpegslp_client.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libclarevtdmt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstorbasemt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstorfilcimmt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstorsilcimmt.sl" missing.
ERROR:   File "/opt/emc/usr/lib/libstorsilmt.sl" missing.
ERROR:   Fileset "SYMCLI.STORBASE,l=/opt/emc,r=T7.1.1.365" had file
         errors.
ERROR:   File "/opt/emc/usr/lib/libstorctrlmt.sl" missing.
ERROR:   Fileset "SYMCLI.STORFULL,l=/opt/emc,r=T7.1.1.365" had file
         errors.

       * Summary of Analysis Phase:
ERROR:       Verify failed SYMCLI.64BIT,l=/opt/emc,r=T7.1.1.365
ERROR:       Verify failed SYMCLI.CFGCHK,l=/opt/emc,r=T7.1.1.365
ERROR:       Verify failed SYMCLI.CORE,l=/opt/emc,r=T7.1.1.365
             Verified      SYMCLI.INFORMIX,l=/opt/emc,r=T7.1.1.365
             Verified      SYMCLI.ORACLE,l=/opt/emc,r=T7.1.1.365
ERROR:       Verify failed SYMCLI.SRMBASE,l=/opt/emc,r=T7.1.1.365
ERROR:       Verify failed SYMCLI.STAR_PERL,l=/opt/emc,r=T7.1.1.365
ERROR:       Verify failed SYMCLI.STORBASE,l=/opt/emc,r=T7.1.1.365
```

```
ERROR:       Verify failed SYMCLI.STORFULL,l=/opt/emc,r=T7.1.1.365
             Verified      SYMCLI.SYBASE,l=/opt/emc,r=T7.1.1.365
             Verified      SYMCLI.SYMCLI,l=/opt/emc,r=T7.1.1.365
             Verified      SYMCLI.SYMRECOVER,l=/opt/emc,r=T7.1.1.365
             Verified      SYMCLI.UDB,l=/opt/emc,r=T7.1.1.365
ERROR:   7 of 13 filesets had Errors.
       * 6 of 13 filesets had no Errors or Warnings.
ERROR:   The Analysis Phase had errors.  See the above output for
         details.


=======  03/02/09 20:35:09 IST  END verify AGENT SESSION (pid=4279)
         (jobid=spea20-4824)
```

HP UNIX-specific issues

For HP Tru64 UNIX Clusters (utilizing clustered filesystems), the executable images can be shared but the Solutions Enabler database, log, and config files must be unique for each node in the cluster. In this environment, each node in the cluster has local needs.

An example post-install script provided in this section should be run as a post installation step of the EMC Solutions Enabler software. It will determine if the node executing the script is part of a Cluster. The system administrator should take the defaults, during the installation, and then execute this script afterward or just execute the command interactively if they know this is a Tru64 Cluster environment.

For each node, the script creates a private area in the cluster for the files that are unique for Solutions Enabler.

Use the following Compaq command (`mkcdsl`) to invoke the script:

```
/usr/sbin/mkcdsl -a -f /var/symapi
```

where:

`-a`
Specifies all members.

`-f`

Forces the overwriting of the existing CDSL or member-specific file or directory. When the force option is used with a copy option, `mkcdsl` will overwrite an existing member-specific file or directory. Without the force (`-f`) option, `mkcdsl` issues an error or message whenever the physical path of the target differs from the specified targetname (for example, when targetname resolution traverses a symbolic link), or when the source for a specified copy option cannot be found. Unless the `-f` option is specified, `mkcdsl` will exit when it encounters a situation that would generate an error message. The `mkcdsl` command issues a warning message if the specified sourcename differs from the calculated sourcename. However, you do not need the `-f` option to stop `mkcdsl` from exiting when it encounters a situation that generates a warning message.

For example:

```
#!/usr/bin/ksh

/sbin/sysconfig -q generic clu_active_member >/dev/null
   2>&1

cluster_member=$?

if [[ $cluster_member -eq 0 ]]
  then
    /usr/sbin/mkcdsl -f -a -c /var/symapi
fi
```

HP OpenVMS-specific issues

Starting with Solutions Enabler V7.1, the default client/server communication security level is SECURE (on platforms that will support it). This can cause communication failures between OpenVMS hosts and non OpenVMS hosts since OpenVMS does not support secure communication. To workaround this, you must change the security level on the non OpenVMS host to ANY. For instructions, refer to "Securing remote transmissions using SSL" on page 120.

IBM AIX-specific issues

This section describes the IBM AIX system issues concerned with Oracle database mapping and rebooting a system.

Oracle database mapping

Oracle 8 database mapping with SYMCLI is supported on 32-bit AIX V4.3 and above.

You may need to create the Oracle library, `libclntsh.so`.

To determine if the library exists for Oracle 8, execute the following:

```
ls $ORACLE_HOME/lib/libclntsh.so
```

If the library does not exist, execute the following command:

```
make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk
   client_sharedlib
```

The Oracle 8 OCI executable is linked dynamically. You must set the following environment variable as follows:

```
setenv LIBPATH $ORACLE_HOME/lib
```

BCV devices lost after reboot

When a system comes back up after a reboot, it will not recognize your mapped BCVs. To work around this problem, you should run the following special BCV script (`mkbcv`):

```
cd /
./inq.AIX | more  (look for no gaps in the numbers, ie..
   rhdisk0, rhdisk1, rhdisk3... - rhdisk2 is missing)
cd /usr/lpp/Symmetrix/bin
./mkbcv -a ALL
cd /
./inq.AIX | more  (look for no gaps in the numbers, ie..
   rhdisk0, rhdisk1, rhdisk2... - rhdisk2 is not
   missing)
```

It is recommended to have `./mkbcv -a ALL` in your AIX boot procedures.

Note: `inq.AIX` can be found on the EMC FTP site.

Windows-specific issues

For Windows platforms in a clustered environment, the disk drive (device) in the Symmetrix array that you assign as gatekeeper must be a minimum of 8 MB in size and have a signature.

In a non-clustered environment, gatekeeper devices smaller than 8 MB will show up in the new Disk Manager as devices with no available information. (Disk Manager just displays the disk number and a blank bar.) The devices are still addressable at the SCSI level, and SYMCLI scripts continue to work. (There may be some implications for device naming, since the Windows Device Manager does not create some of the normal device objects for devices smaller than 8 MB).

G

Solutions Enabler Directories

This appendix contains the directory list for UNIX and Windows installations:

UNIX directories

Table 30 lists the directories for UNIX platforms. Your directories may differ from this list since the location of these directories is configurable at installation.

Table 30 UNIX directories (page 1 of 2)

Contents	Directories	Details
Binaries for executables	/usr/storapi/storbin /usr/storapi/bin	STORCLI binaries. SYMCLI binaries.
Shared libraries	/usr/storapi/shlib	All shared libraries.
Database engines	/usr/storapi/shlib/sql/IBMUDB/ /usr/storapi/shlib/sql/ORACLE/ /usr/storapi/shlib/sql/SYBASE/	IBM database engine. Oracle database engine. Sybase database engine.
Language interfaces	/usr/storapi/interfaces/java/ /usr/storapi/interfaces/xml/c_xml/examples/ /usr/storapi/interfaces/xml/c_xml/include /usr/storapi/interfaces/xml/shlib/ /usr/storapi/interfaces/xml/java_xml/ /usr/storapi/interfaces/xml/java_xml/examples/ /usr/storapi/interfaces/xml/schemas /usr/storapi/interfaces/xml/jni/	Java language interface. XML examples for C programmers. XML API header files. XML shared libraries. Jar files for XML. XML examples for Java programmers. XML schema files. JNI examples and Java docs.
SYMCLI manpages	/usr/storapi/storman/man1 /usr/storapi/storman/man3 /usr/storapi/man/man1 /usr/storapi/man/man3	STORCLI manpages. STORAPI manpages. SYMCLI manpages. SYMAPI and CLARAPI manpages.
Daemons	/usr/symcli/daemons/	Location of the daemon executables.
Configuration database file(s)	/var/symapi/db/	Contains the configuration database file(s) for SYMAPI, CLARAPI, and STORAPI.
SYMAPI environment and system files	/var/symapi/config	Includes licenses, avoidance, options, daemon_options, daemon_users, and nethost files. It is recommended that you back up this directory frequently.
SYMAPI certificate files	/var/symapi/config/cert	Contains server and trusted certificate files and support files for certificate creation. Used for client/server security.

Table 30 UNIX directories (page 2 of 2)

Contents	Directories	Details
Security data	/var/symapi/authz_cache	Acts as a cache of authorization data from attached Symmetrix arrays.
XML Properties	/var/symapi/interfaces/xml/java_xml/	XML properties files for SYMAPI and STORAPI.
Log files	/var/symapi/log	Contains SYMAPI logs and daemon logs.

Windows directories

Table 31 lists the default directories for Windows. Your directories may differ from this list since the location of these directories is configurable at installation.

Table 31 Windows directories (page 1 of 2)

Contents	Directories	Details
Binaries for executables	C:\Program Files\EMC\SYMCLI\storbin C:\Program Files\EMC\SYMCLI\bin	STORCLI binaries. SYMCLI binaries.
Shared libraries	C:\Program Files\EMC\SYMCLI\shlib	All shared libraries.
Database engines	C:\Program Files\EMC\SYMCLI\shlib\sql\IBMUDB C:\Program Files\EMC\SYMCLI\shlib\sql\Oracle C:\Program Files\EMC\SYMCLI\shlib\sql\SQLSERVER	IBM database engine. Oracle database engine. SQL server database engine.
Language interfaces	C:\Program Files\EMC\SYMCLI\interfaces\java C:\Program Files\EMC\SYMCLI\interfaces\xml\c_xml\examples C:\Program Files\EMC\SYMCLI\interfaces\xml\c_xml\include C:\Program Files\EMC\SYMCLI\interfaces\c_xml\shlib C:\Program Files\EMC\SYMCLI\interfaces\dotnet_xml\examples C:\Program Files\EMC\SYMCLI\interfaces\java_xml\examples C:\Program Files\EMC\SYMCLI\interfaces\schemas C:\Program Files\EMC\SYMCLI\interfaces\jni	Java language interface. XML examples for C programmers. XML API header files. XML shared libraries. Jar files for XML. XML examples for Java programmers. XML schema files. JNI examples and Java docs.
SYMCLI manpages	C:\Program Files\EMC\SYMCLI\storman\man1 C:\Program Files\EMC\SYMCLI\storman\man3 C:\Program Files\EMC\SYMCLI\man\man1 C:\Program Files\EMC\SYMCLI\man\man3	STORCLI manpages. STORAPI manpages. SYMCLI manpages. SYMAPI and CLARAPI manpages.
Daemons	C:\Program Files\EMC\SYMCLI\daemons	Location of the daemon executables.
Connectivity directories	C:\Program Files\EMC\SYMCLI\conn C:\Program Files\EMC\SYMCLI\conn\bin C:\Program Files\EMC\SYMCLI\conn\data C:\Program Files\EMC\SYMCLI\conn\etc C:\Program Files\EMC\SYMCLI\conn\nt_lib	Connectivity related binaries, data, and database libraries.
Solutions Enabler Config Checker	C:\Program Files\EMC\SYMCLI\ConfigChecker	Solutions Enabler ConfigChecker binary and files
Configuration database file(s)	C:\Program Files\EMC\SYMAPI\db	Contains the configuration database file(s) for SYMAPI, CLARAPI, and STORAPI.

Table 31 Windows directories (page 2 of 2)

Contents	Directories	Details
SYMAPI environment and system files	C:\Program Files\EMC\SYMAPI\config	Includes licenses, avoidance, options, and server network files. It is recommended that you back up this directory frequently.
SYMAPI certificate files	C:\Program Files\EMC\SYMAPI\config\cert	Contains server and trusted certificate files and support files for certificate creation. Used for client/server security.
Security data	C:\Program Files\EMC\SYMAPI\authz_cache	Acts as a cache of authorization data from attached Symmetrix arrays.
XML properties	C:\Program Files\EMC\SYMAPI\interfaces\xml\java_xml	XML properties files for SYMAPI and STORAPI.
SYMAPI log files	C:\Program Files\EMC\SYMAPI\log	Contains log of significant events.
Providers	C:\Program Files\EMC\SYMCLI\shlib	VSS and VDS providers.

OpenVMS directories

Table 32 lists the default directories for OpenVMS. Your directories may differ from this list since the location of these directories is configurable at installation.

Table 32 OpenVMS directories

Contents	Directories	Details
Binaries for executables	SYMCLI$BIN	STORCLI binaries. SYMCLI binaries.
Shared libraries	SYMCLI$SHLIB	All shared libraries.
SYMCLI manpages	SYMCLI$HELP	STORCLI manpages. STORAPI manpages. SYMCLI manpages. SYMAPI and CLARAPI manpages.
Daemons	SYMCLI$DAEMONS	Location of the daemon executables.
Configuration database file(s)	SYMAPI$DB	Contains the configuration database file(s) for SYMAPI, CLARAPI, and STORAPI.
SYMAPI environment and system files	SYMAPI$CONFIG	Includes licenses, avoidance, options, daemon_options, and netcnfg files. It is recommended that you back up this directory frequently.
SYMAPI log files	SYMAPI$LOG	Contains log of significant events.
Security data	SYMAPI$AUTHZ_CACHE	Acts as a cache of authorization data from attached Symmetrix arrays.

z/OS USS directories

Table 33 lists the USS directories for z/OS. Your directories may differ from this list since the location of these directories is configurable at installation.

Table 33 z/OS directories

Contents	Directories	Details
Configuration database file(s)	/var/symapi/db/	Contains the configuration database file(s) for SYMAPI, CLARAPI, and STORAPI.
SYMAPI environment and system files	/var/symapi/config	Includes licenses, avoidance, options, daemon_options, daemon_users, and nethost files. It is recommended that you back up this directory frequently.
Log files	/var/symapi/log	Contains SYMAPI logs and daemon logs.

H

UNIX Installation Log Files

This appendix describes the UNIX log files created by the Solutions Enabler install script:

Understanding the UNIX installer log files

The Solutions Enabler installer script `se7110_install.sh` creates log files in install root directory `/opt/emc/logs`.

Format

The log files are named using the following convention:

`SE_NI_<V M.m.P>_<TimeStamp>.log`

For example:

`SE_NI_V7.1.1_061017_165942.log`

Where:

`SE`	Solutions Enabler
`NI`	Native installation
`V`	Letter portion of version
`M`	Version major
`m`	Version minor
`P`	Version point
`TimeStamp`	File creation time stamp in the format: *yymmdd_hhmmss*

Log file contents

The log files contain the following information:

- Date
- Script name
- User running the script
- Operating system and hardware type
- Script command line options
- Location of native install (NI) kit if the kit is found
- Previous Install root directory
- Previous working root directory
- Install root directory
- Minimum operating system version required
- Existing operating system version in system
- Installed product version
- Current product Version
- Selected components
- Information on active processes (if any)
- Information on active daemons (if any)

- Information on active components
- Package/fileset/rpm being installed/uninstalled
- List of files installed by package/fileset/rpm only during install
- Successful completion of install /uninstall

Note: In addition to the above information, the log files will also contain operating system-specific information useful in trouble shooting native installations.

Example

The following is an example of a SunOS log file:

```
 Date : Wed Mar 31 11:17:03 IST 2010

 UWS_HOME_REL : .
 [Script Name]    : se7110_install.sh
 [User Name]      : root
 [Main] OS Name   : SunOS
 [Main] HW Type   : sun4v
 [Main] Host Name : spea127

 [Command Line arguments] -install
[Check_App_Path_File] EMC_APPLICATION_PATH file is [/tmp/emc_app_path]
 [Check_NI_Sanity] Running sanity check on native install command [pkginfo]
 [Check_NI_Sanity] Sanity check on native install command [pkginfo] successful.
WORK_DIR : /BRTC/tmp/v7.1.1.0-1026/SunOS/RT
UWS_HOME : /BRTC/tmp/v7.1.1.0-1026/SunOS/RT
[Check_NIKIT_Location] NI Kit Component List :
   64BITCFGCHKCOREDATACOREDATASTORBASEJNIORACLESRMBASESTAR_PERLSTORBASESTORFULLS
   YBASESYMCLISYMRECOVERUDB
[Check_NIKIT_Location] Available kit type : RT

 Solutions Enabler Native Installer[RT] Kit Location :
   /BRTC/tmp/v7.1.1.0-1026/SunOS/RT
[Check_NIKIT_Location] NI Kit architecture : sparc
 [NI Kit Version] : V7.1.1.0
 Previous Install root directory : /opt/emc
 Previous Working root directory : /usr/emc
 Install root directory : /opt/emc
 Working root directory : /usr/emc

 [Check_OS_Version] Checking for OS version compatibility

 [Check_OS_Version] Minimum OS Version Required    : 5.8.0
 [Check_OS_Version] Existing OS Version in System  : 5.10.0

 [Check_OS_Version] Completed
```

```
 [Search_Previous_NI_Install] Checking for previous NI deployment of Solutions
   Enabler
 [Search_Previous_NI_Install] IsPrevInstall[] prevInstall[]
 [Search_Previous_NI_Install] Is previous installation available :N
 [Search_Previous_NI_Install] Is previous data available :N

 [Search_Previous_CI_Install] Checking for previous CI deployment of Solutions
   Enabler
 [Search_Previous_CI_Install] Checking for InstFile[/tmp/emc_installed_found]
 [Search_Previous_CI_Install] Completed HAVE_OLD_SE_CI_INSTALL[FALSE]



 Checking for active processes.....
 Checking for active SYMCLI components...
[Select_Components] Selected Components : datacore datastorbase core storbase
   srmbase storfull symcli star_perl symrecover

 [Get_DataDir_Perm] Setting umask to 22..
 [Get_DataDir_Perm] umask set to 22..
0022
  [Backup_Data_Dir]: Data files directory /usr/emc/API does not exist


 Starting installation of Solutions Enabler......

[Install_SE] Set EMC_APPLICATION_PATH file to [/tmp/emc_app_data_path]


 Non interactive installation using pkgadd.
Previous Version : Return Value : 9
 ----------------------------------
 Installing SYMdcore.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
```

```
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/usr/emc


 Checking for active processes.....


 Checking for active SYMCLI components


Installation of <SYMdcore> was successful.

List of files installed by SYMdcore:
SYMdcore: /usr/emc/API/symapi/NI/config
SYMdcore: /usr/emc/API/symapi/NI/config/README.clarcnfg
SYMdcore: /usr/emc/API/symapi/NI/config/README.daemon_options
SYMdcore: /usr/emc/API/symapi/NI/config/README.daemon_users
SYMdcore: /usr/emc/API/symapi/NI/config/README.netcnfg
SYMdcore: /usr/emc/API/symapi/NI/config/README.options
SYMdcore: /usr/emc/API/symapi/NI/config/SymCLI_ML.xsd
SYMdcore: /usr/emc/API/symapi/NI/config/SymCLI_ML_Element.xsd
SYMdcore: /usr/emc/API/symapi/NI/config/cert/ssl.rnd
SYMdcore: /usr/emc/API/symapi/NI/config/cert/symapisrv_install.cnf
SYMdcore: /usr/emc/API/symapi/NI/config/cert/symapisrv_trust.pem
SYMdcore: /usr/emc/API/symapi/NI/config/cert/symapisrv_trustkey.pem
SYMdcore: /usr/emc/API/symapi/NI/config/clarcnfg
SYMdcore: /usr/emc/API/symapi/NI/config/daemon_options
SYMdcore: /usr/emc/API/symapi/NI/config/daemon_users
SYMdcore: /usr/emc/API/symapi/NI/config/netcnfg
SYMdcore: /usr/emc/API/symapi/NI/config/options
SYMdcore: /usr/emc/API/symapi/NI/config/symapinlck
SYMdcore: /usr/emc/API/symapi/NI/config/syscall_cache
[Set_VarSymapi_Link] Previous persistent data directory [ /usr/emc/API/symapi]
  exists.
[Set_VarSymapi_Link] Removing previous /var/symapi softlink.
[Set_VarSymapi_Link] /var/symapi soft link removed successfully.
[Set_VarSymapi_Link] Creating  /var/symapi.........
[Set_VarSymapi_Link] Running command [ln -s -f /usr/emc/API/symapi
  /var/symapi]...
[Set_VarSymapi_Link] /var/symapi soft link created. [/usr/emc/API/symapi]
[Set_VarSymapi_Link] /var/symapi permssion set to 755
[Set_VarSymapi_Link] Setting SET_VARSYMAPI to TRUE
```

```
Previous Version : Return Value : 9
 ----------------------------------
 Installing SYMdsbase.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/usr/emc


 Checking for active processes.....


 Checking for active SYMCLI components


Installation of <SYMdsbase> was successful.

List of files installed by SYMdsbase:
SYMdsbase: /usr/emc/API/symapi/NI/config/key.pem
SYMdsbase: /usr/emc/API/symapi/NI/config/server.pem
SYMdsbase: /usr/emc/API/symapi/NI/config/stpdef.dat
SYMdsbase: /usr/emc/API/symapi/NI/config/trusted.crt
```

```
Previous Version : Return Value : 9
 ----------------------------------
 Installing SYMcore.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/opt/emc


 Checking for active processes.....


 Checking for active SYMCLI components

 Creating  OPENSSL Certificate....
/usr/symcli/bin/manage_server_cert.sh using /usr/storapi/bin/storssl64 to create
   keys
Generating a 1024 bit RSA private key
........++++++
..................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd
   spea127.lss.emc.com/ST=Massachusetts/C=US/L=Hopkinton/emailAddress=support@em
   c.com/O=EMC Corporation/OU=Storage Platform Enablers and Applications
```

```
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the directory .

Installation of <SYMcore> was successful.

List of files installed by SYMcore:
SYMcore: /opt/emc/SYMCLI
SYMcore: /opt/emc/SYMCLI/V7.1.1
SYMcore: /opt/emc/SYMCLI/V7.1.1/PERL
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/CheckProcess.sh
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/STAR
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/SymRecover
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/manage_server_cert.sh
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/stordaemon
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/storssl
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/storssl64
SYMcore: /opt/emc/SYMCLI/V7.1.1/bin/symlmf
SYMcore: /opt/emc/SYMCLI/V7.1.1/cfgchk
SYMcore: /opt/emc/SYMCLI/V7.1.1/daemons
SYMcore: /opt/emc/SYMCLI/V7.1.1/daemons/storsrvd
SYMcore: /opt/emc/SYMCLI/V7.1.1/doc
SYMcore: /opt/emc/SYMCLI/V7.1.1/doc/README.pseudo_devices
SYMcore: /opt/emc/SYMCLI/V7.1.1/doc/instcmpt.csh
SYMcore: /opt/emc/SYMCLI/V7.1.1/doc/instcmpt.pl
SYMcore: /opt/emc/SYMCLI/V7.1.1/install
SYMcore: /opt/emc/SYMCLI/V7.1.1/install/SYMCLI.cfg
SYMcore: /opt/emc/SYMCLI/V7.1.1/install/SYMCLI.eapprem
SYMcore: /opt/emc/SYMCLI/V7.1.1/install/se7110_install.sh
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/java
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/docs
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/docs/Readme
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example1.txt
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example1.xml
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example1.xsl
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example2.txt
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example2.xml
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example2.xsl
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example3.txt
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example3.xml
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example3.xsl
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example4.xsl
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example4a.txt
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/example4b.txt
SYMcore: /opt/emc/SYMCLI/V7.1.1/interfaces/xml/examples/index
SYMcore: /opt/emc/SYMCLI/V7.1.1/man
SYMcore: /opt/emc/SYMCLI/V7.1.1/man/man1
SYMcore: /opt/emc/SYMCLI/V7.1.1/man/man3
SYMcore: /opt/emc/SYMCLI/V7.1.1/man/man3/storsrvd.3
```

```
SYMcore: /opt/emc/SYMCLI/V7.1.1/man/man3/symlmf.3
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/apps
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/apps/SAP
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_crypto.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_crypto.so.0.9.8
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_crypto64.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_crypto64.so.0.9.8
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_ssl.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_ssl.so.0.9.8
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_ssl64.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libemc_ssl64.so.0.9.8
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/liboslevtdmt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libsnmpevtdmt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libstorapimt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libstorcoremt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libstorpdsmt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libsymapimt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libsymevtdmt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/libsymlvmmt.so
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/sql
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/sql/ASM
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/sql/IBMUDB
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/sql/ORACLE
SYMcore: /opt/emc/SYMCLI/V7.1.1/shlib/sql/SYBASE
SYMcore: /opt/emc/SYMCLI/V7.1.1/storbin
SYMcore: /opt/emc/SYMCLI/V7.1.1/storbin/stordaemon
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3/storapid.3
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3/stordaemon.3
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3/storevntd.3
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3/storgnsd.3
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3/storrdfd.3
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3/storstpd.3
SYMcore: /opt/emc/SYMCLI/V7.1.1/storman/man3/storwatchd.3
SYMcore: /usr/lib/libemc_crypto.so
SYMcore: /usr/lib/libemc_crypto.so.0.9.8
SYMcore: /usr/lib/libemc_crypto64.so
SYMcore: /usr/lib/libemc_crypto64.so.0.9.8
SYMcore: /usr/lib/libemc_ssl.so
SYMcore: /usr/lib/libemc_ssl.so.0.9.8
SYMcore: /usr/lib/libemc_ssl64.so
SYMcore: /usr/lib/libemc_ssl64.so.0.9.8
SYMcore: /usr/lib/liboslevtdmt.so
SYMcore: /usr/lib/libsnmpevtdmt.so
SYMcore: /usr/lib/libstorapimt.so
SYMcore: /usr/lib/libstorcoremt.so
SYMcore: /usr/lib/libstorpdsmt.so
SYMcore: /usr/lib/libsymapimt.so
SYMcore: /usr/lib/libsymevtdmt.so
SYMcore: /usr/lib/libsymlvmmt.so
```

```
SYMcore: /usr/storapi
SYMcore: /usr/symapi
SYMcore: /usr/symapi64
SYMcore: /usr/symapi64mt
SYMcore: /usr/symapimt
SYMcore: /usr/symcli
SYMcore: /usr/symcli64
SYMcore: /usr/symcli64mt
SYMcore: /usr/symclimt

[Repair_Symcli_Storapi_Link]  Application Directory is set to
   [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symcli  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symcli64  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symclimt  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symcli64mt  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/storapi  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symapi  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symapi64  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symapimt  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] /usr/symapi64mt  linked to [/opt/emc/SYMCLI/V7.1.1].
[Repair_Symcli_Storapi_Link] Setting SET_USRSYMCLI to TRUE



Previous Version : Return Value : 9
 ----------------------------------
 Installing SYMstrbse.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
```

```
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/opt/emc


 Checking for active processes.....


 Checking for active SYMCLI daemons...


 Checking for active SYMCLI components


Installation of <SYMstrbse> was successful.

List of files installed by SYMstrbse:
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/daemons/storapid
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/daemons/storevntd
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/daemons/storgnsd
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/daemons/storrdfd
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/daemons/storstpd
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/daemons/storstpd64
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/daemons/storwatchd
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libEmcpegclient.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libEmcpegcommon.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libEmcpegexportclient.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libEmcpegexportserver.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libEmcpeglistener.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libEmcpegslp_client.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libclarevtdmt.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libstorbasemt.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libstorfilcimmt.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libstorsilcimmt.so
SYMstrbse: /opt/emc/SYMCLI/V7.1.1/shlib/libstorsilmt.so
SYMstrbse: /usr/lib/libEmcpegclient.so
SYMstrbse: /usr/lib/libEmcpegcommon.so
SYMstrbse: /usr/lib/libEmcpegexportclient.so
SYMstrbse: /usr/lib/libEmcpegexportserver.so
SYMstrbse: /usr/lib/libEmcpeglistener.so
SYMstrbse: /usr/lib/libEmcpegslp_client.so
SYMstrbse: /usr/lib/libclarevtdmt.so
SYMstrbse: /usr/lib/libstorbasemt.so
SYMstrbse: /usr/lib/libstorfilcimmt.so
SYMstrbse: /usr/lib/libstorsilcimmt.so
SYMstrbse: /usr/lib/libstorsilmt.so


Previous Version : Return Value : 9
```

```
 ----------------------------------
 Installing SYMsrmbse.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/opt/emc


 Checking for active processes.....


 Checking for active SYMCLI daemons...


 Checking for active SYMCLI components


Installation of <SYMsrmbse> was successful.

List of files installed by SYMsrmbse:
SYMsrmbse: /opt/emc/SYMCLI/V7.1.1/daemons/storsrmd
SYMsrmbse: /opt/emc/SYMCLI/V7.1.1/shlib/apps/SAP/emcadaptive.ini
SYMsrmbse: /opt/emc/SYMCLI/V7.1.1/shlib/apps/SAP/libsapacosprep_emc.so
SYMsrmbse: /opt/emc/SYMCLI/V7.1.1/shlib/apps/SAP/sapacosprep.emc.1
SYMsrmbse: /opt/emc/SYMCLI/V7.1.1/shlib/libstormapmt.so
SYMsrmbse: /opt/emc/SYMCLI/V7.1.1/shlib/sql/SYBASE/SymSybs_15
```

```
SYMsrmbse: /usr/lib/libsapacosprep_emc.so
SYMsrmbse: /usr/lib/libstormapmt.so


Previous Version : Return Value : 9
 ----------------------------------
 Installing SYMstrful.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/opt/emc


 Checking for active processes.....


 Checking for active SYMCLI daemons...


 Checking for active SYMCLI components


Installation of <SYMstrful> was successful.

List of files installed by SYMstrful:
SYMstrful: /opt/emc/SYMCLI/V7.1.1/bin/symreplicate
```

```
SYMstrful: /opt/emc/SYMCLI/V7.1.1/shlib/libstorctrlmt.so
SYMstrful: /usr/lib/libstorctrlmt.so


Previous Version : Return Value : 9
 ----------------------------------
 Installing SYMsymcli.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/opt/emc


 Checking for active processes.....


 Checking for active SYMCLI daemons...


 Checking for active SYMCLI components

 Enabling stordaemon...


Installation of <SYMsymcli> was successful.
```

```
List of files installed by SYMsymcli:
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symaccess
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symacl
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symapierr
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symaudit
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symauth
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symbcv
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symcfg
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symcg
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symchg
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symchksum
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symcli
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symclone
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symconfigure
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symconnect
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symdev
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symdg
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symdisk
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symdrv
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symevent
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symfast
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symgate
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symhost
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symhostfs
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/syminq
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symioctl
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symipsec
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symlabel
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symld
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symlv
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symmask
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symmaskdb
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symmigrate
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symmigrate64
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symmir
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symoptmz
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/sympart
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/sympd
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symqos
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symrcopy
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symrdb
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symrdf
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symreturn
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symrslv
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symsan
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symsg
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symsnap
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symstat
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symtier
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/bin/symvg
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symaccess.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symacl.1
```

```
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symapierr.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symaudit.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symauth.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symbcv.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symcfg.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symcg.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symchg.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symchksum.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symcli.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symclient.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symclone.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symconfigure.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symconnect.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symdev.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symdg.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symdisk.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symdrv.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symevent.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symfast.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symgate.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symhost.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symhostfs.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/syminq.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symioctl.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symipsec.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symlabel.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symld.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symlv.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symmask.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symmaskdb.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symmigrate.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symmir.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symoptmz.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/sympart.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/sympd.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symqos.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symrcopy.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symrdb.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symrdf.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symreplicate.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symreturn.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symrslv.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symsan.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symsg.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symsnap.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symstar.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symstat.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symtier.1
SYMsymcli: /opt/emc/SYMCLI/V7.1.1/man/man1/symvg.1


Previous Version : Return Value : 9
```

```
 ----------------------------------
 Installing SYMstarp.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/opt/emc


 Checking for active processes.....


 Checking for active SYMCLI daemons...


 Checking for active SYMCLI components

 Unzipping perl.zip installation...

 Completed STAR installation.


Installation of <SYMstarp> was successful.

List of files installed by SYMstarp:
SYMstarp: /opt/emc/SYMCLI/V7.1.1/PERL/perl.zip
SYMstarp: /opt/emc/SYMCLI/V7.1.1/PERL/unzip
```

```
SYMstarp: /opt/emc/SYMCLI/V7.1.1/bin/STAR/Sanity.plx
SYMstarp: /opt/emc/SYMCLI/V7.1.1/bin/symstar


Previous Version : Return Value : 9
 ----------------------------------
 Installing SYMsymrec.....
 ----------------------------------

Customized adminfile for non-interactive installation

#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident"@(#)default1.704/12/21 SMI"
#
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default

Response file for non-interactive installation

BASEDIR=/opt/emc


 Checking for active processes.....


 Checking for active SYMCLI daemons...


 Checking for active SYMCLI components


Installation of <SYMsymrec> was successful.

List of files installed by SYMsymrec:
SYMsymrec: /opt/emc/SYMCLI/V7.1.1/bin/SymRecover/symrecover.pl
```

```
SYMsymrec: /opt/emc/SYMCLI/V7.1.1/bin/SymRecover/symrecover_FUNCT.pm
SYMsymrec: /opt/emc/SYMCLI/V7.1.1/bin/SymRecover/symrecover_GLOBAL.pm
SYMsymrec: /opt/emc/SYMCLI/V7.1.1/bin/SymRecover/symrecover_MAIN.pm
SYMsymrec: /opt/emc/SYMCLI/V7.1.1/bin/symrecover
SYMsymrec: /opt/emc/SYMCLI/V7.1.1/man/man1/symrecover.1


[Setup_Var_Symapi_Dir] File [/usr/EMC/API/symapi/config/cert/ssl.rnd] found
[Setup_Var_Symapi_Dir] File [/usr/EMC/API/symapi/config/cert/symapisrv_cert.pem]
   found
[Setup_Var_Symapi_Dir] File
   [/usr/EMC/API/symapi/config/cert/symapisrv_install.cnf] found
[Setup_Var_Symapi_Dir] File [/usr/EMC/API/symapi/config/cert/symapisrv_key.pem]
   found
[Setup_Var_Symapi_Dir] File
   [/usr/EMC/API/symapi/config/cert/symapisrv_trust.pem] found
[Set_VarSymapi_Link] Previous persistent data directory [ /usr/EMC/API/symapi]
   exists.
[Set_VarSymapi_Link] Removing previous /var/symapi softlink.
[Set_VarSymapi_Link] /var/symapi soft link removed successfully.
Creating  /var/symapi.........
[Set_VarSymapi_Link] Running command [ln -s -f /usr/EMC/API/symapi
   /var/symapi]...
[Set_VarSymapi_Link] /var/symapi soft link created. [/usr/EMC/API/symapi]
[Set_VarSymapi_Link] /var/symapi permssion set to 755
[Set_VarSymapi_Link] Setting SET_VARSYMAPI to TRUE
 Enabling stordaemon...
[Postinstallation] Set EMC_APPLICATION_PATH file to [/tmp/emc_app_data_path]
Update emc_installed_found[/tmp/emc_installed_found] file.
Update emc_installed_found[/opt/emc/SYMCLI/V7.1.1/install/emc_installed_found]
   file.
Update emc_installed_found[/opt/emc/emc_installed_found] file.
[Postinstall] SET_VARSYMAPI is TRUE
[Postinstall] SET_USRSYMCLI is TRUE
 Enabling stordaemon...
  [Set_Daemon_UID] daemonUID not set.
  [Set_Daemon_UID] Running command [/usr/symcli/bin/stordaemon setuser all -user
   root -v ] .....
  /var/symapi/authz_cache                              Creating Directory

Processing daemon storgnsd
  /var/symapi/gns                                      Creating Directory

Processing daemon storevntd
  /var/symapi/ldb/storevntd                            Creating Directory
  /var/symapi/ldb/storevntd#lock                       Creating File
  /var/symapi/events                                   Creating Directory

Processing daemon storwatchd

Processing daemon storsrvd
```

```
  [Set_Daemon_UID] command [/usr/symcli/bin/stordaemon setuser all -user root -v
   ] ran successfully.
  [Set_Daemon_UID] command [/usr/symcli/bin/stordaemon setuser all -user root -v
   ] return value [0].
  Waiting for daemon to start.  This may take several seconds.
[/opt/emc/SYMCLI/V7.1.1/shlib] exists
Library[libstorpdsmt.so] exists
Testing FullPathLibrary[/opt/emc/SYMCLI/V7.1.1/shlib/libstorpdsmt.so]
[/opt/emc/SYMCLI/V7.1.1/shlib/libstorpdsmt.so] is Type[RT]
 isCFGCHK : FALSE



 Do not forget to run 'symcfg discover' after the installation
 completes and  whenever your configuration changes.

 You may need to manually rediscover remotely connected
 arrays. Please see the installation notes for further
 information.


 #------------------------------------------------------------------------------
 # The following HAS BEEN INSTALLED in /opt/emc via the pkgadd utility.
 #------------------------------------------------------------------------------
  ITEM  PRODUCT                                          VERSION
  01    EMC Solutions Enabler                            V7.1.1.0
        RT KIT
 #------------------------------------------------------------------------------
```

Legal Notices

This appendix contains legal attribution for acknowledging open-source and third-party software copyright, and licensing requirements for the EMC Solutions Enabler V6.5.

OpenSSL copyright information

Redistribution and use in source and binary forms, with or withoutmodification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyrightnotice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.

If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.

This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:

   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Perl licensing information

Solutions Enabler uses Perl and Perl extensions software.

The standard version of code is located at:

http://www.perl.com/pub/a/language/info/software.html

For license information, refer to:

http://dev.perl.org/licenses/artistic.html

XML:: Parser licensing information

Solutions Enabler uses software from the XML Parser and an extension to Perl.

For further information, refer to:

http://search.cpan.org/src/MSERGEANT/XML-Parser-2.34/README

Expat Parser licensing information

Solutions Enabler uses software from the Expat XML Parser as part of the XML::Parser.

For further information, refer to

http://www.libexpat.org/

Info-ZIP licensing information

Solutions Enabler uses Info-ZIP.

For further information, refer to:

ftp://ftp.info-zip.org/pub/infozip/license.html

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

ncFTP licensing information

Solutions Enabler uses software from ncFTP Software, Inc.

For further information about the product and specific instructions on downloading ncFTP for your own purposes, refer to:

http://www.nfctp.com

The Clarified Artistic License

Preamble:

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Distribution fee" is a fee you charge for providing a copy of this Package to another party.

"Freely Available" means that no fee is charged for the right to use the item, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain, or those made Freely Available, or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

   a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major network archive site allowing unrestricted access to them, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

   b. use the modified Package only within your corporation or organization.

   c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

   d. make other distribution arrangements with the Copyright Holder.

   e. permit and encourage anyone who receives a copy of the modified Package permission to make your modifications Freely Available in some specific way.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

   a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

   b. accompany the distribution with the machine-readable source of the Package with your modifications.

   c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

   d. make other distribution arrangements with the Copyright Holder.

e. offer the machine-readable source of the Package, with your modifications, by mail order.

5. You may charge a distribution fee for any distribution of this Package. If you offer support for this Package, you may charge any fee you choose for that support. You may not charge a license fee for the right to use this Package itself. You may distribute this Package in aggregate with other (possibly commercial and possibly nonfree) programs as part of a larger (possibly commercial and possibly nonfree) software distribution, and charge license fees for other parts of that software distribution, provided that you do not advertise this Package as a product of your own. If the Package includes an interpreter, You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of the Standard Version of the Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE

Index

Symbols

A

C

D

E

F

G

T

U

V

W

Z