StorageTek Virtual Storage Manager System

VSM console Planning Guide

Release 1

E59665-04

January 2019

ORACLE

StorageTek Virtual Storage Manager System VSM console Planning Guide, Release 1

E59665-04

Contents

5 oVTCS Operational Considerations

6 ELS Facilities for VSM console

7 Controlling Contaminants

List of Figures

List of Tables

Preface

This publication is intended for customers responsible for doing site planning for Oracle's StorageTek Virtual Storage Manager console (VSM console).

> **Note:** The VSM console server is co-located with a VSM 6 or VSM 7 in a Sun Rack II Model 1242. For site planning information, see the appropriate *VSM Planning Guide*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1

What is StorageTek VSM console?

Oracle's StorageTek Virtual Storage Manager console is a hardware and software appliance made available as an additional feature for the VSM 6 or VSM 7 solution.

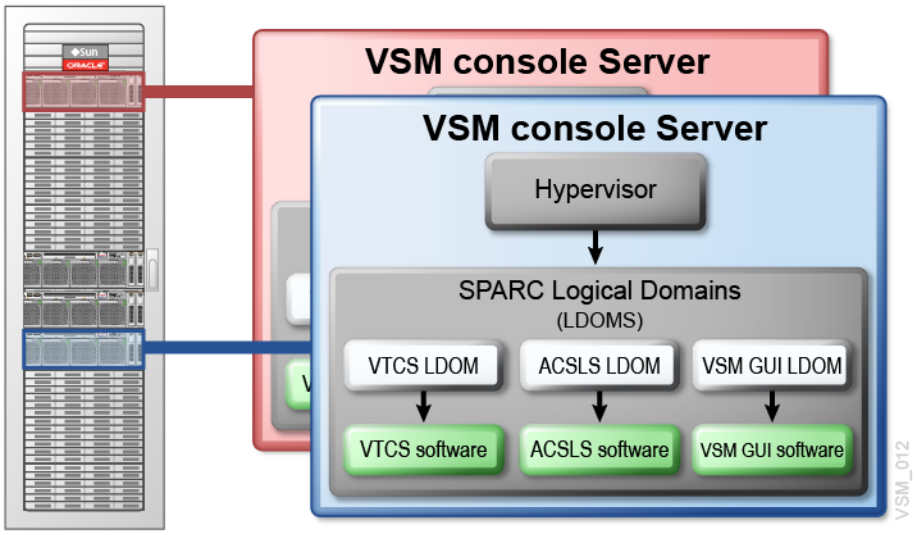
The VSM console consists of dual SOLARIS servers on a T5-2 platform (one primary server and one secondary server for redundancy). These servers are installed into an existing Oracle Sun Rack II Model 1242, and are connected to a VSM installed in the same rack. These servers may be installed within a single VSM rack or on two separate VSM servers.

Note:

- This document includes information specifically related to the VSM console. As VSM console is a supported feature of VSM 6 or VSM 7 and is not available with earlier VSM releases, refer to the appropriate *VSM Planning Guide* for additional information.
- The VSM console appliance is installed by Oracle field services.

Figure 1–1 illustrates the main components of a VSM console server. These components are described in the sections that follow.

Figure 1–1 VSM console Servers



Pre-installed Software

The following Oracle StorageTek software is pre-installed on the VSM console server:

- StorageTek Virtual Tape Control Software (VTCS) Release 7.3 (customized to operate in the Solaris operating environment)

  This software enables management of virtual tape volumes (VTVs). It services virtual tape volume requests from clients connected to the VSM console. The VSM console uses existing VTCS interfaces to provide an easy migration path for existing MVS customers. **This version of VTCS running on the VSM console is known as oVTCS.**

- StorageTek Automated Cartridge System Library Software (ACSLS) Release 8.4 with XAPI server support enabled

  This software services real tape volume requests from clients connected to the VSM console.

- StorageTek VSM GUI Release 1.1 (customized to operate in the Solaris operating environment)

  This software provides a graphical user interface management console for oVTCS on the VSM console.

Managing Logical Domains (LDOMs)

The VSM console platform is designed to run multiple Oracle StorageTek software applications. To accomplish this, you must deploy multiple SPARC Logical Domains (LDOMs) on the VSM console server. Each instance of an Oracle StorageTek software application requires its own LDOM. Each LDOM utilizes its own dedicated SOLARIS operating system and virtual CPUs, memory, disks, and network interfaces.

VSM console provides a control domain known as the Hypervisor, which maps LDOM resources to the physical hardware. Additionally, VSM console provides a menu-driven terminal user interface application (TUI) that enables you to configure and manage the Hypervisor as well as each individual LDOM.

2

VSM console Planning

This chapter provides information about the following VSM console planning topics:

Satisfying Mainframe Host Software Requirements

Using SMC as an MVS client connected to the VSM console, oVTCS on a VSM console server requires the following:

- VSM 6 or VSM 7 as the VTSS
- SMC 7.3 or higher (with the XAPI support), serving as an MVS client to VSM console

Satisfying Network Infrastructure Requirements

If possible, do any configuration of IP addresses, network switch(es) for VLANs or other setup (running cables, and so forth) before the VSM console arrives to minimize the installation time.

Ensure that, for each VSM console server, the network is ready for connection to the VSM console as follows: 1/10GBase-T (1Gb or 10Gb copper RJ45 Ethernet connection) is required on all network switches and routers that are directly attached to VSM console. There are at most two physical Ethernet connections to each VSM console.

> **Note:** During VSM console configuration, you can specify a maximum of two time servers for each VSM console.

Check that you are using the proper (customer-supplied) 1GigE Ethernet cables:

- CAT5 cables and below are not acceptable for GigE transmission.
- CAT5E cable: 90 meters is acceptable if run through a patch panel, 100 meters if straight cable.
- CAT6 cable: 100 meters is acceptable regardless of patch panel configuration.

This release supports a single connection from VSM console to a single router.

Satisfying Serviceability Requirements

The VSM console product uses a standard Oracle service strategy common with other Oracle products. Automated Service Request (ASR) is a feature of Oracle Premier Support for Systems and Oracle Limited Warranty that is designed to automatically request Oracle service when specific hardware faults occur. You can use ASR to manually or automatically request the creation of a service case. You must have a service contract for the device.

Optionally, in combination with ASR, Oracle Support can configure outgoing email containing details about an ASR event. Contact Oracle Support for more information.

The advantages of ASR functionality are documented in the ASR FAQ available on the My Oracle Support site in Knowledge Article Doc ID 1285574.1.

Customer-supplied Information

The VSM console will be configured to allow outgoing ASR and email communication with Oracle Support. To support VSM console outgoing ASR notifications, you need to supply the information in Table 2–1 through Table 2–3 to the installing Oracle Field Engineer.

Table 2–1 General Configuration - Site Information

Configuration Value	Example
Company Name	Company Inc
Site Name	Site A
City	AnyTown

Table 2–2 General Configuration - Contact Information

Configuration Value	Example
First Name	Joe
Last Name	Companyperson
Contact email	joecompanyperson@company.com

Table 2–3 ASR Setup - Oracle Online Account Information

Configuration Value	Example
Customer Oracle CSI Login Name	joecompanyperson@company.com
Customer Oracle CSI Login Password	********

Note: You must log in to My Oracle Support (MOS) and approve the registration of the VSM console. Until this approval is completed, the VSM console is not capable of auto-generating cases through MOS.

For email notification of event and log information, you must also supply a list of email IDs to notify of an ASR alert.

In cases where outgoing communication steps are not completed at the time of installation or not allowed at all, Oracle's options for timely response to events that require support from the Oracle Service team are greatly reduced. You can configure the VSM console to send email containing event and log information directly to a designated customer internal email address. A recipient of this email can then initiate a service request directly with Oracle and forward any emails received from the VSM console to Oracle Support. In this case, you must supply the email address where VSM console emails are sent, where this email address can accept emails of up to 5M.

ASR Configuration Information

ASR is configured with the following information:

- Your MOS credentials
- The URL for ASR events, which is MOS

Determining VSM console Configuration Values

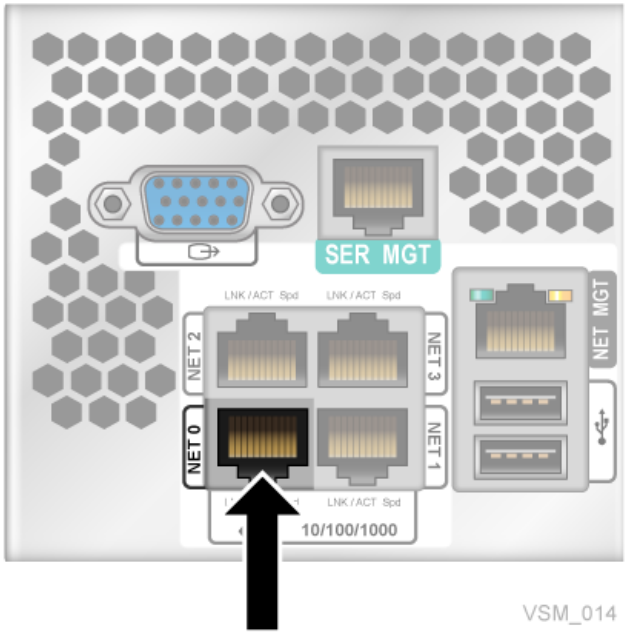
The following sections explain how to determine configuration values for the VSM console.

VSM console Ethernet Ports

Figure 2–1 shows the 1GigE Ethernet ports on the rear of the server.

Note: Attach only through NET0.

Figure 2–1 VSM console 1GigE Ethernet Data Ports



The 1GigE Ethernet ports are general purpose ports that are used for connection to the network for data traffic and management.

Port's Host Name

The value is the machine (host) name for each IP address to be connected to the network. Characters can be alpha-numeric (A-Z, a-z, 0-9) or "." or "-". The first and last characters of the string cannot be "." or "-". The name cannot be all-numeric. Host names (not including the domain names) must be 8 characters or less.

IP Address

There is only one network address for the server. The IP addresses may be IPV4 or IPV6 and use the standard notation.

CIDR

VSM console uses CIDR notation to manage network and host addresses. You must know the CIDR for the network that VSM console attaches to.

Providing Configuration Information to Oracle

Supply this information to Oracle field personnel as they prepare to configure the VSM console for you:

- database format
- data disk size

Selecting the Database Format

Before VSM console configuration, you **must provide the following information to Oracle field personnel**:

- **The database mode, either HSC-based or SQL**
  - An HSC-based database allows both existing MVS VTCS systems and VSM console oVTCS systems to use the same CDS.

    Using the HSC CDS causes a performance limitation.
  - An SQL database can be replicated across multiple systems. The SQL database is designed to be an HA (high availability) solution with an oVTCS node running on both the primary and secondary VSM console nodes.
- **If you choose the SQL database option, you must tell Oracle which oVTCS node is primary and which is secondary.**

  The SQL database can be replicated across multiple systems and is designed to be an HA (high availability) solution with an oVTCS node running on both the primary and secondary VSM console nodes.

Determining the Data Disk Size

During LDOM creation, the VSM console asks for the size of the data disk.

For the initial configuration, you must provide the data disk size information based on the following estimations:

- For an ACSLS LDOM, a total data disk size of approximately 21Gb:
  - 1Gb for the database
  - 10Gb for backup images

  - 10Gb for diagnostic materials, for example, logs and traces. Keep in mind the larger the data disk the more diagnostic materials can be retained.

    Note: Oracle recommends 25Gb to ensure you have allocated enough space.

- For an oVTCS LDOM, **using an example of a configuration containing 1,000,000 VTVs**, a total data disk size of approximately 21Gb:
  - The CDS size depends on the size of the configuration.
    * Assume 500 bytes per VTV
    * 500Mb for 1,000,000 VTVs
  - 5Gb for CDS backups, based on 10 * the CDS size for 1,000,000 VTVs.
  - 15Gb for diagnostic materials, for example, logs and traces. Keep in mind the larger the data disk the more diagnostic materials can be retained.
- For a VSM GUI LDOM, a total data disk size of approximately 48 Gb.

  Note: The disk size can be changed dynamically at a later time if necessary.

3

VSM console System Identifiers

This chapter describes user IDs, passwords, hostnames, IP addresses, and other information you must provide to create a functioning VSM console.

Required VSM console Identifiers

Table 3–1 shows required VSM console identifiers.

Table 3–1 Required VSM console Identifiers

Identifiers:	What You Need:
MOS log in information	■ MOS user ID ■ MOS password
ILOM	■ ILOM user name (`root`) ■ ILOM password (`changeme`) ■ hostname ■ IP address ■ CIDR
VSM Console primary and secondary servers	**For Servers**: ■ hostname ■ IP address ■ CIDR ■ default router ■ DNS servers 1-3 ■ search domains 1-3 **For Virtual Machines (1 to *x*):** ■ type (either ACSLS or VTCS) ■ hostname ■ IP address ■ CIDR

4

VSM console Configuration Scenarios

This chapter shows samples of the following configuration scenarios:

- Database Configurations
- VSM console Configurations

Database Configurations

The following configurations are examples of two database modes:

- SQL Database
- HSC CDS

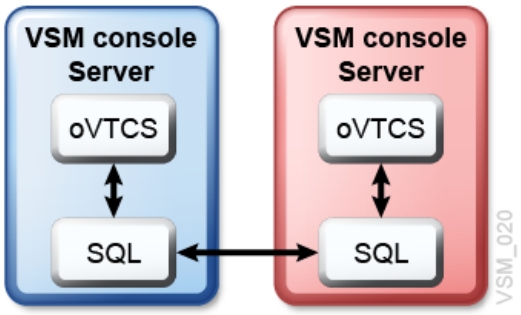
SQL Database

Figure 4–1 shows a typical configuration using SQL database services. This is expected to be the most common configuration.

After the VSM console initial configuration has been performed, load the oVTCS policies through the `oVTCS MGMTDEF ACTIVATE` command and perform an oVTCS configuration. These tasks are executed through the `SMCUUUI` utility. See "Loading the oVTCS Policy Parameter File Using the SMCUUUI Utility" on page 5-4.

Once the `MGMTDEF` and oVTCS configuration is completed, the VSM console systems may be brought up to full service level.

Figure 4–1 Configuration Using SQL Services



HSC CDS

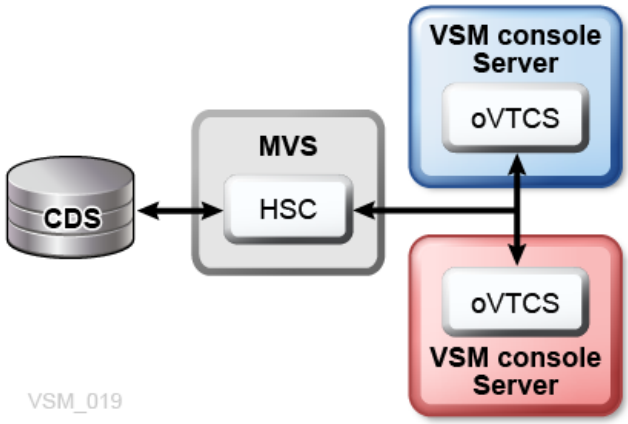
Figure 4–2 shows a configuration using an HSC CDS.

Assuming you have an existing production environment with VTCS using an HSC CDS, and you want to migrate to VSM console, you can perform the procedure below.

> **Note:** You can use this method to transition an existing tapeplex under ELS control to a tapeplex under VSM console control. It enables ELS and VSM console to run on the same hardware and to use the same CDS. Specifically this enables a fallback to an all ELS controlled system if required.

1. Set up VSM console to share an HSC CDS.
2. Create a test system on VSM console.
   - If it fails, remove VSM console, and the CDS retains the correct data.
   - If it succeeds, add a production environment to VSM console.
3. If the test system is running properly, move all production to VSM console and stop VTCS on all MVS systems.
4. Configure VSM console to use SQL database services.

Figure 4–2 Configuration using an HSC CDS



VSM console Configurations

The following configurations are examples for VSM console:

- Configuration for VTVs Only
- Configuration with VLE
- Configuration with a Real Library

In each example, an SMC client is connected to access the VSM console. Though not shown, a VSM system is present in each configuration. Virtual requests are processed by the oVTCS application residing on the VSM console server.

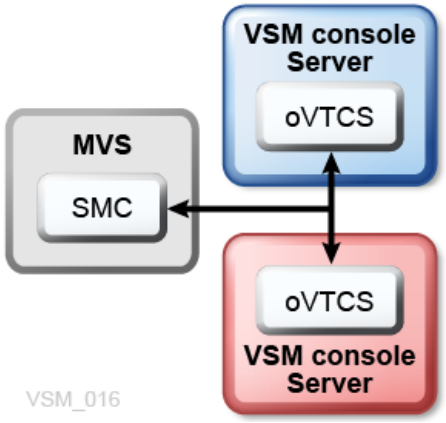
Configuration for VTVs Only

Figure 4–3 shows a configuration with VSM console processing VTVs only.

In this configuration, there is one oVTCS TapePlex on two VSM console servers (one primary and one secondary). The SMC client's SMCCMDS or SMCPARMS data set must define only one SMC `TAPEPlex` statement and two SMC `SERVer` statements, one for each oVTCS TapePlex.

See "Using an MVS Client with the VSM Console" on page 6-1 for more information about connecting the SMC client to the VSM console.

Figure 4–3 Configuration for VTVs Only



Configuration with VLE

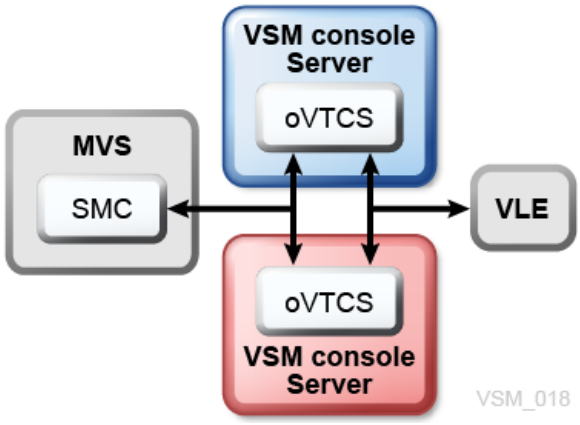
Figure 4–4 shows a configuration with Oracle's Virtual Library Extension (VLE) added.

In this configuration, there is one oVTCS TapePlex on two VSM console servers (one primary and one secondary). The SMC client's SMCCMDS or SMCPARMS data set must define only one SMC `TAPEPlex` statement and one SMC `SERVer` statement for the oVTCS TapePlex.

See "Using an MVS Client with the VSM Console" on page 6-1 for more information about connecting the SMC client to the VSM console.

Additionally, the oVTCS on the VSM console must define VLE as a TapePlex. See "Loading the oVTCS Policy Parameter File Using the SMCUUUI Utility" on page 5-4.

Figure 4–4 Configuration with VLE



Configuration with a Real Library

Figure 4–5 shows a configuration with a library added logically.

In this configuration, there is one oVTCS TapePlex on two VSM console servers (one primary and one secondary). The SMC client's SMCCMDS or SMCPARMS data set must define SMC `TAPEPlex` and `SERVer` definitions for oVTCS on the VSM console.

If the SL library includes non-RTD tape drives, and non-MVC tape cartridges accessible to MVS, then the SMC client's SMCCMDS or SMCPARMS data set must also define SMC TAPEPlex and SERVer definitions for the HSC or ACSLS library server.

If the SL library includes only RTD tape drives and MVC tape cartridges accessible only to oVTCS, then the SMC client's SMCCMDS or SMCPARMS data set does not require SMC TAPEPlex and SERVer definitions for the HSC or ACSLS library server.

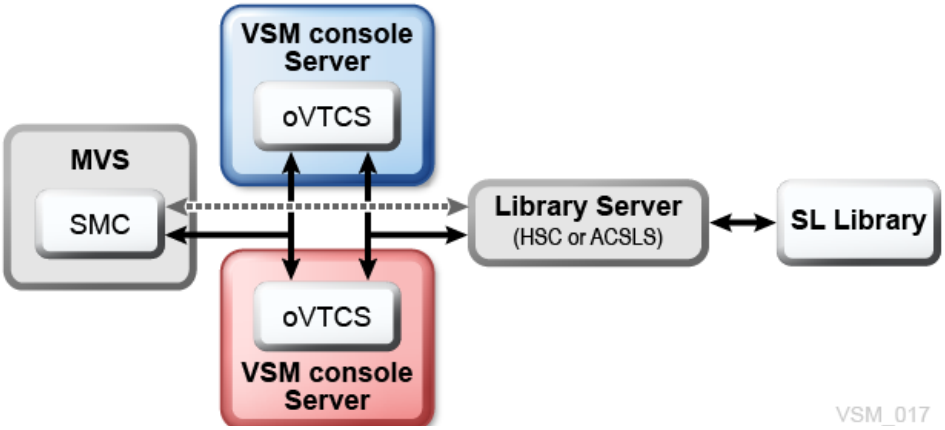
The library server can be HSC or ACSLS:

- If the library server is HSC, then it resides on an MVS host.
- If the library server is ACSLS, it is located on the ACSLS LDOM on the VSM console server.

See "Using an MVS Client with the VSM Console" on page 6-1 for more information about connecting the SMC client to the VSM console.

Additionally, the oVTCS on the VSM console must define the HSC or ACSLS library server as a TapePlex. See "Loading the oVTCS Policy Parameter File Using the SMCUUUI Utility" on page 5-4.

Figure 4–5 Configuration with Real Library



5

oVTCS Operational Considerations

This chapter describes software considerations for oVTCS, the version of VTCS that runs on the VSM console server.

oVTCS Functions

oVTCS represents StorageTek Virtual Tape Control Software (VTCS) Release 7.3, customized to operate on the VSM console in the Solaris operating environment. oVTCS performs the following functions:

- Influences the allocation of Virtual Tape Drives (VTDs).

  Virtual Tape Storage Subsystem (VTSS) is the VSM disk buffer containing virtual volumes (VTVs) and transports. The VTSS is disk device with microcode that enables emulation of 32 or 64 transports. The device can read and write "tape" data from/to disk, and can read and write the data from/to an RTD.

  A Virtual Tape Drive (VTD) is a transport in VSM's Virtual Tape Storage Subsystem (VTSS) that emulates a physical tape cartridge. The data written to a VTD is actually written to disk. The VTSS has 256 VTDs that perform virtual mounts of VTVs.

- Manages the use of Virtual Tape Volumes (VTVs), including migration and recall.
  - Migration is the movement of data from the VTSS to the Real Tape Drive (RTD) where VTVs are stacked onto MVCs.
  - Recall is the movement of VTVs back to the VTSS from the MVC. VSM provides the ability to recall VTVs on demand.
- Manages the use of real tape media and transports used by VSM.

Defining oVTCS Policy Parameters

This section describes the oVTCS policy parameter file and how to activate this file in your VSM console configuration.

oVTCS Policy Parameter File

In a VSM console configuration, oVTCS utilizes an oVTCS policy parameter file to include management and storage class policies for the oVTCS configuration. During startup, oVTCS examines the status of VTVs in the CDS, loads the defined policies, and implements the required actions in order to honor the policies.

When oVTCS is running in a cluster, this parameter file is automatically distributed to each node. The file settings are also persistent across restarts.

The method used to initially load this parameter file is dependent on your configuration.

- See "Loading the oVTCS Policy Parameter File Using the SMCUUUI Utility" on page 5-4.
- See "Loading the oVTCS Policy Parameter File Using the VSM GUI" on page 5-5.

Required Statements

The oVTCS policy parameter file must include at least one instance of each of the following statements:

> **Note:** With the exception of TAPEPLEX, the following statements closely match the ELS usage. Refer to your Oracle StorageTek ELS publications for more information about these statements.

POOLPARM

These statements describe scratch and MVC pools for the instance of oVTCS.

> **Note:** In shared CDS mode, POOLPARM statements in the CDS are not used.

VOLPARM

These statements define the attributes of the various volser ranges and allocates them to POOLPARM statements. These closely match the ELS usage of this statement, different only in that changing the VOLPARM statements does not update the oVTCS configuration.

STORCLAS

These statements define the storage classes. A storage class is a named list of storage attributes that identify performance goals and availability requirements for a data set.

MGMTCLAS

These statements define the management classes. A management class is a collection of management attributes, assigned by the storage administrator, that are used to control the allocation and use of space by a data set.

TAPEPLEX

These statements define the network contact details for other instances of ACSLS, HSC, and VLE. If VSM console has RTDs to migrate to, then a TAPEPLEX statement is required regardless of the type of TAPEPLEX (ACSLS, HSC, or VLE).

If there are multiple systems that VSM console can migrate to, then a separate TAPEPLEX statement for each system is required.

The format of the TAPEPLEX statement is as follows:

```
TAPEPLEX NAME=tapeplex_name SERVer(server [,server] [,server] [,server])
[SUBSYS=subsystem_name]
```

where:

- NAME specifies the name assigned to the TapePlex. This can be an ACSLS, HSC, or VLE system, and must match the TapePlex name assigned by the target ACSLS, HSC, or VLE.

- SERVer specifies one or more server paths to the named TapePlex. You can specify a host name or IP address.
- SUBSYS specifies the name of the HSC MVS subsystem and is required only when the target TapePlex is an HSC, and only when there are multiple HSC subsystems on the same MVS host.

The following is an example of a TAPEPLEX statement for an HSC system, with the SERVer parameter specifying the host name:

```
TAPEPLEX NAME=HSCVTCS SERV(host-name)
```

Alternatively, the SERVer parameter can specify the IP address in place of *host-name*:

The following is an example of a TAPEPLEX statement for a multi-node VLE system, with the SERVer parameter specifying the IP address of each node:

```
TAPEPLEX NAME=VLE1 SERV(ip_address1, ip_address2, ip_address3)
```

In this type of configuration, you may want to perform maintenance on an individual VLE node. The SMC SERVer DISable command is not supported by VSM console. Instead, you must use the following process:

1. In the oVTCS policy parameter file, update the TAPEPLEX statement to remove the P address of the node requiring maintenance.
2. Issue the oVTCS MGMTDEF ACTIVATE command with the updated parameter file.
3. Perform maintenance on the node that was removed.
4. Update the TAPEPLEX statement to re-add the node IP address.
5. Issue the oVTCS MGMTDEF ACTIVATE command to load the updated parameter file.

Optional Statements

The oVTCS parameter file can optionally include the following statements:

> **Note:** The following statements closely match the ELS usage. Refer to your Oracle StorageTek ELS publications for more information about these statements.

OPTION

This statement specifies an identifying string (name) for the file.

MIGRSEL

These statements specify migration request settings for managing migrates to storage classes from VTSSs.

MIGRVTV

These statements specify migration request settings for individual VTV copies processed by immediate migration.

MVCATTR

These statements assign a swap-to RTD device type to an MVC media name. When an error occurs while reading an MVC on an RTD, VTCS may swap the MVC to another RTD to retry the operation.

STORLST

These statements specify lists of storage classes and their corresponding preferencing.

STORSEL
These statements specify storage class usage rules that apply to the Storage Class list and its preferencing specified on a referenced STORLST control statement.

VTSSLST
These statements specify a list of VTSSs and their corresponding preferencing.

VTSSSEL
These statements specify storage class usage rules that apply to the VTSS list and its preferencing specified on a referenced VTSSLST control statement.

CMDEXEC
These statements define commands to be executed at startup, or whenever the parameter file is loaded. It provides an equivalent of the ELS startup command file.

Loading the oVTCS Policy Parameter File Using the SMCUUUI Utility

To activate the oVTCS policy parameter file in a mainframe configuration, use the SMC SMCUUUI utility to issue the oVTCS MGMTDEF command.

oVTCS MGMTDEF Command

The oVTCS MGMTDEF command activates the oVTCS policy parameter file.

From the SMCUUUI utility, specify the name of your oVTCS parameter file, along with the MGMTDEF command statement with the ACTIVATE parameter, in a UUIIN SDD statement, as shown in Example 5–1.

The oVTCS parameter file can exist anywhere as long as the fully qualified path and file name is specified.

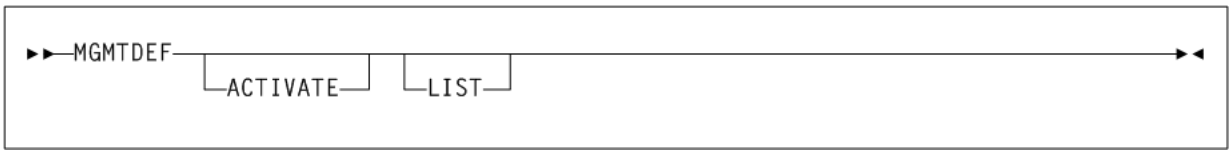
Refer to the *ELS Command, Control Statement, and Utility Reference* for information about the SMCUUUI utility.

Note: The oVTCS MGMTDEF command is a native oVTCS command, and is not related to the SMC MGMTDEF command.

Syntax

Figure 5–1 shows syntax for the oVTCS MGMTDEF command statement:

Figure 5–1 oVTCS MGMTDEF command syntax



Parameters

As shown in Figure 5–1, the oVTCS MGMTDEF command includes the following parameters:

ACTIVATE
optionally, validate the parameters contained in the specified oVTCS parameter file and then activate these parameter settings.

Note: If you do not specify the ACTIVATE parameter, the parameters contained in the specified oVTCS parameter file are only validated.

LIST
optionally, list the parameters that are read from the oVTCS parameter file.

Example

The following example shows the MGMTDEF command with the ACTIVATE parameter as specified in the SMCUUUI utility UUIIN statement:

Example 5–1 MGMTDEF ACTIVATE command

```
//UUIIN    DD  *
    SDD DDNAME(INPARMS) INPUT TEXT
    MGMTDEF ACTIVATE
```

In the example above, the MGMTDEF ACTIVATE command indicates to validate and activate the oVTCS parameter file, named INPARMS.

Loading the oVTCS Policy Parameter File Using the VSM GUI

To activate the oVTCS policy parameter file using the VSM GUI supplied with the VSM console:

1. Start the VSM GUI application.
2. Access the **VSM Console menu**. This menu includes the following options:
   - **Command Line Interface (CLI)**
   - **Configuration/Policy**
   - **Console Log**
3. Select the **VSMc Configuration/Policy** tab. This page enables you to download, edit, and upload an oVTCS Policy file that defines oVTCS policy settings.
4. Select the **tapeplex name** from the menu.
5. Select the **server address** from the menu. Only server addresses configured for the selected tapeplex are listed.
6. Click the **Download** button to specify your policy parameter file and load it into the VSM GUI.
7. Click the **Edit** button to make any desired edits to the file.
8. Click the **Upload** button to activate the file in the specified oVTCS tapeplex.

Refer to the *VSM GUI User's Guide* for more detailed information about using the **VSM Console Menu** to load your configuration settings.

oVTCS Command Considerations

The following list describes oVTCS command considerations:

- CDS Logging is not supported. Therefore, the LOGPOL parameter is not valid in the CONFIg GLOBAL statement.

- If you use SMC 7.3, XAPI security must be disabled by setting the XSECurity parameter of the SMC HTTP command to OFF. This enables RTDs attached to HSC to come online.

  Refer to the *ELS Command, Control Statement, and Utility Reference* for information about the SMC HTTP command.

- oVTCS includes a native MGMTDEF command that enables you to activate the oVTCS policy parameter file in a mainframe configuration. This command is not related to the ELS MGMTDEF command. See "Loading the oVTCS Policy Parameter File Using the SMCUUUI Utility" on page 5-4 for more information.

- You must use CONFIg VTVVOL and CONFIg MVCVOL statements to add VTVs or MVCs to the CDS. You cannot use the POOLPARM or VOLPARM method.

  You can use POOLPARM and VOLPARM for SUBPOOL naming, however, you must use CONFIg VTVVOL and CONFIg MVCVOL to define the volumes.

  Refer to the *ELS Command, Control Statement, and Utility Reference* for information about the POOLPARM and VOLPARM control statements.

  Refer to the *ELS Legacy Interfaces Reference* for more information about CONFIg VTVVOL and CONFIg MVCVOL control statements.

- When issuing mounts from the SMCUUUI utility, use the following conventions:
  - You must specify the full MOUNT keyword. Unlike the ELS command, you cannot abbreviate this keyword.
  - Specify the device address as N, NAME, or DRIVE_NAME=*devaddr*, where *devaddr* is the device address.
  - Specify the volser as V, VOL, or VOLSER=*volser*, where *volser* is the volume serial number or SCRTCH.
  - Specify the subpool as P, POOL, SUBPOOL, or SUBPOOL_NAME=*subpool-name*.

- When issuing dismounts from the SMCUUUI utility, use the following conventions:
  - You can specify the full DISMOUNT keyword or the abbreviated DISM.
  - Specify the device address as N, NAME, or DRIVE_NAME=*devaddr*, where *devaddr* is the device address.
  - Specify the volser as V, VOL, or VOLSER=*volser*, where *volser* is the volume serial number or SCRTCH.

- Handling of tape libraries in a non-shared CDS mode:

  If your configuration does not run in shared CDS mode, the following parameter restrictions apply:

  - RTD statements in the oVTCS configuration must include STORMNGR parameters.
  - In VTSS statements in the oVTCS configuration, the DEFLTACS parameter can only be defaulted.
  - In STORCLAS statements, use of the ACS parameter requires the STORMNGR parameter.
  - In MGMTCLAS statements, the ACSLIST parameter cannot be used.

  If your configuration runs in shared CDS mode, all libraries are considered 'remote' and are therefore part of an independent TapePlex. The name of the TapePlex that is the default library is supplied as part of the database

configuration. Normally, this TapePlex would also supply the CDS. Therefore, the restrictions described above do not apply.

- The oVTCS `TRace` command includes only two options, `ON` and `OFF`.
  - `TRace ON` closes all trace files and opens a new trace file for all running processes. This is the recommended setting.
  - `TRace OFF` stops all tracing.

  Unlike the ELS `TRace` command, you cannot specify specific components to be traced.
- Use the SMC `VMSG` command to obtain VSM console messages

  See "Starting/Stopping the VSM console Message Processor" on page 6-9 for more information about this command.
- Use the HSC `DBSERVER` command to enable the VSM console to share an HSC CDS.

  See "Running the oVTCS CDS Database Server" on page 6-5 for more information about this command.
- Use the SMC `SMCUSMF` utility to off-load SMF-type records from a VSM console server.

  See "Off-loading VSM console SMF Records" on page 6-13 for more information about this utility.
- Use the internal MVS `GETMGPOL` command to return listings of active oVTCS policy statements:
  - Specify the `GETMGPOL` command with no subparameters to return oVTCS `MGMTCLAS` and `STORCLAS` statements.
  - Specify `GETMGPOL MGMTCLAS` to return oVTCS `MGMTCLAS` statements.
  - Specify `GETMGPOL STORCLAS` to return oVTCS `STORCLAS` statements.
  - Specify `GETMPOL FLATDD(`filename`)` to return all oVTCS policy statements. This returns the entire contents of the oVTCS parameter file.

oVTCS Operator and Administrator Commands

oVTCS includes a set of operator and administrator commands. These commands are identical to their ELS VTCS counterpart, with few exceptions as described in "oVTCS Command Considerations" on page 5-5.

Use one of the following methods to issue these commands:

- Use the SMC Route command or `SMCUUUI` utility to send commands from an SMC client to oVTCS on the VSM c.onsole.

  Refer to the *ELS Command, Control Statement, and Utility Reference* for information about the SMC `Route` command and `SMCUUUI` utility.
- Start the VSM GUI application supplied with the VSM console and use the VSMc Command Line Interface (CLI) feature to send commands to oVTCS on the VSM console. Refer to the *VSM GUI User's Guide* for information about using this feature.

  1. Start the VSM GUI application.
  2. Access the **VSM Console menu**.

3. Select the **VSMc Command Line Interface (CLI)** tab. This page enables you to download, edit, and upload an oVTCS Policy file that defines oVTCS policy settings.
4. Select the tapeplex and the appropriate node server address.
5. Enter the oVTCS command in the input text box and click **Submit**.

   The command is recorded in the Command Log and command Output tables.

oVTCS includes the commands listed below. Refer to the *ELS Command, Control Statement, and Utility Reference* for information about each command.

- ARCHive
- AUDIT
- CANcel
- CONSolid
- CONFIg
- DEComp
- DELETSCR
- DISMount
- Display
  - CMD
  - MSG
  - SERVer
  - ACTive
  - CLInk
  - CLUster
  - CONFIG
  - LOCKs
  - MIGrate
  - MVC
  - MVCPool
  - PATH
  - Queue
  - REPlicat
  - RTD
  - SCRatch
  - STORclas
  - STORMNgr
  - TASKs
  - VSCRatch
  - VTD

  - VTSS
  - VTV
- DRMONitr
- EEXPORT
- EXPORT
- INVENTRY
- MEDVERfy
- MERGMFST
- METADATA
- MIGrate
- Mount
- MVCDRain
- MVCMAINT
- MVCPLRPT
- MVCRPt
- RECall
- RECLaim
- RECONcil
- SCRPT
- SET MIGOPT
- TRace
- Vary (CLInk, PATH, RTD, VTSS)
- VLEMAINT
- VTVMAINT
- VTVRPt

oVTCS XAPI Server Component Operator and Administrator Commands

The VSM console includes several XAPI server operator and administrator commands that enable you to manage the XAPI server component that operates within the VSM console.

Use one of the following methods to issue these commands:

- Issuing XAPI Server Commands Using the xapi_startup_file

  The xapi_start_file is a file of XAPI server commands that is read during XAPI server startup.

  This is the preferred method of specifying XCLIENT and XUDB definitions and XSECURITY and MSGLVL specifications.

  As this file is read at startup, there is no need to re-enter these commands if the XAPI server component is restarted.

The default path for the xapi_start_file under VSM console is: `/data/ovtcs/config/xapi_startup_file`.

- Issuing XAPI Server Commands Using the oVTCS_cli XCMD interface

  The `oVTCS_cli` interface may be used to direct commands to the XAPI server component by specifying '`XCMD`' followed by the XAPI server command string.

  For example, to enter the XAPI server '`LOG 0011`' command using the `oVTCS_cli` enter the command:

  ```
  oVTCS_cli ' XCMD LOG 0011'
  ```

  This is the preferred method to enter XAPI server `LIST` and `TRACE` commands when directed by StorageTek Software support.

- Issuing XAPI Server Commands Using the VSM GUI

  You can issue XAPI server commands to the VSM console using the VSM GUI VSM console command line interface option.

  Refer to the *VSM GUI User's Guide* for information about using the VSM console menu options.

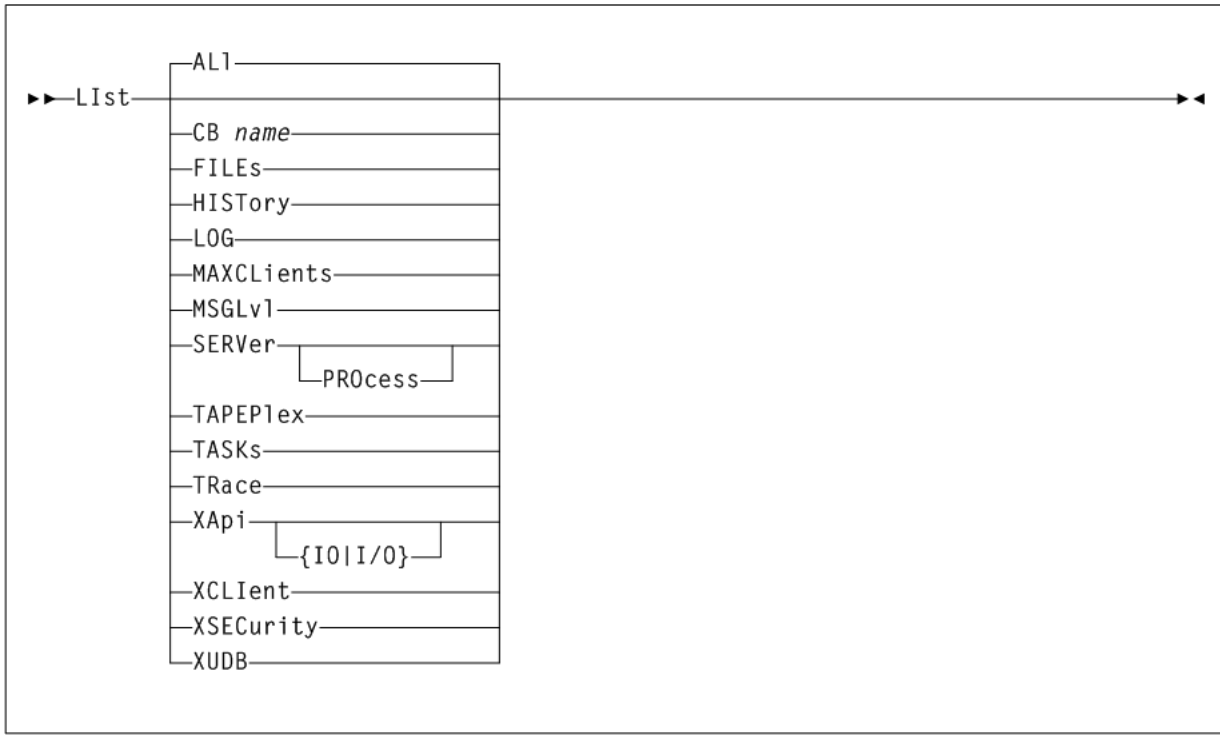
XCMD LIst command

The XCMD `LIst` command displays XAPI server component and environment settings. The XCMD `LIst` command is intended to be used primarily as directed by StorageTek Software Support.

Syntax

Figure 5–2 shows syntax for the `XCMD LIst` command.

Figure 5–2 XCMD LIst command syntax



Parameters

As shown in figure, the `XCMD LIst` command includes the following parameters:

ALl
optionally, displays all XAPI server parameters and environment variables. `ALl` is the default when no parameters are specified on the `LIst` command.

CB *NNNN* or *NNNN-IIII*
optionally, lists the named (*`NNNN`*) XAPI server control block in character hexadecimal. The name (*`NNNN`*) or name-index (*`NNNN-IIII`*) combinations are:

- `HTTPCVT` indicates the main XAPI server shared segment.
- `HTTPGBL` indicates XAPI server global definitions.
- `HTTPREQ-`*`nnn`* indicates the XAPI server request block.
- `HTTPAPI-`*`nnn`* indicates the XAPI server `API` request block.
- `XCLIENTTABLE` indicates the XAPI server `XCLIENT` shared segment.
- `XUDBTABLE` indicates the XAPI server `XUDB` shared segment.

FILEs
optionally lists the XAPI server file paths.

HISTory
optionally lists the XAPI server accept history for the past 24 hours.

LOG
optionally lists the XAPI server `LOG` setting.

MAXCLients
optionally lists the XAPI server `MAXCLients` setting.

MSGLvl
optionally lists the XAPI server `MSGLvl` setting.

SERVer
optionally lists relevant UNIX system release and resource limits, and XAPI server release, version, and environment settings.

> **PROcess**
> when specified with `LIst SERVer`, the `PROcess` keyword optionally lists the individual XAPI system processes.

TAPEPlex
optionally lists the XAPI server TapePlex name.

TASKs
optionally lists the XAPI server system and work tasks, along with their execution statistics.

TRace
optionally lists the XAPI server `TRace` setting.

XApi
optionally lists relevant XAPI server TCP/IP parameters

IO or I/O
when specified with `LIst XApi`, the `IO` (or `I/O`) keyword optionally lists the XAPI server TCP/IP statistics.

XCLIent
optionally lists the XAPI server `XCLIent` definitions.

XSECurity
optionally lists the XAPI server `XSECurity` setting.

XUDB
optionally lists the XAPI server `XUDB` definitions.

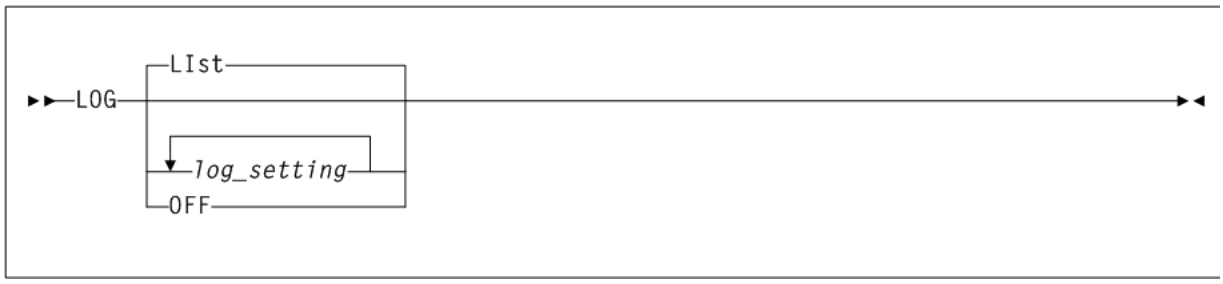
XCMD LOG command

The XCMD `LOG` command displays or changes the XAPI server log settings. XAPI server logging optionally enables TCP/IP requests, TCP/IP responses, XAPI server operator commands, console messages, and errors to the XAPI server log file. The XCMD `LOG` command is intended to be used primarily as directed by StorageTek Software Support.

Syntax

Figure 5–3 shows syntax for the `XCMD LOG` command.

Figure 5–3 XCMD LOG command syntax



Parameters

As shown in figure, the `XCMD LOG` command includes the following parameters:

LIst
optionally, displays XAPI server log settings. `LIst` is the default when no parameters are specified on the `LOG` command.

OFF
optionally, turns off all XAPI server log events.

1 or 0
optionally, turns off or on individual XAPI server log events. A string of up to 16 '`0`' and '`1`' characters may be entered. '`1`' turns on the log event, '`0`' turns off the log event. The position in the entered string controls individual log events as follows:

- `1000000000` to log XAPI server error messages to stdout
- `0100000000` to log XAPI server messages to the log file
- `0010000000` to log XAPI input request errors to the log file

- 0001000000 to log XAPI recv packets to the log file
- 0000100000 to log XAPI send packets to the log file
- 0000010000 to log XCMD commands and responses to the log file

Note:

- Currently string positions 7-16 do not control any XAPI server log settings. If strings longer than 6 characters (but less than 17 characters) are entered, the characters are validated, but subsequently ignored.
- LOG 0 is equivalent to LOG OFF.
- Any entered setting completely replaces the prior log settings: Thus if LOG 010001 is followed by LOG 00011, then following the second LOG command, neither XAPI messages nor XCMD commands and responses will be logged to the log file.
- The environment variable SMCVLOGFILE may be used, if specified before XAPI server startup, to override the default XAPI server log file path.
- To display the location and name of the XAPI server log file, enter the XCMD LIST FILES command.
- The environment variable SMCVLOG may be used, if specified before XAPI server startup, to set the XAPI server log settings.

Example

In the following example, the LOG command logs both XAPI recv and send packets to the log file:

```
LOG 00011
```

XCMD MAXCLients command

The XCMD MAXCLients command is used to set an upper limit for the number of concurrent requests that may be active at any one time. When the MAXCLients limit is reached, any new client requests received by the XAPI server will receive the '503 Service unavailable' response and should be retried by the client.

Syntax

Figure 5–4 shows syntax for the XCMD MAXCLients command.

Figure 5–4 XCMD MAXCLients command syntax

Parameters

As shown in figure, the `XCMD MAXCLients` command includes the following parameters:

LIst
optionally, displays the XAPI server `MAXCLients` setting. `LIst` is the default when no parameters are specified on the `MAXCLients` command.

NNNN
optionally, specifies the maximum number of concurrent requests. Enter a number between 1 and 1000.

XCMD MSGLvl command

The XAPI server `MSGLvl` command is used to determine the messages that are output to `stdout`. Each XAPI server message has a fixed `MSGLvl`. When the XAPI server `MSGLvl` is greater than the message `MSGLvl`, the message is output, otherwise it is suppressed. Thus the higher the XAPI server `MSGLvl`, the more verbose the XAPI server messaging.

Syntax

Figure 5–5 shows syntax for the `XCMD MSGLvl` command.

Figure 5–5 XCMD MSGLvl command syntax

Parameters

As shown in figure, the `XCMD MSGLvl` command includes the following parameters:

LIst
optionally, displays the XAPI server `MSGLvl` setting. `LIst` is the default when no parameters are specified on the `MSGLvl` command.

NN
optionally, specifies the XAPI server `MSGLvl`. Enter a number between 0 and 32 as follows:

- `0` - display startup/shutdown and error messages only.
- `4` - display warning messages
- `8` - display additional system status messages
- `> 8` - display debugging messages; use only at the direction of StorageTek Software Support.

XCMD TRace command

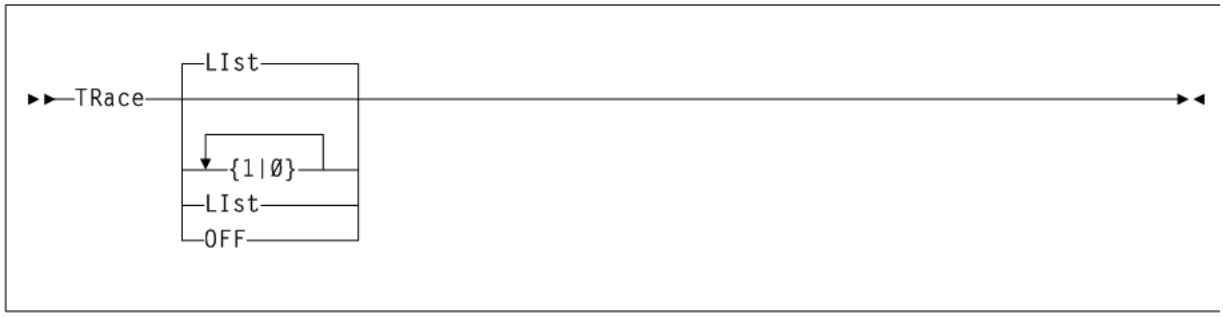
The XCMD `TRace` command displays or changes the XAPI server trace settings. XAPI server tracing optionally enables XAPI server component trace events. The XCMD

TRace command is intended to be used solely as directed by StorageTek Software Support.

Syntax

Figure 5–6 shows syntax for the XCMD TRace command.

Figure 5–6 XCMD TRace command syntax



Parameters

As shown in figure, the XCMD TRace command includes the following parameters:

LIst
optionally, displays XAPI server trace settings. LIst is the default when no parameters are specified on the TRace command.

OFF
optionally, turns off all XAPI server trace events.

1 or 0
optionally, turns off or on individual XAPI server trace events. A string of up to 16 '0' and '1' characters may be entered. '1' turns on the trace event, '0' turns off the trace event. The position in the entered string controls individual trace events as follows:

- 1000000000 to trace XAPI errors
- 0100000000 to trace XAPI TCP/IP component events
- 0010000000 to trace PGMI API component events
- 0001000000 to trace XAPI server thread events
- 0000100000 to trace XAPI server malloc() and free() events
- 0000010000 to trace XAPI server XML parse events
- 0000001000 to trace XAPI server command events
- 0000000100 to trace XAPI server system monitor
- 0000000010 to trace XAPI server XML, CSV, and text output component events
- 0000000001 to trace XAPI server logical file events

Note:

- Currently string positions 11-16 do not control any XAPI server trace settings. If strings longer than 11 characters (but less that 17 characters) are entered, the characters are validated, but subsequently ignored.
- Entering `TRACE 0` is equivalent to entering `TRACE OFF`.
- Any entered setting completely replaces the prior trace settings: Thus if `TRACE 010001` is followed by `TRACE 00011`, then following the second `TRACE` command, neither TCP/IP component events, nor XAPI server `malloc()` and `free()` events will be traced to the XAPI server trace file.
- The environment variable `SMCVTRCFILE` may be used, if specified before XAPI server startup, to override the default XAPI server trace file path.
- To display the location and name of the XAPI server trace file, enter the XCMD `LIST FILES` command.
- The environment variable `SMCVTRACE` may be used, if specified before XAPI server startup, to set the XAPI server trace settings.

Example

In the following example, the `TRace` command traces both XAPI server thread and process events and XAPI server `malloc()` and `free()` events to the XAPI server trace file:

```
TRace 00011
```

XCMD XCLIent command

The XAPI server `XCLIent` command is used to define XAPI clients that use a different protocol version than the default server XAPI protocol.

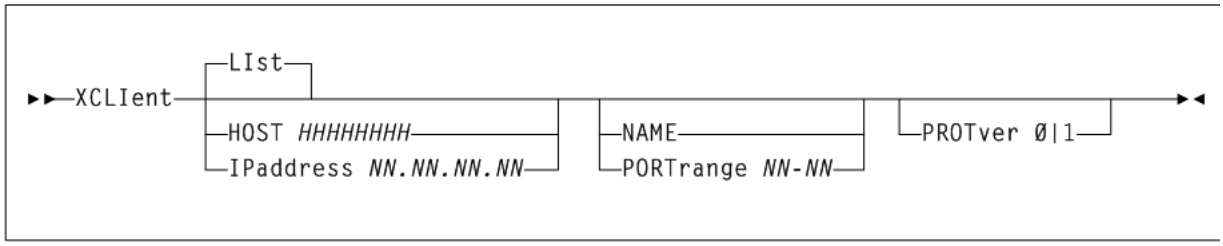
Note:

- The `XCLIent` command is only necessary to define clients that use the older 'unsecured' protocol when XAPI security is enabled (`XSECurity ON`). When `XSECurity ON` is specified, then any XAPI request that originates from a client that is not defined with a `XCLIent` command is assumed to use the newer XAPI security protocol.
- If `XSECurity OFF` is specified, then `XCLIent` definitions are not required.

Syntax

Figure 5–7 shows syntax for the `XCMD XCLIent` command.

Figure 5–7 XCMD XCLIent command syntax



Parameters

As shown in figure, the `XCMD XCLIent` command includes the following parameters:

LIst
optionally, displays XAPI server `XCLIent` definitions. `LIst` is the default when no parameters are specified on the `XCLIent` command.

HOst *HHHHHHHH*
optionally, specifies the IP resolver host name (*HHHHHHHH*) on which the client resides.The `HOst` name must be a resolvable name in the TCP/IP name table. The following rules apply:

- The value must be between 1 and 255 characters in length.
- The first character must be either an alpha character or a digit.
- The last character must be either an alpha character or digit.
- Any character between the first and last must be either an alpha character, digit, hyphen, or dot.

IPaddress *NN.NN.NN.NN*
optionally, specifies the IP address (*NN.NN.NN.NN*) for the client.

NAme *CCCCCCCC*
optionally, specifies the name (*CCCCCCCC*) of the client. If the client is SMC/MVS, the `NAme` specified should be the name returned as `<client_subsystem_name>`. Otherwise the `NAme` specified should be the name returned as `<client_name>`. If `NAme` is specified as '`*`', then any request from the specified `HOst` or `IPaddress` will be defined as using the specified protocol version.

PORTrange *NN-NN*
optionally, specifies the range of client ports (*NN-NN*) from which the XAPI request originates on the specific `HOst` or `IPaddress` that are allowed to use the specified protocol version. Valid ports are 1 to 65535, and the range may contain between 10 and 1000 ports.

PROTver [0|1]
optionally, specifies the protocol version.

- '`0`' indicates the older 'unsecure' protocol.
- '`1`' indicates the new XAPI security protocol version. The default is `0`.

Note:

- HOst and IPaddress are mutually exclusive.
- NAme and PORTrange are mutually exclusive.
- PORTrange should be used if the SMC/MVS TCPIP PORTrange command has been specified to restrict SMC/MVS client ports to the specified range.

XCMD XSECurity command

The XAPI server XSECurity command is used to globally enable or disable the XAPI security protocol for the XAPI server.

When XAPI security protocol is enabled, then user password security will be enforced for any requests received by the XAPI server.

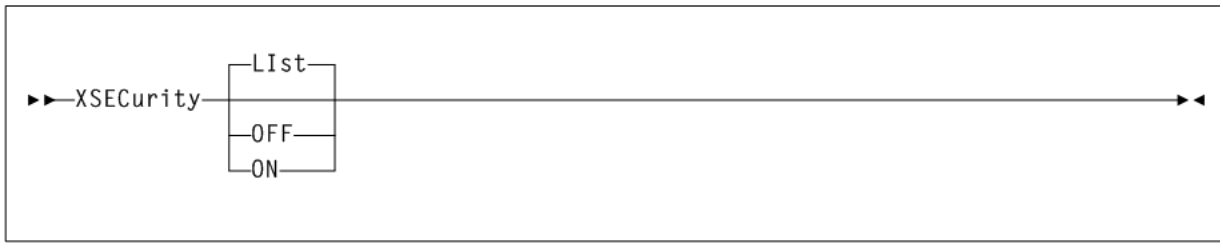
Note:

- When XAPI security is globally enabled, individual clients may be 'exempted' from XAPI security if they are the specified in an XCLIent definition.
- When XAPI security is globally enabled, then the same user and password must be defined on both the client and server. XAPI security users and passwords are normally specified using XUDB definitions, but see the notes in the XUDB command concerning XAPI security users under the Virtual Storage Manager console (VSMc).

Syntax

Figure 5–8 shows syntax for the XCMD XSECurity command.

Figure 5–8 XCMD XSECurity command syntax



Parameters

As shown in figure, the XCMD XSECurity command includes the following parameters:

LIst
optionally, displays the XAPI server XSECurity setting. LIst is the default when no parameters are specified on the XSECurity command.

ON
optionally, enables XAPI security.

OFF
optionally, disables XAPI security.

XCMD XUDB command

The XAPI server `XUDB` command is used add, update, delete, and list XAPI security users. The XAPI security user list is maintained by both the client and server.

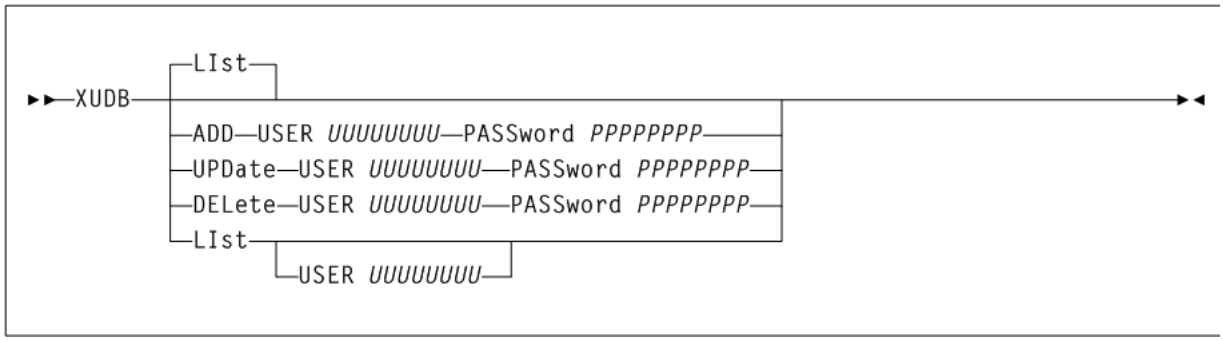
Note:

- The same XUDB `USER` must be defined on both the client and server.
- If `XSECurity OFF` is specified, then XUDB definitions are not required.

Syntax

Figure 5–9 shows syntax for the `XCMD XUDB` command.

Figure 5–9 XCMD XUDB command syntax



Parameters

As shown in figure, the `XCMD XUDB` command includes the following parameters:

LIst
optionally, displays XAPI server XUDB definitions. `LIst` is the default when no parameters are specified on the `XUDB` command.

ADD
specifies that the specified user name and password is to be added to the XAPI security user list.

USER *UUUUUUUU*
specifies the name to be added. The `USER` name need not be a defined UNIX user name, so long as the same name and password are defined in both the client and server. The `USER` name may be up to 20 characters in length.

PASSWORD PPPPPPPP
specified the password for the specified `USER`. The `PASSword` may be up to 20 characters in length.

UPDate
specifies that the specified user name is updated in the XAPI security user list with the specified `PASSword`.

USER *UUUUUUUU*
specifies the name to be updated.

PASSWORD PPPPPPPP
specified the new password for the specified `USER`.

DELete
specifies that the specified user name is deleted in the XAPI security user list.

USER *UUUUUUUU*
specifies the name to be deleted.

LIst
specifies that the specified user name from the XAPI security user list is to be listed.

USER *UUUUUUUU*
specifies the name to list. If `USER` is not specified then all names are listed.

Note:

- When the XAPI server is a component of the Virtual Storage Management console (VSMc), the XAPI server uses VSM console facilities to add, update and delete XAPI security users. While XUDB `ADD` commands may be issued on the VSM console, it is recommended that you use the VSM console TUI to maintain XAPI security users.
- If an XUDB `ADD`, `UPDate`, or `DELete` command is issued on VSM console, the VSM console user database is updated.
- In VSM console, a XUDB `LIST` command will simply state that users are maintained in the VSM console user database.

Viewing the oVTCS Console Log

The VSM GUI supplied with the VSM console enables you to view a running system log for console operator messages generated from the oVTCS instances running on each VSM console server.

To view the console log:

1. Start the VSM GUI application.
2. Access the **VSM Console menu**.
3. Select the **Console Log** tab.
4. Select the tapeplex from the **Tapeplex** menu to populate the Console Log with messages from that tapeplex.

Three types of messages are displayed:

- WTO (Write to Operator)
- WTOR (WTO with Reply)
- HILITE (highlighted WTO)

There are two tables on the Console Log page:

- The WTORs and HILITEs table lists WTOR and HILITE messages for the selected tapeplex. Messages are displayed in chronological order sorted by message type, with the most recent at the bottom.

  Certain fields include a context menu indicator. Right-click on these fields to access the context menu, enabling you to perform actions including replying to WTOR messages or deleting messages.

- The Log table lists WTO messages and replied to or deleted WTOR and HILITE messages. Messages are displayed in chronological order sorted by message type, with the most recent at the bottom.

Refer to the *VSM GUI User's Guide* for more detailed information about using the VSM console Console Log.

6

ELS Facilities for VSM console

This chapter describes the following ELS-related topics that affect VSM console commands, operations, and usage.

Using an MVS Client with the VSM Console

This section describes how to use Oracle's StorageTek Storage Management Component (SMC) as an MVS client connected to the VSM console.

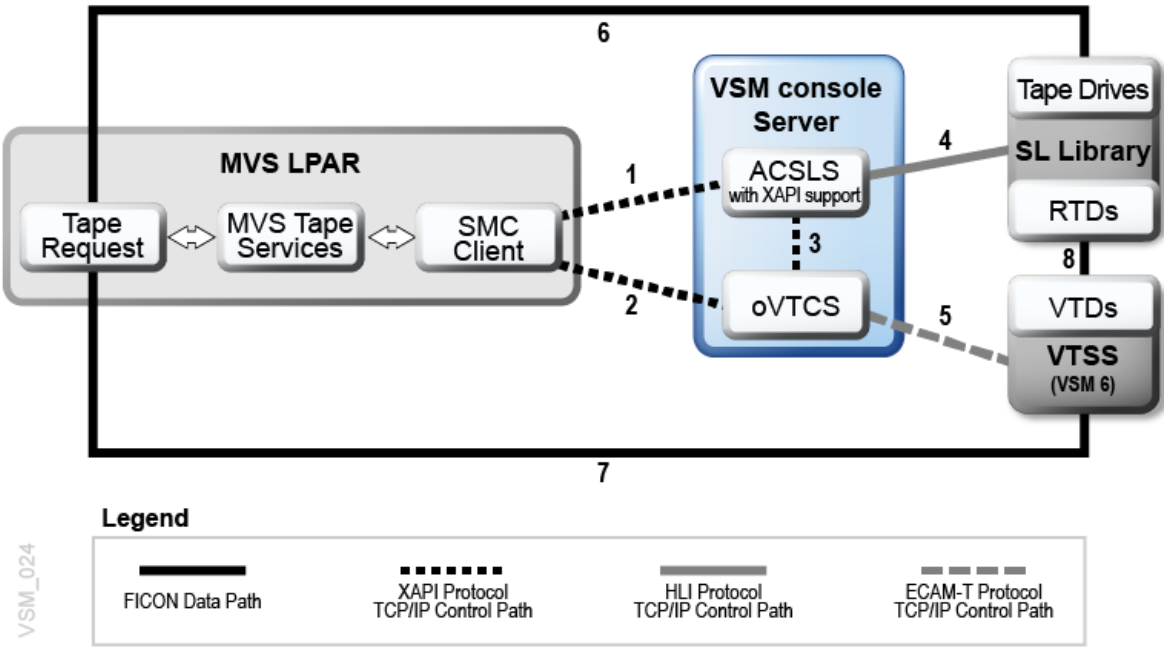
Overview

oVTCS on a VSM console server requires the following:

- VSM 6 or VSM 7 as the VTSS
- SMC 7.3 or higher (with the XAPI support), serving as an MVS client to VSM console.
- ACSLS 8.4 on a VSM console server (not required in a VSM console configuration processing VTVs only, or if the ACSLS is replaced by VLE(s) for multi-volume cartridge (MVC) support)

Figure 6–1 shows the MVS Client/VSM console server data flow. In this example, VSM console is configured with both ACSLS and oVTCS LDOMs to support both native tapes in an SL library and VTVs in a VSM 6.

Figure 6–1 MVS Client Connection to the VSM console Server



As shown in Figure 6–1:

- XAPI protocol over TCP/IP is used for the following paths:
  - Control path from SMC to ACSLS on the VSM console (path 1).
  - Control path from the SMC on MVS to oVTCS on the VSM console (path 2).
  - Control path from oVTCS to ACSLS with XAPI support (path 3).
- HLI protocol over TCP/IP is used for the control path from ACSLS on the VSM console to the SL library (path 4).
- ECAM-T protocol over TCP/IP is used for the control path from the oVTCS software on the VSM console to the VTSS (path 5).
- FICON is used for the following paths:
  - Data Path from MVS to the real tape drives in an SL library (path 6).
  - Data path from MVS to the VTDs in the VTSS (path 7).
  - Data path from the VTDs on the VTSS to the RTDS in the SL library (path 8).

As shown in Figure 6–1, the SMC client to VSM console software control path works as follows:

1. An MVS job sends a tape request to MVS tape allocation and mount or dismount services.
2. The SMC client receives the tape request from MVS tape services.
3. The SMC client, using XAPI protocol over TCP/IP, sends the following:
   - Virtual tape requests to oVTCS on a VSM console server.
   - Real tape requests to ACSLS with XAPI on a VSM console server.

Note:

- If all tape drives in the SL library are oVTCS RTDs, and all tape cartridges in the SL library are oVTCS MVCs, then the SMC client does not require access to the SL library. In this scenario, a control path from the SMC Client to ACSLS with XAPI support (path 1), and data path from MVS to the Tape Drives in the SL Library (path 6) are eliminated. See "Connecting SMC to ACSLS on the VSM console Server" on page 6-4.
- If oVTCS is configured for VTVs only, then there are no oVTCS RTDs and no oVTCS MVCs. In this scenario, there is no SL library, and ACSLS with XAPI Support is not required.
- If all oVTCS MVCs are on VLEs, then ASCLS with XAPI Support is not required.
- If you also have a HSC and VTCS on MVS connected to a separate VTSS that configured as the Cross Tape Replication target from oVTCS on the VSM console, then additional data paths and control paths are required.

Connecting SMC to the VSM Console Server Applications

The following sections describe how to enable the SMC client to connect to the oVTCS and ACSLS applications on the VSM console server. This is dependent on your configuration. See "Overview" on page 6-1.

Connecting SMC to oVTCS on the VSM console Server

You must perform the following steps to configure the XAPI control path from the SMC client to oVTCS on the VSM console. This is path 2 in Figure 6–1.

1. In your SMC client's SMCCMDS or SMCPARMS data set, include SMC `TAPEPlex` and `SERVer` commands to define the oVTCS on the VSM console server as a TapePlex and define the TCP/IP control path between the SMC client and the oVTCS LDOM on the VSM console server.

   For example:

   ```
   TAPEPLEX NAME(VTSP31) ENABLE
   SERVER NAME(VTSP31S) ENABLE TAPEPLEX(VTSP31) +
       HOST (VTSP31.yourhost.com) PORT(7070)
   ```

   This example includes the following:

   - An SMC `TAPEPlex` command that defines an oVTCS TapePlex, `VTSP31`, on the VSM console server.
   - An SMC `SERVer` command that defines a TCP/IP control path to the VSM console, where:
     - The TapePlex name (`VTSP3`) matches the name specified on the `TAPEPlex` statement.
     - The VSM console server name is `VTSP3S`.
     - The VSM console host name address is `VTSP31.`*`yourhost`*`.com`. The `HOST` parameter may be replaced by specifying the `IPADDRESS` parameter instead.

  - The server PORT parameter value is 7070. This value must match the listener port configured for oVTCS on the VSM console (the default value of the oVTCS listener port 7070).

Refer to the ELS publication *Configuring and Managing SMC* for information about the SMCCMDS and SMCPARMS data sets, and the *ELS Command, Control Statement, and Utility Reference* for information about the SMC TAPEPlex and SERVer commands.

2. Configure MVC and virtual volumes.

   Create an oVTCS parameter file to define MVC and virtual volumes and pools and create Management and Storage Classes to route data to VSM 6 or VSM 7 and/or tape drives on a SL Library. Using the SMCUUUI utility, specify the oVTCS MGMTDEF command to load this parameter file. See "Loading the oVTCS Policy Parameter File Using the SMCUUUI Utility" on page 5-4.

Connecting SMC to ACSLS on the VSM console Server

The following describes how to configure the XAPI control path from the SMC client to ACSLS with XAPI support on VSM console. This is path 1 in Figure 6–1.

This procedure may not be required depending upon the how the SL library is configured:

- If your SL library includes non-RTD tape drives, and non-MVC tape cartridges accessible to MVS, then you must configure the XAPI control path from the SMC to ACSLS as documented below.
- If all tape drives in the SL library are RTDs, and all of the tape cartridges in the SL library are MVCs, then the SMC client does not require access to the SL library. In this scenario, you are not required to configure the XAPI control path from the SMC client to ACSLS on the VSM console.

To connect SMC to ACSLS:

In your SMC client's SMCCMDS or SMCPARMS data set, include SMC TAPEPlex and SERVer commands to define the ACSLS on the VSM console server as a TapePlex and define the TCP/IP control path between the SMC client and the ACSLS LDOM on the VSM console server.

For example:

```
TAPEPLEX NAME(ACSLSLIB) ENABLE
SERVER NAME(ACSLSSRV) ENABLE TAPEPLEX(ACSLSLIB) +
    HOST (myhost.hostname.com) PORT(50020)
```

This example includes the following:

- An SMC TAPEPlex command, which defines an ACSLS TapePlex, ACSLIB, on the VSM console server.
- An SMC SERVer command that defines a TCP/IP control path to the VSM console, where:
  - The TapePlex name (ACSLSLIB) matches the name specified on the TAPEPlex statement.
  - The VSM console server name is ACSLSSRV.
  - The VSM console host name address is *myhost.hostname*.com. The HOST parameter may be replaced by specifying the IPADDRESS parameter instead.

– The server PORT parameter value is 50020. This value must match the listener port configured for ACSLS on the VSM console (the default value of the ACSLS listener port is 50020).

Refer to the ELS publication *Configuring and Managing SMC* for information about the SMCCMDS and SMCPARMS data sets, and the *ELS Command, Control Statement, and Utility Reference* for information about the SMC TAPEPlex and SERVer commands.

Running the oVTCS CDS Database Server

The oVTCS CDS database server component proxy enables a client oVTCS running on the VSM console server to act as a local z/OS VTCS so that it can access the z/OS-resident CDS database.

The oVTCS client must have its own "host slot" within the CDS to send and receive broadcast messages, hold locks, and so forth. This requirement imposes the following limitations on executing the oVTCS CDS database server:

- The oVTCS CDS database server cannot execute on a z/OS host that has VTCS executing or that may have VTCS executing. The HSC subsystem must be started with the NOVTCS startup parameter specified in the EXEC statement.
- The oVTCS CDS database server is restricted to communicating with a single oVTCS client; there is a 1-to-1 relationship between client and server. For example, if there are two oVTCS appliances, they each require their own HSC host. As a result, there are two instances of HSC each executing their own oVTCS CDS database server as shown in Figure 6–2:

Figure 6–2 Relationship Between oVTCS Client and oVTCS CDS DB Server

The "Server Host ID" boxes in Figure 6–2 represent the server on which the DBSERVer command is issued (see "DBSERVer command" on page 6-6). It is also the host that supplies the proxy host ID for the oVTCS client CDS. Thus, oVTCS1 is host ID MVSA and oVTCS2 is host ID MVSB. Note that MVSA and MVSB may share the same CDS, but they must have separate host IDs.

- The oVTCS CDS database server requires HSC database services be active. Therefore, the oVTCS CDS database server cannot be started before the HSC has reached BASE service level.
- You can include the DBSERVer START command in the HSC startup parameter file, or you can issue it as an HSC operator command. It is not a UUI/XAPI enabled command.
- The oVTCS CDS database server requires its own TCP/IP port assignment for its socket listener. The port number is specified on the DBSERVer START command. If you execute the SMC HTTP server on the same host as the oVTCS CDS database server, you must specify different port numbers.
- If requested by Oracle StorageTek Software support, use GTF and the TRace VTcs command to enable tracing of the oVTCS CDS database server on z/OS.

DBSERVer command

Interfaces:

- console or utility only
- UUI Support: No

Subsystem Requirements:

Active HSC required. VTCS must not be active.

Description

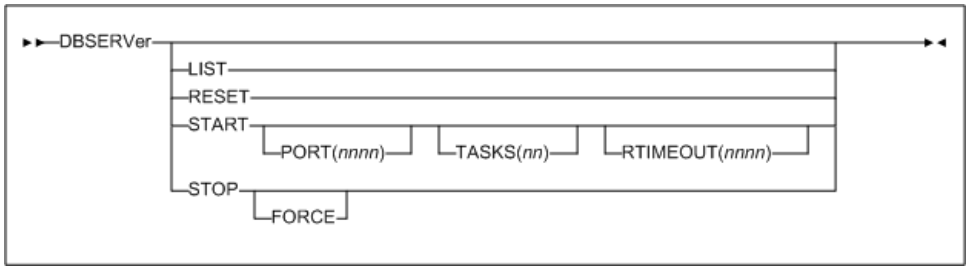
The `DBSERVer` command starts or stops the oVTCS CDS database server. The oVTCS CDS database server services CDS database I/O requests from an oVTCS client.

> **Note:** You can only start an oVTCS CDS database server in an HSC subsystem with no executing VTCS component.

Syntax

The following figure shows the syntax for the DBSERVer command:

Figure 6–3 DBSERVer command syntax



Parameters

LIST
optionally, list the settings and status of the oVTCS CDS database server.

RESET
optionally, force a reset of the oVTCS CDS database server which closes the accepted socket, clear any pending work, and listen again for the oVTCS client to re-connect.

START
optionally, start the oVTCS CDS database server.

PORT(*nnnn*)
optionally, specifies the socket listener port, *nnnn*. If you do not specify `PORT(nnnn)`, the default is 8081.

TASKS(*nn*)
optionally, specifies the maximum number of tasks for asynchronous CDS reads. Allowable values are 1-10. If you do not specify `TASKS(nn)`, the default is 4.

RTIMEOUT(*nnnn*)
optionally, specifies the reserve timeout in seconds, *nnnn*. Allowable values are 1-3600. If you do not specify `RTIMEOUT(nnnn)`, the default is 180.

STOP
optionally, stop the oVTCS CDS database server.

FORCE
optionally, force termination even when the oVTCS CDS database server has not terminated successfully.

oVTCS CDS Database Server and DBSERVer command Messages

SLS0780I

`Cannot start the oVTCS CDS server; CCCCCCCC`

Explanation: The DBSERVER command specified the START option but the oVTCS CDS database server could not be started for the reason stated.

System Action: The command is rejected.

User Response: Correct the condition and resubmit the DBSERVER command.

SLS0781I

`oVTCS CDS database server started on PORT=NNNN`

Explanation: The DBSERVER command specified the START option and was successfully started listening on the specified PORT number.

System Action: None.

User Response: None.

SLS0782I

`Timeout waiting for oVTCS CDS database server startup`

Explanation: The DBSERVER command specified the START option and but the oVTCS CDS database server did not successfully start within the allotted timeout period.

System Action: None.

User Response: Check the console for messages indicating the reason for the startup failure.

SLS0783I

`Cannot {LIST|RESET|STOP} the oVTCS CDS server; not currently active`

Explanation: The DBSERVER command specified the LIST or STOP option but the oVTCS CDS database is not currently active.

System Action: The command is rejected.

User Response: None.

SLS0784I

`Timeout waiting for oVTCS server CCCCCCCC termination`

Explanation: The DBSERVER command specified the STOP option but the indicated oVTCS CDS database server component did successfully terminate within the allotted timeout period.

System Action: None.

User Response: Check the console for messages indicating the reason for the termination failure. If the problem persists, use the DBSERVER STOP FORCE option.

SLS0785I

`oVTCS CDS database server termination complete`

Explanation: The DBSERVER command specified the STOP option and was successfully terminated.

System Action: None.

User Response: None.

SLS0786I

`oVTCS server TCP/IP error; func=`*`CCCCCCCC`*`, errno=`*`NN`*

`{TERMINATING|RESETTING|RETRYING|CONTINUING}`

Explanation: The oVTCS CDS database server encountered a TCP/IP error during its processing.

System Action: Depending upon the action specified, the oVTCS CDS database server will terminate, reset itself, retry the operation, or simply ignore the error and continue.

User Response: If the problem persists, then check the MVS system log for TCP/IP stack issues.

SLS0787I

`oVTCS server transaction error; `*`CCCCCCCC`*

Explanation: The oVTCS CDS database server encountered an error processing an oVTCS transaction or response.

System Action: Communication with the oVTCS client is reset.

User Response: If the problem persists, then contact StorageTek Software Support.

SLS0788I

`oVTCS protocol failure: `*`CCCCCCCC`*

Explanation: The oVTCS CDS database server detected a serious error communicating with the oVTCS client or processing an oVTCS request. This unexpected error or breach in protocol has affected the oVTCS server integrity.

System Action: Communication with the oVTCS client is reset.

User Response: If the problem persists, contact StorageTek Software Support.

SLS0789I

`oVTCS client has held the CDS reserve for `*`NNN`*` seconds`

Explanation: The oVTCS CDS database server detected a long database reserve initiated by the oVTCS client. This is abnormal and exceeds expected CDS reserve duration.

System Action: The CDS reserve is released and communication with the oVTCS client is reset.

User Response: Check log files for the oVTCS client and HSC. If the problem persists, contact StorageTek Software Support.

SLS0790I

`oVTCS CDS server cannot continue; CCCCCCCC`

Explanation: The oVTCS CDS database server encountered a serious error and cannot continue.

System Action: The oVTCS CDS database server terminates.

User Response: Issue the DBSERVER START command with the appropriate parameters to re-start the oVTCS CDS database server, and contact StorageTek Software Support.

SLS0791

`oVTCS CDS server reset complete; awaiting new connection`

Explanation: The oVTCS CDS database server stopped and then restarted itself in response to an unexpected event or operator DBSERVER RESET command.

System Action: The existing oVTCS client connection is closed and the connection process restarted. The oVTCS server is now ready for the oVTCS client to reconnect.

User Response: Check to MVS or HSC logs for the root cause of the reset event.

SLS0792I

`oVTCS client connection accepted from CCCCCCCC`

Explanation: The oVTCS CDS database server accepted a new socket connection from IP address *CCCCCCCC*.

System Action: The oVTCS CDS database server is now ready to process requests from the specified client.

User Response: None.

SLS0793I

```
oVTCS Server status:
oVTCS Server started: DD/MM HH:MM:SS
Socket listener port: NNNNN
CDS reserve time in seconds: NNNN
CDS asynchronous read tasks: NN
Data trace length in bytes: NNNNN
Number of input messages: NNN,NNN,NNN
Number of output messages: NNN,NNN,NNN
Number of input bytes NNN,NNN,NNN{K|M}
Number of output bytes: NNN,NNN,NNN{K|M}
Number of process resets: NNN,NNN,NNN
Client connected MM/DD HH:MM:SS from NN.NN.NN.NN
```

Explanation: The DBSERVER command specified the LIST option.

System Action: The oVTCS CDS database server settings and status are displayed.

User Response: None.

Starting/Stopping the VSM console Message Processor

The SMC VMSG command starts or stops the SMC oVTCS message processor.

VMSG command

Interfaces:

- Console, utility, SMCCMDS data set, SMCPARMS data set
- UUI Support: Yes (no XML/CSV output)

Subsystem Requirements:

Active SMC required.

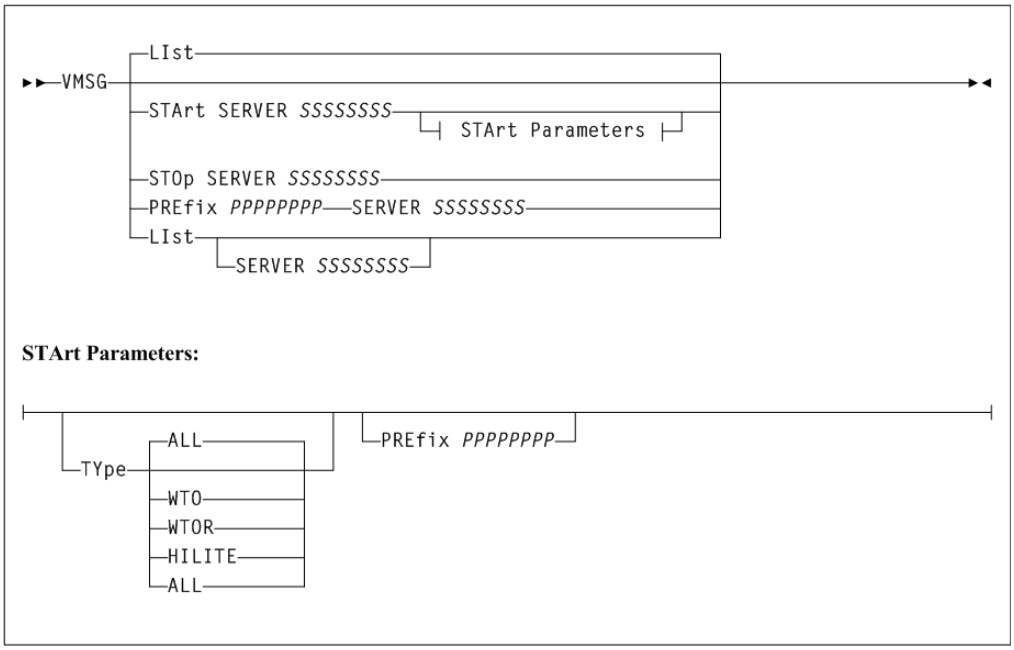
Description

The SMC `VMSG` command is used to start and stop the Virtual Storage Manager console (VSMc) message processor client. The VMSG message processor client allows the local SMC subsystem to receive and reply to messages issued by the remote VSM console server.

Syntax

The following figure shows the syntax for the `VMSG` command:

Figure 6–4 VMSG command syntax



Parameters

LIst

optionally, displays `VMSG` task status information.

SERVER *SSSSSSSS*

optionally, lists only the VMSG processor for the named server. If specified, then the named server must have been previously defined by an SMC `SERVer` command.

`LIst` is the default when the `STArt`, `STOp`, or `PREfix` keywords are not specified.

START

optionally, starts a VMSG message processor client.

SERVER *SSSSSSSS*
specifies the server name for the VMSG processor. The server name is required and must have been previously defined by an SMC `SERVer` command.

TYPE (*type-list*)
optionally, defines a type-list containing one or more type(s) of messages that are received by the `VMSG` client.

Specify one or more of the following in the *type-list*, using a comma to separate values:

ALL
Receive all messages. If `ALL` is specified, it cannot be specified with any other message types. This is the default.

HILITE
Receive highlighted WTO messages.

WTO
Receive non-highlighted WT0 messages.

WTOR
Receive messages that require a reply.

PREfix *PPPPPPPP*
optionally, specifies the message prefix identifier that will identify messages from this VSM console server in the SMC subsystem log. If not specified, then the specified `server` name is used as the message prefix. The prefix can be a maximum of 8 characters in length and can contain any combination of the following:

- A-Z
- 0-9
- @#$,.()+-=<|!;%>?:

STOP
optionally, stops a VMSG message processor client.

SERVER *SSSSSSSS*
specifies the server name for the `VMSG` processor. The server name is required and must have been previously defined by an SMC `SERVer` command.

PREfix *PPPPPPPP*
optionally, specifies the message prefix identifier that will identify messages from this VSM console server in the SMC subsystem log. If not specified, then the specified server name is used as the message prefix. The prefix can be a maximum of 8 characters in length and can contain any combination of the following:

- A-Z
- 0-9
- @#$,.()+-=<|!;%>?:

SERVER *SSSSSSSS*
specifies the server name for the VMSG processor. The server name is required and must have been previously defined by an SMC `SERVer` command.

VMSG Messages

SMC0284

CCCCCCCC command parameter=*PPPPPPPP* value=*VVVVVVVV* is invalid; *RRRRRRRRRR*

Level: 0

Explanation: An invalid value *VVVVVVVV* was specified for the parameter *PPPPPPPP* of the *CCCCCCCC* command. The value was invalid because of the command context.

System Action: None.

User Response: Correct the parameter value and resubmit the command.

SMC0285

VMSG task for server *SSSSSSSS* [START|STOP|message prefix updated]

Level: 0

Explanation: VMSG task for server *SSSSSSSS* was either STARTED, STOPPED, or the message prefix was updated.

System Action: None.

User Response: None.

SMC0286

VMSG server *SSSSSSSS* exception reason: *RRR...RRR*

Level: 4

Explanation: The VMSG task for server *SSSSSSS* encountered an exception while processing the request.

System Action: The VMSG request is retried.

User Response: Investigate the cause of the error. If necessary, stop and restart the VMSG task associated with the server.

SMC0287

PPPPPPPP SSS...SSS

Level: 0

Explanation: This message is received in response to a VMSG task. The *PPPPPPPP* is the specified message prefix or the server name, if the message prefix is not specified.

System Action: None.

User Response: See the associated product for specific messages.

SMC0304

VMSG TASK STATUS:
TAPEPLEX=*CCCCCCCC* SERVER=*CCCCCCCC*
Prefix=*PPPPPPPP* Msg types=*MMM...MMM*
Status: *SSSSSSSS*
Started: *mon dd hh:mm:ss*
Last msg: *mon dd hh:mm:ss*
WTOs=*NNNNNN* WTORs=*NNNNNN* DOMs=*NNNNNN*

Level: 0

Explanation: An SMC VMSG LIst command was issued. The SMC0304 multiline message lists the status of each VMSG task.

System Action: None.

User Response: None.

Off-loading VSM console SMF Records

The SMC SMCUSMF utility enables you to offload SMF records from a VSM console server. The SMF records are placed in a z/OS data set with the following DCB attributes:

- recfm - VB
- lrecl - 27990
- blksize - 27994

Refer to the ELS publication *ELS Programming Reference* for information about these HSC/VTCS SMF records.

Sample JCL

The following JCL sample executes the SMCUSMF utility:

Example 6–1 JCL to Execute the SMCUSMF Utility

```
//jobname JOB (account),REGION=0M
//S1 EXEC PGM=SMCUSMF,PARM='pgmparms'
//STDOUT DD SYSOUT=*
//SMCSMF DD DISP=SHR,DSN=yoursmf.output.dataset
```

Parameters

The following SMCUSMF utility parameters may be specified for *pgmparms* in the sample JCL:

NOHDR
optionally, specifies that the STDOUT report headings and pagination carriage control are not produced.

LINECNT(*nn*)
optionally, specifies STDOUT report pagination line count setting.

nn
the number of lines per page. Valid values are 10-99.

SERVer(*ssss*)
specifies the name of the VSM console SERVER from which SMF records are to offloaded. The specified SERVER must be the ACTIVE server for the VSM console TapePlex. SERVer is a required parameter.

ssss
the server name.

BEGIN(*begin-date*:*begin-time*)
specifies the beginning of the period of SMF record creation. This is the server date and time.

begin-date
the starting date, expressed in *yyyymmdd* format.

yyyymmdd
the beginning date.

TODAY
specifies TODAY as the beginning date.

YESTERDAY
specifies YESTERDAY as the beginning date.

begin-time
the starting time-of-day (24-hour value, expressed in *hhmmss* format. The allowable range for begin-time is 000000 to 235959. The default value is 000000.

END(*end-date*:*end-time*)
optionally, specifies the end of the period of SMF record creation. This is the server date and time.

end-date
the ending date, expressed in *yyyymmdd* format.

yyyymmdd
the ending date.

TODAY
specifies TODAY as the ending date.

YESTERDAY
specifies YESTERDAY as the ending date.

end-time
the ending time-of-day (24-hour value, expressed in *hhmmss* format. The allowable range for the end-time parameter is 000000 to 235959. The default value is 235959.

SMFTYPE(*nnn*)
optionally, specifies the SMF record type for the output records.

nnn
the SMF record type. Valid values are 128 to 255. The default is 255.

Required Data Set Definition (DD) Statements

The following DD statements are required:

STDOUT
The `SMCUSMF` utility reports final completion code and any error messages in the `STDOUT` data set.

SMCSMF
The `SMCUSMF` utility places SMF records in the `SMCSMF` data set. This is a variable blocked file. The DCB attributes must be:

```
DCB=(RECFM=VB,LRECL=27990,BLKSIZE=27994)
```

SMCUSMF Usage

If the VSM console TapePlex is defined with two SERVERS and each server has functioned as the active server for that TapePlex since the last run of SMCUSMF, you must run the following procedure to collect all SMF records from both servers.

With SERVER1 active and SERVER2 inactive:

1. Run `SMCUSMF` with `SERVER(SERVER1)`.
2. Disable SERVER1:

   ```
   SERVER NAME(SERVER1) DISABLE
   ```

   This command makes SERVER2 active.
3. Run `SMCUSMF` with `SERVER(SERVER2)`.
4. Re-enable SERVER1:

   ```
   SERVER NAME(SERVER1) ENABLE
   ```

SERVER2 remains active but SERVER1 is eligible to become the active server if SERVER2 becomes inaccessible. To make SERVER1 the active server once again (if desired), disable SERVER2 and then enable it once SERVER1 has become the active server for the TapePlex.

SMC Messages

The following describes messages issued by SMC. These messages are identified by the "SMC" prefix.

SMC9000

```
Copyright (c) YYYY, YYYY, Oracle and/or its affiliates. All rights reserved.
```

Level: 0

Explanation: The XAPI communications server has been started.

System Action: None.

User Response: None.

SMC9001

```
Communication server initialization starting
```

Level: 0

Explanation: The XAPI communications server startup has commenced.

System Action: None.

User Response: None.

SMC9002

```
Communication server initialization complete
```

Level: 0

Explanation: The XAPI communications server startup has completed.

System Action: None.

User Response: None.

SMC9003

`Communication server release=N.N.N active on host=HHHHHHHH, port=NNNN, TapePlex=PPPPPPPP`

Level: 0

Explanation: The XAPI communications server status message lists the release, host name, listening port number, and TapePlex name. The status message is displayed at startup and once each day after midnight.

System Action: None.

User Response: None.

SMC9004

`Error allocating shared memory segment, key=XXXXXXXX, errno=NN (CCCC...CCCC); { server terminating | RESET specified, continuing | EXCL not specified, continuing}`

Level: 0

Explanation: The XAPI communications server encountered the indicated error while attempting to define a required shared memory segment during XAPI communications server startup.

System Action: Depending upon the XAPI communications server startup option(s), the shared memory segment may be required as exclusive or shared. If the exclusive option was specified (EXCL), the server will terminate. Otherwise, the XAPI communications server will continue startup by sharing (i.e. resetting) the indicated shared memory segment.

User Response: When EXCL is specified, it prevents the startup of a duplicate XAPI communications server when one is already active. If you are certain that the XAPI communications is not already started then you may specify the RESET startup option. Alternatively, you may use UNIX facilities to remove the existing IPC shared memory segment.

SMC9005

`Error attaching shared memory segment, id=XXXXXXXX, errno=NN (CCCC...CCCC); SSSSSSSS terminating`

Level: 0

Explanation: An XAPI communications server task encountered the indicated error while attempting to attach a required shared memory segment during XAPI communications execution.

System Action: The indicated XAPI communications server service, *SSSSSSSS*, terminates.

User Response: Use UNIX services to determine the status of the IPC shared memory segment. If the shared memory segment has been inadvertently removed, then a restart of the XAPI communications server is required.

SMC9006

`Error initializing CCCCCCCC semaphore, errno=NN (CCCC...CCCC); server terminating`

Level: 0

Explanation: The XAPI communications server encountered the indicated error while attempting to initialize a required semaphore during XAPI communications server startup.

System Action: XAPI communications server startup is terminated.

User Response: Use the indicated `errno` and reason to determine why the semaphore could not be initialized.

SMC9007

`Error in` *`EEEEEEEE`* `variable; using { default | truncated } value=`*`VVVV...VVVV`*

Level: 0

Explanation: The XAPI communications server encountered an error obtaining the indicated environment variable, *EEEEEEEE*.

System Action: The default or truncated value, *VVVV...VVVV*, will be used for the indicated environment variable, *EEEEEEEE*.

User Response: Use UNIX services to determine the value of the indicated environment variable, *EEEEEEEE*.

SMC9008

`Error writing WTO mque id=`*`QQQQ...QQQQ`* `errno=`*`NN (CCCC....CCCC)`* `trying printf`

Level: 0

Explanation: The XAPI communications server encountered the indicated error while attempting to queue a message for output by the VSMc WTO message service.

System Action: The XAPI communications server will write the message to `stdout` instead.

User Response: Use UNIX services to determine the status of the indicated IPC message queue, *QQQQ...QQQQ*.

SMC9009

`Error msgsnd diag message queue=`*`QQQQ...QQQQ`* `errno=`*`NN (CCCC...CCCC)`*`; { log service | trace service} disabled trying printf`

Level: 0

Explanation: The XAPI communications server encountered the indicated error while attempting to queue a diagnostic log or trace record for output to the XAPI communications server log and trace service.

System Action: The XAPI communications server log service or trace service will be disabled.

User Response: Use UNIX services to determine the status of the indicated IPC message queue, *QQQQ...QQQQ*.

SMC9010

`Error { creating | opening | reading | writing | retrying } file=`*`FFFF...FFFF`* `errno=`*`NN (CCCC...CCCC)`*`; SSSSSSSS terminating`

Level: 0

Explanation: The XAPI communications server encountered the indicated error processing file, *FFFF...FFFF*.

System Action: The indicated XAPI communications server service, *SSSSSSSS*, is terminated.

User Response: Use the indicated `errno` and reason to determine why the file operation failed.

SMC9011

`{ log | trace } file at` *NNNNN* `bytes`

Level: 8

Explanation: The XAPI communications server has written the indicated number of bytes to the log or trace file.

System Action: None.

User Response: None.

SMC9012

`Unknown message type=`*NNNN* `on diag message queue=`*QQQQ...QQQQ*`; message ignored`

Level: 04

Explanation: The XAPI communications server diagnostic service has encountered an unknown message type, *NNNN*, in its queue, *QQQQ*...*QQQQ*.

System Action: The unknown message is ignored.

User Response: If the problem persists contact StorageTek Software Support.

SMC9013

`Communication server termination starting`

Level: 0

Explanation: The XAPI communications server termination has commenced.

System Action: None.

User Response: None.

SMC9014

`Communication server terminating { work | service } process` *PPPPPPPP=NNNNN*

Level: 0

Explanation: During XAPI communications server termination, the indicated process *PPPPPPPP* (pid=*NNNNN*) did not terminate itself as requested.

System Action: The indicated process is killed and termination continues.

User Response: None.

SMC9015

`Communication server termination complete`

Level: 0

Explanation: The XAPI communications server has completed its termination process.

System Action: None.

User Response: None.

SMC9016

`ftok errno=NN (CCCC...CCCC)` for `{ WTO message queue | diagnostic message queue | HTTPCVT } from path=FFFF...FFFF; server terminating`

Level: 0

Explanation: The XAPI communications server encountered the indicated `ftok` error for the file path, *FFFF...FFFF*.

System Action: XAPI communications server startup is terminated.

User Response: Use the indicated `errno` and reason to determine why the `ftok` operation failed.

SMC9017

`Internal error; file=SSSS...SSSS[NNNN], function=FFFFFFFF, RRRR...RRRR {errno=NN (CCCC...CCCC) }`

Level: 0

Explanation: The XAPI communications server encountered an internal error in source file *SSSS...SSSS* at line *NNNN* in function *FFFFFFFF*. The reason *RRRR...RRRR* and possible `errno` are also displayed.

System Action: The current operation is terminated.

User Response: Contact StorageTek Software Support.

SMC9018

`Error starting work process smcvcvt; no free HTTPREQ`

Level: 0

Explanation: The XAPI communications server received a new transaction request, but the request could not be processed because the server is at its task limit.

System Action: The new transaction request is rejected.

User Response: Distribute the work load among multiple XAPI communication servers.

SMC9019

`Abnormal termination; process=NNNNN, signal=NN (CCCC...CCCC)`

Level: 0

Explanation: The XAPI communications server process *nnnn* was terminated with the unexpected signal NN.

System Action: The current request is terminated.

User Response: Gather the diagnostics indicated in the SMC9020 and SMC9021 messages and contact StorageTek Software Support.

SMC9020

NNN `stack trace entries returned for process=NNNNN`

Level: 0

Explanation: The XAPI communications server process *NNNNN* was terminated.

System Action: *NNN* backtrace entries were available for process *NNNNN* and are listed.

User Response: Gather the diagnostics indicated in the SMC9020 and SMC9021 messages and contact StorageTek Software Support.

SMC9021

```
Core dump { generated to file: FFFF...FFFF |
requested but could not be written |
requested but could not be renamed |
request failed, errno=NN (CCCC...CCCC) }
```

Level: 0

Explanation: An XAPI communications server process was terminated.

System Action: The XAPI communications server requested a core dump image. The core dump image result is displayed.

User Response: Gather the diagnostics indicated in the SMC9020 and SMC9021 messages and contact StorageTek Software Support.

SMC9022

CCCCCCCC `command received`

Level: 8

Explanation: The *CCCCCCCC* operator command was received by the XAPI communications server.

System Action: *CCCCCCCC* command processing continues.

User Response: None.

SMC9023

CCCCCCCC `command RC=`*NN*

Level: 8

Explanation: The *CCCCCCCC* operator command was processed by the XAPI communications server and completed with return code *NN*.

System Action: None.

User Response: None.

SMC9024

CCCCCCCC `is an invalid command`

Level: 0

Explanation: The *CCCCCCCC* operator command was input to the XAPI communications server but *CCCCCCCC* is not a valid command verb.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9025

CCCCCCCC `command requires a value`

Level: 0

Explanation: The *cccccccc* operator command was input to the XAPI communications server without a value but the *CCCCCCCC* command requires a value.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9026

VVVVVVVV is an invalid value for the *CCCCCCCC* command

Level: 0

Explanation: The *CCCCCCC* operator command was input to the XAPI communications server with invalid value *VVVVVVVV*.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9027

KKKKKKKK=VVVVVVVV

Level: 0

Explanation: An operator command was input to the XAPI communications server that resulted in a list of values being displayed. Keyword or command *KKKKKKKK* has value *VVVVVVVV*.

System Action: None.

User Response: None.

SMC9028

Startup parameter *PPPPPPPP* successfully processed

Level: 0

Explanation: The XAPI communications server executable was initiated with command line option *PPPPPPPP* which was successfully processed at startup.

System Action: None.

User Response: None.

SMC9029

Startup parameter *PPPPPPPP* { is invalid | requires a value | contains an invalid value}

Level: 0

Explanation: The XAPI communications server executable was initiated with command line option *PPPPPPPP* which was not successfully processed at startup for the indicated reason.

System Action: The command line option *PPPPPPPP* is rejected but startup continues.

User Response: Correct the command line option(s).

SMC9030

Startup parameter *PPPPPPPP* is mutually exclusive with *XXXXXXXX*

Level: 0

Explanation: The XAPI communications server executable was initiated with command line options *PPPPPPPP* and *XXXXXXXX*, but *PPPPPPPP* and *XXXXXXXX* cannot be specified together.

System Action: The command line option *PPPPPPPP* is accepted, command line option *XXXXXXXX* is rejected, but startup continues.

User Response: Correct the command line option(s).

SMC9031

```
Line parse error={ mismatched or invalid quotes detected |
mismatched or invalid parenthesis detected |
maximum token number exceeded | parameter truncated }
```

Level: 0

Explanation: An operator command was input to the XAPI communications server but the command line could not be processed because of the indicated parse error.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9032

```
XAPI PORT=NNNNN IPADDRESS=NNN.NNN.NNN.NNN HOST=HHHH...HHHH
MAXCLIENTS=NNN XSECURITY={ ON | OFF }
Total: I/Os=NNNNNN bytes=NNNNNN accepts=NNNNNN intervals=NNNNNN
Total: processed input reqs=NNNNNN rejects=NNNNNN
Last:  I/Os=NNNNNN bytes=NNNNNN accepts=NNNNNN
High:  I/Os=NNNNNN bytes=NNNNNN accepts=NNNNNN tasks=NNNNNN
Avg:   I/Os=NNNNNN bytes=NNNNNN accepts=NNNNNN
Total: errs=NNNNNN retries=NNNNNN
Total: maxclient errs=NNNNNN other errs=NNNNNN xsec errs=NNNNNN
```

Level: 0

Explanation: An XAPI communications server XAPI LIST I/O command was received. The current XAPI settings are displayed along with I/O and error statistics.

System Action: None.

User Response: None.

SMC9033

```
Current tasks:
Name          Pid   Tid         Count  Last Time       Status
smcvmai       NNNNN XXXXXXXX  NNNNNN  MM/DD HH:MM:SS  SSSS...SSSS
smcvwts       NNNNN XXXXXXXX  NNNNNN  MM/DD HH:MM:SS  SSSS...SSSS
smcvdts       NNNNN XXXXXXXX  NNNNNN  MM/DD HH:MM:SS  SSSS...SSSS
smcvops       NNNNN XXXXXXXX  NNNNNN  MM/DD HH:MM:SS  SSSS...SSSS
smcvmon       NNNNN XXXXXXXX  NNNNNN  MM/DD HH:MM:SS  SSSS...SSSS
smcvlis       NNNNN XXXXXXXX  NNNNNN  MM/DD HH:MM:SS  SSSS...SSSS
smcvwrk-NNNN NNNNN XXXXXXXX  NNNNNN  MM/DD HH:MM:SS  SSSS...SSSS
```

Level: 0

Explanation: An XAPI communications server XAPI LIST TASKS command was received. The current XAPI communications server system and work tasks are displayed along with their execution counts, and status. Multiple `smcvwrk-`*NNNN* work tasks may be listed depending upon work load and process `hi-water`.

System Action: None.

User Response: None.

SMC9034

```
process reuse required for cmd server; retrying
```

Level: 0

Explanation: An operator command was input to the XAPI communications server but there are no free tasks available to process the request.

System Action: The XAPI communications server tries to find an available reusable task to process the request.

User Response: None.

SMC9035

```
task recovery failed for cmd server; now in single user mode
```

Level: 0

Explanation: An operator command was input to the XAPI communications server but there are no free tasks, and no available reusable tasks available to process the request.

System Action: The XAPI communications server operator command service now processes the request in single task mode.

User Response: None.

SMC9036

```
{ Thread XXXXXXXX | Process NNNNN } active at termination
```

Level: 0

Explanation: During XAPI communications server termination, the indicated thread or process was active at termination after the initial XAPI communications server termination signal.

System Action: The indicated process is killed and termination continues.

User Response: None.

SMC9037

```
Server status:
Server name=CCCC release=N.N.N version=CCCC started on MM/DD ...
TapePlex=CCCCCCCC type={ ACSLS | oVTCS }
RLIMITM=NNN RLIMITS=NNN RLIMITW=NNN
Task mode=MMMM (CCCC...CCCC) signal handling={ VTCS | SMCV }
Work task={ PERMWORK | TERMWORK } (CCCC...CCCC)
System name=SSSS release=NN machine=MMMM ({ little | big } endian)
System version=CCCC...CCCC
rlimit_stack=NNN rlimit_data=NNN rlimit_as=NNN rlimit_nproc=NNN
...more rlimit values
SSCVT shared segment key=XXXXXXXX id=NNNNNN size=NNNN
CVT   shared segment key=XXXXXXXX id=NNNNNN size=NNNN
WTO  message queue  key=XXXXXXXX id=NNNNNN
DIAG  message queue  key=XXXXXXXX id=NNNNNN
```

Level: 0

Explanation: An XAPI communications server `XAPI LIST SERVER` command was received. The current XAPI communications server settings and environment are displayed along its IPC resources.

System Action: None.

User Response: None.

SMC9038

```
control block name:
XXXXXXXX +0000| XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX|CCCC...CCCC|
XXXXXXXX +0010| XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX |CCCC...CCCC|
...
```

Level: 0

Explanation: An XAPI communications server `XAPI LIST CB` command was received. The specified control block is listed in character hexadecimal in 16 byte increments.

System Action: None.

User Response: None.

SMC9039

```
malloc() failure, bytes=NNNN, request=CCCC...CCCC;
{ transaction lost | csv output lost | XML parse failure |
HTTP metadata lost | work task terminated | request terminated }
```

Level: 0

Explanation: The XAPI communications server attempted to `malloc` *NNNN* bytes for request or control block *CCCC...CCCC* but storage was not available.

System Action: The transaction or request is terminated.

User Response: Use appropriate Unix commands to determine memory usage and contact StorageTek Software Support.

SMC9040

```
ACSLS cp_proc_int failure=NNNN; work task terminated
```

Level: 0

Explanation: The XAPI communications server attempted to invoke the ACSLS `cl_proc_init` RPC service during process initiation, but the request failed with the indicated return code.

System Action: The transaction or request is terminated.

User Response: Ensure that ACSLS is active.

SMC9041

```
Communication error: { TCP/IP cccc failure ( reqId=XXXXXXXX ... ) |
Unsuccessful login from CCCC...CCCC port=NNNNN |
TCP/IP bind failure; port=NNNNN, socket=NN, CCCC...CCCC; retrying |
TCP/IP accept failure; port=NNNNN, socket=NN, CCCC...CCCC |
requests=NNN exceeds MAXCLIENTS=NNN;
rejected connection from CCCC...CCCC |
free HTTPREQ error; rejected connection from CCCC...CCCC |
work task start error; rejected connection from CCCC...CCCC |
AF_UNIX accept failure; socket=NN, file=CCCC...CCCC; errno=NN |
```

```
XML parse failure; reqId=XXXXXXXX |
work task start error; retrying |
XAPI work task limit exceeded |
command listener attach failure; start work task error }
```

Level: 0

Explanation: The XAPI communications server received a request but the indicated communication error caused the request to be rejected. Most of the indicated errors are transient errors caused by internal or external resource constraints. In most cases the client will retry the rejected request.

System Action: The transaction or request is rejected.

User Response: None.

SMC9042

```
Invalid format for the CCCCCCCC command
```

Level: 0

Explanation: The *CCCCCCCC* operator command was input to the XAPI communications server but the command contained too many or too few tokens to be a valid command.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9043

```
IPC error: { socketpair failure=NN-NN; errno=NN (CCCC...CCCC) |
sem_init failure; errno=NN (CCCC...CCCC), HTTPTASK=NNNN |
sem_wait failure; errno=NN (CCCC...CCCC), HTTPTASK=NNNN |
sem_timedwait failure; errno=NN (CCCC...CCCC), HTTPTASK=NNNN |
sem_post failure; errno=NN (CCCC...CCCC), HTTPTASK=NNNN }
```

Level: 0

Explanation: The XAPI communications server received a request but the indicated IPC error caused the request to be rejected.

System Action: The transaction or request is terminated.

User Response: Contact StorageTek Software Support.

SMC9044

```
Invalid HOSTNAME specified; header=HHHHHHHH, actual=AAAAAAAA
```

Level: 8

Explanation: The XAPI communications server received a request but the host name specified in the XAPI request header, *HHHHHHHH*, does not match the actual `gethostbyaddr()` host name, *AAAAAAAA*.

System Action: None.

User Response: None.

SMC9045

```
Request id=XXXX pid=NNNN (CCCC...CCCC) cancelled;
RC=NNNN reason=NNNN
```

Level: 0

Explanation: The XAPI communications server received the *CCCC...CCCC* request but the request was terminated within the VSMc PGMI processor with the indicated return and reason codes.

System Action: The request is terminated.

User Response: Check the VSMc logs for the cause of the failure.

SMC9046

`XAPI server not active`

Level: 0

Explanation: An `XCMD` command was input to the ACSLS `cmd_proc` executable, but the XAPI server was not active to receive the command.

System Action: The request is rejected.

User Response: Start the ACSLS XAPI server.

SMC9047

`Startup file=`*`CCCC...CCCC`* `does not exist`

Level: 0

Explanation: The XAPI communications server was started, but the indicated file containing startup and initialization commands does not exist.

System Action: XAPI communications startup continues.

User Response: Move your startup file to the indicated path.

SMC9048

`CCCCCCCC` `command { not allowed from operator |`
`not allowed from file | not allowed from XCMD |`
`not allowed from VSM }`

Level: 0

Explanation: The *CCCCCCCC* operator command was input to the XAPI communications server but the command is not allowed from the indicated command origin.

System Action: The command is rejected.

User Response: Re-enter the command from an allowed origin.

SMC9049

`{ XCLIENT | XUDB } record { for IPADDRESS` *`nnn.nnn.nnn`* `added |`
`for IPADDRESS` *`nnn.nnn.nnn`* `updated |`
`(suppressed) updated |`
`(suppressed) updated in VSMc |`
`(suppressed) added |`
`(suppressed) added in VSMc |`
`(suppressed) exists; updated in VSMc |`
`(suppressed) add error; RC=`*`nn`*`, reason=`*`cccc...cccc`* `|`
`(suppressed) deleted |`
`(suppressed) deleted from VSMc }`

Level: 0

Explanation: An XCLIENT or XUDB operator command was input to the XAPI communications server.

System Action: The XCLIENT or XUDB record is updated, added, or deleted in the XAPI communications server shared memory, or VSMc tables as indicated. The XUDB user information is listed as (suppressed) in the XAPI communications server log.

User Response: None.

SMC9050

```
No { XCLIENT | XUDB } records to list
```

Level: 0

Explanation: An XCLIENT LIST or XUDB LIST operator command was input to the XAPI communications server but there are no records of the specified type to list.

System Action: None.

User Response: None.

SMC9051

```
{ no matching | matching } XUDB record
{ found for update | found for delete | already exists }
```

Level: 0

Explanation: An XUDB ADD, UPDATE, or DELETE operator command was input to the XAPI communications server but the record already exists (for ADD), or does not exist (for UPDATE or DELETE).

System Action: None.

User Response: Correct and re-enter the command.

SMC9052

```
{ No XUDB(s) defined; user(s) defined in VSMc |
No XCLIENT(s) defined; XAPI server using VSMc definitions |
No XUDB(s) or XCLIENT(s) defined; XAPI server will reject
all requests }
```

Level: 0

Explanation: An XUDB DELETE operator command was input to the XAPI communications server with the result that XAPI security user ids are no longer defined.

System Action: In the absence of other application security, such as from VSMc, all incoming requests may be rejected.

User Response: Validate that either XAPI communications server XSECURITY is OFF, or appropriate user(s) are defined in VSMc.

SMC9053

```
Communication server terminating; invalid startup parameters
```

Level: 0

Explanation: The XAPI communications server was started with invalid command line options.

System Action: XAPI communications server startup is terminated.

User Response: Correct the command line options and restart.

SMC9054

Startup file=*CCCC...CCCC* processing starting

Level: 0

Explanation: The XAPI communications server was started and the file of startup and initialization commands, *CCCC...CCCC*, has been opened for processing.

System Action: XAPI communications server startup continues.

User Response: None.

SMC9055

Startup file=*CCCC...CCCC* processing complete; RC=*NN*

Level: 0

Explanation: The XAPI communications server was started and the file of startup and initialization commands, *CCCC...CCCC*, has been processed. The indicated return code is the highest return code for all command(s) processed in the file.

System Action: XAPI communications server startup continues.

User Response: None.

SMC9101

Invalid keyword *KKKKKKKK* for the *CCCCCCCC* command

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified invalid keyword *KKKKKKKK*.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9102

Invalid value *VVVVVVVV* for keyword or tag *KKKKKKKK* of the *CCCCCCCC* command

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified keyword *KKKKKKKK* with an invalid value *VVVVVVVV*.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9103

Keyword or tag *KKKKKKKK* of the *CCCCCCCC* command requires a value

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified keyword *KKKKKKKK* without a required value.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9104

Unexpected format for positional parameter in command *CCCCCCCC*

Level: 0

Explanation: The positional parameter of command *CCCCCCCC* is not correctly formatted.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9105

Duplicate keyword or tag *KKKKKKKK* specified for the *CCCCCCCC* command

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified keyword *KKKKKKKK* multiple times.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9106

Keyword or tag *KKKKKKKK* of the *CCCCCCCC* command is mutually exclusive with keyword or tag *XXXXXXXX* command

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified multiple keywords, two of which (*KKKKKKKK* and *XXXXXXXX*) are mutually exclusive.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9107

Keyword or tag *KKKKKKKK* of the *CCCCCCCC* command requires keyword or tag *RRRRRRRR* command

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified keyword *KKKKKKKK*, but not the required co-requisite keyword *RRRRRRRR*.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9108

Keyword or tag *KKKKKKKK* of the *CCCCCCCC* command is required command

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that did not specify the required keyword *KKKKKKKK*.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9109

`Invalid range VVVV...VVVV for keyword KKKKKKKK of the CCCCCCCC command`

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified a range value *VVVV...VVVV* for keyword *KKKKKKKK*. However the range value is invalid either because the left value is higher than the right value, or the left and right values have different formats.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9110

`Unrecognized XML tag=TTTTTTTT for the CCCCCCCC command`

Level: 0

Explanation: The XAPI communications server encountered a input request in XML format containing a tag, *TTTTTTTT*, that is not recognized as valid for the *CCCCCCCC* command. This message can be caused when the current software level does not support a tag that was valid in an earlier level, or has not been upgraded to support a new tag.

System Action: The individual parameter is ignored, but command processing continues.

User Response: Verify that the command is correctly specified.

SMC9111

`Value=VVVVVVVV is invalid type for keyword or tag=KKKKKKKK in command=CCCCCCCC`

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that specified an invalid value type for keyword *KKKKKKKK*.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9112

`Keyword or tag=KKKKKKKK may not have a value in command=CCCCCCCC`

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that contained a value for a keyword or XML tag that does not allow a value.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9113

`Length of value=VVVV...VVVV is invalid for keyword or tag=KKKKKKKK in command=CCCCCCCC`

Level: 0

Explanation: The XAPI communications server encountered a command, *CCCCCCCC*, that contained a keyword value *VVVV...VVVV* that was too long.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9114

Error parsing XML values for XML tag=*TTTTTTTT* in command=*CCCCCCCC*; RC=*NNNN*

Level: 0

Explanation: The XAPI communications server encountered an XML format command that contained a value or parse error related to the listed tag. The parse return code is included in the message for diagnostics.

System Action: The command is rejected.

User Response: Correct and re-enter the command.

SMC9115

Error: *EEEE...EEEE; AAAA...AAAA*

Level: 0

Explanation: During XAPI communications server processing, the *EEEE...EEEE* error occurred.

System Action: The *EEEE...EEEE* error caused the system to take the resulting action, *AAAA...AAAA*.

User Response: Contact StorageTek Software Support.

SMC9998

CCCC...CCCC

Level: 12

Explanation: The XAPI communications server has issued a diagnostic message.

System Action: None.

User Response: None.

SMC9999

Error: *EEEE...EEEE; AAAA...AAAA*

Level: 12

Explanation: The XAPI communications server has issued a diagnostic message.

System Action: None.

User Response: None.

oVTCS Messages

The following describes messages issued by oVTCS.

SLS8000I

Unexpected exception thrown: *RRRRRRRRR*

Explanation: A software error has been detected and this has caused an exception to be raised. The reason *RRRRRRRRR* gives details of the error that occurred.

System Action: If necessary, a dump of the process will be taken. Attempts will also be made to recover the failing task or thread.

User Response: Because the error is unexpected, it is possible that any recovery action may not be successful. Therefore check the system and if necessary restart the process that suffered the failure. Refer the problem to StorageTek Software Support.

SLS8001I

PPPPPPPP/NNNN: Uncaught exception terminating thread

Explanation: A software error has been detected in process the *PPPPPPPP* with an Id of *NNNN* and this has caused an exception to be raised. There is no recovery routine active to recover from the error.

System Action: The effected task or thread will be terminated. Typically no additional recovery actions will be performed.

User Response: More than likely, the effected process will need to be restarted. Check for additional messages that might also be a trigger for this error message. Refer the problem to StorageTek Software Support.

SLS8002I

PPPPPPPP/NNNN: Abnormal thread termination: *RRRRRRRRR*

Explanation: A software thread has decided to terminate unexpectedly within process *PPPPPPPP* with an Id of *NNNN*. The reason *RRRRRRRRR* gives details of the triggering error.

System Action: The effected task or thread will be terminated. A dump may also be taken. Typically no additional recovery actions will be performed.

User Response: More than likely, the effected process will need to be restarted. Check for additional messages that might also be a trigger for this error message. Refer the problem to StorageTek Software Support.

SLS8003I

PPPPPPPP/NNNN: Call to terminate()

Explanation: A software error has occurred that means that the run-time environment for the process *PPPPPPPP* with an Id of *NNNN* has decided that it cannot continue.

System Action: The effected process will be terminated. A dump may also be taken. Depending upon circumstances, the system may restart the failed process.

User Response: Check that the failing process restarted. If not then perform a manual stop and start of the process. Check for additional messages that might also be a trigger for this error message. Refer the problem to StorageTek Software Support.

SLS8004I

PPPPPPPP/NNNN: Unexpected exception called

Explanation: A software error has been detected in process *PPPPPPPP* with an Id of *NNNN* and this has caused an exception to be raised. There is no recovery routine active to recover from the error.

System Action: The effected task or thread will be terminated. Typically no additional recovery actions will be performed.

User Response: More than likely, the effected process will need to be restarted. Check for additional messages that might also be a trigger for this error message. Refer the problem to StorageTek Software Support.

SLS8005I

PPPPPPPP/*NNNN*: Call to terminate()

Explanation: A software error has occurred that means that the run-time environment for the process *PPPPPPPP* with an Id of *NNNN* has decided that it cannot continue.

System Action: The effected process will be terminated. A dump may also be taken. Depending upon circumstances, the system may restart the failed process.

User Response: Check that the failing process restarted. If not then perform a manual stop and start of the process. Check for additional messages that might also be a trigger for this error message. Refer the problem to StorageTek Software Support.

SLS8006I

Cannot create TCP/IP socket: *RRRRRRRRR*

Explanation: An attempt was made to create a socket for TCP/IP communications. This failed with the error code *RRRRRRRRR*.

System Action: The function attempting to create the socket will fail.

User Response: This maybe a network or a server resource problem. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8007I

Cannot bind to port *NNNN*: *RRRRRRRRR*

Explanation: An attempt was made to bind to port *NNNN* for TCP/IP communications. This failed with the error code *RRRRRRRRR*. If this reason is 'address already in use', then it probably means that a server component has not completed termination before the replacement was started.

System Action: The function attempting to bind to the port will fail.

User Response: This maybe a network or a server resource problem. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8008I

Listen on port NNNN failed: RRRRRRRRR

Explanation: An attempt was made to listen for connections to port *NNNN* for TCP/IP communications. This failed with the error code *RRRRRRRRR*.

System Action: The function attempting to listen to the port will fail.

User Response: This maybe a network or a server resource problem. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8009I

Cannot create AF_UNIX socket: *RRRRRRRRR*

Explanation: An attempt was made to create a socket for internal communications as a server. This failed with the error code *RRRRRRRRR*.

System Action: The function attempting to create the socket will fail.

User Response: This is probably a server resource problem. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8010I

Cannot bind to file FFFFFFFF: *RRRRRRRRR*

Explanation: An attempt was made to bind to file *FFFFFFFF* for internal communications. This failed with the error code *RRRRRRRRR*. If this reason is 'address already in use', then it probably means that a server component has not completed termination before the replacement was started.

System Action: The function attempting to bind to the file will fail.

User Response: This is probably a server resource problem. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8011I

Listen on file *FFFFFFFF* failed: *RRRRRRRRR*

Explanation: An attempt was made to listen for connections to file *FFFFFFFF* for internal communications. This failed with the error code *RRRRRRRRR*.

System Action: The function attempting to bind to the file will fail.

User Response: This is probably a server resource problem. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8012I

Failed lookup of *HHHHHHHH*:*PPPP*: *RRRRRRRR*

Explanation: An attempt was made to resolve the network address of *HHHHHHHH* and port *PPPP* and this failed with the error code *RRRRRRRRR*.

System Action: The function attempting to resolve the address will fail.

User Response: Check that *HHHHHHHH* is either a valid IPv4 address, IPv6 address or DNS address. Short form DNS addresses are only valid if it can be converted to a full address by using one of the DNS search suffixes. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8013I

Failed connect to *HHHHHHHH*:*PPPP*: *RRRRRRRR*

Explanation: An attempt was made to connect to the network address *HHHHHHHH* and port *PPPP* and this failed with the error code *RRRRRRRRR*. If this reason is 'connection refused', then it probably means that a server component is not running. If this reason is 'no route to host' or 'connection timed out', then it is probably some kind of network problem.

System Action: The function attempting to connect to the address will fail.

User Response: Check that the system *HHHHHHHH* is up. Then check that the network route to system is correct and functional. If the target system is on another sub-net, then the default router must be configured correctly. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8014I

`Cannot create AF_UNIX socket`

Explanation: An attempt was made to create a socket for internal communications as a client.

System Action: The function attempting to create the socket will fail.

User Response: This is probably a server resource problem. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8015I

`Failed connect to FFFFFFFF: RRRRRRRR`

Explanation: An attempt was made to connect to file *FFFFFFFF* for internal communication and this failed with the error code *RRRRRRRR*. If this reason is 'connection refused', then it probably means that a server component is not running.

System Action: The function attempting to connect to the address will fail.

User Response: Check that the service that should connected to file *FFFFFFFF* is up. Additional messages produced around the same time should also give an indication of what function is effected.

SLS8016I

`Accept on port NNNN failed: RRRRRRRR`

Explanation: An attempt was made to accept a connection on port *NNNN* and this failed with the error code *RRRRRRRR*.

System Action: The function attempting to accept the connection will fail. Typically this will cause an additional failure within the server component attempting the connection.

User Response: Additional messages produced around the same time should also give an indication of what function is effected.

SLS8017I

`Connection from SSSSSSSS`

Explanation: A server has received a TCP/IP connection request from the system *SSSSSSSS*.

System Action: Depending upon the nature of the connection, processing will proceed within the server component.

User Response: This is informational only.

SLS8018I

`PPPPPPPP: PGMI server running`

Explanation: The process *PPPPPPPP* has started an instance of the server component for handling command execution.

System Action: Processing of commands now commences.

User Response: None.

SLS8019I

`Process PPPP trace state is SSSS`

Explanation: The tracing state of process *PPPP* has now changed to *SSSS*.

System Action: If 'Off', then tracing has been disabled. If 'On', then the file to which tracing is now being done is reported.

User Response: None.

SLS8020I

PPPPPPPP/*NNNN*: Normal shutdown complete

Explanation: The process *PPPPPPPP* with an Id of *NNNN* has shut down as a result of a service level change.

System Action: None.

User Response: None.

SLS8021I

Dump written to *FFFFFFFF*

Explanation: A software error has occurred and a dump of the failing process has been written to file *FFFFFFFF*.

System Action: If possible, recovery routines will be invoked. Depending upon the nature and reason for the error, the recovery may or may not be successful.

User Response: Additional messages produced around the same time should also give an indication of what function is effected. Refer the problem to StorageTek Software Support.

SLS8022I

Response to message *NNNN* was *RRRRRRRR*

Explanation: The REPLY command has been used against the outstanding message with the Id of *NNNNN*. The response text was *RRRRRRRR*.

System Action: The function awaiting the response to the message will be woken up and passed the relevant text.

User Response: None.

SLS8023I

Message *NNNN* has been deleted

Explanation: The outstanding message with the Id of *NNNNN* has been deleted by the system.

System Action: None.

User Response: None.

SLS8024I

Manual dump of process *PPPP* initiated: *TTTTTTTT*

Explanation: A manual dump of process *PPPP* has been initiated via a command. The reason for the dump was given as *TTTTTTTT*.

System Action: After the dump has been taken, processing should continue.

User Response: Assuming this is at the request of StorageTek Software Support, use the DIAGS command or the GUI to add the generated dump to the relevant support bundle.

SLS8025I

Security failure on *CCCCCCCC* command matching rule on line *LLLL RRRRRRRR*

Explanation: A user attempted to execute *CCCCCCCC* command when matching the PERMIT/ALLOW rule on line *LLLL*. *RRRRRRRR* is additional resources that maybe triggering the failure.

System Action: Execution of the command will be rejected.

User Response: Contact your system administrator and get them to review the PERMIT/ALLOW rules for your User Id.

SLS8026I

Security warning on *CCCCCCCC* command matching rule on line *LLLL RRRRRRRR*

Explanation: A user attempted to execute *CCCCCCCC* command when matching the PERMIT/ALLOW rule on line *LLLL*. *RRRRRRRR* is additional resources that maybe triggering the failure. The matching rule is currently set to just issue a warning.

System Action: Execution of the command continues.

User Response: Contact your system administrator and get them to review the PERMIT/ALLOW rules for your User Id.

SLS8027I

Process *PPPPPPPP*/*NNNN* received termination request

Explanation: The process *PPPPPPPP* with an Id of *NNNN* has received a request to shut down as a result of a service level change.

System Action: Any work the process is performing will be quiesced before the shut down request is honoured. Typically from this point, new requests will be rejected.

User Response: None.

SLS8028I

PPPPPPPP/*NNNN*: Communication failure writing *TTTTTTTT* to logger: *RRRRRRRR*

Explanation: An internal communications error has occurred when process *PPPPPPPP* with an Id of *NNNN* tried to send a message of type *TTTTTTTT* to the logger process. The reason for the failure is *RRRRRRRR*. The most likely reason for this problem, is the logger process failing.

System Action: The process *PPPPPPPP* will reset the connection and attempt to reconnect to the logger process. If the process is not available, then it will retry until the process becomes available. Some messages may have been lost as a result of this failure.

User Response: Check for additional messages that might also be a trigger for this error message.

SLS8029I

Command *CCCCCCCC*: Internal Comms error: *RRRRRRRR*

Explanation: When executing the command *CCCCCCCC*, an internal communication failure with the reason *RRRRRRRR* occurred. This failure could have a number of reasons that are perfectly normal. A typical reason is a external command being cancelled or a service level change whilst a command is executing.

System Action: Depending upon the timing, the command *CCCCCCCC* may or may not continue executing.

User Response: Check for additional messages that might also be a trigger for this error message. Also review what any client systems were doing at the time.

SLS8030I

Connection *NNNNNNNN* authorisation failure: *RRRRRRRR*

Explanation: When authorizing a client over a TCP/IP connection using the name *NNNNNNNN*, there was a failure with the reason *RRRRRRRR*. The previous SLS8017 message will give the network address of the client.

System Action: The connection request will be aborted after a short period of time.

User Response: Check the configuration and setup of the client. Also review what any client systems were doing at the time.

SLS8031I

PPPPPPPP/*NNNN*: Communication failure to CDS proxy: *RRRRRRRR*

Explanation: The process *PPPPPPPP* with an Id of *NNNN* suffered an internal communication failure with the CDS proxy process. This can only occur in shared CDS mode where the CDS manages the connections to the ELS host.

System Action: This will typically cause follow-on errors as communications from the process to the CDS have been compromised and I/Os may have been lost.

User Response: Check the connection to the ELS host from the 'dbserv' process and confirm that the process is running. It maybe necessary to reset things by dropping and raising the service level.

SLS8032I

Connected to CDS server *SSSSSSSS*

Explanation: The CDS proxy process has established a connection with the ELS system with then network address of *SSSSSSSS*.

System Action: Access to the CDS is now permitted and processing will continue.

User Response: None.

SLS8033I

Communication failure to CDS server: *RRRRRRRR*

Explanation: The CDS proxy process has suffered a communications failure talking to the ELS host with a reason of *RRRRRRRR*.

System Action: This will typically cause follow-on errors as communications from other processes to the CDS have been compromised and I/Os may have been lost.

User Response: Check the connection to the ELS host from the 'dbserv' process and confirm that the process is running. Also check that the ELS host is running and that the DBSERVER command has been correctly issued. It maybe necessary to reset things by dropping and raising the service level.

SLS8034I

Trying to reconnect to CDS server *SSSSSSSS*

Explanation: The CDS proxy process is trying to reconnect to the ELS host *SSSSSSSS*.

System Action: This will be done indefinitely until the connection has been established. Until this point, some functions that will require access to the CDS will fail and others will hang.

User Response: Check that the ELS host is running and that the DBSERVER command has been correctly issued.

SLS8035I

`Communication failure to CDS client: RRRRRRRR`

Explanation: The CDS proxy process has detected that one of its client processes has dropped an internal connection for the reason *RRRRRRRR*.

System Action: Processing continues and the client's I/O requests will be dropped.

User Response: Check the other processes for error messages that may give an indication as to the source of the problem.

SLS8036I

`CDS version: VVVVVV Primary DSN: DDDDDDDD`

Explanation: The CDS proxy has successfully connected to the ELS system that is serving up the CDS. The version of ELS is *VVVVVV* and the primary CDS is *DDDDDDDDDD*

System Action: Processing continues.

User Response: None.

SLS8037I

`Cannot create shared memory segment`

Explanation: The process was unable to create the shared memory segment that is used for communication between processes.

System Action: The process will abort its start-up and take a dump.

User Response: Check the other processes for error messages that may give an indication as to the source of the problem. Try rebooting the LDOM.

SLS8038I

`Cannot attach shared memory segment`

Explanation: The process was unable to attach to the shared memory segment that is used for communication between processes. It is possibly incompatible.

System Action: The process will abort its start-up and take a dump.

User Response: Check the other processes for error messages that may give an indication as to the source of the problem. Try rebooting the LDOM.

SLS8039I

`Local configuration change to host NNNN with name SSSSSSSS`

Explanation: The local configuration has been updated and the system has been allocated an Id of *NNNN* and a name of *SSSSSSSS*.

System Action: Processing continues.

User Response: None.

SLS8040I

`Configuration connection from NNNN as host SSSSSSSS`

Explanation: A cluster connection has been received from the system with an Id of *NNNN* and a name of *SSSSSSSS*.

System Action: Processing continues. This includes ensuring that both systems have correc6t configuration information.

User Response: None.

SLS8041I

Configuration connection to *NNNN* as host *SSSSSSSS* using *AAAAAAAA*

Explanation: A cluster connection is been attempted to the system with an Id of *NNNN* and a name of *SSSSSSSS* using the network address *AAAAAAAA*.

System Action: Processing continues. This includes ensuring that both systems have correc6t configuration information.

User Response: None.

SLS8042I

Configuration sent to *NNNN* as host *SSSSSSSS*

Explanation: This system has deduced that it has a more current configuration than the system with the Id of *NNNN* and the name of *SSSSSSSS*. It is therefore uploading the changed configuration to this target in order to bring both systems into line.

System Action: Processing continues. Once the configuration update has been received, notifications will be sent to the various processes to cause them to read in the update.

User Response: None.

SLS8043I

Configuration received from *NNNN* as host *SSSSSSSS*

Explanation: This system with the Id of *NNNN* and the name of *SSSSSSSS* has deduced that it has a more current configuration than this system. It is therefore downloading the changed configuration to this system in order to bring both systems into line.

System Action: Processing continues. Once the configuration update has been received, notifications will be sent to the various processes to cause them to read in the update.

User Response: None.

SLS8044I

Configuration connection to # *NNNN* as host *SSSSSSSS* terminated: *RRRRRRRR*

Explanation: The cluster connection to the system with the Id of *NNNN* and the name of *SSSSSSSS* has terminated for the reason of *RRRRRRRR*.

System Action: Processing continues. Depending upon the reason for the termination, other error messages maybe posted.

User Response: If this is not an expected condition, then check for error messages that may give an indication as to the source of the problem.

SLS8045I

PPPPPPPP/*NNNN*: SQL error: *EEEEEEEE* Return Code: *RRRR*

Explanation: The process *PPPPPPPP* with an Id of *NNNN* has suffered an internal SQL of *EEEEEEEE* when accessing the CDS. The return code from the operation was *RRRR*.

System Action: This will typically cause follow-on errors as access to the CDS will have been compromised and I/Os may have been lost. Depending upon the nature of the error, some retries maybe attempted before giving up. on the operation.

User Response: Check the other processes for error messages that may give an indication as to the source of the problem. It maybe necessary to reset things by dropping and raising the service level. In a clustering environment where there are two systems, errors can be posted if communications to the primary system is lost.

SLS8046I

`CDS unavailable: RRRRRRRR`

Explanation: The current operation was unable to complete because the CDS is unavailable because of the reason *RRRRRRRR*.

System Action: The function attempting to access the CDS will fail.

User Response: Check the connection to the ELS host from the 'dbserv' process and confirm that the process is running. It may be necessary to reset things by dropping and raising the service level.

SLS8047I

`DS access error: RRRRRRRR`

Explanation: The current operation was unable to complete because the CDS access returned the error with a reason of *RRRRRRRR*.

System Action: The function attempting to access the CDS will fail.

User Response: Check for additional messages that might also be a trigger for this error message.

SLS8048I

`PPPPPPPP: Parameter change detected`

Explanation: The process *PPPPPPPP* has been notified of a change to the main parameter file,

System Action: The process will read and process the updated parameter file.

User Response: This is informational only.

SLS8049I

`Restart of process PPPPPPPP detected`

Explanation: The process *PPPPPPPP* has been restarted after previously suffering an uncontrolled termination.

System Action: Processing continues.

User Response: None.

SLS8050I

`Received=RRRR/rrrrrrrr, Sent=SSSS/ssssssss, Duplicate reads=DDDD, Cache reads=CCCC`

Explanation: This reports the statistics for accessing the CDS via an ELS host and is normally issued when the 'dbserv' process is shut down.

The *RRRR/rrrrrrrr* value is the number of messages received from the ELS host and the number of bytes transferred.

The *SSSS/ssssssss* value is the number of messages sent to the ELS host and the number of bytes transferred.

The *DDDD* value is the number of read requests to the ELS host that were suppressed because the same request was already in-flight.

The *CCCC* value is the number of read requests to the ELS host that could be satisfied from a client side cache.

System Action: Processing continues.

User Response: None.

SLS8051I

`Number of CDS I/O operations=`*IIII*`, Response time=`*TTTTTTT*

Explanation: This reports the statistics for accessing the CDS via an ELS host and is normally issued by each process when it stops doing I/O to the CDS. The *IIII* value is the number of CDS I/O operations performed. The *TTTTTTTT* value is the average response time of the CDS I/O requests. At times this may appear lower than expected because some requests are services from a cache or because the request is a duplicate of an existing request.

System Action: Processing continues.

User Response: None.

SLS8052I

`Configuration connection from` *NNNN* `as host` *SSSSSSSS* `rejected`

Explanation: This system with the Id of *NNNN* and the name of *SSSSSSSS* attempted to connect to this system in order to form part of a cluster. This was rejected because the system is not deemed to be part of the cluster.

System Action: Processing continues on this system. On the system attempting to connect, it should drop into a failed state and refuse to start. This is to avoid a split-brain scenario with the CDS.

User Response: Investigate the history of the two systems. Check for additional messages that might also be a trigger for this error message.

When attempting to recover from this situation it is important that a full picture of the state and history is obtained. If this is not done, then it is easy to make the wrong decision and destroy the CDS contents.

SLS8053I

`Allowing MySQL to start:` *RRRRRRRR*

Explanation: The cluster control process has deemed that it is OK to permit access to the SQL based CDS. The reason for the decision is given as *RRRRRRRR*.

System Action: Processing continues and the internal SQL database is started up.

User Response: None.

SLS8054I

`Configuration shutdown - Possible split-brain condition`

Explanation: The cluster control process has detected a condition whereby continuing to start could result in a split-brain condition.

System Action: The clustering process shuts down and this will in turn shut down all other dependent process. Access to the internal SQL CDS is disabled.

User Response: Investigate the history of the systems. Check for additional messages that might also be a trigger for this error message upon this and any other systems.

When attempting to recover from this situation it is important that a full picture of the state and history is obtained. If this is not done, then it is easy to make the wrong decision and destroy the CDS contents.

SLS8055I

`Cluster connection to NNNN down. In single server mode`

Explanation: The connection from the cluster control process to the host with the Id of *NNNN* can either not be established or has failed.

System Action: Processing continues. This does mean that the CDS is only running with a single copy active and that any subsequent failure could be fatal.

Depending upon the reason for the connection loss, there maybe transient errors reported.

User Response: Investigate the history of the systems. Check for additional messages that might also be a trigger for this error message upon this and any other systems.

SLS8056I

`Database state compromised - Check service levels`

Explanation: A check of the internal SQL database state has indicated that not all of the required parts are functional.

System Action: Processing continues. This does mean that the CDS is only running with a single copy active and that any subsequent failure could be fatal.

User Response: Investigate the history of the systems. Check for additional messages that might also be a trigger for this error message upon this and any other systems.

Check that the service levels are correct upon each member of the cluster and adjust if necessary.

SLS8057I

`Process PPPPPPPP/NNNN running`

Explanation: The process *PPPPPPPP* with an Id of *NNNN* has started executing.

System Action: None.

User Response: None.

7

Controlling Contaminants

This chapter discusses the following topics:

Environmental Contaminants

Control over contaminant levels in a computer room is extremely important because tape libraries, tape drives, and tape media are subject to damage from airborne particulates. Most particles smaller than ten microns are not visible to the naked eye under most conditions, but these particles can be the most damaging. As a result, the operating environment must adhere to the following requirements:

- ISO 14644-1 Class 8 Environment.
- The total mass of airborne particulates must be less than or equal to 200 micrograms per cubic meter.
- Severity level G1 per ANSI/ISA 71.04-1985

Oracle currently requires the ISO 14644-1 standard approved in 1999, but will require any updated standards for ISO 14644-1 as they are approved by the ISO governing body. The ISO 14644-1 standard primarily focuses on the quantity and size of particulates as well as the proper measurement methodology, but does not address the overall mass of the particulates. As a result, the requirement for total mass limitations is also necessary as a computer room or data center could meet the ISO 14644-1 specification but still damage equipment because of the specific type of particulates in the room. In addition, the ANSI/ISA 71.04-1985 specification addresses gaseous contaminations as some airborne chemicals are more hazardous. All three requirements are consistent with the requirements set by other major tape storage vendors.

Required Air Quality Levels

Particles, gasses and other contaminants may impact the sustained operations of computer hardware. Effects can range from intermittent interference to actual component failures. The computer room must be designed to achieve a high level of cleanliness. Airborne dusts, gasses and vapors must be maintained within defined limits to help minimize their potential impact on the hardware.

Airborne particulate levels must be maintained within the limits of *ISO 14644-1 Class 8 Environment*. This standard defines air quality classes for clean zones based on airborne particulate concentrations. This standard has an order of magnitude less particles than standard air in an office environment. Particles ten microns or smaller are harmful to most data processing hardware because they tend to exist in large numbers, and can easily circumvent many sensitive components' internal air filtration systems. When computer hardware is exposed to these submicron particles in great numbers they endanger system reliability by posing a threat to moving parts, sensitive contacts and component corrosion.

Excessive concentrations of certain gasses can also accelerate corrosion and cause failure in electronic components. Gaseous contaminants are a particular concern in a computer room both because of the sensitivity of the hardware, and because a proper computer room environment is almost entirely recirculating. Any contaminant threat in the room is compounded by the cyclical nature of the airflow patterns. Levels of exposure that might not be concerning in a well ventilated site repeatedly attack the hardware in a room with recirculating air. The isolation that prevents exposure of the computer room environment to outside influences can also multiply any detrimental influences left unaddressed in the room.

Gasses that are particularly dangerous to electronic components include chlorine compounds, ammonia and its derivatives, oxides of sulfur and petrol hydrocarbons. In the absence of appropriate hardware exposure limits, health exposure limits must be used.

While the following sections will describe some best practices for maintaining an ISO 14644-1 Class 8 Environment in detail, there are some basic precautions that must be adhered to:

- Do not allow food or drink into the area.
- Cardboard, wood, or packing materials must not be stored in the data center clean area.
- Identify a separate area for unpacking new equipment from crates and boxes.
- Do not allow construction or drilling in the data center without first isolating sensitive equipment and any air targeted specifically for the equipment. Construction generates a high level of particulates that exceed ISO 14644-1 Class 8 criteria in a localized area. Dry wall and gypsum are especially damaging to storage equipment.

Contaminant Properties and Sources

Contaminants in the room can take many forms, and can come from numerous sources. Any mechanical process in the room can produce dangerous contaminants or agitate settled contaminants. A particle must meet two basic criteria to be considered a contaminant:

- It must have the physical properties that could potentially cause damage to the hardware.

- It must be able to migrate to areas where it can cause the physical damage.

The only differences between a potential contaminant and an actual contaminant are time and location. Particulate matter is most likely to migrate to areas where it can do damage if it is airborne. For this reason, airborne particulate concentration is a useful measurement in determining the quality of the computer room environment. Depending on local conditions, particles as big as 1,000 microns can become airborne, but their active life is very short, and they are arrested by most filtration devices.

Submicron particulates are much more dangerous to sensitive computer hardware, because they remain airborne for a much longer period of time, and they are more apt to bypass filters.

Operator Activity

Human movement within the computer space is probably the single greatest source of contamination in an otherwise clean computer room. Normal movement can dislodge tissue fragments, such as dander or hair, or fabric fibers from clothing. The opening and closing of drawers or hardware panels or any metal-on-metal activity can produce metal filings. Simply walking across the floor can agitate settled contamination making it airborne and potentially dangerous.

Hardware Movement

Hardware installation or reconfiguration involves a great deal of subfloor activity, and settled contaminants can very easily be disturbed, forcing them to become airborne in the supply air stream to the room's hardware. This is particularly dangerous if the subfloor deck is unsealed. Unsealed concrete sheds fine dust particles into the airstream, and is susceptible to efflorescence -- mineral salts brought to the surface of the deck through evaporation or hydrostatic pressure.

Outside Air

Inadequately filtered air from outside the controlled environment can introduce innumerable contaminants. Post-filtration contamination in duct work can be dislodged by air flow, and introduced into the hardware environment. This is particularly important in a downward-flow air conditioning system in which the subfloor void is used as a supply air duct. If the structural deck is contaminated, or if the concrete slab is not sealed, fine particulate matter (such as concrete dust or efflorescence) can be carried directly to the room's hardware.

Stored Items

Storage and handling of unused hardware or supplies can also be a source of contamination. Corrugated cardboard boxes or wooden skids shed fibers when moved or handled. Stored items are not only contamination sources; their handling in the computer room controlled areas can agitate settled contamination already in the room.

Outside Influences

A negatively pressurized environment can allow contaminants from adjoining office areas or the exterior of the building to infiltrate the computer room environment through gaps in the doors or penetrations in the walls. Ammonia and phosphates are often associated with agricultural processes, and numerous chemical agents can be produced in manufacturing areas. If such industries are present in the vicinity of the data center facility, chemical filtration may be necessary. Potential impact from

automobile emissions, dusts from local quarries or masonry fabrication facilities or sea mists should also be assessed if relevant.

Cleaning Activity

Inappropriate cleaning practices can also degrade the environment. Many chemicals used in normal or “office” cleaning applications can damage sensitive computer equipment. Potentially hazardous chemicals outlined in "Cleaning Procedures and Equipment" on page 7-8 should be avoided. Out-gassing from these products or direct contact with hardware components can cause failure. Certain biocide treatments used in building air handlers are also inappropriate for use in computer rooms either because they contain chemicals, that can degrade components, or because they are not designed to be used in the airstream of a re-circulating air system. The use of push mops or inadequately filtered vacuums can also stimulate contamination.

It is essential that steps be taken to prevent air contaminants, such as metal particles, atmospheric dust, solvent vapors, corrosive gasses, soot, airborne fibers or salts from entering or being generated within the computer room environment. In the absence of hardware exposure limits, applicable human exposure limits from OSHA, NIOSH or the ACGIH should be used.

Contaminant Effects

Destructive interactions between airborne particulate and electronic instrumentation can occur in numerous ways. The means of interference depends on the time and location of the critical incident, the physical properties of the contaminant and the environment in which the component is placed.

Physical Interference

Hard particles with a tensile strength at least 10% greater than that of the component material can remove material from the surface of the component by grinding action or embedding. Soft particles will not damage the surface of the component, but can collect in patches that can interfere with proper functioning. If these particles are tacky they can collect other particulate matter. Even very small particles can have an impact if they collect on a tacky surface, or agglomerate as the result of electrostatic charge build-up.

Corrosive Failure

Corrosive failure or contact intermittence due to the intrinsic composition of the particles or due to absorption of water vapor and gaseous contaminants by the particles can also cause failures. The chemical composition of the contaminant can be very important. Salts, for instance, can grow in size by absorbing water vapor from the air (nucleating). If a mineral salts deposit exists in a sensitive location, and the environment is sufficiently moist, it can grow to a size where it can physically interfere with a mechanism, or can cause damage by forming salt solutions.

Shorts

Conductive pathways can arise through the accumulation of particles on circuit boards or other components. Many types of particulate are not inherently conductive, but can absorb significant quantities of water in high-moisture environments. Problems caused by electrically conductive particles can range from intermittent malfunctioning to actual damage to components and operational failures.

Thermal Failure

Premature clogging of filtered devices will cause a restriction in air flow that could induce internal overheating and head crashes. Heavy layers of accumulated dust on hardware components can also form an insulative layer that can lead to heat-related failures.

Room Conditions

All surfaces within the controlled zone of the data center should be maintained at a high level of cleanliness. All surfaces should be periodically cleaned by trained professionals on a regular basis, as outlined in "Cleaning Procedures and Equipment" on page 7-8. Particular attention should be paid to the areas beneath the hardware, and the access floor grid. Contaminants near the air intakes of the hardware can more easily be transferred to areas where they can do damage. Particulate accumulations on the access floor grid can be forced airborne when floor tiles are lifted to gain access to the sub-floor.

The subfloor void in a downward-flow air conditioning system acts as the supply air plenum. This area is pressurized by the air conditioners, and the conditioned air is then introduced into the hardware spaces through perforated floor panels. Thus, all air traveling from the air conditioners to the hardware must first pass through the subfloor void. Inappropriate conditions in the supply air plenum can have a dramatic effect on conditions in the hardware areas.

The subfloor void in a data center is often viewed solely as a convenient place to run cables and pipes. It is important to remember that this is also a duct, and that conditions below the false floor must be maintained at a high level of cleanliness. Contaminant sources can include degrading building materials, operator activity or infiltration from outside the controlled zone. Often particulate deposits are formed where cables or other subfloor items form air dams that allow particulate to settle and accumulate. When these items are moved, the particulate is re-introduced into the supply airstream, where it can be carried directly to hardware.

Damaged or inappropriately protected building materials are often sources of subfloor contamination. Unprotected concrete, masonry block, plaster or gypsum wall-board will deteriorate over time, shedding fine particulate into the air. Corrosion on post-filtration air conditioner surfaces or subfloor items can also be a concern. The subfloor void must be thoroughly and appropriately decontaminated on a regular basis to address these contaminants. Only vacuums equipped with High Efficiency Particulate Air (HEPA) filtration should be used in any decontamination procedure. Inadequately filtered vacuums will not arrest fine particles, passing them through the unit at high speeds, and forcing them airborne.

Unsealed concrete, masonry or other similar materials are subject to continued degradation. The sealants and hardeners normally used during construction are often designed to protect the deck against heavy traffic, or to prepare the deck for the application of flooring materials, and are not meant for the interior surfaces of a supply air plenum. While regular decontaminations will help address loose particulate, the surfaces will still be subject to deterioration over time, or as subfloor activity causes wear. Ideally all of the subfloor surfaces will be appropriately sealed at the time of construction. If this is not the case, special precautions will be necessary to address the surfaces in an on-line room.

It is extremely important that only appropriate materials and methodology are used in the encapsulation process. Inappropriate sealants or procedures can actually degrade the conditions they are meant to improve, impacting hardware operations and

reliability. The following precautions should be taken when encapsulating the supply air plenum in an on-line room:

- Manually apply the encapsulant. Spray applications are totally inappropriate in an on-line data center. The spraying process forces the sealant airborne in the supply airstream, and is more likely to encapsulate cables to the deck.
- Use a pigmented encapsulant. The pigmentation makes the encapsulant visible in application, ensuring thorough coverage, and helps in identifying areas that are damaged or exposed over time.
- It must have a high flexibility and low porosity to effectively cover the irregular textures of the subject area, and to minimize moisture migration and water damage.
- The encapsulant must not out-gas any harmful contaminants. Many encapsulants commonly used in industry are highly ammoniated or contain other chemicals that can be harmful to hardware. It is very unlikely that this out-gassing could cause immediate, catastrophic failure, but these chemicals will often contribute to corrosion of contacts, heads or other components.

Effectively encapsulating a subfloor deck in an on-line computer room is a very sensitive and difficult task, but it can be conducted safely if appropriate procedures and materials are used. Avoid using the ceiling void as an open supply or return for the building air system. This area is typically very dirty and difficult to clean. Often the structural surfaces are coated with fibrous fire-proofing, and the ceiling tiles and insulation are also subject to shedding. Even before filtration, this is an unnecessary exposure that can adversely affect environmental conditions in the room. It is also important that the ceiling void does not become pressurized, as this will force dirty air into the computer room. Columns or cable chases with penetrations in both the subfloor and ceiling void can lead to ceiling void pressurization.

Exposure Points

All potential exposure points in the data center should be addressed to minimize potential influences from outside the controlled zone. Positive pressurization of the computer rooms will help limit contaminant infiltration, but it is also important to minimize any breaches in the room perimeter. To ensure the environment is maintained correctly, the following should be considered:

- All doors should fit snugly in their frames.
- Gaskets and sweeps can be used to address any gaps.
- Automatic doors should be avoided in areas where they can be accidentally triggered. An alternate means of control would be to remotely locate a door trigger so that personnel pushing carts can open the doors easily. In highly sensitive areas, or where the data center is exposed to undesirable conditions, it may be advisable to design and install personnel traps. Double sets of doors with a buffer between can help limit direct exposure to outside conditions.
- Seal all penetrations between the data center and adjacent areas.
- Avoid sharing a computer room ceiling or subfloor plenum with loosely controlled adjacent areas.

Filtration

Filtration is an effective means of addressing airborne particulate in a controlled environment. It is important that all air handlers serving the data center are

adequately filtered to ensure appropriate conditions are maintained within the room. In-room process cooling is the recommended method of controlling the room environment. The in-room process coolers re-circulate room air. Air from the hardware areas is passed through the units where it is filtered and cooled, and then introduced into the subfloor plenum. The plenum is pressurized, and the conditioned air is forced into the room, through perforated tiles, which then travels back to the air conditioner for reconditioning. The airflow patterns and design associated with a typical computer room air handler have a much higher rate of air change than typical comfort cooling air conditioners so air is filtered much more often than in an office environment. Proper filtration can capture a great deal of particulates. The filters installed in the in-room, re-circulating air conditioners should have a minimum efficiency of 40% (Atmospheric Dust-Spot Efficiency, ASHRAE Standard 52.1). Lowgrade pre-filters should be installed to help prolong the life of the more expensive primary filters.

Any air being introduced into the computer room controlled zone, for ventilation or positive pressurization, should first pass through high efficiency filtration. Ideally, air from sources outside the building should be filtered using High Efficiency Particulate Air (HEPA) filtration rated at 99.97% efficiency (DOP Efficiency MILSTD-282) or greater. The expensive high efficiency filters should be protected by multiple layers of pre-filters that are changed on a more frequent basis. Low-grade pre-filters, 20% ASHRAE atmospheric dust-spot efficiency, should be the primary line of defense. The next filter bank should consist of pleated or bag type filters with efficiencies between 60% and 80% ASHRAE atmospheric dust-spot efficiency.

Table 7–1 ASHRAE 52-76

	Fractional Efficiencies %	**Fractional Efficiencies %**	**Fractional Efficiencies %**
Dust spot efficiency %	**3.0 micron**	**1.0 micron**	**0.3 micron**
25-30	80	20	<5
60-65	93	50	20
80-85	99	90	50
90	>99	92	60
DOP 95	--	>99	95

Low efficiency filters are almost totally ineffective at removing sub-micron particulates from the air. It is also important that the filters used are properly sized for the air handlers. Gaps around the filter panels can allow air to bypass the filter as it passes through the air conditioner. Any gaps or openings should be filled using appropriate materials, such as stainless steel panels or custom filter assemblies.

Positive Pressurization and Ventilation

A designed introduction of air from outside the computer room system will be necessary in order to accommodate positive pressurization and ventilation requirements. The data center should be designed to achieve positive pressurization in relation to more loosely controlled surrounding areas. Positive pressurization of the more sensitive areas is an effective means of controlling contaminant infiltration through any minor breaches in the room perimeter. Positive pressure systems are designed to apply outward air forces to doorways and other access points within the data processing center in order to minimize contaminant infiltration of the computer room. Only a minimal amount of air should be introduced into the controlled environment. In data centers with multiple rooms, the most sensitive areas should be

the most highly pressurized. It is, however, extremely important that the air being used to positively pressurize the room does not adversely affect the environmental conditions in the room. It is essential that any air introduction from outside the computer room is adequately filtered and conditioned to ensure that it is within acceptable parameters. These parameters can be looser than the goal conditions for the room since the air introduction should be minimal. A precise determination of acceptable limits should be based on the amount of air being introduced and the potential impact on the environment of the data center.

Because a closed-loop, re-circulating air conditioning system is used in most data centers, it will be necessary to introduce a minimal amount of air to meet the ventilation requirements of the room occupants. Data center areas normally have a very low human population density; thus the air required for ventilation will be minimal. In most cases, the air needed to achieve positive pressurization will likely exceed that needed to accommodate the room occupants. Normally, outside air quantities of less than 5% make-up air should be sufficient (ASHRAE Handbook: Applications, Chapter 17). A volume of 15 CFM outside air per occupant or workstation should sufficiently accommodate the ventilation needs of the room.

Cleaning Procedures and Equipment

Even a perfectly designed data center requires continued maintenance. Data centers containing design flaws or compromises may require extensive efforts to maintain conditions within desired limits. Hardware performance is an important factor contributing to the need for a high level of cleanliness in the data center.

Operator awareness is another consideration. Maintaining a fairly high level of cleanliness will raise the level of occupant awareness with respect to special requirements and restrictions while in the data center. Occupants or visitors to the data center will hold the controlled environment in high regard and are more likely to act appropriately. Any environment that is maintained to a fairly high level of cleanliness and is kept in a neat and well organized fashion will also command respect from the room's inhabitants and visitors. When potential clients visit the room they will interpret the overall appearance of the room as a reflection of an overall commitment to excellence and quality. An effective cleaning schedule must consist of specially designed short-term and long-term actions. These can be summarized as follows:

Table 7–2 Cleaning Schedule

Frequency	Task
Daily Actions	Rubbish removal
Weekly Actions	Access floor maintenance (vacuum and damp mop)
Quarterly Actions	Hardware decontamination
	Room surface decontamination
Bi-Annual Actions	Subfloor void decontamination
	Air conditioner decontamination (as necessary)

Daily Tasks

This statement of work focuses on the removal of each day's discarded trash and rubbish from the room. In addition, daily floor vacuuming may be required in Print Rooms or rooms with a considerable amount of operator activity.

Weekly Tasks

This statement of work focuses on the maintenance of the access floor system. During the week, the access floor becomes soiled with dust accumulations and blemishes. The entire access floor should be vacuumed and damp mopped. All vacuums used in the data center, for any purpose, should be equipped with High Efficiency Particulate Air (HEPA) filtration. Inadequately filtered equipment cannot arrest smaller particles, but rather simply agitates them, degrading the environment they were meant to improve. It is also important that mop-heads and dust wipes are of appropriate nonshedding designs.

Cleaning solutions used within the data center must not pose a threat to the hardware. Solutions that could potentially damage hardware include products that are:

- Ammoniated
- Chlorine-based
- Phosphate-based
- Bleach enriched
- Petro-chemical based
- Floor strippers or re-conditioners

It is also important that the recommended concentrations are used, as even an appropriate agent in an inappropriate concentration can be potentially damaging. The solution should be maintained in good condition throughout the project, and excessive applications should be avoided.

Quarterly Tasks

The quarterly statement of work involves a much more detailed and comprehensive decontamination schedule and should only be conducted by experienced computer room contamination-control professionals. These actions should be performed three to four times per year, based on the levels of activity and contamination present. All room surfaces should be thoroughly decontaminated including cupboards, ledges, racks, shelves and support equipment. High ledges and light fixtures and generally accessible areas should be treated or vacuumed as appropriate. Vertical surfaces including windows, glass partitions, doors, etc. should be thoroughly treated. Special dust cloths that are impregnated with a particle absorbent material are to be used in the surface decontamination process. Do not use generic dust rags or fabric cloths to perform these activities. Do not use any chemicals, waxes or solvents during these activities.

Settled contamination should be removed from all exterior hardware surfaces including horizontal and vertical surfaces. The unit's air inlet and outlet grilles should be treated as well. Do not wipe the unit's control surfaces as these areas can be decontaminated by the use of lightly compressed air. Special care should also be taken when cleaning keyboards and life-safety controls. Specially treated dust wipes should be used to treat all hardware surfaces. Monitors should be treated with optical cleansers and static-free cloths. No Electro-Static Discharge (ESD) dissipative chemicals should be used on the computer hardware, since these agents are caustic and harmful to most sensitive hardware. The computer hardware is sufficiently designed to permit electrostatic dissipation thus no further treatments are required. After all of the hardware and room surfaces have been thoroughly decontaminated, the access floor should be HEPA vacuumed and damp mopped as detailed in the Weekly Actions.

Biennial Tasks

The subfloor void should be decontaminated every 18 months to 24 months based on the conditions of the plenum surfaces and the degree of contaminant accumulation. Over the course of the year, the subfloor void undergoes a considerable amount of activity that creates new contamination accumulations. Although the weekly above floor cleaning activities will greatly reduce the subfloor dust accumulations, a certain amount of surface dirt will migrate into the subfloor void. It is important to maintain the subfloor to a high degree of cleanliness since this area acts as the hardware's supply air plenum. It is best to perform the subfloor decontamination treatment in a short time frame to reduce cross contamination. The personnel performing this operation should be fully trained to assess cable connectivity and priority. Each exposed area of the subfloor void should be individually inspected and assessed for possible cable handling and movement. All twist-in and plug-in connections should be checked and fully engaged before cable movement. All subfloor activities must be conducted with proper consideration for air distribution and floor loading. In an effort to maintain access floor integrity and proper psychrometric conditions, the number of floor tiles removed from the floor system should be carefully managed. In most cases, each work crew should have no more than 24 square feet (six tiles) of open access flooring at any one time. The access floor's supporting grid system should also be thoroughly decontaminated, first by vacuuming the loose debris and then by damp-sponging the accumulated residue. Rubber gaskets, if present, as the metal framework that makes up the grid system should be removed from the grid work and cleaned with a damp sponge as well. Any unusual conditions, such as damaged floor suspension, floor tiles, cables and surfaces, within the floor void should be noted and reported.

Activity and Processes

Isolation of the data center is an integral factor in maintaining appropriate conditions. All unnecessary activity should be avoided in the data center, and access should be limited to necessary personnel only. Periodic activity, such as tours, should be limited, and traffic should be restricted to away from the hardware so as to avoid accidental contact. All personnel working in the room, including temporary employees and janitorial personnel, should be trained in the most basic sensitivities of the hardware so as to avoid unnecessary exposure. The controlled areas of the data center should be thoroughly isolated from contaminant producing activities. Ideally, print rooms, check sorting rooms, command centers or other areas with high levels of mechanical or human activity should have no direct exposure to the data center. Paths to and from these areas should not necessitate traffic through the main data center areas.

A

Customer-supplied Information for Field Personnel

Table A–1 shows a chart you can fill out and give to field personnel. It provides a list of items you assign as you go through the planning process.

Table A–1 Customer-supplied Information

Value:	What You Entered:
Company Name	
Site Name	
City	
First Name	
Last Name	
Contact Email	
Customer Oracle CSI Login Name	
Customer Oracle CSI Login Password	
Ethernet Port Host Name	
Server IP Address	
Network CIDR	
VTCS Database Mode (if SQL, indicate which VTCS node is primary)	
ACSLS or VTCS LDOM data disk size	
MOS User ID	
MOS Password	
ILOM Hostname	
ILOM IP Address	
ILOM CIDR	
Server Hostname	
Server IP Address	
Server CIDR	
Server Default Router	

Table A–1 (Cont.) Customer-supplied Information

Value:	What You Entered:
DNS Servers 1-3	
Server Search Domains 1-3	
Virtual Machine Type (ACSLS or VTCS)	
Virtual Machine Hostname	
Virtual Machine IP Address	
Virtual Machine CIDR	

Index

P

R

S

V