

# StorageTek VM Client

---

## Installation, Configuration, and Administration Guide

Version 7.1



Part Number: E22058-03  
December 2013

Submit comments about this document to [STP\\_FEEDBACK\\_US@ORACLE.COM](mailto:STP_FEEDBACK_US@ORACLE.COM).

Oracle welcomes your comments and suggestions for improving this book. Contact us at [STP\\_FEEDBACK\\_US@ORACLE.COM](mailto:STP_FEEDBACK_US@ORACLE.COM). Please include the title, part number, issue date, and revision.

Copyright © 2013, 2012, 2011 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related software documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Table of Contents

<b>List of Figures</b>	7
<b>List of Tables</b>	9
<b>Preface</b>	11
Related Documentation	11
Oracle's StorageTek Enterprise Library Software (ELS) 7.1	11
Documentation, Support, and Training	11
Conventions for User Readability	12
Typographic	12
Syntax Flow Diagrams	12
Control Statements	16
<b>What's New</b>	17
Revision 03	17
<b>1 Introduction</b>	19
<b>2 Installing the VM Client</b>	21
Overview	21
Preparing for Installation	21
IBM VMSES/E	21
VM Client Installation Package	21
Software and Hardware Requirements	22
Application Program Interface Verification	23
DASD Storage and User ID Requirements	23
Installing the VM Client using VMSES/E	25
IBM VMSES/E	25
Summary of Installation Steps	25
Step 1: Determine VM Client Resource Requirements	26
Step 2: Allocate VM Client Resources	27
Step 3: Install VM Client Product Files	28
Step 4: Build the VM Client Executable Code	29
Step 5: Create the VM Client Service Machine	29
Step 6: Tailor VM Client Service Machine Files	30
Step 7: Test the VM Client	30
Step 8: Place the VM Client into Production	30

Installing VM Client Corrective Service .....	31
IBM VMSES/E .....	31
Summary of Installation Steps .....	31
Step 1: Prepare to Receive Service .....	32
Step 2: Receive Service .....	33
Step 3: Apply Service .....	33
Step 4: Build New Levels .....	33
Step 5: Place the New Service into Production .....	34
<b>3 Starting the VM Client .....</b>	<b>35</b>
Overview .....	35
SMCBINT Module Parameters .....	35
TRACE Keyword Value Pair .....	36
OPERATOR Keyword Value Pair .....	36
MAXRC Keyword Value Pair .....	36
VM Client Command Files .....	36
SMCPARMS .....	37
SMCCMDS .....	37
VM Client Customer Exits .....	37
CP DETACH support.....	38
<b>4 VM Client Commands .....</b>	<b>39</b>
Overview .....	39
Issuing VM Client Commands .....	39
VM Client Commands .....	40
AUTHorize .....	40
CMS .....	42
COMMtest .....	43
CP .....	45
DISMOUNT .....	46
Display DRive .....	47
Display RC .....	49
Display Volume .....	50
DRIVemap .....	51
DUMP .....	53
DUMPOpts .....	54
EXIT .....	55
Help .....	56
Llist .....	57
LOGdisk .....	59
MOunt .....	61
MSGDef .....	63
OPERator .....	66
OPTion .....	67
POOLmap .....	68
READ .....	71
RESYNChronize .....	72
Route .....	73

SERVer .....	74
SMCUDBX .....	78
SMCUSCR .....	81
TAPEPlex .....	82
TCPip .....	85
TRace .....	88
<b>5 ELS 7.1 Server Considerations .....</b>	<b>89</b>
Overview .....	89
SMC HTTP Server Component .....	89
Scratch Subpools .....	89
VTCS Management Classes .....	90
VM:Tape Allocation Exit .....	90
<b>6 VM Client Messages .....</b>	<b>91</b>
Overview .....	91
Message Listing .....	91
<b>7 VM Client Tape Management Interface .....</b>	<b>123</b>
Overview .....	123
TMS Responsibilities.....	125
User Interface .....	125
Tape Resource Allocation .....	125
Operator Interface .....	125
TMS Decision Points .....	126
TMS Initialization .....	126
Drive Allocation .....	126
Scratch Allocation .....	126
Volume Movement .....	126
Returning a Volume to Scratch Status .....	126
TapePlex Information Returned to the TMS .....	127
Configuration Information: .....	127
Volume Status: .....	127
Volume Location: .....	127
Eligible Drives .....	127
Movement Status and Error Codes .....	127
LSM and ACS Status .....	128
Inter-user Communications Vehicle (IUCV) Considerations .....	129
Additional Considerations .....	131
TMS and VM Client Interaction.....	132
TMS to VM Client Initial Connection .....	133
Initial Connection Dialog .....	133
Drive Allocation .....	133
Allocation Interaction .....	133
Allocation Dialog .....	134
Termination of Allocation Interface .....	135
Operator Message Processing .....	135
Operator Message Interaction .....	135
Operator Message Dialog .....	135

Termination of Operator Message Interface .....	136
PROP-Detected Dismount .....	136
The ACSRQ Macro.....	139
Introduction .....	139
ACSRQ Requests .....	139
ACSRQ Macro Syntax .....	140
DISMOUNT .....	143
EJECT .....	146
MOUNT .....	151
MOVE .....	155
QCAP .....	159
QCONFIG .....	163
QDRIVES .....	166
QDRLIST .....	169
QSCRATCH .....	174
QVOLUME .....	178
QVOLUME .....	182
SCRATCH .....	184
SELSCR .....	187
UNSCRATCH .....	191
Interface Data Areas.....	194
SLX Macro .....	194
<b>A MEDIA, RECtech, and MODEL Values .....</b>	<b>225</b>
Overview .....	225
Media Type (MEDia) .....	226
Recording Technique (RECtech).....	229
MODEL Type (MODEL) .....	232
<b>Index .....</b>	<b>1</b>

---

## List of Figures

<b>FIGURE 1-1</b>	VM Client Data Flow .....	20
<b>FIGURE 4-1</b>	AUTHorize syntax .....	40
<b>FIGURE 4-2</b>	CMS syntax .....	42
<b>FIGURE 4-3</b>	COMMtest syntax .....	43
<b>FIGURE 4-4</b>	CP syntax .....	45
<b>FIGURE 4-5</b>	DISMount syntax .....	46
<b>FIGURE 4-6</b>	Display DRive syntax .....	47
<b>FIGURE 4-7</b>	Display RC syntax .....	49
<b>FIGURE 4-8</b>	Display Volume syntax .....	50
<b>FIGURE 4-9</b>	DRIVemap syntax .....	51
<b>FIGURE 4-10</b>	DUMP syntax .....	53
<b>FIGURE 4-11</b>	DUMPOpts syntax .....	54
<b>FIGURE 4-12</b>	EXIT syntax .....	55
<b>FIGURE 4-13</b>	Help syntax .....	56
<b>FIGURE 4-14</b>	Llst syntax .....	57
<b>FIGURE 4-15</b>	LOGdisk syntax .....	59
<b>FIGURE 4-16</b>	MOunt syntax .....	61
<b>FIGURE 4-17</b>	MSGDef syntax .....	63
<b>FIGURE 4-18</b>	OPERator syntax .....	66
<b>FIGURE 4-19</b>	OPTion syntax .....	67
<b>FIGURE 4-20</b>	POOLmap syntax .....	68
<b>FIGURE 4-21</b>	READ syntax .....	71
<b>FIGURE 4-22</b>	RESYNChronize syntax .....	72
<b>FIGURE 4-23</b>	Route syntax .....	73
<b>FIGURE 4-24</b>	SERVer syntax .....	74
<b>FIGURE 4-25</b>	SMCUDBX syntax .....	79
<b>FIGURE 4-26</b>	SMCUSCR syntax .....	81
<b>FIGURE 4-27</b>	TAPEPlex syntax .....	82
<b>FIGURE 4-28</b>	TCPip syntax .....	85
<b>FIGURE 4-29</b>	TRace syntax .....	88
<b>FIGURE 7-1</b>	Initial Connection Dialog .....	133
<b>FIGURE 7-2</b>	Scenario A - Normal Dismount .....	137
<b>FIGURE 7-3</b>	Scenario B - Dismount Already Processed Automatically .....	138
<b>FIGURE 7-4</b>	ACSRQ syntax .....	140
<b>FIGURE 7-5</b>	ACSRQ DISMOUNT request syntax .....	143
<b>FIGURE 7-6</b>	ACSRQ EJECT request syntax .....	147
<b>FIGURE 7-7</b>	ACSRQ MOUNT request syntax .....	151
<b>FIGURE 7-8</b>	ACSRQ MOVE request syntax .....	155
<b>FIGURE 7-9</b>	ACSRQ QCAP request syntax .....	159
<b>FIGURE 7-10</b>	ACSRQ QCONFIG request syntax .....	163
<b>FIGURE 7-11</b>	ACSRQ QDRIVES request syntax .....	166
<b>FIGURE 7-12</b>	ACSRQ QDRLIST request syntax .....	170
<b>FIGURE 7-13</b>	ACSRQ QSCRATCH request syntax .....	174

<b>FIGURE 7-14</b>	ACSRQ QVOLUME request syntax .....	178
<b>FIGURE 7-15</b>	ACSRQ QVOLUSE request syntax .....	182
<b>FIGURE 7-16</b>	ACSRQ SCRATCH request syntax .....	184
<b>FIGURE 7-17</b>	ACSRQ SELSCR request syntax .....	187
<b>FIGURE 7-18</b>	ACSRQ UNSCRATCH request syntax .....	191



---

## List of Tables

<b>TABLE 2-1</b>	VM Client Installation Media Contents.....	22
<b>TABLE 2-2</b>	VSMC710A Minidisk Layout.....	24
<b>TABLE 2-3</b>	VM Client Service Machine Files .....	30
<b>TABLE 7-1</b>	SLX Macro - Parameter Matrix .....	195
<b>TABLE 7-2</b>	SLX Record Format .....	196
<b>TABLE 7-3</b>	ACSINT Record Format .....	213
<b>TABLE 7-4</b>	IUB Record Format .....	222
<b>TABLE A-1</b>	Media Types .....	226
<b>TABLE A-2</b>	Media Type Defaults .....	228
<b>TABLE A-3</b>	Recording Techniques .....	229
<b>TABLE A-4</b>	Recording Technique Defaults.....	231
<b>TABLE A-5</b>	Model Types .....	232



---

---

## Preface

This publication describes how to install, configure, and manage Oracle's StorageTek VM Client software. It is intended for storage administrators, system programmers and operators responsible for maintaining the VM Client.

## Related Documentation

### Oracle's StorageTek Enterprise Library Software (ELS) 7.1

- *Introducing ELS*
- *Installing ELS*
- *ELS Syntax Quick Reference*
- *ELS Messages and Codes*
- *ELS Programming Reference*
- *ELS Legacy Interfaces Reference*
- *Configuring HSC and VTCS*
- *Managing HSC and VTCS*
- *Configuring and Managing SMC*
- *ELS Disaster Recovery and Offsite Data Management Guide*

## Documentation, Support, and Training

Function	URL
Oracle Home	<a href="http://oracle.com">http://oracle.com</a>
Documentation	<a href="http://oracle.com/technetwork/indexes/documentation/index.html">http://oracle.com/technetwork/indexes/documentation/index.html</a>
Support	<a href="http://www.oracle.com/us/support/044752.html">http://www.oracle.com/us/support/044752.html</a>
Training	<a href="http://www.oracle.com/us/education/selectcountry-new-079003.html">http://www.oracle.com/us/education/selectcountry-new-079003.html</a>

# Conventions for User Readability

## Typographic

Some JCL examples in this guide include *italic* type. Italic type is used to indicate a variable. You must substitute an actual value for these variables.

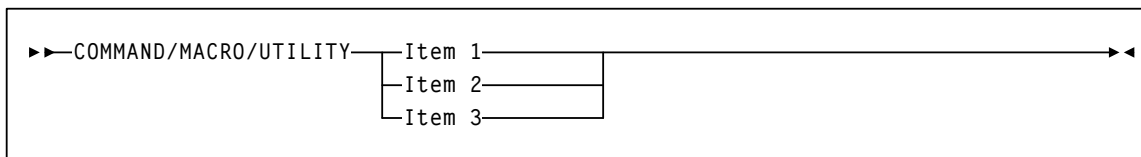
The use of mixed upper and lower case characters for commands, control statements, and parameters indicates that lower case letters may be omitted to form abbreviations. For example, you may simply enter TAPEP when executing the TAPEPlex command.

## Syntax Flow Diagrams

Syntax flow diagramming conventions include the following:

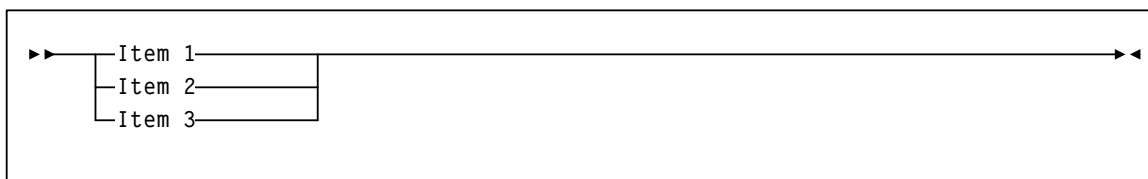
### Flow Lines

Syntax diagrams consist of a horizontal base line, horizontal and vertical branch lines, and the text for a command, control statement, macro, or utility. Diagrams are read left to right, and top to bottom. Arrows indicate flow and direction.



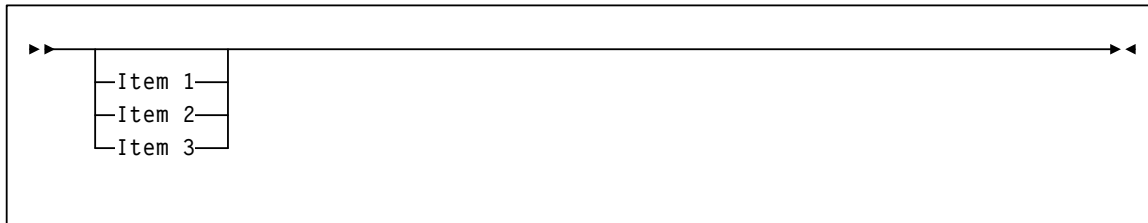
### Single Required Choice

Branch lines (without repeat arrows) indicate that a single choice must be made. If one of the items to choose from is positioned on the baseline of the diagram, one item must be selected.



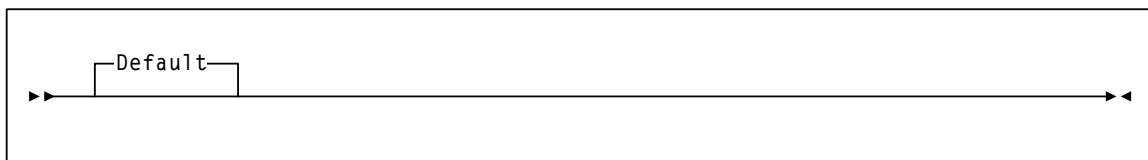
## Single Optional Choice

If the first item is positioned on the line below the baseline, one item may be optionally selected.

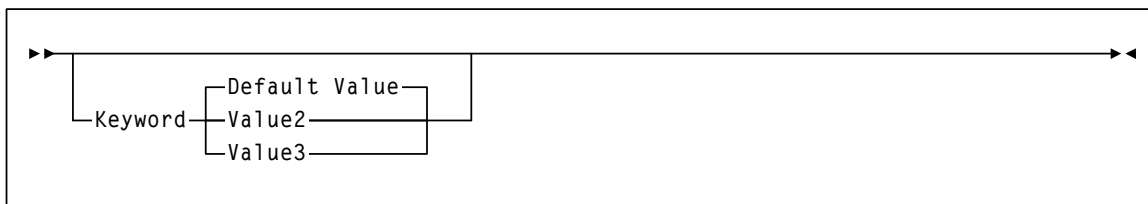


## Defaults

Default values and parameters appear above the baseline.

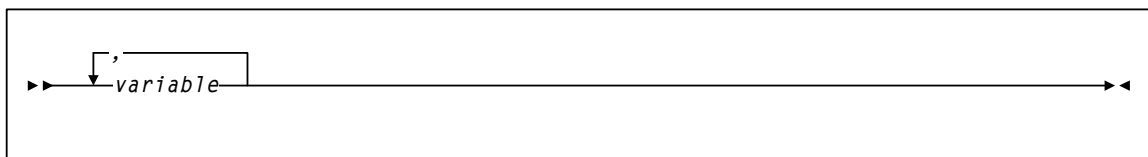


Some keyword parameters provide a choice of values in a stack. When the stack contains a default value, the keyword and the value choices are placed below the base line to indicate that they are optional, and the default value appears above the keyword line.



## Repeat Symbol

A repeat symbol indicates that more than one choice can be made or that a single choice can be made more than once. The following example indicates that a comma is required as the repeat delimiter.



## Keywords

All command keywords are shown in all upper case or in mixed case. When commands are not case sensitive, mixed case implies that the lowercase letters may be omitted to form an abbreviation.

## Variables

Italic type is used to indicate a variable.

## Alternatives

A bar ( | ) is used to separate alternative parameter values.

## Optional

Brackets [ ] are used to indicate that a command parameter is optional.

## Delimiters

If a comma (,), a semicolon (;), or other delimiter is shown with an element of the syntax diagram, it must be entered as part of the statement.

## Ranges

An inclusive range is indicated by a pair of elements of the same length and data type, joined by a dash. The first element must be strictly less than the second element.

A hexadecimal range consists of a pair of hexadecimal numbers (for example, 0A2-0AD, or 000-0FC).

A decimal range consists of a pair of decimal numbers (i.e., 1-9, or 010-094). Leading zeros are not required. The decimal portion is referred to as an incremental range. The character positions of the incremental portion of both range elements must match, and the non incremental characters of the first element must be identical to those of the second element.

A numeric VOLSER range (*vol-range*) consists of a pair of VOLSER elements containing a decimal numeric portion of 1 to 6 digits (for example, ABC012-ABC025, or X123CB-X277CB). The decimal portion is referred to as an incremental range. The following additional restrictions apply:

- The character positions of the incremental portion of both range elements must match.
- The non incremental characters of the first element must be identical to those of the second element.
- You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
- If a VOLSER range contains more than one decimal portion, any portion is valid as the incremental range. For example:

A00B00	the largest range that can be specified is A00B00 through A99B99.
A0B0CC	the largest range that can be specified is A0B0CC through A9B9CC.
000XXX	the largest range that can be specified is 000XXX through 999XXX.

An alphabetic VOLSER range (*vol-range*) consists of a pair of VOLSER elements containing an incremental portion of 1 to 6 characters (for example, 000AAA-000ZZZ, or 9AAA55-9ZZZ55). This portion is referred to as an incremental range. The following additional restrictions apply:

- The character positions of the incremental portion of both range elements must match.
- The non incremental characters of the first element must be identical to those of the second element.
- You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
- The alphabetic portion of the VOLSER range is defined as being from character A to Z. To increment multi-character sequences, each character increments to Z. For instance, ACZ is part of the AAA-AMM range. Examples are:

A00A0-A99A0	increments VOLSERs A00A0 through A09A0, then A10A0 through A99A0.
9AA9A-9ZZ9A	increments VOLSERs 9AA9A through 9AZ9A, then 9BA9A through 9ZZ9A.
111AAA-111ZZZ	increments VOLSERs 111AAA through 111AAZ, then 111ABA through 111ZZZ
999AM8-999CM8	increments VOLSERs 999AM8 through 999AZ8, then 999BA8 through 999CM8
A3BZZ9-A3CDE9	increments VOLSERs A3BZZ9 through A3CAA9, then A3CAB9 through A3CDE9
AAAAAA-AAACCC	increments VOLSERs AAAAAA through AAAAAZ, then AAAABA through AAACCC
CCCNNN-DDDNNN	increments VOLSERs CCCNNN through CCCNNZ, then CCCNOA through DDDNNN <sup>1</sup>

1. Caution: This is a very large range.

The number of volumes in an alphabetic VOLSER range depends on the number of elements in the incrementing portion of the VOLSER range. For an A to Z range in each character position, the number of volumes can be calculated by 26 to the power of the number of positions that are being incremented.

A-Z	26 <sup>1</sup>	26
AA-ZZ	26 <sup>2</sup>	676
AAA-ZZZ	26 <sup>3</sup>	17,576

AAAA-ZZZZ	$26^4$	456,976
AAAAA-ZZZZZ	$26^5$	11,881,376
AAAAAA-ZZZZZZ	$26^6$	308,915,776

## Lists

A list consists of one or more elements. If more than one element is specified, the elements must be separated by a comma or a blank space, and the entire list must be enclosed in parentheses.

## Blanks

Keyword parameters and values may be separated by any number of blanks.

## Control Statements

The standard syntax conventions for control statements are as follows:

- The only valid control statement information area is from column 1 to column 72. Columns 73-80 are ignored.
- Parameters may be separated by one or more blanks or a comma.
- A value is associated with a parameter by an equal (=) sign or by enclosing the value in parentheses, and concatenating it immediately after the parameter.
- Case (upper or lower) is ignored in actual control statements.
- Continuations are supported by including a plus (+) sign at the end of the line to be continued. A control statement is terminated if the statement is not continued.
- /\* and \*/ can be used to enclose comments in the job stream. Comments can be continued over multiple lines, but cannot be nested.

PARMLIB members **must** include a /\*...\*/ comment as the **first** control statement. Otherwise, the old format is assumed. Comments in the old format must begin with an asterisk (\*) in column 1.

For definition data sets (e.g., VOLATTRs, UNITATTRs and TAPEREQs), comments **must** be in the new format (/\*...\*/).

- Asterisk (\*) comments are **not** allowed.
- A /\*...\*/ comment in the first line is **not** required.
- The maximum length for a control statement is 1024 characters.



---

# What's New

## Revision 03

- The VM Client OPTion command specifies VM Client system options.  
See [“OPTion” on page 67](#) for more information.
- The VM Client SMCUDBX command supports scratch synchronization between the VM Client host and the HSC CDS. SMCUDBX accesses the client tape management system to produce an extract file that can be read as input by the VM Client Scratch command (SMCUSCR).  
See [“SMCUDBX” on page 78](#) for more information.
- The VM Client SMCUSCR utility command generates scratch transactions from an extract file created by the VM Client SMCUDBX command  
See [“SMCUSCR” on page 81](#) for more information.
- The revised VM Client POOLmap command is provides a convenient (and recommended) method to associate a management class or subpool number with an ELS scratch subpool name. Additionally, this command provides validation of the subpool and management class names with the ELS server.  
See [“POOLmap” on page 68](#) for more information.
- The VM Client includes new messages:  
[SMC0831](#), [SMC0832](#), [SMC0833](#), [SMC0834](#), [SMC0835](#), [SMC0836](#), [SMC0837](#)
- The VM Client includes changed messages:  
[SMC0827](#)
- Added support for HP and IBM LTO6 tape drives.  
See [Appendix A, “MEDia, RECtech, and MODel Values”](#).
- Added support for Oracle’s StorageTek T10000D tape drive.  
See [Appendix A, “MEDia, RECtech, and MODel Values”](#).



---

## Introduction

VM Client allows a client running on a VM system to request real and virtual tape services from a StorageTek TapePlex server executing on MVS.

Refer to the publication *Introducing ELS 7.1* to learn about the MVS software used to manage the TapePlex server.

VM Client software provides the following features:

- IUCV interface to accept VM Tape Management Interface (VMTMI) requests

Oracle's StorageTek VM Client replaces VM/HSC as the component that provides an interface to allow VM Tape Management Systems (TMS) to use the StorageTek Automatic Cartridge System (ACS). In addition, the VM Client provides an interface to allow VM Tape Management Systems to use the StorageTek Virtual Storage Manager (VSM).

VM Client acts as the target for VMTMI requests originating from the VM TMS. All responses are returned to the TMS in VMTMI format. VM Client uses the VM Inter-user Communications Vehicle (IUCV) to communicate with the TMS service machines.

Not all VMTMI requests are supported by the VM Client. See [Chapter 7, “VM Client Tape Management Interface”](#) for a complete list of supported VMTMI requests.

- A TCP/IP interface to MVS based TapePlexes (HSCs)

StorageTek Enterprise Library Software (ELS) 7.1 provides an XML interface (XAPI) to control the Storagetek ACS and VTCS systems. XAPI communications occur via TCP/IP. VM Client converts VMTMI requests to the new XAPI format. XAPI responses are converted to VMTMI responses.

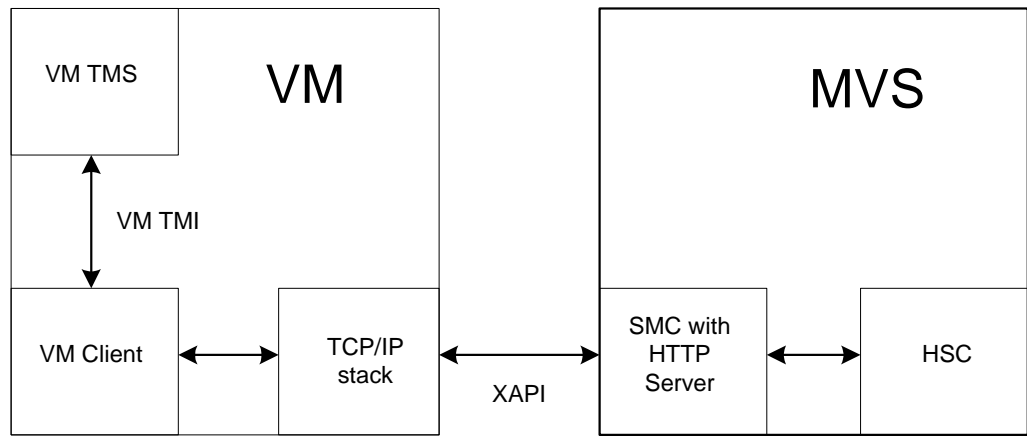
**Note** – VM Client can **only** communicate with **ELS 7.1** systems (SMC/HSC/VTCS).

- Operator commands to control the VM Client

See [Chapter 4, “VM Client Commands”](#) for information about VM Client commands.

**FIGURE 1-1** illustrates the VM Client data flow.

1. A VM Tape Management System (TMS) request is sent to the VM Client via the VM Tape Management Interface (VMTMI).
2. The VM Client converts the VMTMI request to XAPI format and routes the request to the MVS server using TCP/IP communications.
3. SMC/HSC/VTCS software on the MVS server processes the request and returns all responses to the VM Client in XAPI format.
4. The VM Client converts the XAPI responses to VMTMI format and routes these responses to the TMS.



**FIGURE 1-1** VM Client Data Flow

---

## Installing the VM Client

### Overview

This chapter describes how to install VM Client software and corrective service.

### Preparing for Installation

#### IBM VMSES/E

VM Client is installed using IBM's VMSES/E (Virtual Machine Serviceability Enhancements Staged/Enhanced), included as part of IBM VM/ESA.

Refer to the IBM publication *VMSES/E Introduction and Reference* for information about VMSES/E operation.

### VM Client Installation Package

The VM Client installation package includes the following materials:

- VM Client installation file, 9840B/9940B tape, or CD-ROM (SERVLINK envelope) containing VM Client software.
- VM Client corrective service file or 9840B/9940B tape containing PTFs, HOLDDATA, cover letters, and summary information released since the installation media was created.

**Note** – For CD-ROM installations, VM Client software and corrective service are provided on the same CD-ROM.

[TABLE 2-1 on page 22](#) lists the files or SERVLINK groups included on the VM Client installation media.

**TABLE 2-1** VM Client Installation Media Contents

<b>Tape File / CD-ROM SERVLINK group</b>	<b>Description</b>
1	Tape Header
2	Tape Header
3	VSMC710A (Product Identifier File)
4	VSMC710A MEMO (Memo to Users)
5	Service Apply Lists (AXLIST)
6	PARTLISTs
7	Service (DELTA)
8	Service (APPLY)
9	Base Code (BASE)
10	Help Files (HELP)

## Software and Hardware Requirements

VM Client software and hardware requirements are as follows:

### Software Requirements

#### ***Operating System:***

- Any IBM supported version of IBM z/VM

#### ***Independent Software Vendor Products (optional):***

- ACF/VTAM
- CA-DYNAM/TLMS for z/VM
- DFSORT
- EPIC VSE
- Multi-Image Manager (MIM)
- SYNCSORT
- VM:Tape (see [“Application Program Interface Verification”](#) below)
- IBM Tape Manager for z/VM

## Hardware Requirements

VM Client requires an IBM or compatible processor capable of running IBM z/VM (any IBM supported version).

## MVS Requirements

VM Client communicates with a StorageTek TapePlex server executing on MVS. This server is managed by Oracle's StorageTek ELS software including SMC, HSC, and optionally, VTCS.

VM Client requires that all ELS software be at release 7.1.

Refer to *Installing ELS* for information about ELS hardware and software requirements.

## Application Program Interface Verification

If you use VM:Tape as your tape management system, ensure that VM:Tape release 1.8 or later is installed.

Additionally, you **must** use the VM Client AUTHorize command to add VM:Tape to the VM Client authorized operators list. See [“AUTHorize” on page 40](#).

If any other tape management system is currently installed, you may need to write special routines to communicate with the VM Client. See [Chapter 7, “VM Client Tape Management Interface”](#) for more information about the interface to tape management systems, including recommended allocation and message processing, commands and responses, data areas, and interrupt handling required to communicate with the TapePlex.

## DASD Storage and User ID Requirements

VM Client requires you to add both the VM Client installation userid and VM Client service machine userid to your VM directory. The requirements of these userids are further defined during the VM Client installation process.

The default VM Client installation userid is VSMC710A. It is recommended that you use this default userid to install and service VM Client.

If you choose to change the name of the VM Client installation userid, you must create a Product Parameter Override (PPF). See [“Creating a PPF Override File” on page 27](#) for more information.

[TABLE 2-2 on page 24](#) describes the VSMC710A minidisk layout.

**TABLE 2-2** VSMC710A Minidisk Layout

Owner (User ID)	Default Address	Size (3390 Cyl)	Disk Name and Description
VSMC710A	2B2	20	BASE Contains all VM Client base code
VSMC710A	2C2	5	LOCALSAM Contains customization files.
VSMC710A	2D2	20	DELTA Contains services files.
VSMC710A	2A6	10	APPLY TEST Contains AUX files and software inventory tables that represent the test service level of VM Client.
VSMC710A	2A2	10	APPLY PRODUCTION Contains AUX files and software inventory tables that represent the production service level of VM Client.
VSMC710A	29D	10	BUILD4 Contains HELP files.
VSMC710A	201	20	BUILD1 Test build disk for VM Client. Contains load libraries and modules.
VSMC710A	202	20	BUILD2 Production build disk for VM Client. Contains load libraries and modules.
VSMC710A	191	10	INST191 Installation user ID 191 minidisk
VMSMC	191	10	VMSMC191 VM Client Service Machine 191 minidisk
VMSMC	200	20	VMSMCRUN VM Client Service Machine RUN disk



# Installing the VM Client using VMSES/E

## IBM VMSES/E

VM Client is installed using IBM's VMSES/E (Virtual Machine Serviceability Enhancements Staged/Enhanced), a component of IBM VM/ESA.

VMSES/E includes the VMFINS installation aid, designed to make installation of products consistent.

Refer to the IBM publication *VMSES/E Introduction and Reference* for more information about VMSES/E operation.

## Summary of Installation Steps

The following is a summary of installation steps. Each step is described in detail in the sections that follow.

### Step 1: Determine VM Client Resource Requirements

Use the VMFINS command to load several VMSES/E files from the product tape to obtain VM Client resource requirements.

### Step 2: Allocate VM Client Resources

Use the information obtained from the previous step to allocate the appropriate minidisks and userids needed to install and use VM Client.

### Step 3: Install VM Client Product Files

Use the VMFINS command to load the VM Client product files from tape to the BASE minidisk.

### Step 4: Build the VM Client Executable Code

Use the VMFINS command to build the VM Client test BUILD minidisk.

### Step 5: Create the VM Client Service Machine

Create a directory entry for the VM Client Service Machine.

### Step 6: Tailor VM Client Service Machine Files

Edit the VM Client Service Machine samples.

### Step 7: Test the VM Client

Test the VM Client on the test build disk.

### Step 8: Place the VM Client into Production

After VM Client testing, copy the VM Client files from the test build disk to the production build disk.

## Step 1: Determine VM Client Resource Requirements

Use the VMFINS command to determine VM Client resource requirements.

1. Log on as the installer/planner.

Use any userid with read access to MAINT 5E5 and write access to the 51D disk that will contain the VM Client software inventory.

2. If installing from the VM Client installation tape, mount the tape and attach it to the userid at virtual address 181. The VMFINS EXEC requires the tape drive to be at this address.
3. Enter the following commands to establish read access to the VMSES/E code:

```
LINK MAINT 5E5 5E5 RR
ACCESS 5E5 B
```

4. Enter the following commands to establish write access to the Software Inventory disk, MAINT 51D in this example.

```
LINK MAINT 51D 51D M
ACCESS 51D D
```

**Note –**

- The Software Inventory disk can be the system Software Inventory disk. It is recommended that a 51D disk be allocated to the maintenance/install userid (VSMC710A).
  - If another user is currently linked to the Software Inventory disk in write mode (R/W), the LINK command will fail. If this occurs, direct the other user to re-link to the Software Inventory disk in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not proceed until you establish a Read/Write link to the Software Inventory (51D) disk.
5. Load the VM Client product control files to the 51D minidisk.

If installing from the VM Client installation tape, enter the following:

```
VMFINS INSTALL INFO ( NOMEMO
```

If installing from the VM Client installation CD-ROM (SERVLINK envelope), enter the following:

```
VMFINS INSTALL INFO ( NOMEMO ENV VSMC710A
```

The INSTALL INFO command loads various product control files and creates the VMFINS PRODLIST file.

6. Obtain resource planning information for VMCLIENT.

If installing from the VM Client installation tape, enter the following:

```
VMFINS INSTALL PPF VSMC710A VMCLIENT ( NOMEMO PLAN
```

If installing from the VM Client installation CD-ROM (SERVLINK envelope), enter the following:

```
VMFINS INSTALL PPF VSMC710A VMCLIENT ( NOMEMO PLAN ENV
VSMC710A
```

The file, VMFINS PLANINFO, is created on the A-disk. This file contains information about the userids and minidisks required to install VM Client.

7. Review the install message log file, \$VMFINS \$MSGLOG. All installation messages are written to the installation user's A-disk. Correct any errors before proceeding.

## Creating a PPF Override File

If the maintenance or service machine userids must be changed, use the following procedure to create a PPF override file:

1. At the following prompt, enter 1.

```
VMFINS2601R Do you want to create an override for :PPF VSMC710A
VMCLIENT :PRODID VSMC710A%VMCLIENT?
```

```
Enter 0 (No), 1 (Yes) or 2 (Exit)
```

2. At the following prompt, enter 0.

```
VMFMKO2917R Do you want to use the defaults for this product?
```

```
Enter 0 (No), 1 (Yes) or 2 (Exit)
```

3. Update the VM Client INSTALL User and/or VM Client SERVER user on the Make Override Panel. Press F3 to exit this panel.

4. Select 2 - Save as... and enter file name of the override file.

The override \$PPF and PPF files are copied to the D-disk (51D).

The override PPF now replaces the VSMC710A PPF file. Substitute this override PPF for all instances of VSMC710A in all remaining installation steps.

## Step 2: Allocate VM Client Resources

Use the planning information in the VSMC710A PLANINFO file to create the VSMC710A user directory entry.

1. Create the VSMC710A user directory entry.

The VSMC710A user directory is located at the bottom of the PLANINFO file. These entries contain the links and privilege classes necessary for the VSMC710A userid. Use the directory entry found in PLANINFO as a model for the VSMC710A directory entry.

2. Add the MDISK statements to the directory entry for VSMC710A. The minidisk layout can be found in the PLANINFO file.
3. Add the VSMC710A directory entry to the system directory. Change the password for VSMC710A from XXXXXX to a valid password, in accordance with your security guidelines.
4. Place the new directory online.

## Step 3: Install VM Client Product Files

Use the VMFINS command to build the VM Client test BUILD minidisk.

1. Logon to the installation userid VSMC710A, created in Step 2.
2. Create a PROFILE EXEC that contains the ACCESS commands for MAINT 5E5 and 51D minidisks.

```
XEDIT PROFILE EXEC A
===> input /**/
===> input 'access 5e5 b'
===> input 'access 51d d'
===> file
```

3. Execute the profile to access MAINT's minidisks.

```
PROFILE
```

4. Establish write access to the Software Inventory disk, if it is not already linked R/W.

```
LINK MAINT 51D 51D M
ACCESS 51D D
```

**Note** – If another user is currently linked to the Software Inventory disk in write mode (R/W), the LINK command will fail. If this occurs, direct this user to re-link to the Software Inventory disk in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not proceed until you establish a Read/Write link to the Software Inventory (51D) disk.

5. Copy the VMSES/E files created on the installer/planner 191 disk to the VSMC710A 191 disk. This will place all the VM Client VMSES/E files in one location. The files to copy are:
  - VSMC710A PLANINFO
  - VSMC710A PRODLIST
6. If receiving the corrective service from tape, the VM Client installation tape must be mounted and attached to VSMC710A at virtual address 181. The VMFINS EXEC requires the tape drive to be at this address.
7. Install VM Client

If installing from the VM Client installation tape, enter the following:

```
VMFINS INSTALL PPF VSMC710A VMCLIENT (NOMEMO NOLINK
OVERRIDE NO
```

If installing from the VM Client installation CD-ROM (SERVLINK envelope), enter the following:

```
VMFINS INSTALL PPF VSMC710A VMCLIENT (NOMEMO NOLINK ENV
VSMC710A OVERRIDE NO
```

The NOLINK option indicates that VMFINS is not to link the maintenance minidisks, only access them if they are not already accessed.

The OVERRIDE NO option indicates that VMFINS is not to create an override PPF. If an override PPF was created above, replace VSMC710A with the name of the PPF created.

8. Review the install message log file, \$VMFINS \$MSGLOG. All installation messages are written to the installation user's A-disk. If necessary, correct any errors before proceeding.

## Step 4: Build the VM Client Executable Code

Use the VMFINS command to build the VM Client test BUILD minidisk.

1. Enter the following command to build the test BUILD minidisk.

```
VMFINS BUILD PPF VSMC710A VMCLIENT (ALL NOLINK
```

2. Review the install message log (\$VMFINS \$MSGLOG). All installation message logs are written to the installation user's A-disk. Correct any errors before proceeding.

**Note** – The following messages in the VMFINS BUILD log are normal:

- VMFBDC2178I  
Object =.HELPMMSG cannot be rebuilt because it is not serviced
- VMFBDC2178I  
Object =.HELPSMC cannot be rebuilt because it is not serviced
- VMFINB2173I  
No verification exec found for this product

## Step 5: Create the VM Client Service Machine

Create a directory entry for the VM Client Service Machine.

The VM Client Service Machine must be able to issue the following:

- CP MSGNOH command (Privilege Class B)
- CP QUERY TAPE ALL command (Privilege Class B)
- Diagnose 000 Store Extended-Identification Code (Privilege Class ANY)
- Diagnose 094 to create a VM dump (Privilege Class ANY)
- Diagnose 0A0 to obtain ACI group name for a given userid (Privilege Class ANY)

Refer to the VSMC710A PLANINFO file for machine and minidisk requirements.

## Step 6: Tailor VM Client Service Machine Files

Copy the following files from the LOCALSAM minidisk to the VM Client service machine 191 minidisk (VMSMC191):

**TABLE 2-3** VM Client Service Machine Files

Sample Name	Operational Name	Use
SMCPARMS SAMPLE	SMCPARMS (FILE)	Start parameters
SMCCMDS SAMPLE	SMCCMDS (FILE)	Start commands
SMCSTART EXEC	SMCSTART EXEC	VM Client start EXEC
SMCPRO SAMPLE	PROFILE EXEC	VM Client PROFILE EXEC

The SMCPARMS and SMCCMDS contain start parameters and commands. The file names of these files must match FIILDEF names in the SMCSTART EXEC.

See [“VM Client Command Files” on page 36](#) for information about updating VM Client parameter and command files.

## Step 7: Test the VM Client

VM Client is now ready for testing. The executable code to be tested is on the VSMC710A 201 minidisk. The options to run the VM Client are:

- LINK to VSMC710A 201 and issue SMCSTART.
- Copy VSMC710A 201 to VMSMC 200 minidisk and issue SMCSTART. The sample PROFILE EXEC (SMCPRO SAMPLE) assumes this option.

## Step 8: Place the VM Client into Production

After VM Client testing, enter the following commands to copy the VM Client files from the test disk (201) to the production disk (202):

```
VMFSETUP VSMC710A VMCLIENT
```

```
VMFCOPY ** fm1==fm2 (PRODID VSMC710A%VMCLIENT SPRODID  
VSMC710A%VMCLIENT OLDDATE REPLACE
```

*fm1* is the file mode of the TEST build disk (BUILD1 - 201)

*fm2* is the file mode of the PRODUCTION build disk (BUILD2 – 202)

The VM Client service machine will need to be modified to use the PRODUCTION version. The options are:

- LINK to VSMC710A 202 and issue SMCSTART.
- Copy VSMC710A 202 to VMSMC 200 minidisk and issue SMCSTART. The sample PROFILE EXEC (SMCPRO SAMPLE) assumes this option.

# Installing VM Client Corrective Service

## IBM VMSES/E

VM Client corrective service is installed using IBM's VMSES/E (Virtual Machine Serviceability Enhancements Staged/Enhanced), a component of IBM VM/ESA.

Refer to the IBM publication *VMSES/E Introduction and Reference* for more information about VMSES/E operation.

## Summary of Installation Steps

### Step 1: Prepare to Receive Service

Use the VMFMRDSK command to clear the alternate apply disk before receiving new service. This allows you to easily remove the new service if a serious problem is encountered.

### Step 2: Receive Service

The VMFREC command receives service from the delivery media and places it on the Delta disk.

### Step 3: Apply Service

The VMFAPPLY command updates the VMSES/E version vector table (VVT), which identifies the service level of all the serviced parts. In addition, AUX files are generated from the VVT for parts that require them.

### Step 4: Build New Levels

The build task generates the serviced level of an object and places the new object on a test BUILD (201) disk.

### Step 5: Place the New Service into Production

After being tested, the service is placed into production by copying the new service to the production disk (202).

## Step 1: Prepare to Receive Service

1. Log on to the VM Client service ID VSMC710A.
2. Establish write access to the Software Inventory disk, if it is not already linked R/W.

```
LINK MAINT 51D 51D M
ACCESS 51D D
```

**Note** – If another user is currently linked to the Software Inventory disk in write mode (R/W), the LINK command will fail. If this occurs, direct this user to re-link to the Software Inventory disk in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not proceed until you establish a Read/Write link to the Software Inventory (51D) disk.

3. If receiving corrective service from the VM Client corrective service tape, mount the tape and attach it to VSMC710A at virtual address 181. The VMFINS EXEC requires the tape drive to be at this address.
4. Establish the correct minidisk access order.

```
VMFSETUP VSMC710A VMCLIENT
```

VSMC710A is the PPF shipped with the product. If you have your own PPF override, you must substitute your PPF name for VSMC710A.

5. Receive the documentation. VMFREC with the INFO option loads the documentation and displays a list of all products on the delivery media.

If installing from the VM Client corrective service tape, enter the following:

```
VMFREC INFO
```

or if installing from the VM Client installation CD-ROM (SERVLINK envelope), enter the following:

```
VMFREC INFO ( ENV filename
```

6. Review the receive message log file, \$VMFREC \$MSGLOG, for warning and error messages.

```
VMFVIEW RECEIVE
```

7. Clear the alternate APPY disk to ensure that you have a clean disk for new service.

```
VMFMRDSK VSMC710A VMCLIENT APPLY
```

VSMC710A is the PPF shipped with the product. If you have your own PPF override, you must substitute your PPF name for VSMC710A.

8. Review the merge message log file, \$VMFMRD \$MSGLOG. If necessary, correct any errors before proceeding.

```
VMFVIEW MRD
```



## Step 2: Receive Service

1. Receive the service.

If installing from the VM Client corrective service tape, enter the following:

```
VMFREC PPF VSMC710A VMCLIENT
```

or if installing from the VM Client installation CD-ROM (SERVLINK envelope), enter the following:

```
VMFREC PPF VSMC710A VMCLIENT ( ENV filename
```

2. Review the receive message log file, \$VMFREC \$MSGLOG. If necessary, correct any errors before proceeding.

```
VMFVIEW RECEIVE
```

## Step 3: Apply Service

1. Apply the new service.

```
VMFAPPLY PPF VSMC710A VMCLIENT
```

This command applies the service that you just received. The version vector table is updated with all serviced parts and all necessary AUX files are generated.

2. Review the apply message log file, \$VMFAPP \$MSGLOG. If necessary, correct any errors before proceeding.

```
VMFVIEW APPLY
```

## Step 4: Build New Levels

1. Build the Build Status Table with serviced parts.

```
VMFBLD PPF VSMC710A VMCLIENT (STATUS
```

2. Review the build message log file, \$VMFAPP \$MSGLOG. If necessary, correct any errors before proceeding.

```
VMFVIEW BUILD
```

3. Rebuild VM Client serviced parts.

```
VMFBLD PPF VSMC710A VMCLIENT (SERVICED
```

4. Review the build message log file, \$VMFAPP \$MSGLOG. If necessary, correct any errors before proceeding.

```
VMFVIEW BUILD
```

## Step 5: Place the New Service into Production

After VM Client testing, copy the VM Client files from the test disk (201) to the production disk (202).

```
VMFSETUP VSMC710A VMCLIENT
```

```
VMFCOPY ** fm1==fm2 (PRODID VSMC710A%VMCLIENT OLDDATE REPLACE
```

*fm1* is the file mode of the TEST build disk (BUILD1 - 201).

*fm2* is the file mode of the PRODUCTION build disk (BUILD2 – 202).

Additionally, you must modify the VM Client service machine to use the PRODUCTION version. The options are:

- LINK to VSMC710A 202 and issue SMCSTART.
- Copy VSMC710A 202 to VMSC 200 minidisk and issue SMCSTART. The sample PROFILE EXEC (SMCPRO SAMPLE) assumes this option.

---

## Starting the VM Client

### Overview

The VM Client executes in its own CMS virtual machine, called the VM Client service machine. [Chapter 2, “Installing the VM Client”](#) describes the setup of the VM Client service machine, and the installation of the VM Client service machine software.

Once the VM Client service machine is installed, the VM Client is started by issuing the distributed SMCSTART command. The SMCSTART command is an EXEC that initializes the VM Client virtual machine environment, and then executes the SMCBINT module.

The SMCBINT module reads the VM Client command files, and initializes the VM Client environment to receive TMI commands from other virtual machines. See [“VM Client Command Files” on page 36](#) for more information about the VM Client command files.

### SMCBINT Module Parameters

In addition to the VM Client command files, the SMCBINT module accepts optional command line parameters. The SMCBINT optional command line parameters allow VM Client OPERATOR and VM Client TRACE to be set before the VM Client command files are read, or to enable VM Client MAXRC processing at startup.

SMCBINT optional command line parameters are entered as keyword value pairs following the module name. For example, the following is an example of specifying SMCBINT parameters:

```
SMCBINT TRACE ON OPERATOR VMOPER MAXRC 4
```

**Note** – Any errors encountered during command line parameter processing will result in termination of the VM Client initialization process.

Each of the allowed keyword value pairs are described in the following sections.

## TRACE Keyword Value Pair

**TRACE** {ON|OFF}

specifies whether VM Client trace processing is to be activated before VM Client command file processing. ON specifies that VM Client TRACE is to be enabled as early as possible.

## OPERATOR Keyword Value Pair

**OPERATOR** *userid*

specifies whether a VM Client operator is to be set before VM Client command file processing. *userid* specifies the name of the virtual machine to receive VM Client messages.

## MAXRC Keyword Value Pair

**MAXRC** *nn*

specifies whether VM Client MAXRC processing is to be activated. MAXRC processing determines whether the VM Client system is to be terminated on startup when the specified command return code is exceeded. If MAXRC is not specified, VM Client will always attempt to complete its initialization regardless of any startup command failure(s). This is the default behavior. The value *nn* specifies the highest allowed return code. If a VM Client command executed from the SMCPARMS or SMCCMDS data set exceeds this value, then the SMC0236 and SMC0237 messages will be produced, and the VM Client will be terminated. Allowable values are 0, 4, 8, and 12.

## VM Client Command Files

VM Client commands can be specified in the SMCPARMS and SMCCMDS command files. During VM Client initialization, these command files are read and the commands included in these files are executed. By convention, the command files are named as follows:

- SMCPARMS FILE A1
- SMCCMDS FILE A1

The FILEDEFs for SMCPARMS and SMCCMDS in the SMCSTART EXEC can be modified if different file names are to be used.

The control statements in the VM Client command files must conform to the standard conventions for control statements as described in [“Control Statements” on page 16](#).

## SMCPARMS

The SMCPARMS command file is read first. It is used for user-configured items that cannot be changed while the VM Client is active. SMCPARMS **cannot** be reprocessed using the READ command.

The following is a sample SMCPARMS member entry:

```
OPERATOR ID (nnnn)

LOGDISK ON

MSGDEF CASE (MIXED)

TCPIP TCPNAME (tcpname)
```

## SMCCMDS

The SMCCMDS command file is used for user-configured items that can be changed while the VM Client is active. SMCCMDS can be reprocessed using the VM Client READ command. See [“READ” on page 71](#) for more information about this command.

The following is a sample SMCCMDS member entry:

```
TAPEPLEX NAME (ttttttttt)

SERVER NAME (ssssssss) TAPEPLEX (ttttttttt) PORT (pppp) +

IPADDRESS (nn.nn.nn.nn)
```

## VM Client Customer Exits

The VM Client provides the following customer exits which may be implemented as CMS EXECs.

- SMCXIT00 Midnight Exit

If this exit is implemented, it executes each night at midnight.

See the installation sample SMCXIT00.samp for input parameters (if any), capabilities, and installation instructions.

- SMCXIT01 Command Authorization Exit

If this exit is implemented, it executes when a VM Client command or TMI request is received and the VM Client command or TMI request is not authorized by an appropriate VM Client AUTHorize command.

See [“AUTHorize” on page 40](#) for information about the VM Client AUTHorize command. See the installation sample SMCXIT01.samp for input parameters (if any), capabilities, and installation instructions.

**Note** – VMSES/E installation does **not** install these exits. Refer to the individual customer exit samples for installation instructions.

## CP DETACH support

The SMCPROP EXEC is provided for use as a PRogrammable OPerator (PROP) action routine to process all

TAPE raddr DETACHED....

messages, which CP sends to the system console.

VM (CP) causes a "Rewind Unload" command to be executed on any tape drive DETACHed by the CP commands LOGOFF, FORCE, or DETACH. This would leave any StorageTek DETACHed TapePlex volume in a "selected" state while still residing in an automated TapePlex transport. The volume would be unavailable to any requestor until removed (or DISMOUNTed) from the drive.

SMCPROP EXEC is an "action routine" which may be invoked from the VM PROP service to automatically issue VM Client DISMOUNT commands when a TapePlex transport is DETACHed from a virtual machine. This enables the volume to be available (that is, unselected) earlier than would otherwise be possible. RTABLE SAMPLE is provided to use the SMCPROP action routine with PROP. Refer to the IBM publication *CMS Planning and Administration* for information about the PROP service.

SMCPROP EXEC can also be used with VMOPERATOR to trap DETACH messages. LOGTABLE SAMPLE contains sample VMOPERATOR LOGTABLE statements to assist you. Consult VMOPERATOR documentation for information about tailoring the samples to your needs. The SMCPROP EXEC must be updated if the VM Client service machine userid is not VMSCMC.

SMCPROP EXEC must be available to the PROP/VMOPERATOR machine to operate properly.

---

## VM Client Commands

### Overview

This chapter describes the VM Client operator commands and the methods used to issue them.

### Issuing VM Client Commands

Use the following methods to issue VM Client commands:

- Specify VM Client commands in the SMCPARMS or SMCCMDS files to be processed at startup. See [“VM Client Command Files” on page 36](#) for more information.
- Send VM Client commands to the VM Client service machine using the CP Special Message (SMSG) facility.

Issue the following command from any virtual machine authorized to issue commands to the VM Client service machine:

```
CP SMSG userid command-string
```

where:

- *userid* is the name of the VM Client service machine defined in the CP directory.
  - *command-string* is a character string containing any valid VM Client command.
- Log on to the VM Client service machine and issue commands from the connected console.

# VM Client Commands

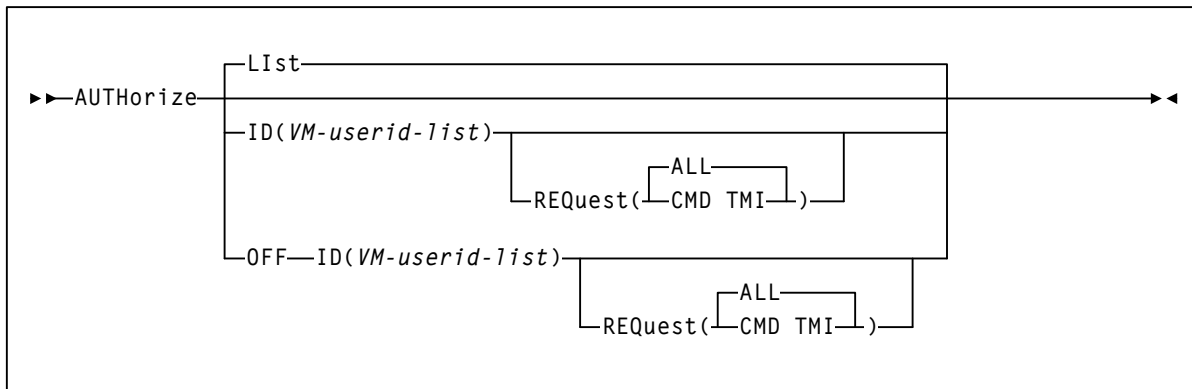
## AUTHorize

The AUTHorize command is used to identify VM userids that are authorized to execute TMI and VM Client command requests.

### Note –

- The customer exit SMCXIT01 is provided to add override authorization capabilities to the VM Client. If an un-authorized VM userid executes a VM Client command or TMI request, the SMCXIT01 exit may be used to override the absence of a matching AUTHorize command and provide the necessary authorization.
- There is no default SMCXIT01 EXEC exec installed as part of the VM Client installation.
- See the distributed SMCXIT01.samp for a sample customer exit SMCXIT01 and installation instructions.

### Syntax



**FIGURE 4-1** AUTHorize syntax

### Parameters

#### Lst

optionally, lists all current AUTHorize mappings.

- Lst is the default when no other parameters are specified.
- Lst may be specified with other parameters. In this case Lst is applied after all other parameters are processed.

#### ID

optionally, specifies the VM userids to be authorized.

*vm-userid-list*

a list of VM userids.



**REQuest**

optionally, specifies the VM Client privileges to be authorized. One or all of the following must be specified.

**CMD**

Authorize command requests received via the SMSG interface.

**TMI**

Authorize Tape Management Interface API requests.

**ALL**

Authorize CMD and TMI requests. This is the default if REQuest is not specified.

**OFF**

optionally, removes AUTHorize entries for the specified VM userid(s).

**ID**

specifies the VM userids for which authorization is to be removed. This parameter is required. A parameter value of "\*" will remove all AUTHorize entries.

***vm-userid-list***

a list of VM userids.

**REQuest**

optionally, specifies the VM Client privileges to be removed. One or all of the following must be specified:

**CMD**

Authorize command requests received via the SMSG interface.

**TMI**

Authorize Tape Management Interface API requests.

**ALL**

Authorize CMD and TMI requests. This is the default if REQuest is not specified.

**Example**

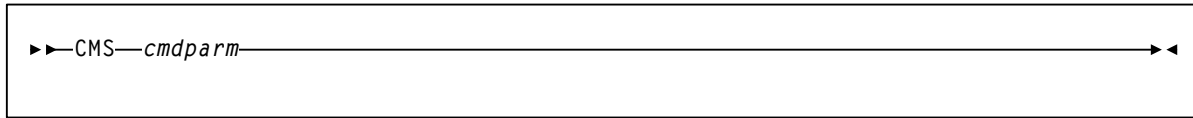
In the following example, the AUTHorize command is used to authorize userid VMTAPE to execute TMI requests, and list all AUTHORIZE entries.

```
AUTH REQ(TMI) ID(VMTAPE) LIST
```

## CMS

The CMS command is used to transmit commands to the VM CMS (Conversational Monitor System) program environment without leaving VM Client.

### Syntax



**FIGURE 4-2** CMS syntax

### Parameters

*cmdparm*

any valid CMS command and parameter string.

### Example

In the following example, the CMS command is used to specify a query of the file definitions in effect.

```
CMS QUERY FILEDEF
```

## COMMtest

The COMMtest command is used to test communications path(s) to one or more servers by executing a QUERY SERVER command to the specified communications path(s) and summarizing the results.

### Syntax

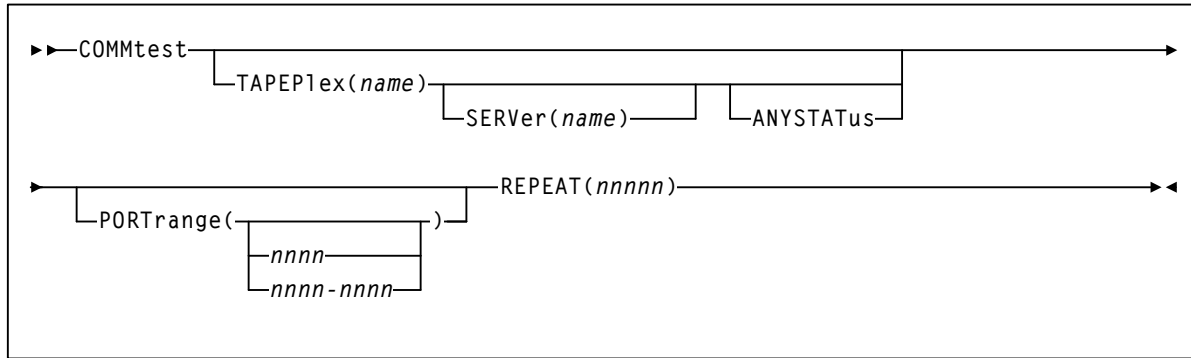


FIGURE 4-3 COMMtest syntax

### Parameters

#### TAPEPlex

optionally, specifies the TapePlex for the communication test. If you do not specify this parameter, communication is tested for all non-disabled TapePlexes.

##### *name*

the TapePlex name as defined by the VM Client TAPEPlex command. The following rules apply:

- The value must be between 1 and 8 characters in length.
- The first character must be either an alpha character or 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, or hyphen.

#### SERVer

optionally, specifies the server path for the communication test. If you do not specify this parameter, communication is tested for all non-disabled server paths for the named TapePlex.

##### *name*

the server path name as defined by the VM Client SERVer command. The following rules apply:

- The value must be between 1 and 8 characters in length.
- The first character must be either an alpha character or 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, or hyphen.

**Note** – See “SERVer” on page 74 for more information about the SERVer command.

#### ANYSTATus

optionally, communication is tested for **all** communication paths, including paths that have been disabled by an operator command or by the VM Client.

#### PORTrange

optionally, specifies that communication for a remote server path be tested from the specified port or range. The specified PORTrange may be different from the TCPip PORTrange specification to allow testing of a firewall setup.

*nnnnn* or *nnnn-nnnn*

the port number or port number range to be used for communication.

Each port number can have a value of 1-65535. However, The maximum port number range that can be specified is 100 (e.g. 6401-6500).

If omitted, a port in the defined TCPip PORTrange is used. If no such port is defined, any ephemeral port is used. If a port range is specified, then communication is attempted on each port number.

#### REPEAT

optionally, specifies the number of times to repeat the communication test. Valid values for *nnnnn* are 1 to 99999.

### Example

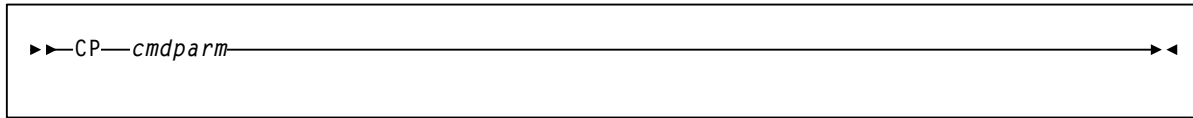
In the following example, the COMMtest command is used to test the communications path to TapePlex PRODHSC1 using SERVer PATHHSC1, and repeat the XAPI QUERY SERVER communication operation 100 times.

```
COMMTEST TAPEPLEX (PRODHSC1) SERVER (PATHHSC1) REPEAT (100)
```

## CP

The CP command is used to transmit commands to the VM CP (Control Program) environment without leaving the VM Client.

### Syntax



**FIGURE 4-4** CP syntax

### Parameters

*cmdparm*

any valid CP command and parameter string.

### Example

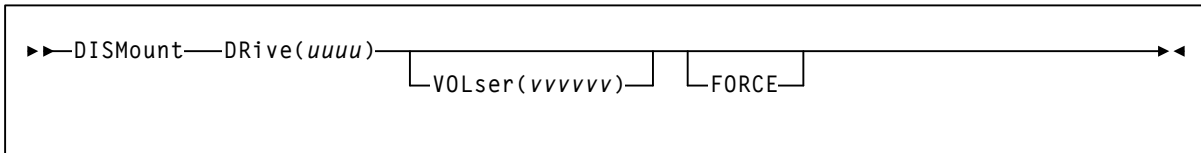
In the following example, the CP command is used to specify a query of the CPLEVEL attributes in effect.

```
CP QUERY CPLEVEL
```

## DISMount

The DISMount command dismounts a volume from a drive.

### Syntax



**FIGURE 4-5** DISMount syntax

### Parameters

#### DRive

specifies a tape drive address of the transport from which the volume is to be dismounted. This parameter is required.

*uuuu*

the tape drive address. If the DRIVEMAP command is used, this will be the CLIENT address. See [“DRIVemap” on page 51](#).

#### VOLser

optionally, specifies the volume serial (volser) for the volume to dismount. If this parameter is not specified, the volume currently mounted on the device is dismounted.

*vvvvvv*

the volume serial.

#### FORCE

optionally, specifies that the device is to be unloaded before the volume is dismounted. This parameter is not valid for virtual drives.

### Example

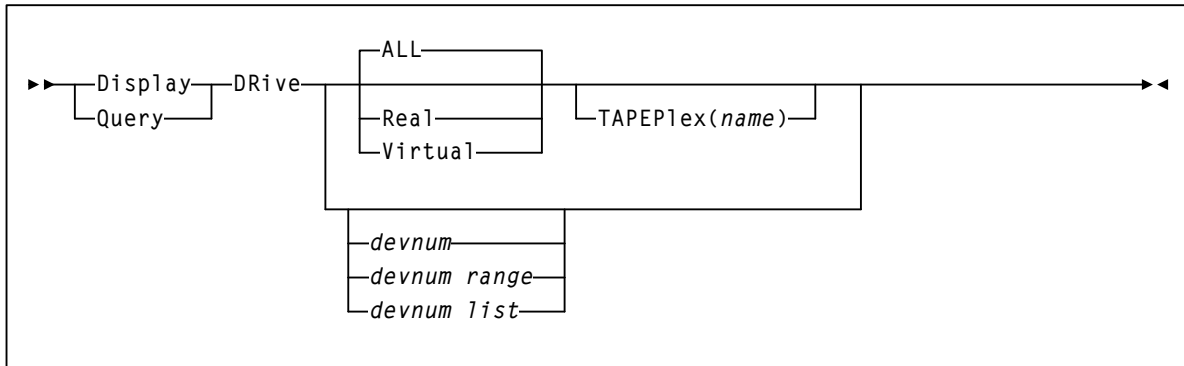
In the following example, the DISMount command is used to dismount volume AAA001 from drive 2900.

```
DISMOUNT DRIVE(2900) VOLSER(AAA001)
```

## Display DRive

The Display DRive command is used to request VM Client drive attribute and TapePlex ownership information.

### Syntax



**FIGURE 4-6** Display DRive syntax

### Parameters

#### *devnum, devnum-range, devnum-list*

indicates the device number(s) to be displayed. Each device number must be a valid hexadecimal address in the format *ccuu*. If the DRIVEMAP command is used this will be the CLIENT address(es).

#### **TapePlex**

optionally, lists only devices owned by the specified TapePlex. If this parameter is not specified, then owned devices for all TapePlexes are displayed.

#### *name*

the TapePlex name. The following rules apply:

- The value must be between 1 and 8 characters in length.
- The first character must be either an alpha character or 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, or hyphen.

#### **ALL**

optionally, lists all devices (real, virtual, and nonlib) owned by the specified TapePlex. This is the default if no parameters are specified.

#### **Real**

optionally, lists only “real” (i.e. nonvirtual) devices owned by the specified TapePlex.

**Virtual**

optionally, lists only virtual devices owned by the specified TapePlex.

**Example**

In the following example, the Display DRive command is used to list only the “real” (i.e. non virtual) devices known to the VM Client.

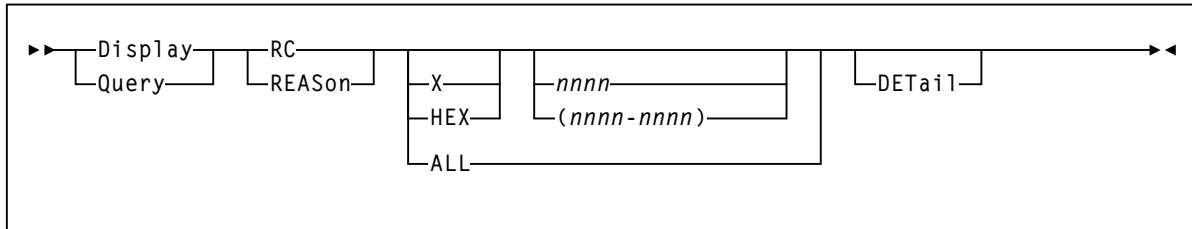
```
DISPLAY DRIVE REAL
```



## Display RC

The Display RC command is used to display information about the meaning of an SMC return or reason code, or an HSC/VTCS UII reason code.

### Syntax



**FIGURE 4-7** Display RC syntax

### Parameters

#### X or HEX

optionally, specifies that the reason/return code value or range is specified as a hexadecimal number.

#### *nnnn, nnnn-nnnn*

optionally, lists the return codes for which the explanation is to be displayed.

- If X or HEX is specified, the value may contain hexadecimal characters 0-9 and A-F.
- If X or HEX is not specified, the value may contain only numeric characters.

A range specified from the console may contain a maximum of 16 values.

#### ALL

optionally, indicates that all defined return or reason codes are to be listed. ALL is permitted only from a utility.

**Note** – ALL and H/HEX are mutually exclusive.

#### DETail

optionally, indicates that detailed information about the requested codes is to be listed.

### Example

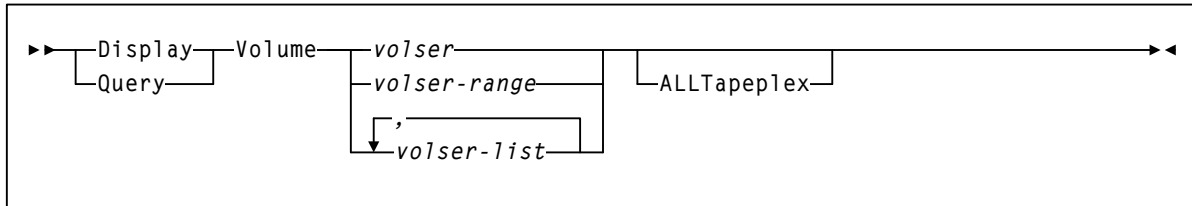
In the following example, the Display RC command is used to display information for SMC return code 302.

```
DISPLAY RC 302
```

## Display Volume

The Display Volume command is used to request volume attribute and TapePlex ownership information.

### Syntax



**FIGURE 4-8** Display Volume syntax

### Parameters

*volser, volser-range, volser-list*

indicates the volser, volser range, or volser list to be processed. If multiple volumes are specified, only the first 100 are displayed.

**ALLTapeplex**

optionally, specifies that all active TapePlexes are queried for the specified volser(s). If specified, multiple display lines may be listed for the same volser if it is defined in multiple TapePlexes.

If this parameter is not specified, the Display Volume command queries TapePlexes in the order they are defined and lists only the first occurrence of the volume.

### Example

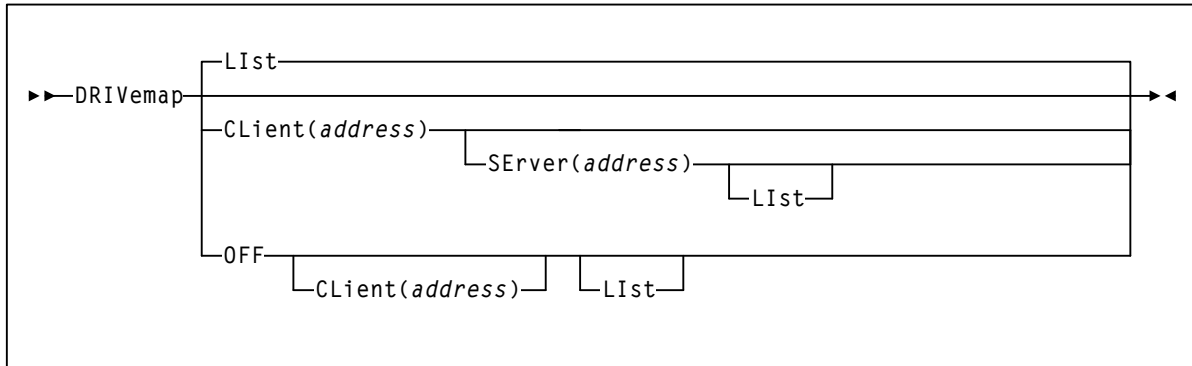
In the following example, the Display Volume command is used to list volume serial number EVT100 that is found in any TapePlex accessible from the VM Client.

```
DISPLAY VOLUME EVT100 ALLTAPEPLEX
```

## DRIVemap

The DRIVemap command is used to map VM Client device addresses to server drive addresses. This command allows users to specify different addresses on the VM Client and on the server host for the same TapePlex real or virtual devices.

### Syntax



**FIGURE 4-9** DRIVemap syntax

### Parameters

#### List

optionally, lists all current DRIVemap mappings.

- List is the default when no parameters are specified on the DRIVemap command. In this case, all DRIVemap entries are listed.
- List may be specified with other parameters. In this case, the List is applied **after** the other parameters are processed.

#### OFF

optionally, removes all DRIVemap entries. If this parameter is specified with the CLient parameter, then only the matching CLient DRIVemap mappings are removed. The address specification (list or range) must match the defining specification exactly.

#### CLient

optionally, specifies the device numbers mapped by the DRIVemap command.

#### *address*

the device number, device number range, or device number list. Each device number is a hexadecimal number.

## SErver

optionally specifies the device numbers that are defined on the HSC server.

### *address*

the device number, device number range of device number list. The device number is a hexadecimal number.

- If CLient is specified without the OFF parameter, then SErver is required.
- If both CLient and SErver are specified, then the CLient parameter must specify an equivalent address list or range as specified by the server parameter.

**Note** – Any VM Client commands entered that reference a device address (such as DISPLAY DRIVE, DISMOUNT, or MOUNT) must specify the client device address (or the address as known by the VM Client).

## Example

In the following example, the DRIVemap command is used to map client device addresses 180-188 to server device addresses 280-288.

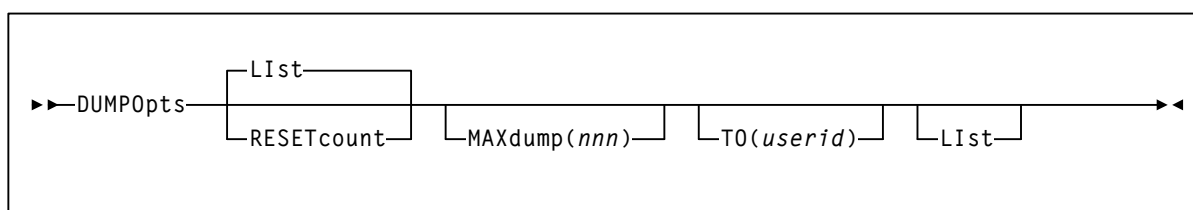
```
DRIVEMAP CLIENT(180-188) SERVER(280-288)
```



## DUMPOpts

The DUMPOpts command is used to specify or reset the maximum number of VMDUMP dumps to be generated. This command helps prevent VM spool space being exhausted in the unlikely event that a severe cycle of abnormal terminations occurs.

## Syntax



**FIGURE 4-11** DUMPOpts syntax

## Parameters

**RESETcount**

optionally, resets the number of dumps generated to zero.

## MAXdump

optionally, sets the threshold count for the number of dumps to allow before dump processing is disabled.

nnnn

the number of dumps. This is a decimal number from 0 to 999. The default is 50.

TO

optionally, specifies the userid to receive the dump.

## userid

the userid. This must be a defined VM userid. If an asterisk (\*) is entered, it is translated to the VM Client service machine id.

The default is the userid of the VM Client service machine.

## List

optionally, lists current DUMPOpts settings, including DUMPS TAKEN, DUMPS MAX COUNT, and USERID.

### Example

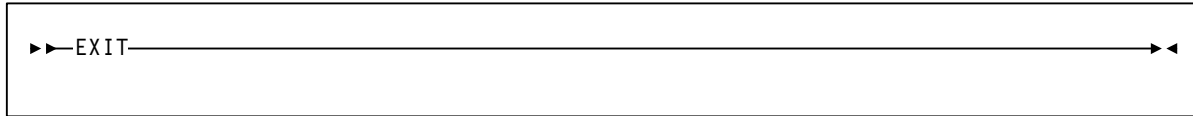
In the following example, the DUMPOpts command is used to reset the number of dumps generated to zero and set the threshold value to ten.

```
DUMPOPTS RESETCOUNT MAXDUMP(10)
```

# EXIT

The EXIT command is used to terminate the VM Client service machine.

## Syntax



**FIGURE 4-12** EXIT syntax

## Parameters

None.

## Help

The `Help` command is used to display VM Client command and message information.

**Note** – If you enter the Help command without any parameters, information is displayed for all available VM Client commands.

## Syntax



**FIGURE 4-13** Help syntax

## Parameters

*command-name*

optionally, a VM Client command name.

*nnnnn*

optionally, the four-digit numerical portion of a VM Client message identifier. Leading zeros are not required.

**nnnn-nnnn**

optionally, a range of VM Client messages specified using the four-digit numerical portion of the message identifier.

**SMC<sub>nnnn</sub>**

optionally, a full VM Client message identifier.

**SMC<sub>nnnn</sub>-SMC<sub>nnnn</sub>**

optionally, a range of VM Client messages specified using full message identifiers.

### Example

In the following example, the `Help` command is used to display information for VM Client message SMC0228.

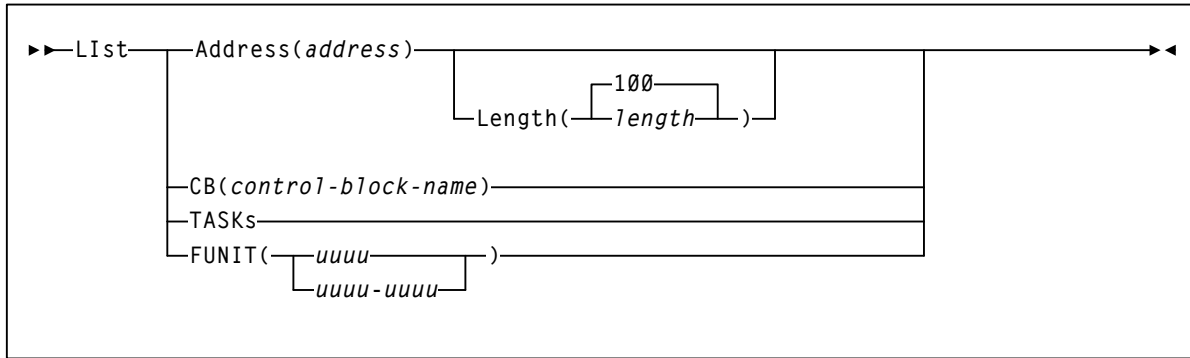
HELP SMC0228



## List

The Llist command is used to display storage contents within the VM Client virtual machine. This command is for diagnostic purposes and should be used only as directed by StorageTek Software Support.

### Syntax



**FIGURE 4-14** Llist syntax

### Parameters

#### Address

optionally, specifies the address at which to begin listing VM Client memory contents.

*address*

a hexadecimal address.

#### Length

optionally, specifies the length of VM Client memory (bytes) to be listed. Length is allowed only if Address is specified.

*length*

a hexadecimal length between 1 and FFFF. Memory is listed beginning at the location specified in the Address parameter. The default value is 100 (decimal 256).

#### CB

optionally, specifies the internal VM Client control block to be listed.

*control-block-name*

the control block name.

VM Client control blocks are listed for diagnostic purposes. Specify *control-block-name* only as directed by StorageTek Software Support.

#### TASKs

optionally, lists the active VM Client system tasks.

## **FUNIT**

optionally, lists VM Client control blocks associated with the specified unit address(es).

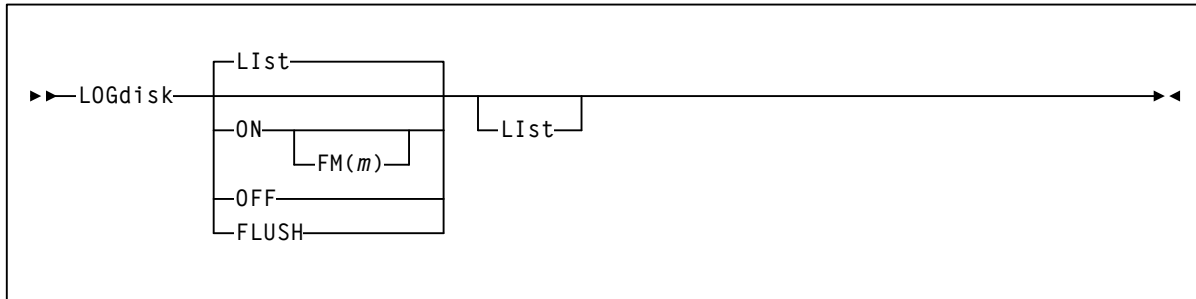
*uuuuu* or *uuuuu-uuuuu*

a single unit address or range of unit addresses.

# LOGdisk

The LOGdisk command is used to control logging of console output to a disk file named *YYYYMMDD LOG*.

## Syntax



**FIGURE 4-15** LOGdisk syntax

## Parameters

### List

optionally, displays current LOGdisk settings.

- List is the default when no parameters are specified on the LOGdisk command.
- List may be specified with other parameters. In this case, the List is applied **after** the other parameters are processed.

### ON

optionally, enables the logging of console output with the listed options. When logging is on, all commands and messages are logged.

### FM

specifies the File Mode to receive the disk log file. The FM must specify a RW minidisk. This parameter is **only** valid with the ON parameter.

*m*

the file mode. This value must be an alphabetic character. The default is 'A'.

### OFF

optionally, disables the logging of console output with the listed options. The log file is closed.

## FLUSH

optionally, flushes the log file. The file is closed and reopened.

**Note** – All messages are written to the VM Client service machine console. The handling of the VM Client service machine console can be controlled by the CP SPOOL command. It is recommended that the VM Client service machine console be started in the PROFILE EXEC and spooled to a maintenance ID. For example:

```
CP SPOOL CON START TO MAINT
```

## Example

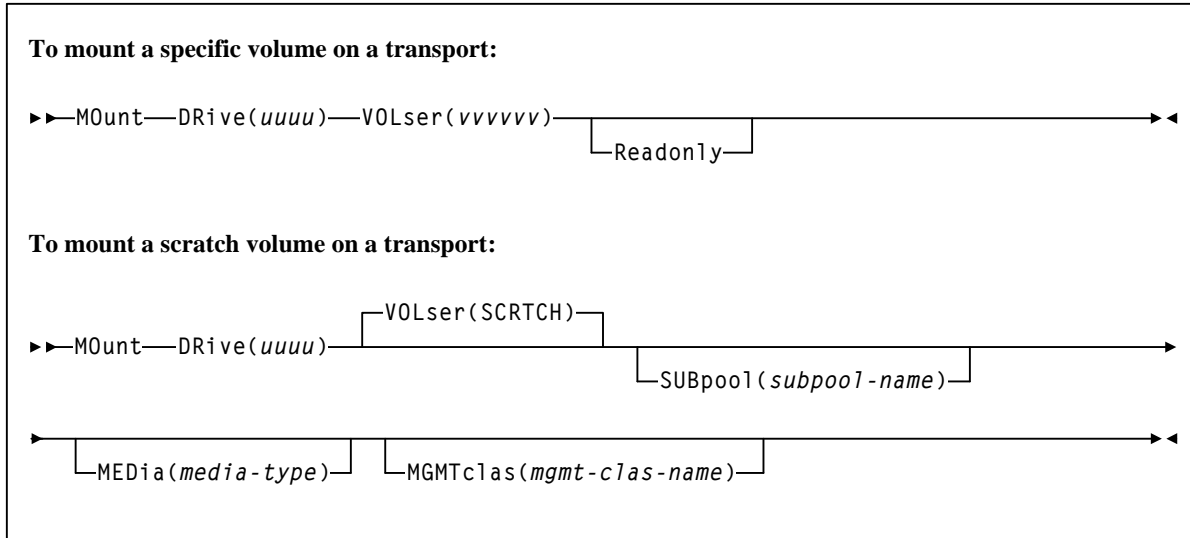
In the following example, the LOGdisk command is used to enable logging to a disk file.

```
LOGDISK ON
```

# MOunt

The MOunt command mounts a volume on a drive.

## Syntax



**FIGURE 4-16** MOunt syntax

## Parameters

### DRive

specifies a tape drive address of the transport on which the volume is to be mounted.

*uuuu*

the tape drive address. If the DRIVemap command is used, this is the CLIENT address.

### VOLser

optionally, specifies the volume to be mounted. If this parameter is not specified, a scratch volume is mounted.

*vvvvvv*

the volume serial. Specify SCRTCH for a scratch volume.

### Readonly

optionally, specifies that the volume is to be mounted for read-only access. This parameter is only valid for a specific mount.

**SUBpool**

optionally, specifies that a scratch volume is to be taken from a scratch subpool. If this parameter is not specified, then the behavior is dependent upon how scratch pools are defined on the HSC server. See the HSC MOUNT command description for details. This parameter is only valid for a scratch mount.

*ssss*

the subpool name.

**MEDia**

optionally, specifies the type of media for the scratch volume. The specified media must be compatible with the request DRIVE. This parameter is only valid for a scratch mount.

*mmmm*

the media type.

See [Appendix A, “MEDia, RECtech, and MODel Values”](#) for a list of valid *media-type* values.

**Note** – If MEDia is not specified, the next scratch volume is selected without regard to media type.

**MGMTclas**

optionally, specifies a Management Class defined in the HSC/VTCS MGMTclas control statement. This parameter is only valid for a scratch mount.

*cccc*

the Management Class name.

**Example**

In the following example, the MOUNT command is used to mount volume AAA001 on drive 2900.

```
MOUNT DRIVE(2900) VOLSER(AAA001)
```

## MSGDef

The MSGDef command defines the appearance of VM Client system messages, and controls which messages are displayed and suppressed.

### Syntax

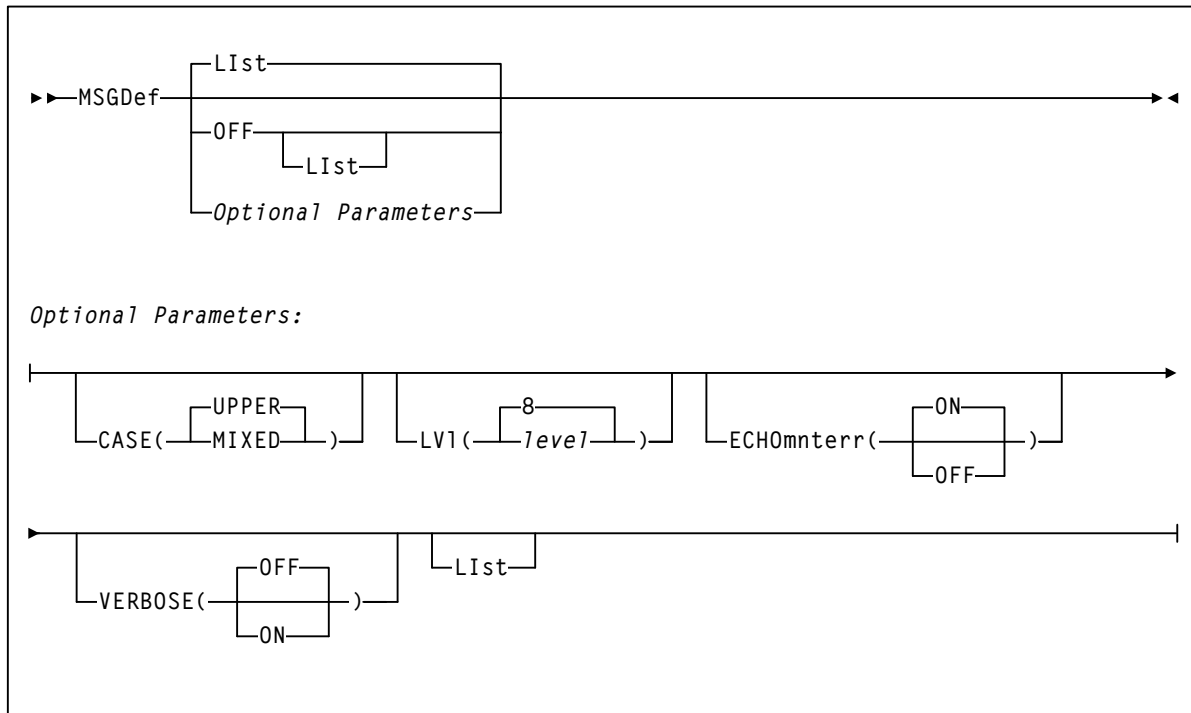


FIGURE 4-17 MSGDef syntax

### Parameters

#### List

optionally, lists current default VM Client message settings.

- List is the default when **no** other parameters are specified on the MSGDef command.
- List may be specified with other parameters. In this case, the List is generated **after** the other parameters are processed.

#### OFF

optionally, resets all MSGDef values to original VM Client default settings. Specify List with this parameter to list these settings.

**CASE**

optionally, specifies the message case. Valid values are UPPER or MIXED.

**UPPER**

specifies upper case. This is the default.

**MIXED**

specifies mixed case.

**LVI**

optionally, specifies the default level used to control which VM Client messages are displayed and suppressed.

*level*

the default level. Valid values include the following:

0	Display error messages only.
4	Display error and warning messages from the VM Client service machine.
8	Display all VM Client service machine messages and allocation job log warning messages. This is the default if the MSGDef parameter is not specified.

If no value is specified, the default value of 8 is used.

**Note** – Levels higher than 8 are used for diagnostic purposes and should **only** be specified as directed by StorageTek Software Support.

**ECHOmnterr**

optionally, specifies whether mount errors generated by the HSC are echoed directly to the console for the VM Client.

**ON**

Mount errors generated by the HSC are echoed to the console for the VM Client. This is the default.

**OFF**

Mount errors generated by the HSC are not echoed to the console for the VM Client.



**VERBOSE**

optionally, specifies whether SMC0190 and SMC0191 messages are displayed whenever VM Client settings are altered.

**ON**

SMC0190 and SMC0191 messages are displayed.

**OFF**

SMC0190 and SMC0191 messages are **not** displayed. This is the default.

**Example**

In the following example, the MSGDef command is used to specify that messages appear in mixed case, and that only error and warning messages from the VM Client service machine are displayed.

```
MSGD CASE(MIXED) LV1(4)
```

## OPERator

The OPERator command specifies the virtual machine to receive VM Client messages.

### Syntax



**FIGURE 4-18** OPERator syntax

### Parameters

#### Lst

optionally, displays current operator settings.

- Lst is the default when **no** other parameters are specified for the OPERator command.
- Lst may be specified with other parameters. In this case, the list is generated **after** the other parameters are processed.

#### ID

optionally, specifies the name of the virtual machine to receive VM Client messages.

##### *vm-userid*

the userid of the virtual machine. This must be a defined VM userid. If an asterisk (\*) is entered, it is translated to the VM Client service machine id.

The default is the userid of the VM Client service machine.

### Example

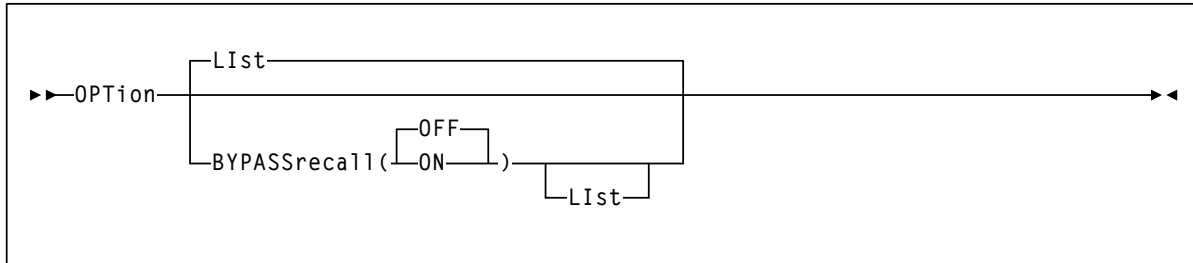
In the following example, the OPERator command is used to specify the OPER machine to receive messages.

```
OPERATOR ID(OPER)
```

## OPTion

The OPTion command specifies VM Client system options.

### Syntax



**FIGURE 4-19** OPTion syntax

### Parameters

#### Lst

optionally, displays current VM Client system settings.

- Lst is the default when **no** other parameters are specified for the OPTion command.
- Lst may be specified with other parameters. In this case, the list is generated **after** the other parameters are processed.

#### BYPASSrecall

optionally, specifies whether a specific MOUNT of a scratch virtual volume (VTV) will bypass the RECALL of the VTV if the VTV is MIGRATED..

##### OFF

The RECALL of scratch VTV for a specific MOUNT will not be bypassed. This is the default.

##### ON

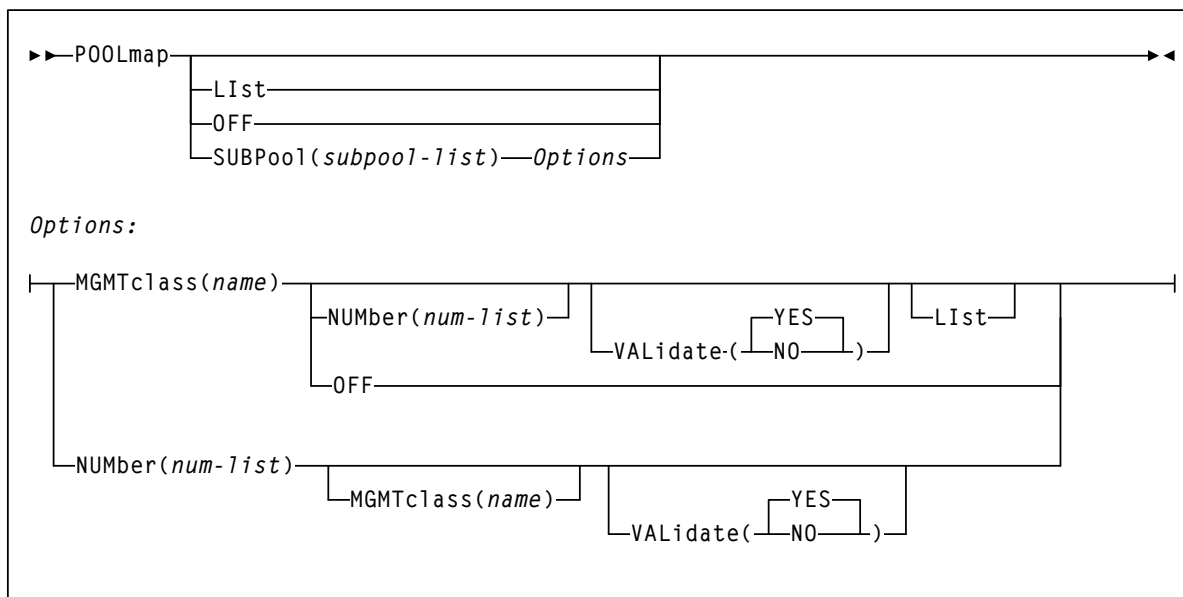
The RECALL of a scratch VTV for a specific MOUNT will be bypassed.

## POOLmap

VM Tape Management Systems (TMS) that use the VM Client Tape Management Interface (VMTMI) typically only specify a subpool name for scratch requests; and in some cases the TMS only supplies subpool numbers.

The POOLmap command provides a convenient (and recommended) method to associate a management class or subpool number with an ELS scratch subpool name. Additionally, this command provides validation of the subpool and management class names with the ELS server, and therefore must be executed **after** TCP/IP communications with the server are established.

### Syntax



**FIGURE 4-20** POOLmap syntax

## Parameters

### Lst

optionally, lists all current POOLmap mappings.

- Lst is the default when no other parameters are specified.
- Lst may be specified with other parameters. In this case, Lst is applied after all other parameters are processed.

### OFF

optionally, removes all POOLmap entries.

If this parameter is specified with the MGMTclass or SUBPool parameters, then only the matching POOLmap entries are removed.

**Note** – OFF is mutually exclusive with NUMber and VALidate parameters.

### SUBPool

specifies the scratch subpool names defined to the ELS server.

#### *subpool-list*

one or more scratch subpool names to be associated with the specified management class name.

**Note** – If OFF is specified, the SUBPool parameter value of "\*" represents all POOLmap entries.

### MGMTclass

optionally, specifies the VTCS management class name to be associated with a given subpool name. This parameter is optional if NUMber is also specified.

#### *name*

the 1-8 character alphanumeric management class name.

### NUMber

optionally, specifies the number to be associated with a given subpool name. This parameter is optional if MGMTCLASS is specified.

#### *num-list*

one or more 1-3 character numbers. Valid values are 1-255.

This number is usually defined by the VM TMS. The number of entries for the NUMber parameter must match the number of entries for the SUBPool parameter.

**Note** – NUMber is mutually exclusive with OFF.

### VALidate

optionally, requests that the ELS server validate the subpool name and/or management class. The default is YES.

**Note** – VALidate is mutually exclusive with OFF.

## Examples

In the following example, the POOLmap command assigns the management class DAILY to subpools VSM1 and VSM2.

```
POOLMAP MGMT(DAILY) SUBPOOL(VSM1,VSM2)
```

In the next example, the POOLmap command assigns number values to 9840 scratch subpools TEST9840 and PROD9840 and lists mappings.

```
POOLMAP SUBP(TEST9840,PROD9840) NUMBER(1,2) LIST
```

For a virtual scratch subpool that requires number values, ensure that the NUMBER value count matches the SUBPool count, as shown in the following example:

```
POOLMAP MGMT(DAILY) SUBP(VCART1,VCART2) NUM(3,4)
```

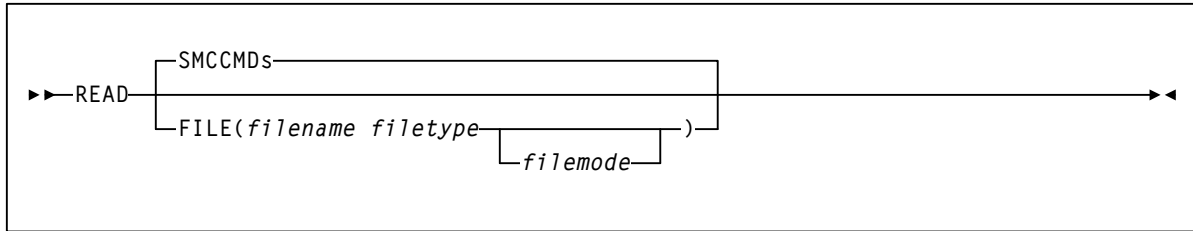
In the next example, the POOLmap command specifies that the management class and subpool name are not to be validated with the ELS server.

```
POOLMAP MGMT(MONTHLY) SUBP(NEW_VSM_3) VAL(NO)
```

# READ

The READ command is used to enter a series of commands using an input data set instead of console commands.

## Syntax



**FIGURE 4-21** READ syntax

## Parameters

### SMCCMDS

optionally, re-processes commands contained in the data set specified in the SMCCMDS FILEDEF of the VM Client START procedure.

### FILE

optionally, specifies the file to READ.

#### *filename*

the file name.

#### *filetype*

the file type.

#### *filemode*

the file mode. The default is 'A'.

## Example

In the following example, the READ command is used to process commands in the SMCCMDS FILEDEF of the VM Client startup EXEC.

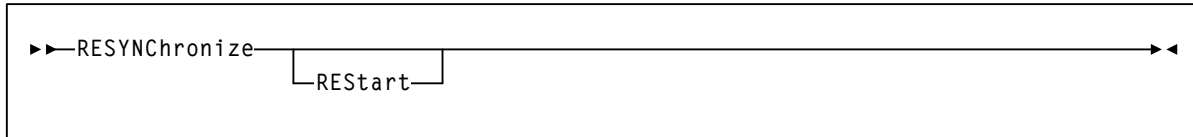
```
READ SMCCMDS
```

## RESYNChronize

The RESYNChronize command is used to re-establish connections to all defined TapePlexes for the purpose of acquiring drive configuration information from all TapePlexes.

This action is automatically performed when the VM Client first activates a new path to a TapePlex, or when an HSC server reports a configuration change.

### Syntax



**FIGURE 4-22** RESYNChronize syntax

### Parameters

#### REStart

optionally, starts the RESYNChronize attempt at the first server, regardless of the last active path.

### Example

In the following example, the RESYNChronize command is used to restart communications from the first server.

```
RESYNC RESTART
```





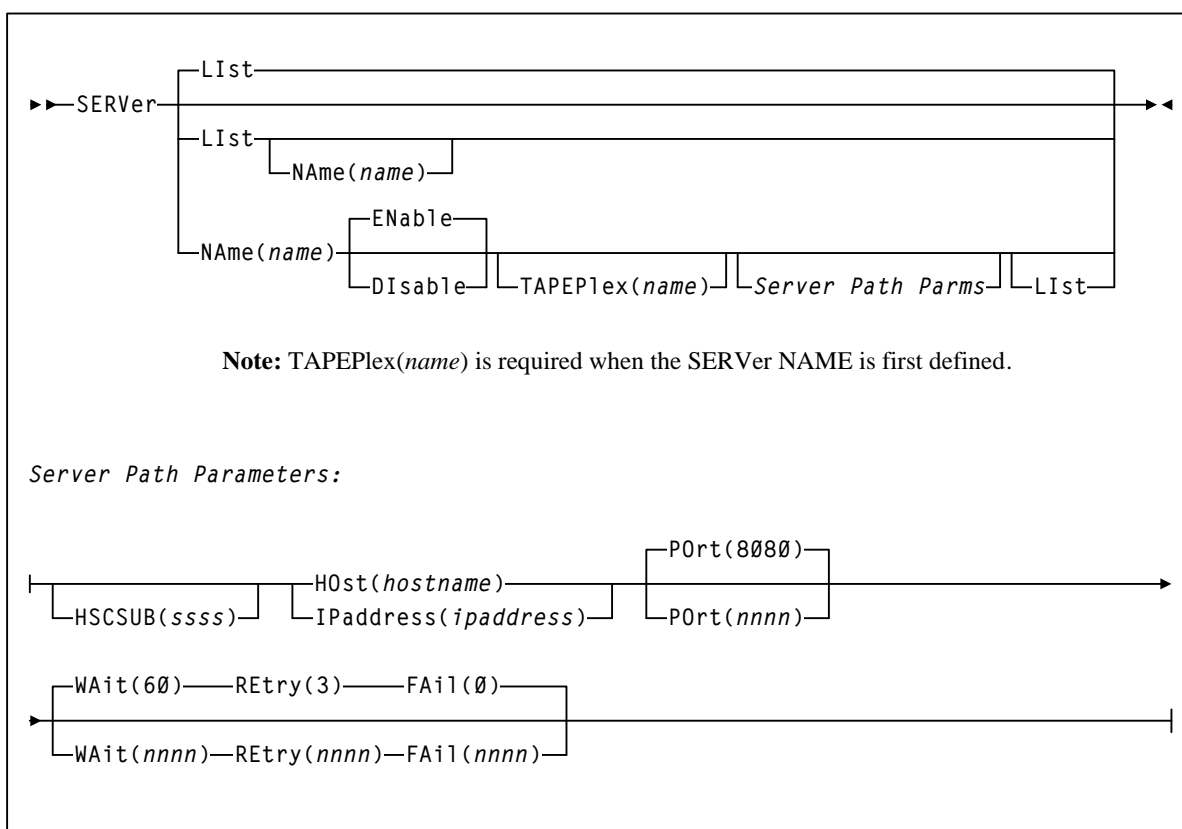
## SERVer

The SERVer command defines a named path to a remote library server. The SERVer command describes the communication path to a StorageTek HTTP server. The SERVer command can also list Servers defined to the VM Client.

### Note –

- Before a SERVer is defined the TapePlex that it references must be defined using the TAPEPlex command.
- The TapePlex name associated with a SERVer cannot be changed. See [“TAPEPlex” on page 82](#) for more information.

### Syntax



**FIGURE 4-24** SERVer syntax

## Parameters

### List

optionally, displays status information for TapePlex server paths.

- List is the default when no parameters are specified on the SERVER command. In this case, all library server paths are listed.
- List may be specified with other parameters. When specified with parameters other than NAME, the List is generated **after** the other parameters are processed.

### NAME

optionally, specifies a TapePlex server path for which status is displayed.

*name*

the server path name.

### NAME

optionally, specifies the communication path or route to the TapePlex server.

*name*

an identifier for the path parameters. This name is reported in any communication error messages. The following rules apply:

- The value must be one to eight characters in length.
- The first character must be an alpha character or 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, or hyphen.

### ENable

optionally, enables the specified server path to be selected for mount requests.

### Disable

optionally, disables the specified server path. If this is the only path to the TapePlex, the TapePlex is unavailable for mount requests.

**TAPEPlex**

optionally, specifies the TapePlex name associated with the ACS hardware configuration. The TAPEPlex parameter must be specified when a new server is defined.

***name***

the TapePlex name. This name is reported in any TapePlex server error messages. The following rules apply:

- The value must be one to eight characters in length.
- The first character must be an alpha character or 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, or hyphen.

**Note** – You can define multiple paths to a single TapePlex.

**HSCSUB**

optionally, specifies the name of the HSC subsystem that represents the TapePlex associated with the server. This parameter is required only when there is more than one HSC subsystem executing on the server host (HSC running in MULT mode).

***ssss***

the HSC subsystem name.

**HOst**

optionally, specifies the IP resolver host name of the TapePlex server. For DNS lookup, the VM Client must have access to the TCPIP DATA file.

***hostname***

the name of the remote host.

**Note** – HOst and IPaddress are mutually exclusive.

**IPaddress**

optionally, specifies the IP address of the TapePlex server.

***ipaddress***

the IP address of the remote host.

**Note** – IPaddress and HOst are mutually exclusive.

**POrt**

optionally, specifies the server port.

***nnnn***

the server port, from 0-65535. The default is 8080.

**WAI**

optionally, specifies the maximum default wait time for requests before the VM Client times out the request.

*nnnn*

the wait time in seconds, from 0-9999. The default is 60.

**Note** – The default wait time does not apply to mount, dismount, eject, or move requests which have a default timeout value of 10 minutes, 10 minutes, 24 hours, and 1 hour respectively.

**REtry**

optionally, specifies the number of retry attempts for any single request before the task is allowed to resume, and a failure recorded.

*nnnn*

the number of retries, from 0-9999. The default is 3.

**FAil**

optionally, specifies the maximum number of failures after successful communication is established, before the specific server path is disabled or placed out of service.

*nnnn*

indicates the number of failures. The default is 0.

If 0 is specified, the named SERVER will never be automatically disabled due to communications errors.

This value should be specified if there are no backup SERVER paths to a named library.

The FAIL limit count only applies after successful communication has been established on this SERVER path.

**Example**

In the following example, the SERVER command is used to add a server named DENVER1 for TapePlex DENVER.

```
SERVER NAME (DENVER1) TAPEPLEX (DENVER) IP (11.22.33.44)
PORT (7777)
```

## SMCUDBX

The SMCUDBX command supports scratch synchronization between the VM Client host and the HSC CDS. SMCUDBX accesses the client tape management system to produce an extract file that can be read as input by the VM Client Scratch command (SMCUSCR).

When the VM Tape Management System controls the scratch selection, the virtual volumes can be left in non-scratch status as the volumes are returned to scratch by the TMS. This may lead to extraneous migrates and recalls as the virtual volumes are reused. A scratch synchronization between the TMS and library server will eliminate the extra migrate and recalls of the scratch virtual volumes.

If the VM client host and HSC server share a tape management system, the HSC SLUCONDB utility may be run using the common tape management data base as input.

If, however, the VM client host uses a tape management system that is not accessible to the HSC host, the SMCUDBX command can be used to extract scratch volumes from the VM tape management system. The SMCUDBX output extract file can be used as input on the HSC host to the SLUCONDB utility or as input to the VM Client Scratch (SMCUSCR) command.

The SMCUSCR command supports timing synchronization by comparing the GMT time of the SMCUDBX run with the last select time of the HSC CDS volume record. If the volume was selected after the VM Client scratch extract, it is not eligible for scratch by SMCUSCR or SLUCONDB.

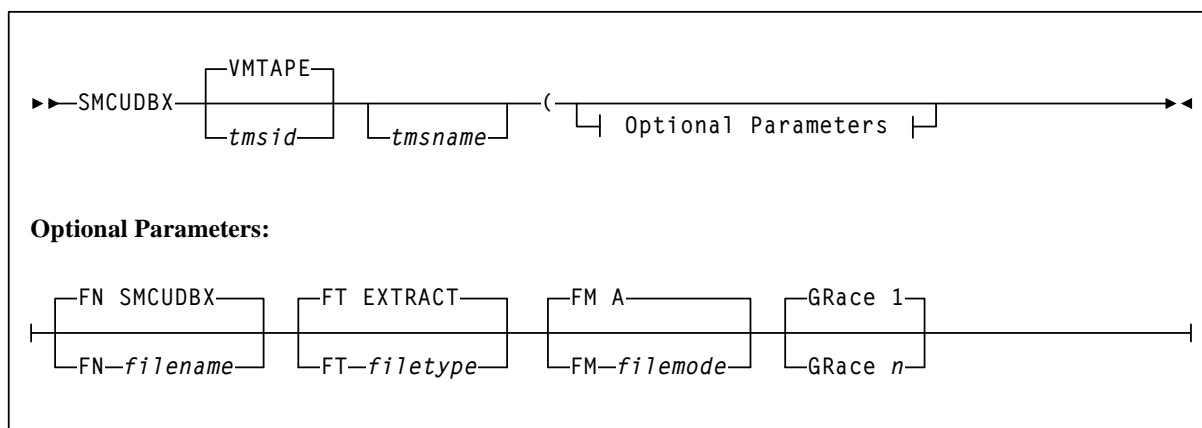
SMCUDBX processes Tape Management system scratch queries and selects volume serial numbers for processing.

If your installation uses a different tape management system, you can develop a program to produce an extract file in the following format:

Position	Field	Format
1	Scratch volume serial	CL6
7	GMT of run	CL8
15	Unused	CL2

The time field in the scratch extract is a character representation of the first four bytes of the output of the MVS STCK macro. For example, the STCK macro produces a result of X'B0912345 6789ABCD', the GMT field of the scratch extract file contains the character value of C'B0912345'.

## Syntax



**FIGURE 4-25** SMCUDBX syntax

## Parameters

### *tmsid*

the Tape Management System. The supported systems are:

- VMTAPE - CA VMTape. This is the default.
- TAPEMANAGER - IBM Tape Manager for z/VM

### *tmsname*

optionally, the userid of the tape management system service machine. The defaults are:

- VMTAPE - when VMTAPE is specified for *tmsid*.
- TMTMM - when TAPEMANAGER is specified for *tmsid*.

**Note** – If the *tmsname* is to be specified, the *tmsid* must also be specified.

### *FN filename*

optionally, specifies the file name of the extract file. The default is SMCUDBX.

### *FT filetype*

optionally, specifies the file type of the extract file. The default is EXTRACT.

### *FM filemode*

optionally, specifies the file mode of the extract file. The default is A.

### *GRace n*

optionally, specifies a grace period to scratch tapes. Tapes that have been used in this grace period will not be scratched. Valid entries are 0-99 days. The default is 1 day.

The VM user id executing the SMCUDBX command may need authorization from the TMS server to obtain scratch volume information:

- For CA VM:Tape, OPERATOR authorization is required. The CA VM:Tape LIST SCRATCH and QUERY VERSION commands are issued by the SMCUDBX command
- For IBM Tape Manager for z/VM, authorization to issue the TAPEQRY with the SYS option is required.

The VM user id executing the SMCUDBX command requires access to the Tape Management system program to process commands:

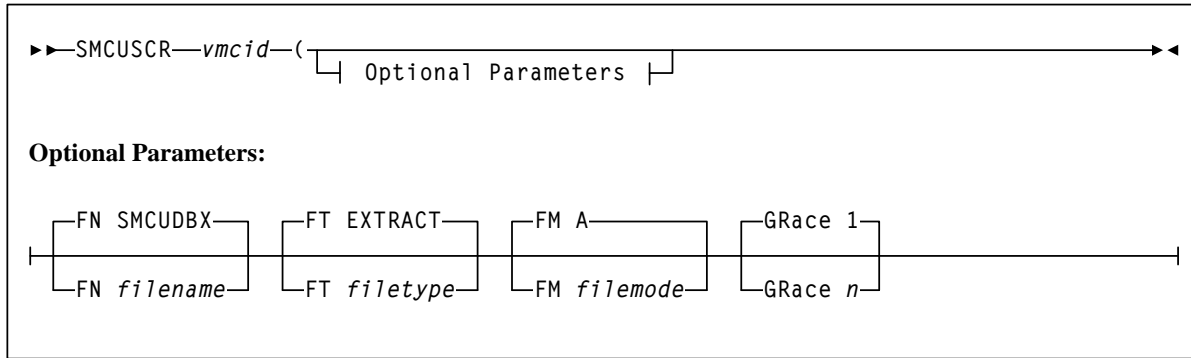
- For CA VM:Tape, VMTAPE or the module renamed to the VMTAPE service machine id.
- For IBM Tape Manager for z/VM, TAPCMD.



## SMCUSCR

The SMCUSCR utility generates scratch transactions from an extract file created by the VM Client SMCUDBX command.

### Syntax



**FIGURE 4-26** SMCUSCR syntax

### Parameters

**vmcid**

the VM Client user id.

**FN filename**

optionally, specifies the file name of the extract file. The default is SMCUDBX.

**FT filetype**

optionally, specifies the file type of the extract file. The default is EXTRACT.

**FM filemode**

optionally, specifies the file mode of the extract file. The default is A.

The VM user id executing the SMCUSCR command requires the following:

- authorization from the VM Client server to run the TMI commands
- access to the SMCTMIRQ MODULE on the VM Client run disk

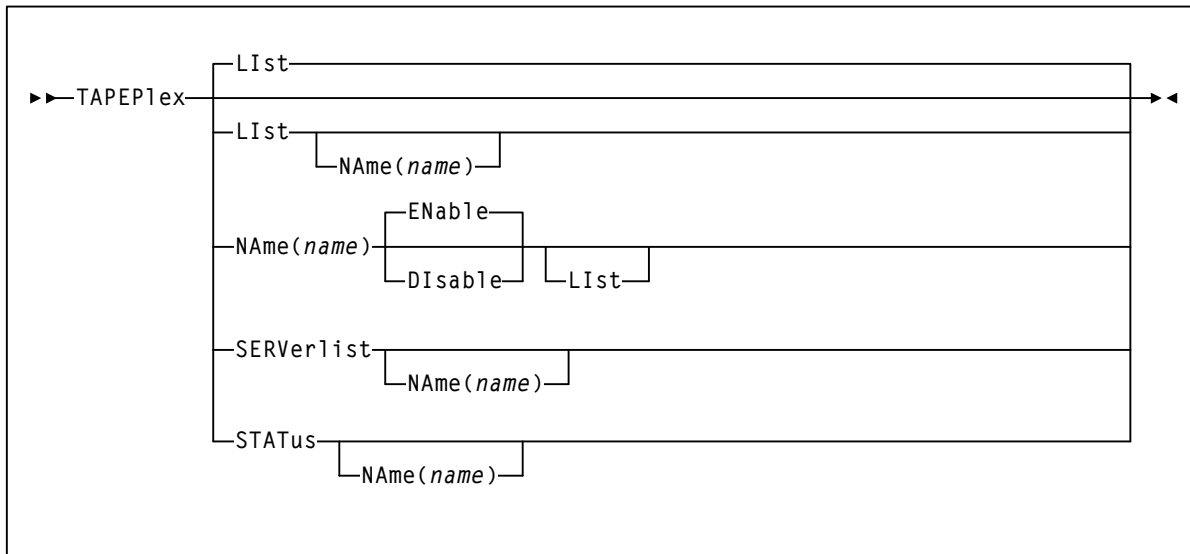
## TAPEPlex

The TAPEPlex command defines a TapePlex; a specific StorageTek tape hardware configuration normally represented by a single CDS.

**Note –**

- TAPEPlex and SERVer commands are required to access HSC TapePlexes.
- The TAPEPlex command can also list TapePlexes that VM Client tries to communicate with, and report their status.

### Syntax



**FIGURE 4-27** TAPEPlex syntax

### Parameters

**List**

optionally, lists the specified TapePlex.

**NAME**

specifies the TapePlex name to be listed.

*name*

the TapePlex name.

**Note –**

- List is the default when no parameters are specified on the TAPEPlex command. In this case, all TapePlexes are listed.
- List may be specified with other parameters. In this case, the named TapePlex is listed after the other parameters are processed.

**NAME**

optionally, specifies the TapePlex name to be defined or modified.

*name*

the TapePlex name. This name is reported in any TapePlex error message. The following rules apply:

- The value must be between one and eight characters in length.
- The first character must be either an alpha character or 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 or hyphen.

**ENABle**

enables the specified TapePlex to be selected for mount requests. This is the default.

**DISable**

disables the specified TapePlex. The TapePlex is **not** used for any mount requests.

**SERVerlist**

optionally, lists defined TapePlexes, their attributes and associated servers. The SERVerlist parameter may also be specified with the NAME parameter to limit the display to a single TapePlex.

**NAME**

specifies the TapePlex name for which servers are to be listed.

*name*

the TapePlex name.

**STATus**

optionally, lists current status of all TapePlexes, or a single named TapePlex. The TapePlex status indicates whether a TapePlex is active, inactive, or disabled. For an active TapePlex, the status lists the name of the current server. STATus does **not** perform a RESYNChronize command.

**NAME**

specifies the TapePlex name for which status information is to be listed.

*name*

the TapePlex name.

## Example

In the following example, the TAPEPlex command is used to define a TapePlex named DENVER (assuming it is not already defined).

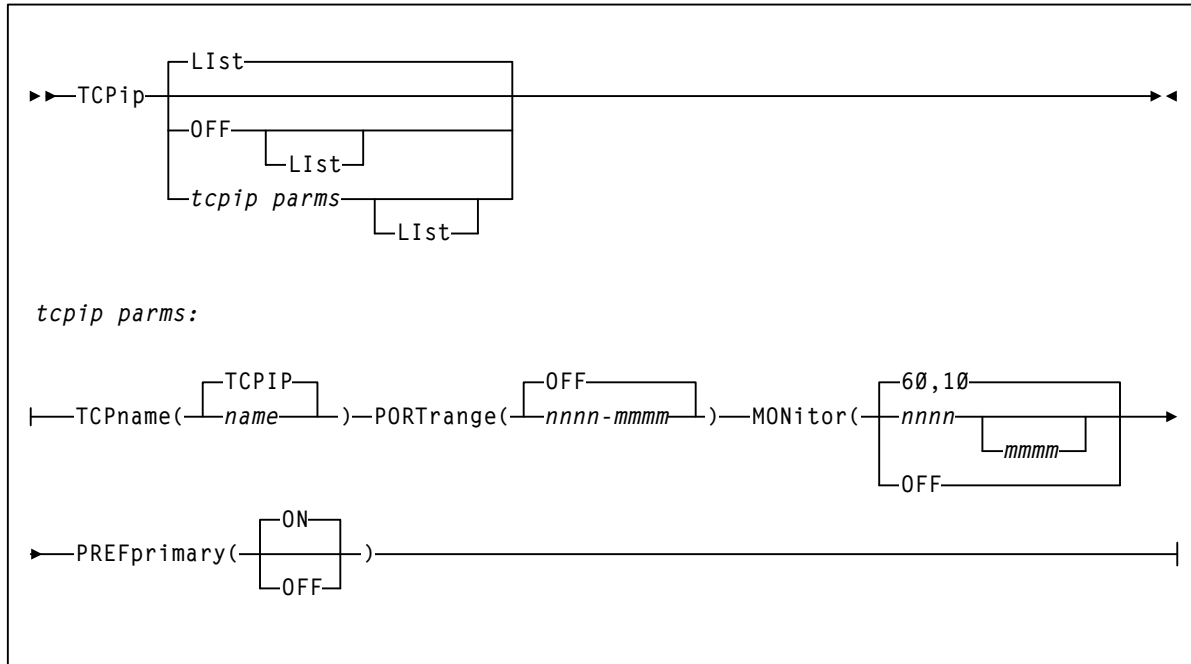
```
TAPEPLEX NAME (DENVER)
```

**Note** – A SERVER command must be specified to define a communications path to TapePlex DENVER. See [“SERVER” on page 74](#) for an example.

## TCPip

The TCPip command alters and/or lists current settings for your TCP/IP communications environment. It allows you to direct TCP/IP requests to a specific TCP/IP stack on a VM host. The TCPip command can be issued at any time.

### Syntax



**FIGURE 4-28** TCPip syntax

### Parameters

#### List

optionally, displays the current TCP/IP settings. If a PORTrange is specified, List also displays the currently bound port numbers and the high-water port numbers indicating the largest number of concurrent communication subtasks executing at one time.

- List is the default when no parameters are specified on the TCPip command.
- List may be specified with other parameters. In this case, the List is generated **after** the other parameters are processed.

#### OFF

optionally, specifies that system defaults are used for VM Client TCP/IP communications.

**TCPname**

optionally, specifies the TCP/IP service machine on a VM host.

***name***

The userid of the TCP/IP service machine on VM to target TCP/IP communications. The default is TCPIP.

**PORTrange**

optionally, specifies a range of ports to be used by the VM Client to bind() sockets on the client when communicating on remote server paths.

When PORTrange is defined, the VM Client binds client sockets to one of the ports within the specified PORTrange and will **not** use client ports outside the PORTrange. This allows the VM Client to operate behind a firewall that restricts communication to known ports. A unique port is required for each concurrent subtask requiring communication services for a volume lookup, mount, etc. If a PORTrange is not defined, then any ephemeral port is used by the VM Client.

Only one PORTrange can be active at a time, but you can dynamically re-define the PORTrange even if the new PORTrange overlaps with the old PORTrange.

***nnnn-mmmm***

the port number range.

Each port number can have a value of 1-65535. However:

- The **minimum** port number range that can be specified is **10** (e.g., 6401-6410).
- The **maximum** port number range that can be specified is **1000** (e.g., 6401-7400).

**OFF**

PORTrange logic is disabled. As a result, any ephemeral port is used. This is the default.

**Note –**

- It is recommended that if you specify a PORTrange, you specify a PORTrange that does not conflict with TCP/IP well-known ports.
- It is recommended that if you specify a PORTrange, you specify a PORTrange greater than the anticipated number of concurrent subtasks requesting communication services. For most installations, a PORTrange of 40 ports is sufficient. However, if SMC0128 messages are produced with a return code indicating “no free port” then a larger PORTrange is required.
- The TCPip LIST command may be used to display the high-water port number, indicating the largest number of concurrent communication subtasks executing at one time.

**MONitor**

optionally, specifies the communication monitor subtask scan interval and communication monitor subtask message interval.

***nnnn***

the monitor scan interval in seconds. The communication monitor wakes every *nnnn* seconds to perform library communication validation. Specify a value between 10 and 9999. The default is 60.

It is recommended that you preserve the default setting of 60 to enable a monitor scan every minute. A value that is too low can potentially degrade performance when inactive libraries exist. A value that is too high can delay a return to the primary server if PREFPRIMARY(ON) is specified.

***mmmm***

optionally, the monitor scan interval in number of scans. Communication error messages are displayed according to this interval. Specify a value between 0 and 9999. The default is 10.

The default MONITOR(60,10) setting specifies a monitor scan interval of 60 seconds, and a monitor message interval of 10 scans. A scan is performed every minute, but error messages are only produced once every 10 scans.

An *mmmm* value of 0 disables all non-fatal or non-disabling error messages issued by the communication monitor subtask. However, errors resulting in the disabling of a server communication path are still issued.

**OFF**

optionally, disables the communication monitor task.

**PREFprimary**

optionally, enables or disables automatic primary server switching. Automatic primary server switching requires the communication monitor subtask to be active. If MONITOR(OFF) is specified, primary server switching is disabled.

**ON**

enables automatic primary server switching. The communication path automatically switches back to the primary server from any secondary server when that primary server becomes available. This is the default.

**OFF**

disables automatic primary server switching.

**Example**

In the following example, the TCPip command is used to direct TCP/IP requests to a VM service machine named TCPIP using any ephemeral port.

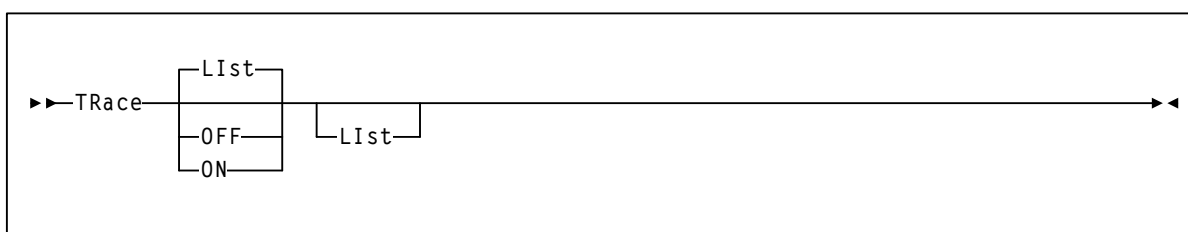
```
TCPIP TCPNAME(TCPIP) PORTRANGE(OFF)
```

## TRace

The TRace command enables VM Client tracing. The VM Client trace file is written to the TRACE FILEDEF file.

**Note** – This command may impact system performance and should be used only as directed by StorageTek Software Support.

### Syntax



**FIGURE 4-29** TRace syntax

### Parameters

#### List

optionally, lists current VM Client trace settings.

- List is the default when no parameters are specified on the TRace command.
- List may be specified with other parameters. In this case the list is generated **after** the other parameters are processed.

#### OFF

optionally, disables VM Client tracing.

#### ON

optionally, enables VM Client tracing.

### Example

In the following example, the TRace command is used to enable VM Client tracing.

```
TRACE ON
```



---

## ELS 7.1 Server Considerations

### Overview

The VM Client is a thin-client and relies on the resources available within an ELS 7.1 server TapePlex connected via TCP/IP. This chapter describes ELS server considerations.

### SMC HTTP Server Component

The VM Client specifies a server IP address and port number defined in the ELS 7.1 SMC configuration. Refer to the publication *Configuring and Managing SMC* for information about how to configure an HTTP server using the SMC HTTP command.

The following is an example of the SMC HTTP command:

```
HTTP START PORT 4242
```

### Scratch Subpools

Unlike VM/HSC, the VM Client does not allow management of scratch subpools. Instead, scratch subpools are defined and managed on the ELS 7.1 server.

A tape management system (TMS), such as VM:Tape, cannot use TMI requests to define scratch subpools. A TMS must use existing scratch subpools defined on the ELS 7.1 server. In the case of VM:Tape, an attempt to define scratch subpools fails yet VM:Tape initialization continues.

The following is an example of VM:Tape initialization:

```
VMTHSC693I The HSC interface is connecting.
VMTHSC000I Sending to VMCLIENT: QCONFIG with wrong length
VMTHSC000I Sending to VMCLIENT: QCONFIG with right length
VMTHSC000I Sending to VMCLIENT: QDRIVES
VMTHSC999I VMCLIENT completed QDRIVES command successfully
VMTHSC000I Sending to VMCLIENT: DEFSCR 5 4
VMTHSC697E HSC server VMCLIENT ACSRQ=DEFSCR RC=16
Reason=00001004
VMTHSC721E HSC scratch pool initialization failed.
VMTHSC704I The interface to HSC is ready for use.
```

If VM:Tape attempts to use a scratch subpool that is not defined to the HSC server, the VM Client replies with a nonzero TMI return code and reason code. For example:

```

VMTHSC000I Sending to VMCLIENT: QSCRATCH TEST
VMTHSC697E HSC server VMCLIENT ACSRQ=QSCRATCH
          RC=16 Reason=00008036
VMTHSC698R 'QSCRATCH TEST ' to VMCLIENT failed;
          Enter RETRY, CANCEL, or NOARM;; Reply 1

```

It is the customer's responsibility to keep scratch subpool names and volumes within the subpools in sync with the VM TMS.

## VTCS Management Classes

It is highly recommended that you use the VM Client POOLmap command to associate a VTCS management class with a scratch subpool that contains virtual tape volumes (VTVs). When the VM Client receives a TMI request that specifies a scratch subpool name, the VM Client will use the POOLmap management class for selecting and mounting VTVs.

See [“POOLmap” on page 68](#) for more information about the POOLmap command.

## VM:Tape Allocation Exit

If the VM:Tape configuration contains multiple tape drives and these tape drives are defined in different ACSs within a TapePlex, scratch mount requests will require the installation of a VM:Tape allocation exit. As an example, the configuration may include two 9840C tape drives, one defined in ACS00 and the other defined in ACS01.

To install a VM:Tape allocation exit, complete the following steps:

1. Copy the SMCVMTAP sample file to VMTAPE's 191 minidisk with a filename and filetype of SMCVMTAP EXEC. You may need to modify the VM Client virtual machine ID in this file.
2. Modify the VMTAPE configuration file to enable the allocation exit. For example:

```
EXIT ALLOCATE SMCVMTAP EXEC
```

3. Modify VMTAPE's PROFILE EXEC to access the VM Client run disk and execute the SMCALLOC EXEC to load the necessary programs into storage. Make sure you include these commands prior to invoking the VMTAPE startup EXEC. For example:

```

CP LINK VSMC710A 202 202 RR
ACCESS 202 J
EXEC SMCALLOC

```

---

## VM Client Messages

### Overview

This chapter describes system messages issued by VM Client. These messages are identified by the "SMC" prefix.

### Message Listing

**SMC0000**

{{CCCCCCCC}} command string

**Level:** 0

**Explanation:** The VM Client received an input command from an operator virtual machine. The virtual machine ID, if available, is listed followed by the command string.

**System Action:** None

**User Response:** None

**SMC0001**

VM Client *Vn.n.n* system initializing

**Level:** 0

**Explanation:** The VM Client version *n.n.n* system initialization process has begun.

**System Action:** None

**User Response:** None

**SMC0002**

CCCCCCCCCCCC failed; return code=XXXX, reason code=XXXX

**Level:** 0

**Explanation:** Operating system facility CCCCCCCCCCCC completed with the specified non-zero return code and reason code.

**System Action:** Depending on the type of error, initiation/termination may try to continue.

**User Response:** Look for IBM related messages in the SYSLOG and refer to the appropriate IBM documentation for the explanation.

### SMC0005

Invalid command CCCCCCCC [at line *nnnn* of SMCCMDS | SMCPARMS]

**Level:** 0

**Explanation:** An undefined command, CCCCCCCC, was encountered by the VM Client.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

### SMC0013

TRACE settings:  
CCCC....CCCC

**Level:** 0

**Explanation:** The TRACE command was specified with the LIST keyword. The SMC0013 multiline message lists the current settings for the VM Client.

**System Action:** None

**User Response:** None

### SMC0014

Unmatched [quote | or invalid parenthesis] detected; command ignored  
[at line *nnnn* of SMCCMDS | SMCPARMS]

**Level:** 0

**Explanation:** The VM Client encountered a command containing an unterminated quoted string, or invalid or unmatched parenthesis.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

### SMC0015

Invalid keyword KKKKKKKK for the CCCCCCCC command  
[at line *nnnn* of SMCCMDS | SMCPARMS]

**Level:** 0

**Explanation:** The VM Client encountered a command that specified an invalid keyword KKKKKKKK.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

**SMC0016**

Invalid value VVVVVVVVV for keyword KKKKKKKK of the CCCCCCCC command  
[at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** The VM Client encountered a command that specified keyword KKKKKKKK with an invalid value VVVVVVVVV.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

**SMC0017**

Keyword KKKKKKKK of the CCCCCCCC command requires a value  
[at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** The VM Client encountered a command that specified keyword KKKKKKKK without an accompanying value (required by most keywords).

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

**SMC0019**

Duplicate keyword KKKKKKKK specified for the CCCCCCCC command  
[at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** The VM Client encountered a command that specified the same keyword, KKKKKKKK, more than once.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

**SMC0020**

Keyword KKKKKKKK1 of the CCCCCCCC command is mutually exclusive with keyword KKKKKKKK2 [at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** The VM Client encountered a command CCCCCCCC that specified multiple keywords, two of which (KKKKKKK1 and KKKKKKK2), are mutually exclusive.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

**SMC0022**

Invalid format or missing keywords for the CCCCCCCC command  
[at line *nnnn* of SMCCMDS | SMCPARMS]

**Level:** 0

**Explanation:** The VM Client encountered a command CCCCCCCC that contained either too many or too few keywords in the command line.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

**SMC0023**

CCCCCCCC command successfully processed  
[at line *nnnn* of SMCCMDS | SMCPARMS]

**Level:** 0

**Explanation:** The CCCCCCCC command was successfully validated and processed by the VM Client.

**System Action:** None

**User Response:** None

**SMC0024**

VM Client system initialization complete; RC=*nn*

**Level:** 0

**Explanation:** The VM Client system initialization process has completed with the return code indicated and the VM Client system is now ready to receive requests.

**System Action:** None

**User Response:** None

**SMC0025**

No {CCCCCCCC | control block} entries to list

**Level:** 0

**Explanation:** Command CCCCCCCC was specified with the LIST keyword. However, no entries were found in the VM Client queue for the specified command.

**System Action:** None

**User Response:** None

**SMC0027**

Keyword *KKKKKKK1* of the *CCCCCCCC* command requires keyword *KKKKKKK2* [at line *nnnn* of *SMCCMDS*|*SMCPARMS*]

**Level:** 0

**Explanation:** The VM Client encountered a command that specified keyword *KKKKKKK1*, but not the required co-requisite keyword, *KKKKKKK2*.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct, or enter the corrected command.

**SMC0029**

*CCCCCCCC* command processing error; [matching entry not found|command line truncated; will be ignored|parameter truncated; command ignored] [at line *nnnn* of *SMCCMDS*|*SMCPARMS*]

**Level:** 0

**Explanation:** An error was found processing the *CCCCCCCC* command.

**System Action:** Processing continues. The command is ignored.

**User Response:** Ensure that the syntax in the command data set is correct and enter the corrected command.

**SMC0034**

VM Client startup parameter *PPPPPPPP* must have a value

**Level:** 0

**Explanation:** During initialization, the VM Client system initialization program encountered a valid execution parameter, but it was not specified as a keyword value pair, and a value is required.

**System Action:** The VM Client system terminates.

**User Response:** Correct the VM Client system initialization exec to specify the correct execution parameters.

**SMC0035**

Error processing VM Client startup parameter *PPPPPPPP*; *CCCCCCCCCCCC*

**Level:** 0

**Explanation:** During initialization, the VM Client system initialization program encountered an error in the execution parameter string. The string *CCCCCCCCCCCC* indicates the type of error encountered.

**System Action:** The VM Client system terminates.

**User Response:** Correct the VM Client system initialization startup exec to specify the correct execution parameter string.

### SMC0036

VM Client startup parameter *PPPPPPPP* successfully processed

**Level:** 0

**Explanation:** During VM Client initialization, the execution parameter *PPPPPPPP* was successfully verified and processed.

**System Action:** None

**User Response:** None

### SMC0037

Invalid VM Client startup parameters; system terminating

**Level:** 0

**Explanation:** During initialization, the VM Client system initialization program detected an error processing the execution parameter string.

**System Action:** The VM Client system terminates.

**User Response:** Look for VM Client related messages in the service machine log. Associated messages may be (but are not limited to) SMC0033 SMC0034, or SMC0035.

### SMC0041

{Command | Comment} beginning at line *nnnn* of {SMCCMDS | SMCPARMS} is unterminated

**Level:** 0

**Explanation:** A command or comment beginning at line *nnnn* of an input command file ended with a continuation character (+), but no continuation was found.

**System Action:** Processing continues. The command containing the unterminated string is ignored.

**User Response:** Ensure that the syntax in the command data set is correct.

### SMC0053

\*\*\*\* VM Client U1099 ABEND AT CCCCCCCC*n* \*\*\*\*

**Level:** 0

**Explanation:** A VM Client task has abended in module CCCCCCCC at abend sequence number *n*.

**System Action:** If the abend occurs in the processing of a TMI request, the request will not be processed.

**User Response:** Save the associated logs and dumps, and contact StorageTek Software Support.



**SMC0056**

*nn* bytes:

AAAAAAAA +0000 | XX..XX XX..XX XX..XX XX..XX | CC..CC |

**Level:** 0

**Explanation:** A VM Client Llst command was issued. The SMC0056 multiline message lists the *nn* bytes of storage in translated hexadecimal (XX.XX) and character (CC.CC) format, each line listing the next 16 bytes of storage beginning at hexadecimal address AAAAAAAAAA.

**System Action:** None

**User Response:** None

**SMC0057**

No {SMCPARMS|SMCCMDS} DDNAME statement found

**Level:** 0

**Explanation:** During VM Client initialization, the specified SMCPARMS DD or SMCCMDS DD was not present in the VM Client startup exec.

**System Action:** Initialization continues.

**User Response:** None

**SMC0058**

Error opening {DDNAME {SMCPARMS|SMCCMDS}|DSNAME DDDDDDDD}

**Level:** 0

**Explanation:** The VM Client encountered a READ command, but the specified DDNAME or DSNAME could not be opened.

**System Action:** The READ command is ignored.

**User Response:** Look for IBM related messages in the log, and refer to the appropriate IBM documentation for more information.

**SMC0060**

I/O error reading {DDNAME {SMCPARMS|SMCCMDS}|DSNAME DDDDDDDD}

**Level:** 0

**Explanation:** The VM Client received an I/O error attempting to read the SMCPARMS or SMCCMDS data set specified in the VM Client startup exec or a data set specified on a READ command.

**System Action:** The indicated data set is not processed.

**User Response:** Specify the correct data set name.

### SMC0061

Command beginning at line *nnnn* of {SMCCMDS|SMCPARMS} is too long; input ignored

**Level:** 0

**Explanation:** The VM Client encountered a multi-line command beginning at line *nnnn* of the specified file. This command exceeds 1024 characters in length.

**System Action:** Processing continues. The entire multi-line is ignored.

**User Response:** Ensure that the command data set has the correct syntax.

### SMC0062

Command CCCCCCCC [with parameter PPPPPPPP] is not allowed [{from console|at line *nnnn* of SMCCMDS|SMCPARMS}]

**Level:** 0

**Explanation:** The VM Client encountered a command or a command parameter that is not supported for the indicated command origin.

**System Action:** The command is ignored.

**User Response:** Issue the command from a valid command origin.

### SMC0063

MSGDEF settings:  
CCCC....CCCC

**Level:** 0

**Explanation:** A MSGDEF command has been issued with the LIST keyword. The SMC0063 multi-line message lists the current settings for the VM Client.

**System Action:** None

**User Response:** None

### SMC0084

*MMM DD YYYY HH:MM:SS UUUUUUUUU* active on hostid *VVVVVVVVV*

**Level:** 0

**Explanation:** The Date (*MMMM DD YYYY*), time (*HH:MM:SS*), service machine userid (*UUUUUUUUU*), and hostid (*VVVVVVVVV*) are displayed once a day at midnight and during VM Client initialization.

**System Action:** None

**User Response:** None

**SMC0086**

SMC system tasks

A(PCE)	Thread	Use	C-S	Userid	Last
-----	-----	-----	---	-----	-----
AAAAAAA	TTTTTT	UUUUUU	C-S	UUUUUUUU	TTTTTTTT

**Level:** 0

**Explanation:** A VM Client LIst TASKs command was issued. The SMC0086 multiline message lists the SMCPCE address, thread ID, use count, and current status information for all VM Client tasks.

**System Action:** None

**User Response:** None

**SMC0088**

Unable to [acquire/release] resource CCCCCCCC; attempt by VVVVVVVV  
XXXXXXXXX1 owned by XXXXXXXX2

**Level:** 0

**Explanation:** A shared VM Client resource could not be acquired or freed successfully. A task servicing virtual machine VVVVVVVV is attempting to acquire or free the resource, but cannot because another task holds the resource.

**System Action:** The request for virtual machine VVVVVVVV may not be processed correctly.

**User Response:** Contact StorageTek Software Support.

**SMC0093**

TCPIP SETTINGS:  
CCCC....CCCC  
TCPIP TCPNAME=CCCCCCCC

**Level:** 0

**Explanation:** A TCPIP LIst command was issued. The SMC0093 multi-line message lists the current settings for the VM Client system.

**System Action:** None

**User Response:** None

### SMC0113

SERVER=SSSSSSSS CCCCCCCC  
Status={active | never active | inactive | disabled}  
Errors=nnnn  
Messages=nnnn  
Retries=nnnn

**Level:** 0

**Explanation:** A SERVER command has been issued with the LIST keyword. The SMC0113 multi-line message lists the server settings and status for each server defined to the VM Client. Refer to the SERVER command for the parameter descriptions.

- **STATUS** indicates the status of the server.
  - **active** indicates that this server is the current path through which the specified TapePlex will be accessed.
  - **never active** indicates that communication was never attempted on this path, or was attempted but never succeeded.
  - **inactive** indicates that another server path is active, or that the last communication attempt on this path failed.
  - **disabled** indicates that the server has been disabled by a discovered incompatibility or by an operator command.
- **Errors** indicates the total number of errors on this server.
- **Messages** indicates the number of logical messages (volume lookup requests, mounts, dismounts) on this server path.
- **Retries** indicates how many message retries have been attempted.

**System Action:** None

**User Response:** None

**SMC0116**

Cannot find TAPEPLEX P P P P P P P P for SERVER S S S S S S S S [at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** A SERVER command was issued with a TapePlex name that was not previously defined.

**System Action:** The server is not added or updated.

**User Response:** Specify a TAPEPLEX command to define the TapePlex, then specify the SERVER command.

**SMC0117**

Cannot change TAPEPLEX name for existing SERVER S S S S S S S S [at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** A SERVER command was issued with the NAME of an existing server and a TapePlex name, but the TapePlex name of the existing server did not match the TapePlex name in the new command.

**System Action:** The command is rejected.

**User Response:** Omit the TapePlex name, change the TapePlex name to match the existing server, or change the server name to add a new server to the specified TapePlex.

**SMC0119**

SERVER C C C C C C C C now disabled

**Level:** 0

**Explanation:** The VM Client detected TCP/IP errors in excess of the FAIL count. See the preceding SMC0128/SMC0129 messages for the reason for the disable.

**System Action:** None. If there are no additional server paths defined for the associated library, the library hardware is no longer accessible.

**User Response:** Correct the problem with the TCP/IP network, server, or host operating system, and re-ENABLE the SERVER.

**SMC0123**

Drive range mismatch between CLIENT(XXXX1-XXXX2) and SERVER (XXXX3-XXXX4)

**Level:** 0

**Explanation:** A DRIVemap command was issued. One of the specified CLient range did not match the format of the corresponding Server range.

**System Action:** None

**User Response:** Reissue the command, ensuring that the CLient parameter and the Server parameter have corresponding formats and number of drives.

### SMC0128

TapePlex error:  
 {Fatal comm error detected |  
 Initialization error number *nn* or {*nn* | unlimited} |  
 Comm error number *nn* of {*nn* | unlimited}  
 Comm error limit exceeded}  
 USER=UUUUUUUUU TASK=XXXXXXXXXX {MSG=XXXXXXXXX}  
 TAPEPLEX=TTTTTTTTT SERVER=SSSSSSSS REQUEST=FFFF  
 {Client {IP=NNN.NNN.NNN.NNN} socket=NN port={*nnnn* | ANY}}  
 {Server IP=NNNN.NNNN.NNNN.NNNN port=*nnnn*}  
 {Bytes out=*nnnn* in=*nnnn*}  
 {Error=EEEE....EEEE}  
 {Reason=RRRR....RRRR}  
 {Response from STK HTTP server follows: HHHH...HHHH}  
 VM Client comm RC=*nnnn*

#### Level: 4

**Explanation:** The VM Client encountered an interface or communication error attempting to communicate with a TapePlex. The SMC0128 multiline message first lists the VM userid, transaction type, and TapePlex name associated with the error followed by the communication error and reason strings.

Examples of the reason strings include:

- Specific TCP/IP function errors (connect, send, recv, etc.)
- Data errors (incomplete or invalid data response)
- HSC server function errors
- SMC HTTP server errors

**Note** – Certain communication errors may result in a display of the entire HTTP server response as follows:

Response from HTTP server follows:  
 HTTP 1.0 401 Unauthorized

- If the message indicates Comm error limit (*nnn*) exceeded then the SMC0128 message will be followed by an SMC0119 message and the server path will be disabled by the VM Client.
- If the message indicates an Initialization error then the error occurred before any successful communication to the named server path. Such errors are not counted against the cumulative error count on the server path, and will not result in the named server being automatically disabled by the VM Client. Also, Initialization error messages will not be generated for every request, but will only be generated at 5 minute intervals until the path is successfully activated.

**System Action:** The allocation or mount event may not be processed by the VM Client.

**User Response:** Use the specified error reason to determine the cause of the problem.

**SMC0129**

{ERROR|WARNING}: No cartridge transport(s) for XXXX1- [XXXX2] for  
{UNITATTR|DRIVEMAP} {ADDRESS|CLIENT}

**Level:** 0

**Explanation:** A UNITATTR or DRIVEMAP command was issued specifying a device XXXX1 or range XXXX1-XXXX2. None of the specified devices is an MVS-defined cartridge transport device.

**System Action:**

- If the message indicates an ERROR, the command is not processed.
- If the message indicates a WARNING, the VM Client stores the value and may use it to translate addresses for non-MVS-defined RTD devices.

**User Response:** Review the devices specified in the command and re-issue the command if they are incorrect.

**SMC0133**

TAPEPLEX=PPPPPPPP  
CCCC....CCCC  
Status={disabled|active|inactive|never active}  
Requests=nnnn  
[SERVER=SSSSSSSS  
Status={disabled|active|inactive|never active}]

**Level:** 0

**Explanation:** A TAPEPLEX command was issued with the LIST keyword. The SMC0133 multiline message lists parameters and status for each TapePlex defined to the VM Client. Optionally, if the SERVERlist keyword was specified, the server status for all servers associated with this TapePlex is also displayed.

- TapePlex **status** indicates the status of the TapePlex.
  - **disabled** indicates that the TapePlex has been disabled by an operator command.
  - **active** indicates that the last communication to this TapePlex was successful.
  - **inactive** indicates that a communication path to this TapePlex is no longer active, although one was previously active.
  - **never active** indicates that a communication path to this TapePlex was never successfully established.
- Requests indicates the total number of requests (configuration, volume lookup, mount, dismount, and swap) that were directed to the specified TapePlex.

If the SERVER keyword was specified, then each server path defined for this TapePlex will also be displayed, along with its status.

**System Action:** None

**User Response:** None

### SMC0135

Mount/dismount message from TAPEPLEX=PPPPPPPP SERVER=SSSSSSSS

**Level:** 4

**Explanation:** The ECHOMNTERR(ON) option is in effect. A mount or dismount was performed on an HSC TapePlex but did not complete successfully. The SMC0135 message indicates the TapePlex name and server name where the mount or dismount was requested. The SMC0136 message merely echoes the HSC server mount or dismount message on the VM Client.

**Note –** Messages SMC0135 and SMC0136 are issued for all mount and dismount errors if the message level is 12 or greater regardless of the ECHOMNTERR setting, and for all HSC mount and dismount messages if the message level is 16 or greater.

**System Action:** None

**User Response:** Correct the problem indicated in the HSC mount or dismount message.

### SMC0136

HSC mount or dismount message

**Level:** 4

**Explanation:** The ECHOMNTERR(ON) option is effect. A mount or dismount was performed on an HSC TapePlex but did not complete successfully. The SMC0135 message indicates the TapePlex name and server name where the mount or dismount was requested. The SMC0136 message merely echoes the HSC server mount or dismount message on the VM Client.

**System Action:** None

**User Response:** Correct the problem indicated in the HSC mount or dismount message.

### SMC0160

Invalid range XXXX1-XXXX2 for keyword ADDRESS of the UNITATTR command

**Level:** 0

**Explanation:** A UNITATTR command has been issued specifying a device range XXXX1-XXXX2, where XXXX1 is larger than XXXX2.

**System Action:** The UNITATTR command does not process this device range.

**User Response:** Reissue the command specifying a valid range.

### SMC0161

Restoring all default settings for the CCC...CCC command

**Level:** 0

**Explanation:** The CCC...CCC command has been issued with the OFF parameter. All CCC...CCC values have been restored for the VM Client system.

**System Action:** None

**User Response:** None



**SMC0162**

CCC...CCC object successfully {added | updated | deleted}

**Level:** 0

**Explanation:** The CCC...CCC command has been successfully processed.

**System Action:** None

**User Response:** None

**SMC0163**

DRIVEMAP settings:

CLIENT=XXXX1{-XXXX2} SERVER=XXXX3{-XXXX4}

**Level:** 0

**Explanation:** A DRIVEMAP command has been issued with the LIST keyword. The SMC0163 multiline message lists the currently active DRIVEMAPs. One line is produced for each client/server range.

**System Action:** None

**User Response:** None

**SMC0164**

CLIENT range XXXX1{-XXXX2} not found for the OFF keyword of the DRIVEMAP command

**Level:** 0

**Explanation:** The DRIVEMAP command has been issued with the OFF parameter and CLIENT parameter. No matching DRIVEMAP range matching the CLIENT parameter was found.

**System Action:** None

**User Response:** None

**SMC0165**

Keyword {CLIENT/SERVER} range XXXX1{-XXXX2} overlaps with previous DRIVEMAP entry

**Level:** 0

**Explanation:** A DRIVEMAP command was issued containing a client or server range that overlaps a range specified on a previously issued DRIVEMAP command.

**System Action:** The DRIVEMAP command is not processed.

**User Response:** Issue the DRIVEMAP LIST command to view the list of currently active DRIVEMAP ranges. Correct the DRIVEMAP command to specify a new range. Or, use the DRIVEMAP CLIENT(XXXX1-XXXX2) OFF command to deactivate the existing overlapping range and re-specify the command with unique ranges.

### SMC0166

Excessive READ depth at line *nn* of DSN *DDDDDDDD*

**Level:** 0

**Explanation:** A READ command was issued from a file. However, too many command files are already open, and the read command depth has been exceeded. Read command depth is defined as the number of files that can be open simultaneously due to imbedded Read commands.

**System Action:** The READ command is not processed.

**User Response:** Restructure your command files to reduce the READ command depth and ensure that the files referenced do not contain a recursive loop.

### SMC0167

CCCCCCC summary:  
TAPEPLEX *PPPPPPPP* is {disabled | inactive | active on  
server *SSSSSSSS*}  
{All TAPEPLEX(s) active |  
*n* of *n* TAPE TAPEPLEX(s) active |  
WARNING: All TAPEPLEX(s) inactive |  
WARNING: No TAPEPLEX(s) defined |  
WARNING: No TAPEPLEX(s) enabled}

**Level:** 0

**Explanation:** The CCCCCCCC command was issued and a TapePlex resynchronization was performed. Each TapePlex is represented by a line in the multiline WTO displaying its status.

**System Action:** None

**User Response:** None

### SMC0172

Specified TAPEPLEX *PPPPPPPP* not  
{defined | HSC | active | enabled | valid for UI}

**Level:** 0

**Explanation:** An VM Client command was issued specifying TAPEPLEX *PPPPPPPP*. However, the command cannot be completed because the TAPEPLEX is either not defined to the VM Client, or is not eligible.

**System Action:** The command is not processed.

**User Response:** Either name a valid TAPEPLEX, or correct the TAPEPLEX status and reissue the command.

**SMC0173**

Response from {TAPEPLEX|STORMNGR} *PPPPPPPP*:  
*CCCC....CCCC*  
Response RC=*nn*

**Level:** 0

**Explanation:** A VM Client Route command was issued that specified TAPEPLEX or STORMNGR *PPPPPPPP*. The SMC0173 message lists the TAPEPLEX or STORMNGR name, followed by the response, terminated by an SMC0173 message displaying the command return code.

**System Action:** None

**User Response:** None

**SMC0175**

Communication initialized on TAPEPLEX=*name* SERVER=*name*

**Level:** 0

**Explanation:** The VM Client has successfully communicated with the specified TapePlex for the first time.

**System Action:** Processing continues.

**User Response:** None

**SMC0176**

No active TAPEPLEX(s) for DISPLAY command

**Level:** 0

**Explanation:** A VM Client DISPLAY command has been entered. However, the VM Client cannot establish communication with any TapePlex.

**System Action:** If the DISPLAY (or QUERY) VOLUME was entered, the command terminates as there are no TapePlexes to direct the request. If the DISPLAY (or QUERY) DRIVE command was entered, the command continues although the drive information may not reflect TapePlex ownership.

**User Response:** None

### SMC0177

VM Client {DISPLAY|QUERY} VOLUME

Volser	TapePlex	Location	Media	Rectech	Scr	Volume	Data
VVVVVV	PPPPPPPP	{AA:LL}	MMMMMMMM	RRRRRRRR	SSS	DDDDDDDD	

**Level:** 0

**Explanation:** A VM Client DISPLAY (or QUERY) VOLUME command was entered. The SMC0177 message(s) list the volsers that match the request. The displayed Rectech for a volume reflects a combination of the volume's media type, server VOLATTR settings (if any) and volume data such as density. For example, a volume with a displayed Rectech of STK1RC may have a server VOLATTR that specifies a RECTECH of STK1RC or may be known to have been mounted as a scratch on a 9840C drive. The Scratch status will be displayed as "Yes" or "No". The Volume Data for a volume reflects known density and encryption characteristics of the volume as stored in the HSC CDS. Volume Data DEN=1 through DEN=3 means low, high, and highest density respectively.

**System Action:** None

**User Response:** None

### SMC0178

VM Client {DISPLAY|QUERY} DRIVE

Addr	TapePlex	Location	Model	Serv	VM Client	Status
AAAA	PPPPPPPP	{AA:LL:PP:DD}	MMMMMMMM	SSSS	CCCCCCCC	

**Level:** 0

**Explanation:** A VM Client DISPLAY (or QUERY) DRIVE command was entered. The SMC0178 message(s) list the drives that match the request. The address AAAA reflects the drive address as it is known to CP; If there are DRIVEMAP entries for the drive, the address SSSS reflects the drive address as it is known to the server.

**System Action:** None

**User Response:** None

### SMC0189

CCCCCCCC entry EEEEEEE not found for {list|update|delete}

**Level:** 0

**Explanation:** A CCCCCCCC command was entered specifying that entry EEEEEEE be either listed, deleted, or updated. However, no entry matching EEEEEEE was found.

**System Action:** None

**User Response:** Issue the CCCCCCCC command with the LIST option to list all CCCCCCCC entries. Then re-issue the command specifying the correct entry name.

**SMC0190**

CCCCCCCC OOOOOOOO set to {ON|OFF|XXXXXXXX}

**Level:** 0

**Explanation:** A CCCCCCCC command was entered specifying that option OOOOOOOO be set to ON, OFF, or the specified value XXXXXXXX. If multiple options were specified on a single CCCCCCCC command, then multiple SMC0190 messages are issued, one for each specified option.

**System Action:** None

**User Response:** None

**Note –** SMC0190 messages are displayed only if MSGDef VERBose(ON) is specified.

**SMC0191**

CCCCCCCC OOOOOOOO set to {ON|OFF|XXXXXXXX} for entry EEEEEEEE

**Level:** 0

**Explanation:** A CCCCCCCC command was entered specifying that option OOOOOOOO be set to ON, OFF, or the specified value XXXXXXXX for the CCCCCCCC entry EEEEEEEE. If multiple options were specified on a single CCCCCCCC command, then multiple SMC0191 messages will be issued, one for each specified option.

**System Action:** None

**User Response:** None

**Note –** SMC0191 messages are displayed only if MSGDef VERBose(ON) is specified.

**SMC0195**

READ processing started for {SMCPARMS|SMCCMDS|*data set name*}

**Level:** 0

**Explanation:** The VM Client has begun processing commands in the named file.

**System Action:** None

**User Response:** None

**Note –** SMC0195 messages are displayed only if VM Client MSGDef VERBose(ON) specified.

### SMC0196

READ processing complete; RC=*nn*  
from {SMCPARMS|SMCCMDS|*data set name*}

**Level:** 0

**Explanation:** The VM Client has completed processing commands in the named file. The highest return code for any command is *nn*.

**System Action:** None

**User Response:** None

**Note** – SMC0196 messages are displayed only if VM Client MSGDef VERBose(ON) specified.

### SMC0203

COMMTTEST:

USER=UUUUUUUUU TASK=XXXXXXXXXXXXXXXXX {MSG=XXXXXXXXX}  
TAPEPLEX=LLLLLLLL SERVER=SSSSSSSS REQUEST=FFFF  
Client {IP=NNN.NNN.NNN.NNN} socket=NN port={*nnnn*|ANY}  
Server IP=NNNN.NNNN.NNNN.NNNN port=*nnnn*  
Bytes out=*nnnn* in=*nnnn*  
Error=EEEE....EEEE  
Reason=RRRR....RRRR  
{Response from STK HTTP server follows: HHHH...HHHH}  
Current LIBPATH status=  
{active|inactive|never active|disabled}  
VM Client comm RC=*nnnn* elapsed time=*nn.nn*

**Level:** 0

**Explanation:** A COMMtest command was entered. The SMC0203 message is displayed for each communication path attempted.

**System Action:** None

**User Response:** None

### SMC0204

No eligible COMMPATH(s) found

**Level:** 0

**Explanation:** A COMMtest command was entered, but the specified TAPEPlex, SERVER, and status parameters resulted in no eligible communication paths selected for the test.

**System Action:** None

**User Response:** Correct and reissue the COMMtest command.

**SMC0205**

Disabling bind to PORTRANGE *nnnn-nnnn*; any ephemeral port will be used

**Level:** 0

**Explanation:** A TCPip PORTrange (OFF) command was entered. Sockets will no longer be bound to the fixed port range of *nnnn-nnnn*, but any ephemeral port will be used.

**System Action:** None

**User Response:** None

**SMC0206**

No PORTRANGE currently defined

**Level:** 0

**Explanation:** A TCPip PORTrange (OFF) command was entered but there is currently no active PORTrange specified to disable.

**System Action:** None

**User Response:** None

**SMC0207**

Specified SERVER *SSSSSSSS* not {found | defined for TAPEPLEX=*TTTTTTTT*}

**Level:** 0

**Explanation:** A COMMtest command was entered specifying a specific TapePlex and server. However, the server is either not defined to the VM Client, or is not defined for the specified TapePlex.

**System Action:** None

**User Response:** Correct and reissue the COMMtest command.

**SMC0226**

Path switch from server=*SSSSSSSS* to *PPPPPPPP* for TAPEPLEX=*TTTTTTTT*

**Level:** 4

**Explanation:** The VM Client automatically switched the communication path from the secondary server *SSSSSSSS* to the primary server *PPPPPPPP* for TAPEPLEX *TTTTTTTT*.

**System Action:** Processing continues.

**User Response:** None

**SMC0227**

Keyword *KKKKKKKK* of the *CCCCCCCC* command ignored; *RRRRRRRR*

**Level:** 0

**Explanation:** The *CCCCCCCC* command specified a keyword that is no longer acceptable. Keyword *KKKKKKKK* may be obsolete in the current version of the product, or it may be unacceptable in the current processing environment.

**System Action:** Keyword *KKKKKKKK* and any associated value are discarded, but the remainder of the command is still processed.

**User Response:** If the keyword is obsolete in the current release, delete the keyword from the command as it may be flagged in error in subsequent releases, invalidating the entire command.

**SMC0228**

Copyright *nnnn*, *nnnn*, Oracle and/or its affiliates.  
All rights reserved.

**Level:** 0

**Explanation:** The VM Client system is initializing.

**System Action:** Processing continues.

**User Response:** None

**SMC0232**

Warning: No TAPEPLEX command processed

**Level:** 0

**Explanation:** The VM Client system has completed initialization, but no TAPEPLEX commands were found in either the SMCPARMS or SMCCMDS data set.

**System Action:** Processing continues

**User Response:** Enter TAPEPLEX and SERVER commands.



**SMC0236**

CCC...CCC command RC=XX exceeds MAXRC=NN at startup

**Level:** 0

**Explanation:** The VM Client was started with the MAXRC startup parameter and during VM Client initialization the CCC...CCC command returned a completion code that exceeded the MAXRC specification.

**System Action:** Processing continues for the remainder of the commands specified in the SMCPARMS or SMCCMDS data set. However, VM Client system initialization will be terminated with the SMC0237 message at the completion of the SMCPARMS or SMCCMDS processing.

**User Response:** Correct the specified CCC...CCC command and restart the VM Client.

**Note –** Multiple SMC0236 messages may be produced at startup as all VM Client commands in the SMCPARMS and SMCCMDS data sets are processed at startup, regardless of prior SMC0236 messages.

**SMC0237**

VM Client terminating due to MAXRC=*nn* exceeded at startup

**Level:** 0

**Explanation:** The VM Client was started with the MAXRC startup parameter, and during VM Client initialization an SMC0236 message was issued indicating that an VM Client command in the SMCPARMS or SMCCMDS data set returned a completion code that exceeded the MAXRC specification.

**System Action:** The VM Client terminates.

**User Response:** Review the VM Client log for the SMC0236 message(s) indicating the commands in error, correct the indicated commands, and restart the VM Client.

**SMC0242**

Cannot add STORMNGR CCC...CCC before TAPEPLEX(es)

**Level:** 0

**Explanation:** STORMNGR commands must be entered after TAPEPLEX commands.

**System Action:** Processing continues.

**User Response:** Enter TAPEPLEX commands before STORMNGR commands.

### SMC0243

CCCCCCCC command specifies {TAPEPLEX|STORMNGR} NNNNNNNN; but NNNNNNNN is a {STORMNGR|TAPEPLEX} [at line *nnnn* of {SMCCMDS|SMCPARMS}]

**Level:** 0

**Explanation:** The CCCCCCCC command was entered and specified the named TAPEPLEX or StorageTek Storage Manager. However, NNNNNNNN is not the type of entity described.

**System Action:** The command is not processed.

**User Response:** Change the entity type from TAPEPLEX to STORMNGR or vice versa, and reissue the command.

### SMC0244

METADATA command not supported for {non-UUI origin|non-XML responses|command CCCC}

**Level:** 0

**Explanation:** A METADATA command was processed, but it is invalid for one of the following reasons:

- non-UUI origin  
indicates that the command was received from an operators console, or from the SMCPARMS or SMCCMDS data sets. The METADATA command is only allowed from the UUI interface; either from the SMCUUI or SMCUSIM utilities, or from the UUI programmatic interface.
- non-XML responses  
indicates that the METADATA command originated from the UUI interface, but that XML responses were not requested. METADATA is only valid as a XML response.
- command CCCC  
indicates that the specified command CCCC does not produce XML output, so metadata is not available.

**System Action:** None

**User Response:** Correct the METADATA command.

### SMC0245

Code *nnnn* (X'*xxxx*): ssssssssss

**Level:** 0

**Explanation:** A display RC command was processed that specified reason code *nnnn*, or hex reason code *xxxx*. The corresponding reason is displayed. If the DETAIL option was specified, the reason code explanation is also displayed.

**System Action:** None

**User Response:** None

### SMC0300

Message | Command *nnnnn* Help Text:

**Level:** 0

**Explanation:** A Help command has been issued. The help text for the Message or Command is listed.

**System Action:** None

**User Response:** None

### SMC0301

HELP for XXXXXX not found

**Level:** 0

**Explanation:** A Help command has been issued. The subject XXXXXX is not found.

**System Action:** None

**User Response:** Re-enter the Help command with a valid subject.

### SMC0302

XXXXXX is an invalid range

**Level:** 0

**Explanation:** A Help command has been issued. The subject XXXXXX is an invalid range.

**System Action:** None

**User Response:** Re-enter the Help command with a valid subject.

### SMC0805

VM Client failed setting ANCHOR: rc=*nnn*

**Level:** 0

**Explanation:** A non-zero return code was received executing the CMS ANCHOR SET macro.

**System Action:** VM Client terminates.

**User Response:** Contact StorageTek Software Support.

### SMC0806

TCP/IP server available: id=YYYYYY

**Level:** 0

**Explanation:** VM Client successfully connected with TCP/IP id, YYYYYY.

**System Action:** None

**User Response:** None

**SMC0807**

TCP/IP server is unavailable; id=YYYYYY errno=NNN errmsg

**Level:** 0

**Explanation:** VM Client TCP/IP functions returned an error trying to establish a socket connection with YYYYYY.

**System Action:** None

**User Response:** Correct the TCP/IP id specified in the SMCPARMS file and re-start VM Client. If the TCP/IP id is correct, contact StorageTek Software Support.

**SMC0810**

Dynamic Allocation NOT supported

**Level:** 0

**Explanation:** The READ command attempted to dynamically allocate a file to read.

**System Action:** The command is not processed.

**User Response:** Re-enter the READ command, specifying the SMCCMDS file from the startup FILEDEF.

**SMC0811**

cp command

**Level:** 0

**Explanation:** The output from the CP command is displayed.

**System Action:** None

**User Response:** None

**SMC0812**

LOG command requires CONSOLE or DISK

**Level:** 0

**Explanation:** The LOG command requires that CONSOLE or DISK must be entered.

**System Action:** Command is not processed.

**User Response:** Reenter the LOG command with the proper options.

**SMC0813**

*mmmm+nnnn* - *aaaaa* - PSW data *pppp*  
 Event *eeee* - Data *dddd* - Thread *nnnn*  
 Data at PSW addr - *xxxx<>xxxx*

**Level:** 0

**Explanation:** An abend has been detected.

- *mmmm+nnnn* indicates the abending module and displacement
- *aaaa* indicates the abend code
- *pppp* indicates the PSW at abend
- *eeee* indicates the address of the VMERROR data return by the error event
- *dddd* indicates the address of the Data Area returned by the error event
- *nnnn* indicates the abending thread id
- *xxxx<>xxxx* indicates the data at the PSW address
- *<>* marks the PSW address

The registers at abend follow.

**System Action:** The command or TMI request is terminated.

**User Response:** Contact StorageTek Software Support.

**SMC0814**

Dump sent to *nnnnn* *ddd* dumps remain

**Level:** 0

**Explanation:** The system produced a storage dump in response to a DUMP command or because of an abending program.

**System Action:** The dump file is transferred to the userid *nnnnn*.

**User Response:** If the number of dumps allowed (*ddd*) approaches 0, issue the DUMPOPTS RESET command to reset the number of dumps allowed and/or change the maximum dump limit. If this message is not in response to a DUMP command, contact StorageTek Software Support.

**SMC0815**

Dump not taken due to dump Max Count - *nnnn*

**Level:** 0

**Explanation:** The maximum number of dumps allowed by the DUMPOPTS command was produced.

**System Action:** This dump request is ignored. Until a DUMPOPTS RESET command is received, no dump requests are honored.

**User Response:** Issue the DUMPOPTS RESET command to reset the number of dumps taken and/or change the maximum dump limit.

**SMC0816**

DUMPOPTS settings:  
DUMPS TAKEN=*tttt*  
DUMPS MAX COUNT= *mmmmm*  
TO=*uuuuuuuuuu*

**Level:** 0

**Explanation:** A DUMPOpts command has been issued with the LIST keyword. The DUMPOPTS settings are listed.

**System Action:** None

**User Response:** None

**SMC0817**

LOG settings:  
CONSOLE=ON|OFF  
CLASS=*c*  
TO=*uuuuuuuuuu*  
DISK=ON|OFF  
FM=*a*

**Level:** 0

**Explanation:** A LOG command has been issued with the LIST keyword. The LOG settings are listed.

**System Action:** None

**User Response:** None

**SMC0818**

OPERATOR settings:  
ID=*uuuuuuuuuu*

**Level:** 0

**Explanation:** A OPERator command has been issued with the LIST keyword. The OPERATOR settings are listed.

**System Action:** None

**User Response:** None

**SMC0819**

Disk FM *m* is READONLY

**Level:** 0

**Explanation:** The disk specified in the FM parameter is READONLY. The disk must be writable.

**System Action:** The command is not processed.

**User Response:** Re-enter the command with proper parameters.

### SMC0820

Disk FM *m* is not defined

**Level:** 0

**Explanation:** The disk specified in the FM parameter is not defined.

**System Action:** The command is not processed.

**User Response:** Re-enter the command with proper parameters.

### SMC0821

*UUUUUUU* is not a defined VM userid

**Level:** 0

**Explanation:** The value entered is not a defined VM userid.

**System Action:** The command is not processed.

**User Response:** Re-enter the command with proper parameters.

### SMC0822

XXXXXXXX has initiated VM Client termination

**Level:** 0

**Explanation:** An EXIT command has been received from the XXXXXXXX source.

**System Action:** VM Client is starting termination processing.

**User Response:** None

### SMC0823

*UUUUUUUUU* is not authorized for VM Client ZZZ requests

**Level:** 0

**Explanation:** A VM Client ZZZ request received from VM userid *UUUUUUUUU* has been denied. The userid is not authorized to execute ZZZ requests.

**System Action:** VM Client ignores the request and continues processing.

**User Response:** Use the AUTHORIZE command to allow VM userids to execute VM Client TMI and command requests. Another option is to use the VM Client verification customer exit, SMCXIT01.

### SMC0824

MOUNT | DISMOUNT command failed; RC=NNNN - Reason=MMMM

**Level:** 0

**Explanation:** The MOUNT or DISMOUNT command failed with a Return Code of NNNN. The Reason Code (MMMM) may be the HSC message number that describes the failure reason.

**System Action:** The command failed.

**User Response:** Correct the problem indicated in the HSC mount or dismount message.

### SMC0825

FORCE parameter invalid for virtual drive *DDDD*

**Level:** 0

**Explanation:** The DISMOUNT FORCE parameter is not supported for virtual devices.

**System Action:** The dismount is not processed.

**User Response:** Re-issue the DISMOUNT command without the FORCE parameter. If necessary, issue the CMS TAPE RUN or CP DETACH command to unload the drive.

### SMC0826

Authorized users:  
*UUUUUUUUU* Requests: *req1 req2 ...*

**Level:** 0

**Explanation:** An AUTHorize command has been issued with the LIST keyword. The authorized users and the types of requests they are authorized for are displayed.

**System Action:** None

**User Response:** None

### SMC0827

POOLMAP SCRATCH	MANAGEMENT SUBPOOL
SUBPOOL	CLASS NUMBER
<i>PPPPPPPPPPPP</i>	<i>MMMMMMMMM NNN</i>

**Level:** 0

**Explanation:** A POOLmap command has been issued with the LIST keyword. The scratch subpool names, *PPPPPPPPPPPP*, are displayed with their corresponding management class names, *MMMMMMMMM*, and number values, *NNN*. If "n/a" is displayed, the value has not been defined.

**System Action:** None

**User Response:** None

### SMC0828

POOLMAP validation failed; *reason*

**Level:** 0

**Explanation:** The POOLmap command received an error validating the management class and subpool name with the HSC server.

The possible reasons are:

- invalid management class
- invalid subpool name
- HSC server unavailable

**System Action:** None

**User Response:** Correct the invalid parameter and re-issue the command.



#### SMC0829

Mount of volume *VVVVVV* complete on drive *DDDD*

**Level:** 0

**Explanation:** Tape volume, *VVVVVV*, has been successfully mounted on tape drive *DDDD*.

**System Action:** None

**User Response:** None

#### SMC0831

VM Client id *SSSSSSS* must be 1-8 characters

**Explanation:** The VM Client id specified in the command must be 1-8 characters.

**System Action:** The command is not processed.

**User Response:** Re-enter the command with proper parameters.

#### SMC0832

VM Client id *SSSSSSS* not active

**Explanation:** The VM Client id specified in the command is not active

**System Action:** The command is not processed.

**User Response:** Re-enter the command with proper parameters.

#### SMC0833

TMS id is missing

**Explanation:** The TMS id must be specified in the command. The valid TMS ids are:

- VMTAPE
- TAPEMANAGER

**System Action:** The command is not processed.

**User Response:** Re-enter the command with proper parameters.

#### SMC0834

TMS *SSSSSSSS* is invalid

**Explanation:** The TMS id specified in the command is invalid. The valid TMS ids are:

- VMTAPE
- TAPEMANAGER

**System Action:** The command is not processed.

**User Response:** Re-enter the command with proper parameters.

### SMC0835

SSSSSSSS module is not available

**Explanation:** Module SSSSSSSS is not available. This module is required for the operation of the command or utility.

**System Action:** The command is not processed.

**User Response:** Investigate the reason that the module is not available.

### SMC0836

TAPEMANAGER TAPCMD module not found

**Explanation:** An SMCUDBX command was entered with the *tsname* of TAPEMANAGER. The TAPCMD module to process TAPEMANAGER is not available.

**System Action:** The command is not processed.

**User Response:** Re-enter the command after the TAPEMANAGER TAPCMD module is available.

### SMC0837

File name | type | mode FFFFF is invalid

**Explanation:** The file name, type, or mode FFFFF is invalid.

**System Action:** The command is not processed.

**User Response:** Re-enter the command with the correct file name, type, or mode.

### SMC9999

MMMMMMMMM Variable text

**Level:** 12, 16, 20, 24, 28

**Explanation:** SMC9999 messages are intended for StorageTek Software Support problem determination and resolution. MMMMMMMMM is the name of the issuing module.

**System Action:** None

**User Response:** None. A message level (LVL) of 12 or higher should generally be specified only when directed by StorageTek Software Support.

---

## VM Client Tape Management Interface

### Overview

This chapter describes the VM Client Tape Management Interface (VMTMI).

The complex of resources managed by StorageTek software, including library resources for real cartridges and VSM resources for virtual volumes, is referred to as a **TapePlex**.

A TapePlex is defined as the hardware managed by a single HSC CDS. VM Client 7.1 provides access to a single TapePlex. VM Client uses TCP/IP to route transactions to an HSC server executing on z/OS. Multiple instances of HSC on multiple hosts can be defined as servers to provide redundancy. Transactions between VM Client and the HSC server use an XML-based API called the XAPI. The VM Client software converts Tape Management Interface transactions into XAPI format for interpretation by the server, and converts the output of these transactions into TMI format for the response.

Because the Tape Management Interface now interacts with a client component, which in turn interfaces with HSC (and VTCS) on z/OS, some TMI commands that were supported in previous releases are no longer supported. In some cases, command functionality cannot be supported in a client/server environment, while other commands may be supported in future releases.

The following TMI commands are **not** supported for VM Client 7.1:

- DEFSCR and DEFPOOL

These commands are superseded by the HSC POOLPARM/VOLPARM feature introduced in ELS 7.0. Refer to the publication *Configuring HSC and VTCS* for more information about this feature.

- QEJECT
- QREQUEST
- SETOPER

VM Client does not support operator responses; therefore, all commands are processed as though the SETOPER command was issued. Conditions that previously would have generated WTOR messages are returned as error messages.

- STOP

In addition, the VM Client does not support the use of TAPEREQ lookup keys, including job name, step name, program name, and data set name, to select media and recording technique values.

A general description of the interfaces between the tape management system (TMS) and VM Client includes the following topics:

- [TMS Responsibilities](#)

This section describes the services a TMS provides.

- [TMS Decision Points](#)

This section describes where TapePlex interaction assists TMS services for TapePlex-managed resources.

- [TapePlex Information Returned to the TMS](#)

This section describes information returned as a result of TapePlex interaction.

- [Inter-user Communications Vehicle \(IUCV\) Considerations](#)

This section describes the parameters used with the IUCV macro.

- [TMS and VM Client Interaction](#)

This section describes various scenarios involving TMS to VM Client interaction.

# TMS Responsibilities

The Tape Management System has three major functions:

- User interface
- Tape resource allocation
- Operator interface

## User Interface

Normally, an end user requests TapePlex facilities indirectly by requesting services from a Tape Management System (TMS). Such requests are routed by the TMS to the VM Client, and then to the TapePlex server. Direct interaction between an end user and the VM Client only occurs if the end user issues VM Client commands directly using the VM Special Message (SMSG) facility, or invokes the VMTMI directly. See the distributed VMTMI SAMPLE for an example.

## Tape Resource Allocation

The resources under TMS control include:

- Transports

The TMS normally has ownership of transports for Automatic Volume Recognition (AVR) and also assigns transports to users requesting tape services. The TMS determines the availability of tape transports for allocation requests. The TMS also knows the media type and density any transport supports. VM Client assists the TMS in selecting TapePlex-controlled transports, when necessary.

- Data sets

The TMS maps data sets to tape volumes and may map external labels to internal labels. The TapePlex contains no such information.

- Scratch volumes

The TMS is the final authority concerning the scratch status of volumes. This status also includes scratch subpool membership. In order to automate mount processing for a TMS generating "nonspecific" mounts (requests for scratch volumes that do not specify VOLSERS), the TapePlex also retains its own scratch status information. The TMS scratch status list is not considered a list of all available scratches, but rather as a subset of the total number of available scratch volumes.

- Specific volumes

The TMS controls which users have access to any specific volume. The TapePlex handles volumes it controls at the request of an authorized operator or the TMS.

## Operator Interface

Tape mounting, dismounting, and scratch pool selection is handled through message traffic between the TMS and the operator. The VM Client uses information supplied in messages to the operator to direct mounts, dismounts, etc.

## TMS Decision Points

The TapePlex server can influence TMS decisions when TapePlex services are available to the TMS via the VM Client. The TapePlex server influences the TMS decisions at the following points:

### TMS Initialization

When the TMS is initialized, have the TMS establish an IUCV path to the VM Client machine to determine if both the VM Client and the TapePlex server are operational and communicating. If the VM Client machine is not operational at TMS startup, establish an IUCV path as soon as possible after VM Client and TapePlex initialization. It is possible to establish and break connection for each transaction, but this causes unnecessary processing. To use IUCV efficiently, a path must be established and maintained throughout the TMS communications session.

### Drive Allocation

At allocation time, the VM Client can provide information about media and location of specific volumes and scratch counts. The TMS can use the result of queries to select compatible optimum drives for an allocation request.

### Scratch Allocation

If the TMS requests, the VM Client can provide the VOLSER of a scratch volume prior to a MOUNT request. This selection can be rejected or used on the subsequent mount. If this information is not needed, a nonspecific MOUNT requests causes a scratch volume to be selected.

### Volume Movement

When a mount, dismount, or other movement of a volume is required, the TMS decides whether the TapePlex performs the action or if a manual operation is required. The VM Client returns status information for volume movement requests.

### Returning a Volume to Scratch Status

It is necessary to keep the TMS and TapePlex scratch status synchronized. The earliest time is at dismount, the latest when a TMS scratch pull list is generated. TMI requests and VM Client commands are available to coordinate this activity.

# TapePlex Information Returned to the TMS

TapePlex information returned includes:

## Configuration Information:

- Maximum number of transports in the largest ACS
- Number of transports under library control
- Number of ACSs
- Number of LSMs
- Response area sizes

## Volume Status:

- Volume in a cell
- Volume in a drive
- Volume not in library
- Volume inaccessible
- Volume location uncertain (errant)

## Volume Location:

- ACSid
- LSMid
- Panel location
- Row location
- Column location

## Eligible Drives

Drives eligible from a TapePlex perspective are those in the same ACS or VTSS as the volume, and with a recording technique compatible with the volume. Not taken into account is the actual drive availability (attachability, online status).

## Movement Status and Error Codes

Volume movement requests return a code indicating the success of the operation. If an error occurs, an additional code corresponding to the message issued to the operator is also returned.

## LSM and ACS Status

LSM status is either online or offline. Online indicates that automated mounting can take place. Offline indicates only manual mounting is possible.

ACS status is connected or disconnected. Disconnected indicates that the ACS is not accessible from this host and any activity must be handled from another host.

- **VOLSERs for Scratch Management**

If scratch selection is requested, a VOLSER marked as scratch in the TapePlex control data set is returned. This selection causes the volume to be marked as nonscratch.

- **Library Notation for Virtual Drives and Volumes**

Using VM Client allows the TMS to access virtual drives and virtual volumes. The Tape Management Interface maps each VTSS to an ACS ID and a set of LSM IDs so that the TMS can handle virtual drives and volumes like real volumes. VM Client provides the VTSS to ACS mapping and ensures that ACS IDs that represent VTSSs do not overlap with real ACSs. Volume movement operations, such as EJECT, MOVE, and ENTER, are not allowed for virtual volumes.



# Inter-user Communications Vehicle (IUCV) Considerations

The inter-user communication vehicle (IUCV) is an IBM-supplied communications interface.

**Note** – VMTMI SAMPLE is a sample program illustrating the use of the VM Client Tape Management Interface. It can be found on the MAINTSTK userid.

To use IUCV to issue requests, follow these steps:

1. Establish a connection to the VM Client service machine using the IUCV CONNECT function.
  - a. Only authorized virtual machines may issue commands to the VM Client service machine. To obtain permission, the virtual machine issuing the TMI or VM Client command must be given privileges by a VM Client AUTHORIZE command issued to the VM Client service machine either in the SMCPARMS or SMCCMDS file at startup, or from a previously authorized virtual machine.
  - b. In addition, the virtual machine must be authorized to CP via an IUCV control statement in its CP directory entry. This is typically done by a systems programmer or administrator. Make sure that the OPTION MAXCONN specifies enough paths for your usage.
  - c. To establish this IUCV connection (path), the IUCV macro is issued with the following parameters:

```
IUCV CONNECT,
      PRMLIST=addr,      * address of IUCV parm list
      USERID=addr,      * address of CL8 'userid'
      USERDATA=addr,    * address of CL16 'ddname'
      PRMDATA=NO         * no parm data in IPARML
```

where:

*userid* indicates the name of the VM Client service machine.

*ddname* indicates the name of the VM Client IUCV interface that is requested for the connection. It is a 16 byte area as follows:

```
DC CL8'SLSTLMS'      ddname
DC CL8' '            reserved
```

On execution of the function, check the PSW condition code. If the condition code is 0, save the path ID from the IPARML area passed to the macro. The program must wait for the VM Client service machine to IUCV ACCEPT the pending connection before sending any messages. If no "connection complete" or "path severed" is returned, either the VM Client is not active, is not fully initialized, or the IUCV CONNECT requestor specified invalid parameters.

2. Send the message to the VM Client service machine using the IUCV SEND function. Specify the following parameters to the IUCV macro:

```
IUCV SEND,  
    PATHID=adpid,  
    TYPE=2WAY,  
    BUFLen=buflen,  
    RBUF=reply,  
    RLEN=reply length
```

where:

***adpid***

is the address of a data area containing the IUCV path ID.

**TYPE=2WAY**

specifies that an IUCV reply is expected.

***buffer***

is the address of a buffer containing any valid TMI request.

***buflen***

is the length of "buffer".

***reply***

is the address of the buffer containing the reply.

***reply length***

is the length of the reply buffer.

3. When you are finished using a connection to the VM Client service machine, release the path using the IUCV SEVER function.

The following IUCV parameters are not supported for this interface:

**TYPE=1WAY**

IUCV REPLY must be issued by the VM Client.

**TRGCLS= TRGCLS**

is ignored by the VM Client.

**DATA=PRMMSG CP**

does not allow the SEND to occur.

**PRMMSG=address**

CP does not allow the SEND to occur.

**PRTY=YES**

CP uses this to alter queuing to the VM Client service machine. The VM Client service machine does not give the message any special handling.

All other IUCV parameters may be used as desired.

## Additional Considerations

Three fields in the IUCV parameter list (IPARML) deserve special mention:

- USERID=
- USERDTA=
- UWORD=

**Note** – Refer to the IBM publication *VM/SP System Facilities for Programming* or *VM/XA CP Programming Services* for additional information about the IUCV interface and the parameters listed below.

### USERID

This parameter specifies the name of the service machine running the VM Client. Oracle recommends that your userid be alterable, rather than hard-coded, to facilitate changes.

### USERDTA

This parameter is used to specify the name of the process in the service machine receiving TMS communication (ACSINT). This must be SLSTLMS.

### UWORD

This parameter specifies a word which will appear in R0 at interrupt time. It is useful for establishing addressability to a common data area. This contains an area listing pending requests, path status, and configuration values that are referenced in different routines.

## TMS and VM Client Interaction

VM Client requires several points of interface with a tape management system (TMS):

- TMS to VM Client initial connection
- Drive Allocation
- Operator message processing

The TMS provides a front end between VM Client and the user, maintaining allocation, data set, and scratch pool services. VM Client provides the TMS with mount/dismount handling and scratch volume selection, and influences the allocation of those volumes and drives under TapePlex control.

An invocation macro (ACSRQ) prepares a call to the Inter-User Communications Vehicle (IUCV) to communicate between the TMS and VM Client. The IUCV call itself is made by the TMS.

The following tape management system facilities support the communication:

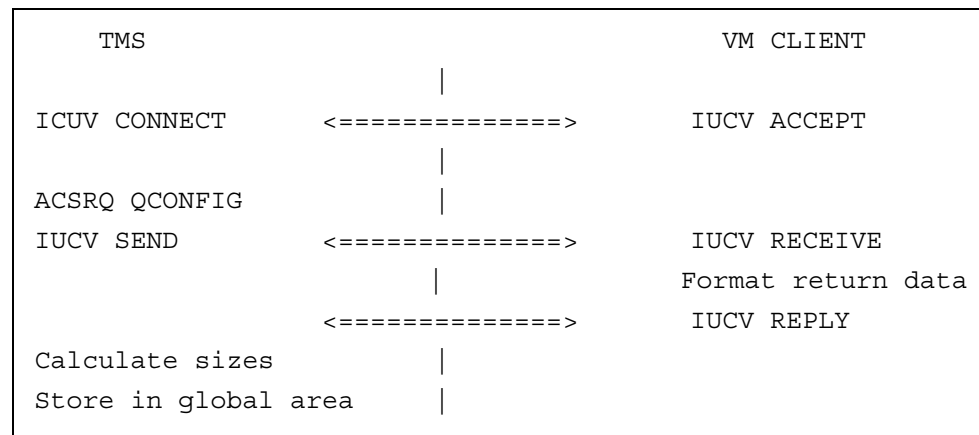
- An interface at allocation time to supply device type and scratch information
- An interface at message time to handle the message normally displayed to the operator
- A list of transport drive addresses and corresponding media and location information
- An IUCV interrupt handler

## TMS to VM Client Initial Connection

At initial connection time it is useful to determine the size of the returned data areas for a few of the longer responses. These vary depending on the TapePlex configuration. These areas are then allocated before their required use.

### Initial Connection Dialog

At connection time, a QCONFIG request should be issued to determine the size of the reply data areas that are needed for other requests.



**FIGURE 7-1** Initial Connection Dialog

## Drive Allocation

While the TMS is fully responsible for drive allocation, VM Client and the TapePlex assist in this process by presenting a list of drives in order of their suitability to satisfy mounts of specific and scratch volumes.

The "Allocation Dialog" section describes the interaction between the TMS allocation interface and the VM Client.

### Allocation Interaction

The TMS service machine receives a request from a virtual machine to mount a volume and invokes the allocation interface routine.

If an IUCV path to the VM Client service machine is not established, the tape management system attempts to establish one. If the attempt fails, no allocation assistance takes place, a return code indicates that condition, and a flag may be set to inform the message interface that operator message processing cannot take place since no special allocation has been done.

## Allocation Dialog

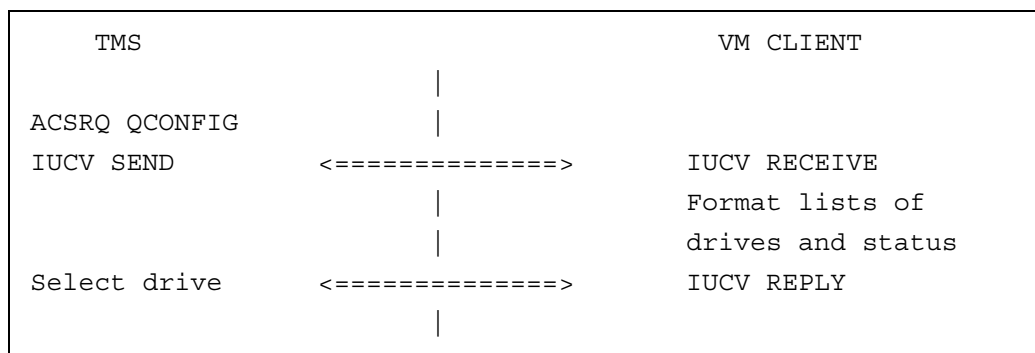
1. The TMS allocation interface sends a request, using an IUCV message, to the VM Client service machine:

```
ACSRQ QDRLIST,VOLSER=voladr
```

An alternate request is:

```
ACSRQ QDRLIST,VOLSER=voladr,COUNT=,LIST=
```

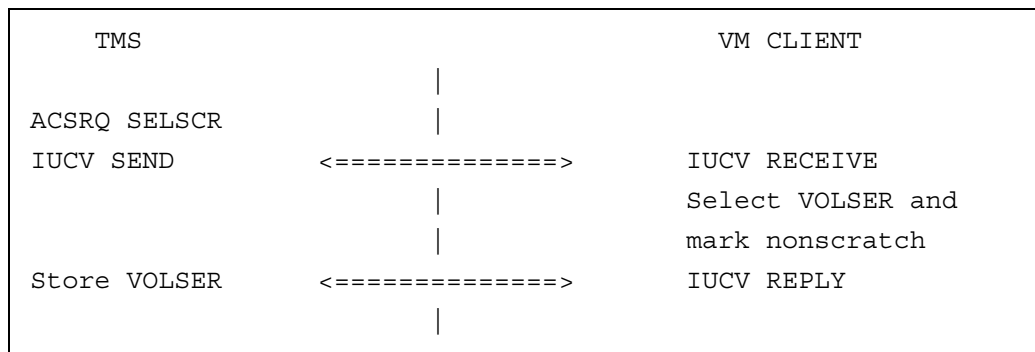
The request above includes the COUNT= and LIST= parameters. These two parameters describe a list of devices considered eligible by the TMS. This information is passed to the VM Client.



2. The allocation interface waits for an IUCV REPLY.
3. A VM Client routine collects the data, formats the response, and issues an IUCV REPLY. The IUCV REPLY is mapped by the SLX macro.
4. The TMS IUCV support functions notify the waiting allocation process that a response has been received.
5. The allocation interface routine then reformats the reply into a drive preference list (TMS dependent format), comparing the reply to the TMS-managed available drives, and leaves the interface.
6. If scratch selection is needed, issue:

```
ACSRQ SELSCR,DRIVE=drivadr
```

This returns a VOLSER and marks the volume as nonscratch in the control data set, or indicates that no scratch volume is available.



## Termination of Allocation Interface

Control is returned to the TMS when the allocation interface routine completes processing.

## Operator Message Processing

This section describes the interactions between the TMS message interface and the VM Client. The “[Operator Message Dialog](#)” section outlines the processing of a MOUNT request. Other message interface point requests (e.g. DISMOUNT, SCRATCH), follow a similar sequence of events and are not separately described. The TMS may suppress or change the message based on the completion of the process.

A DISMOUNT request is issued in response to conditions detected by the TMS that require a volume dismount (e.g. the mounted volume is not the one requested), or in the case where the TMS always dismounts volumes after use.

A SCRATCH request is issued by the TMS to return "work" volumes to scratch status. The HSC control data set on the server is updated to reflect these changes.

## Operator Message Interaction

The TMS service machine receives a request from a virtual machine to mount a volume. The TMS service machine must have an IUCV path established to send commands to the VM Client. A drive has already been selected.

## Operator Message Dialog

1. The message interface code determines that the request is for a drive.
2. The message interface may send a request, using an IUCV message, to the VM Client service machine to obtain location information.  
`ACSRQ QVOLUME,VOLSER=voladr`
3. The message interface waits for an IUCV REPLY.
4. The VM Client determines the volume status, adds the location data, and issues an IUCV REPLY. Note: If volume status information is retained from the allocation routine, the four previous steps may be omitted.
5. If the volume is in the TapePlex, the operator message interface sends a request, using an IUCV message to the service machine, specifying the volume to be mounted and the drive to be used.

`ACSRQ MOUNT,VOLSER=volser,DRIVE=drivadr,PROTECT=`

6. The message interface waits for an IUCV REPLY.
7. The VM Client MOUNT routine requests the server to perform the mount, formats a success/failure response, and issues a reply. The IUCV REPLY to the originating message is mapped by the SLX macro.

**Note** – A MOUNT request directed to a transport drive containing an unloaded volume causes a dismount of that volume followed by the requested mount.

8. TMS IUCV support routines notify the waiting message interface that a response has been received.
9. The message interface routine examines the reply to determine if the mount was successful, sets an appropriate return code, and leaves the interface.

## Termination of Operator Message Interface

Control returns to the tape management system when the message interface routine completes processing.

## PROP-Detected Dismount

When a StorageTek drive attached to a virtual machine is detached, or the virtual machine is logged off, a DETACH message is issued to the VM system operator. If a drive becomes detached while a library or virtual volume is mounted, the TMS may not be notified, and would not issue a normal DISMOUNT message to the VM Client service machine.

To properly handle the dismount, a VM PROP (PRogrammable OPerator) facility should intercept certain messages and process accordingly.

**Note** – Like the TMS machine, the PROP machine must be authorized by the VM Client service machine for commands. The function called using the PROP RTABLE must have the name of the VM Client service machine available.

SMCPROP EXEC, LOGTAPE SAMPLE, and RTABLE SAMPLE are supplied as examples. The EXECs can be used unchanged or modified to suit the environment. These should be set up to execute similar to the following sequence:

1. PROP detects the DETACH message and invokes a routine (SMCPROP EXEC) to check if a library (or virtual) volume was previously mounted on the drive (saved using SMCPROP EXEC). If so, execute the following command:  

```
CP SMSG vmclientuser DISMOUNT DRIVE cuu
```
2. SMCPROP EXEC issues the command using the CP SMSG interface to the VM Client service machine.
3. VM Client receives the dismount request.



Scenario A - Normal Dismount

If the volume is on the drive, the dismount is processed normally and the process is complete.

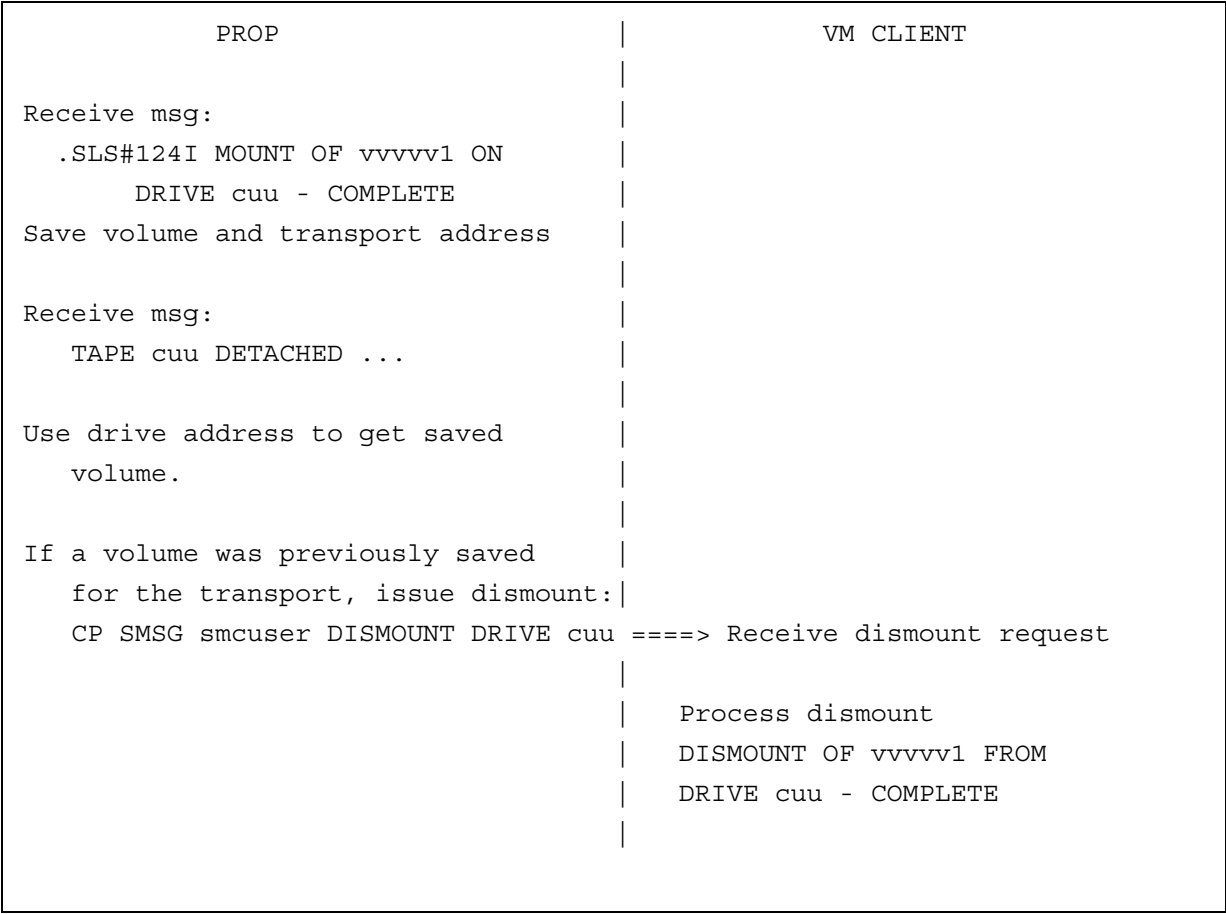


FIGURE 7-2 Scenario A - Normal Dismount

**Scenario B - Dismount already done automatically**

If the drive has already been reallocated and a mount request is issued, VM Client finds the previous volume on the drive and automatically starts dismount processing for that volume. When this automatic dismount completes, the new volume is mounted.

PROP	VM CLIENT
Receive msg:	
.SLS#124I MOUNT OF vvvvvv1 ON	
DRIVE cuu - COMPLETE	
Save volume and transport address	
Receive msg:	
TAPE cuu DETACHED ...	
Use transport address to get	Receive mount for vvvvvv2
save volume.	from TMS
	Issue msg:
	.SLS##81I VOLUME vvvvvv1
	FOUND MOUNTED
	ON DRIVE cuu
	- ATTEMPTING DISMOUNT.
	Process automatic dismount
	of vvvvvv1
	Process mount of vvvvvv2
Issue dismount:	
CP SMSG smcuser DISMOUNT DRIVE cuu	=====> Receive dismount for vvvvvv1
	from PROP
Receive msg:	
.SLS0124I MOUNT OF vvvvvv2 ON	
DRIVE cuu - COMPLETE	
Save volume and transport address.	

**FIGURE 7-3** Scenario B - Dismount Already Processed Automatically

# The ACSRQ Macro

## Introduction

The tape management system interface uses the ACSRQ invocation macro to prepare an IUCV message that contains a request for the VM Client. After the ACSRQ macro returns control to its caller, the TMS routine must issue an IUCV SEND.

## ACSRQ Requests

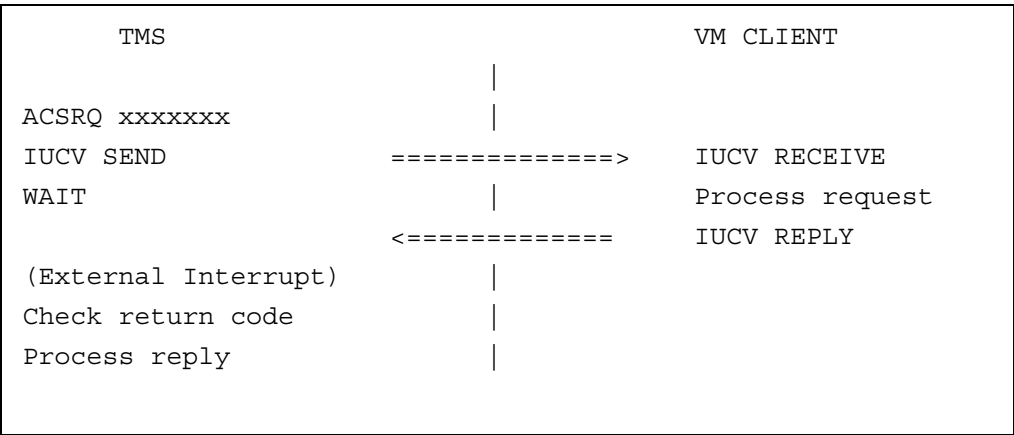
The types of requests used to interact with the library include:

- query information
- set environment parameters
- volume processing

Invoke all VM Client requests through the ACSRQ macro instruction. In general, specify the name of the function to be performed, the address of the data area to be sent (ACSINT), and the other keyword parameters required.

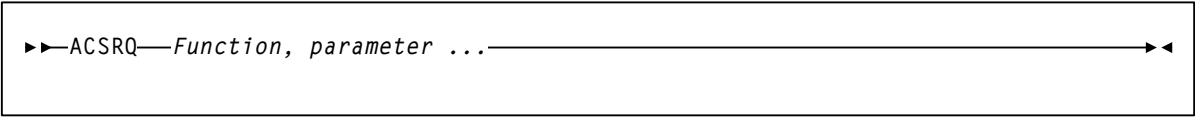
The ACSRQ macro builds the ACS Interface Block (ACSINT) and optionally builds an IPARML for an IUCV SEND. An IUCV instruction referencing the IPARML which has been built should be coded after the ACSRQ macro. The receipt of the ACSINT invokes the proper routines in the VM Client and returns information to the sender using an IUCV REPLY.

The IUCV restrictions documented in the IBM publications *VM/SP System Facilities for Programming* and *VM/XA CP Programming Services* apply. Due to the data area sizes, PRMMMSG is not supported. BUFLIST and ANSLIST are also not supported.



See “SLX Macro Mapping” on page 196 for mapping of the reply area.

## ACSRQ Macro Syntax



►►ACSRQ—*Function, parameter ...*◄◄

**FIGURE 7-4** ACSRQ syntax

where function is one of the following choices, each described in this chapter:

### **DISMOUNT**

Dismount volume

### **EJECT**

Eject volume

### **MOUNT**

Mount volume

### **MOVE**

Move volumes

### **QCAP**

Get CAP information

### **QCONFIG**

Get configuration data

### **QDRIVES**

Get drive information

### **QDRLIST**

Get list of drives for mount

### **QSCRATCH**

Get LSM scratch counts

### **QVOLUME**

Get volume status

### **QVOLUSE**

Return mounted volume information

### **SCRATCH**

Return a volume to scratch status

### **SELSCR**

Select a scratch volume

**UNSCRATCH**

Change the status of a volume to non-scratch.

and parameter describes the parameter(s) valid for specified functions.

,ACCT1=*acct1addr*  
 ,ACCT2=*acct2taddr*  
 ,ACSID=*acsidaddr*  
 ,ACSINT=*acsintaddr*  
 ,CAP=*capidaddr*  
 ,COL=*coladdr*  
 ,COUNT=*countaddr*  
 ,DRIVE=*driveaddr*  
 ,HOSTID=*hostidaddr*  
 ,IPARML=YES  
 ,LIST=*listaddr*  
 ,LSM=*lsmidaddr*  
 ,MEDIA=*medaddr*  
 ,MGMTCLS=*mgmtcls*  
 ,NOTIFY=INSDEL/NOINSDEL  
 ,PAN=*paneladdr*  
 ,PATHID=*pathadr*  
 ,PROTECT=YES  
 ,RECTECH=*recaddr*  
 ,ROW=*rowaddr*  
 ,RSPADDR=*bufadr*  
 ,RSPLN=*buflen*  
 ,SCRATCH=YES  
 ,SUBPOOL=*subpooladdr*  
 ,TEXT=*textaddr*  
 ,TOLSM=*lsmidaddr*  
 ,TOPAN=*paneladdr*  
 ,USER=*useridaddr*  
 ,VOLSER=*voladdr*

The tape management interface (TMI), which allows users to request query information, volume movement, and scratch volume control services from the VM Client, includes requests that allow media and recording technique to be specified.

The requests that can utilize media and recording technique information include:

- MOUNT
- QDRLIST
- QSCRATCH
- SELSCR

These requests are described on the following pages.

The TMI determines media and recording technique values for a request by using MEDia and RECtech parameters directly.

**Note** – If DSECT=YES is specified, no other functions or parameters are valid. An ACSINT DSECT is built.

## DISMOUNT

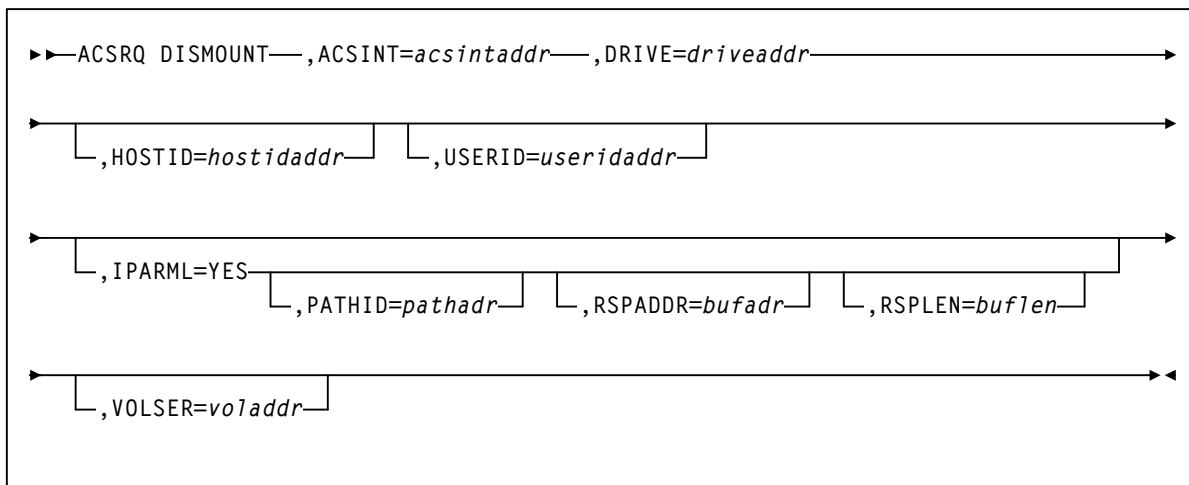
The DISMOUNT request causes a cartridge to be removed from a specific drive. A library cartridge is moved to an LSM cell (selected by the HSC) and becomes available for future requests. A virtual volume remains resident in the VTSS and uses MGMTCLAS definitions to determine migration, replication, and buffer residency policies.

### Considerations

The success of a DISMOUNT request depends on whether the volume has received a REWIND/UNLOAD CCW. If the drive hasn't yet received a REWIND/UNLOAD CCW, the DISMOUNT request is aborted.

A DISMOUNT request may cancel a previous MOUNT request for the same drive.

### Syntax



**FIGURE 7-5** ACSRQ DISMOUNT request syntax

## Parameters

### ACSINT

specifies the address of the data area being sent to the VM Client service machine. This parameter is required.

#### *acsintaddr*

an RX-type data address or the number of the register containing the data address. ACSRQ references the ACSINT at this address when filling in the data.

### DRIVE

specifies the drive from which a volume should be dismounted. This parameter is required.

#### *driveaddr*

the address of the 2-byte drive specification (*ccua*). Specify an RX-type address of the data or the number of the register containing the data address.

### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks. If this parameter is not specified, the host ID executing the request is used.

#### *hostidaddr*

an RX-type host ID address or the register (2) - (12) containing the host ID address.

### IPARML=YES

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

### PATHID

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

#### *pathadr*

an RX-type data address or the number of the register containing the data address. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

### RSPADDR

optionally, specifies the address of the IUCV answer buffer.

#### *bufadr*

an RX-type data address or the number of the register containing the data address. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.



**RSPLEN**

optionally, specifies the address of the length of the IUCV answer buffer.

*buflen*

an RX-type address of the 2-byte field or the number of the register containing the data address. This data is used in the IUCV SEND ANSLEN statement.

RSPLEN is valid only if IPARML=YES is specified. If RSPLEN is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZDISM by QCONFIG.

**USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

*useridaddr*

an RX-type address of the data or the number of the register containing the data address.

If USER is not specified, the userid executing the request is used.

**VOLSER**

optionally, specifies the VOLSER of the volume to be dismounted.

*voladdr*

the address of a 6-character volume label; either an RX-type address of the data or the number of the register containing the data address.

An error will occur if the mounted volume has a different VOLSER.

**Request Response**

The response to a DISMOUNT request is generated when all cartridge movement associated with the request has completed. The response contains a Reply Header and a Message Text Element. The reason code in the Reply Header (SLXSRC) is a binary message number that indicates which HSC message was issued when the DISMOUNT request completed. The Message Text Element contains the complete text of the message specified by the reason code.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

## EJECT

The EJECT request initiates the removal of one or more (up to 500) cartridges from the library. The cartridges are moved from LSM cells to the highest preference CAP or to a CAP specified in the request, so they can be retrieved by an operator.

### Considerations

A request to eject a virtual volume, or a volume not defined in the control data set is considered to be invalid.

The length of the response may vary considerably, depending on the number of volumes specified in the request. Several values are available in the response from a QCONFIG request for use in determining the appropriate answer buffer length for a particular EJECT request. These values include:

- SLXZEJC1 contains the length of an EJECT response for a single volume. Use this value for the answer buffer length when an EJECT request specifies either VOLSER= or COUNT=1.
- SLXXVOLL contains the length of a single Volume Information Element and SLXXMSGSL contains the length of a single Message Text Element. When an EJECT request specifies COUNT=n, then the answer buffer length is computed using the formula:  $((n-1) * (SLXXVOLL + SLXXMSGSL)) + SLXZEJC1$ .
- SLXZEJCT contains the length of an EJECT response when the maximum number of VOLSERS (500) is specified in the request list. Use this value for the answer buffer length when the above formula cannot be used, and when the requestor can afford to commit a large amount of storage (approximately 78KB) to the request.

## Syntax



**FIGURE 7-6** ACSRQ EJECT request syntax

## Parameters

### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the data address. ACSRQ references the ACSINT at this address when filling in the data.

### CAP

optionally, specifies the address of the CAP used to satisfy the request.

#### *capidaddr*

the RX-type address of the data or the number of the register containing the data address.

The format is *AALLCC00*, where *AA* is the ACS number (00-99 decimal), *LL* is the LSM number (00-99 decimal), and *CC* is the CAP number in decimal. These identifiers are always followed by 00.

## COUNT

optionally, specifies the address of a 2-byte field containing the number of VOLSERs in the list designated by the LIST parameter.

### *countaddr*

an RX-type address of the data or the number of the register that contains the address of the data.

COUNT is required with the LIST parameter and is mutually exclusive with the VOLSER parameter.

## HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

### *hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

## IPARML=YES

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

## LIST

optionally, specifies the address of the list of the elements.

### *listaddr*

an RX-type address of the data or the number of a register that contains the data address. Each element in this list is a 6-byte VOLSER.

A special form of this parameter, LIST=\* indicates to ACSRQ that the list is already appended to the ACSINT data area, and does not need to be moved.

LIST is required with the COUNT parameter and is mutually exclusive with the VOLSER parameter.

## LSM

optionally, specifies the address of the LSMid from which the volumes are ejected. If the CAP is not available in the specified LSM, the request fails. If the user does not specify LSM, the HSC chooses a single CAPid in the ACS of the first volume in the list.

The format of an LSMid is *AALL*, where *AA* is the ACS number (decimal) and *LL* is the LSM number (decimal). For example, 0102 is ACS 01, LSM 02.

### *lsmidaddr*

an RX address of the data or the number of the register containing the address of the LSMid.

**PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

***pathadr***

an RX-type address of the data or the number of the register containing the data address. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

**RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

***bufadr***

an RX-type address of the data or the number of the register containing the data address. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

**RSPLN**

optionally, specifies the address of the length of the IUCV answer buffer.

***buflen***

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLN statement.

RSPLN is valid only if IPARML=YES is specified.

If RSPLN is not specified, the subsequent IUCV SEND must specify it. The response length must be determined by one of the methods described in ["Considerations"](#) above.

**SEQ**

optionally, specifies whether or not CAP eject processing fills the CAP cells sequentially or by home location distance.

**NO**

Do not fill CAP cells sequentially. Instead, fill CAP cells by home location distance. This is the default.

specifies that the EJECT process order the requested volumes by home location. EJECT fills the CAP or magazine (for the SL8500) according to the volume home location distance to the CAP; that is, volumes closest to the CAP are ejected first.

## YES

Fill CAP cells sequentially.

specifies that the EJECT process place cartridges in the CAP beginning with the topmost available CAP cell and continuing sequentially.

**Note** – The SEQ parameter is effective for all LSM types but is used primarily for the SL8500 environment. If sequential order is desired for other LSM types, you must code SEQ=YES.

## TEXT

optionally, specifies a 32-character text string for association with the request.

### *textaddr*

an RX-type address of the data or the number of the register containing the address of the data.

## USER

optionally, specifies an 8-byte userid associated with a console ID for the request.

### *useridaddr*

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

## VOLSER

optionally, specifies the address of a 6-character volume label.

### *voladdr*

an RX-type address of the data or the number of the register containing the address of the data.

VOLSER is mutually exclusive with the LIST and COUNT parameters. Either VOLSER or LIST and COUNT must be specified.

## Request Response

The response from an EJECT request consists of one Reply Header and one Volume Information Element, and one Message Text Element for each VOLSER that was specified in the request. Volume Information Elements and Message Text Elements appear in the same order as the VOLSERS in the request.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

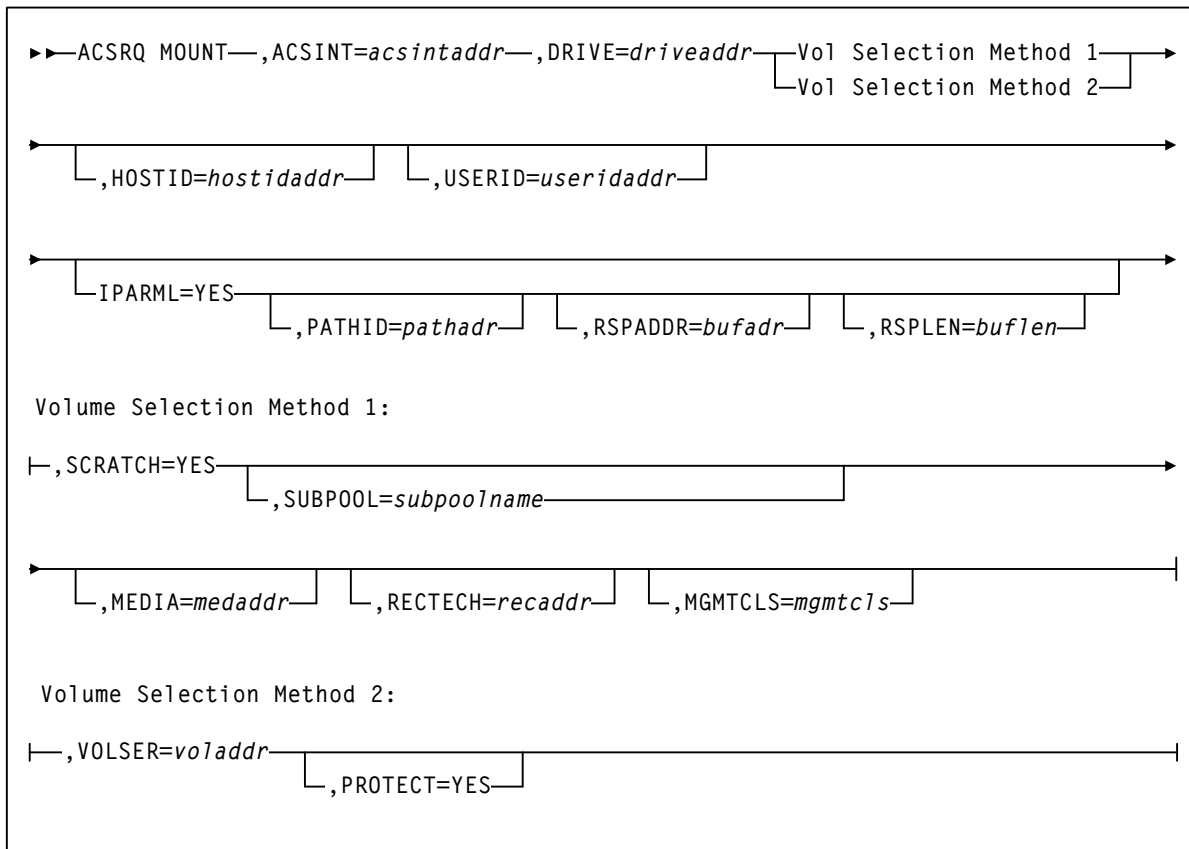
## MOUNT

The MOUNT request causes a volume to be mounted on a specific drive.

### Considerations

An automatic dismount will occur when a MOUNT request is directed to a drive that contains an unloaded cartridge.

### Syntax



**FIGURE 7-7** ACSRQ MOUNT request syntax

## Parameters

### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the data address. ACSRQ references the ACSINT at this address when filling in the data.

### DRIVE

specifies the drive on which the volume is to be mounted; the address of the 2-byte drive specification (ccua). This parameter is required.

#### *driveaddr*

an RX-type address of the data or the number of the register containing the data address.

### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

#### *hostidaddr*

either an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

### MEDIA

optionally, specifies the address of an 8-byte character field containing the media type of the cartridge to be mounted.

If MEDIA is not specified, the next compatible scratch cartridge is mounted without regard to media type.

#### *medaddr*

an RX-type address of the data or the register (2) - (12) containing the address of the data.

### MGMTCLS

optionally, specifies the address of an eight character field containing the management class left justified and padded with blanks.

#### *mgmtclas*

an RX-type address of the data or the register (2) - (12) containing the address of the data.

If MGMTCLS is not specified, but SUBPOOL is specified, then MGMTCLS may be set based upon the VM Client POOLmap command.



**IPARML=YES**

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

**PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

**PROTECT=YES**

optionally, specifies that the volume should be write protected. If PROTECT=YES is not specified, the physical position of the thumbwheel determines whether the volume is write protected.

PROTECT=YES is valid only with VOLSER.

**RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

***bufadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

**RSPLEN**

optionally, specifies the address of the length of the IUCV answer buffer.

***buflen***

an RX-type address of the 2-byte field or the number of the register containing the data address. This data is used in the IUCV SEND ANSLEN statement.

RSPLEN is valid only if IPARML=YES is specified. If RSPLEN is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZMDM by QCONFIG.

**SCRATCH=YES**

optionally, specifies that the request is for a nonspecific (scratch) volume. A scratch VOLSER is selected at this time and mounted on the specified transport.

Either SCRATCH=YES or VOLSER must be specified.

## **SUBPOOL**

optionally, specifies the address of a 13-character field containing the name of the scratch subpool.

### ***subpoolname***

an RX-type address of the data or the number of the register containing the data address. Note that SCRPOOL (subpool index) is no longer supported; you must use the SUBPOOL parameter to select a scratch pool.

SUBPOOL is valid only if SCRATCH=YES is specified. If SUBPOOL is specified, but MGMTCLS is not specified, then MGMTCLS may be set based upon the VM Client POOLmap command.

## **USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

### ***useridaddr***

an RX-type address of the data or the number of the register containing the data address.

If USER is not specified, the userid executing the request is used.

## **VOLSER**

optionally, specifies the address of a 6-character volume label.

### ***voladdr***

an RX-type address of the data or the number of the register containing the data address.

Either VOLSER or SCRATCH=YES must be specified. This parameter specifies the VOLSER of the volume to be mounted. Either VOLSER or SCRATCH=YES must be specified.

## **Request Response**

The response to a MOUNT request is generated when all cartridge movement associated with the request has completed. The response contains a Reply Header, a Message Text Element, and if the request specified SCRATCH=YES, a Volume Information Element. The reason code in the Reply Header (SLXSRC) is a binary message number that indicates which HSC message was issued when the MOUNT request completed. The Message Text Element contains the complete text of the message specified by the reason code. The Volume Information Element is present when the request specified SCRATCH=YES and describes the scratch volume that was mounted.

## MOVE

The MOVE request causes a volume to be moved to a specific location in the ACS.

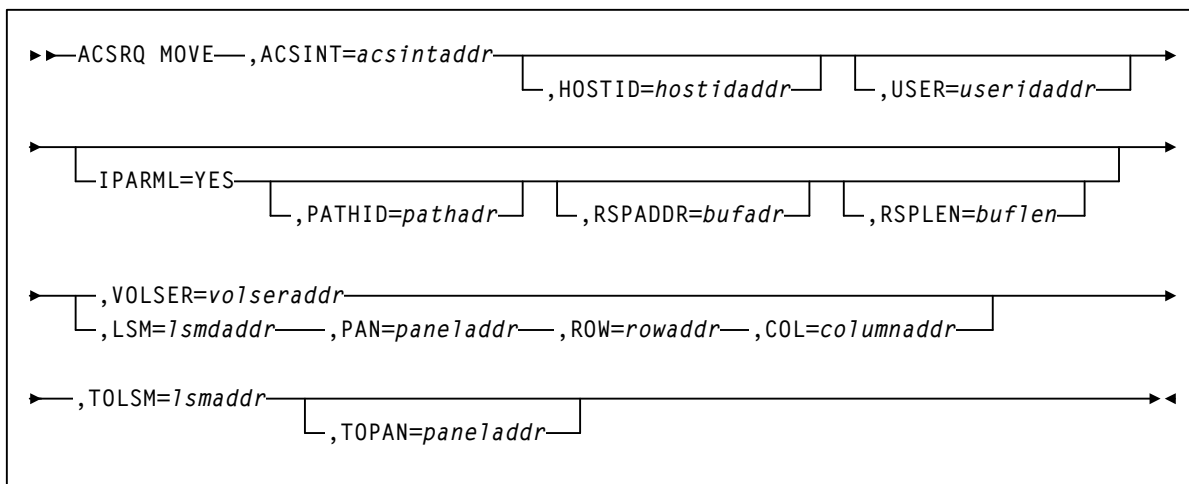
### Considerations

The MOVE function allows movement of a single volume to another location within an ACS. The destination of moved volumes may be the same LSM or a different LSM.

The MOVE functions provides volume movement as well as improved tape management control.

A request to move a virtual volume, or a volume not defined in the control data set is considered to be invalid.

### Syntax



**FIGURE 7-8** ACSRQ MOVE request syntax

## Parameters

### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the data address. ACSRQ references the ACSINT at this address when filling in the data.

### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

#### *hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If the parameter is not specified, the host ID executing the request is used.

### USER

optionally, specifies an 8-byte userid associated with a console ID for the request.

#### *useridaddr*

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

### IPARML=YES

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

### PATHID

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

#### *pathadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

### RSPADDR

optionally, specifies the address of the IUCV answer buffer.

#### *bufadr*

an RX-type address of the data or the number of the register containing the data address. This data is used in the IUCV SEND ANSBUF # statement.

RSPADDR is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

**RSPLEN**

optionally, specifies the address of the length of the IUCV answer buffer.

***buflen***

an RX-type address of the 2-byte field or the number of the register containing the data address. This data is used in the IUCV SEND ANSLEN statement.

RSPLEN is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZMOVE by QCONFIG.

**VOLSER**

specifies the address of a volume number.

***voladdr***

an RX address of the VOLSER or the register (2-12) containing the address of the VOLSER.

This parameter is required unless LSM is specified.

**LSM**

optionally, specifies the address of an LSMid. The format of an LSMid is AALL, where AA is the ACS number (decimal) and LL is the LSM number (decimal). For example, 0110 is ACS 01, LSM 10.

***lsmaddr***

an RX-type address of the LSMid or the register (2) - (12) containing the address of the LSMid.

The COL, PAN, and ROW parameters must accompany the LSM parameter. This parameter is required if VOL is not specified.

**PAN**

optionally, specifies the address of a panel number. Format of the panel number is *pp*, where *pp* is a decimal number.

***paneladdr***

an RX address of the panel or the register (2-12) containing the address of the panel number.

PAN is required if LSM is specified.

**ROW**

optionally, specifies the address of a row number. Format of the row number is *rr* where *rr* is a decimal number.

***rowaddr***

an RX address of the row or the register (2-12) containing the address of the row number.

ROW is required if LSM is specified.

## COL

optionally, specifies the address of a column number. Format of the column number is *cc*, where *cc* is a decimal number.

### *coladdr*

Specify an RX address of the column or the register (2-12) containing the address of the column number.

COL is required if LSM is specified.

## TOLSM

specifies the address of the LSMid where the volume is moved. The LSMid is two hexadecimal bytes in the format AALL, where AA is the ACS number (00-FF hexadecimal) and LL is the LSM number (LL is 00-17 hexadecimal). This parameter is required.

### *lsmaddr*

an RX-type address of the LSMid or the register (2) - (12) containing the address of the LSMid.

## TOPAN

optionally, specifies the address of a panel number. This parameter is required.

### *paneladdr*

an RX address of the panel or the register (2-12) containing the address of the panel number.

## Request Response

The response to a MOVE request is generated when all cartridge movement associated with the request has completed. The response contains a Reply Header, a Message Text Element, and if the request was successful, one Volume Information Element. The reason code in the Reply Header (SLXSRC) is a binary message number that indicates which HSC message was issued when the MOVE request completed. The Message Text Element contains the complete text of the message specified by the reason code.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

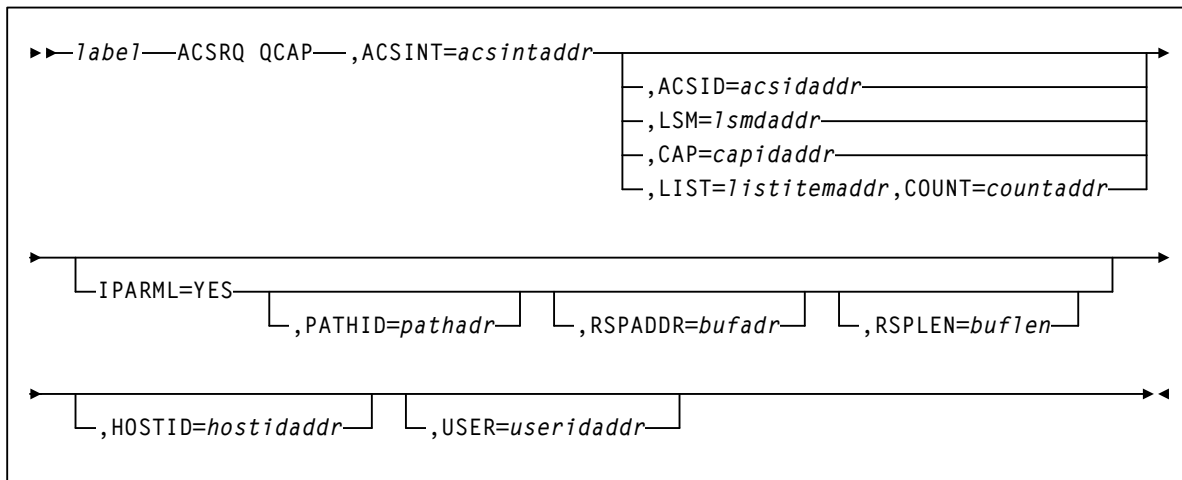
## QCAP

This request is used to query the capacity and status of a CAP.

### Considerations

If ACSID, LSM, CAP, or LIST and COUNT, are not specified, the data returned is for all CAPs.

### Syntax



**FIGURE 7-9** ACSRQ QCAP request syntax

### Parameters

#### ACSID

optionally, specifies the address of the ACS used to satisfy the request. The format of *acsidaddr* is AA, where AA is the ACS number (decimal). For example, 01, designates ACS 01.

#### *acsidaddr*

optionally, an RX-type address of the data or the number of the register containing the address of the data.

If ACSID is specified, information about all CAPs in the ACS is returned.

#### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data.

## **CAP**

optionally, specifies the address of the CAP used to satisfy the request.

### ***capidaddr***

the RX-type address of the data or the number of the register containing the data address.

The format of *capidaddr* is AALLCC00, where AA is the ACS number (decimal), LL is the LSM number (decimal), and CC is the CAP number. These identifiers are always followed by 00.

If CAP is specified, information about the specified CAP is returned.

## **COUNT**

optionally, specifies the address of a 2-byte field containing the number of CAPIDs in the list designated by the LIST parameter.

### ***countaddr***

an RX-type address of the data or the number of the register that contains the address of the data.

COUNT is required with the LIST parameter.

## **HOSTID**

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

### ***hostidaddr***

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If the parameter is not specified, the host ID executing the request is used.

## **IPARML=YES**

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

## **LIST**

optionally, specifies the address of the list of CAPs to be queried.

### ***listitemaddr***

an RX-type address of the data or the number of the register containing the address of the data.

If LIST is specified, information about all CAPs in the list is returned.



**LSM**

optionally, specifies the address of an LSMid. The format of an LSMid is AALL, where AA is the ACS number (00-FF hexadecimal) and LL is the LSM number (LL is 00-17 hexadecimal). For example, 0102 is ACS 01, LSM 02. All values are in hexadecimal format.

***lsmaddr***

an RX-type address of the LSMid or the register (2) - (12) containing the address of the LSMid.

If LSM is specified, the returned data is for the specific LSM. If ACSID, LSM, CAP, or LIST and COUNT, are not specified, the data returned is for all CAPs.

**PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

**RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

***bufadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

**RSPLEN**

optionally, specifies the address of the length of the IUCV answer buffer.

***buflen***

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLEN statement.

RSPLEN is valid only if IPARML=YES is specified. If RSPLEN is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZQSCR.

## **USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

### ***useridaddr***

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

## **Request Response**

The response to the QCAP request contains a Reply Header and a CAP information element containing information about each CAP requested.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

## QCONFIG

The QCONFIG request is used to obtain summary information about the TapePlex configuration and the recommended answer buffer lengths for other TMS interface requests.

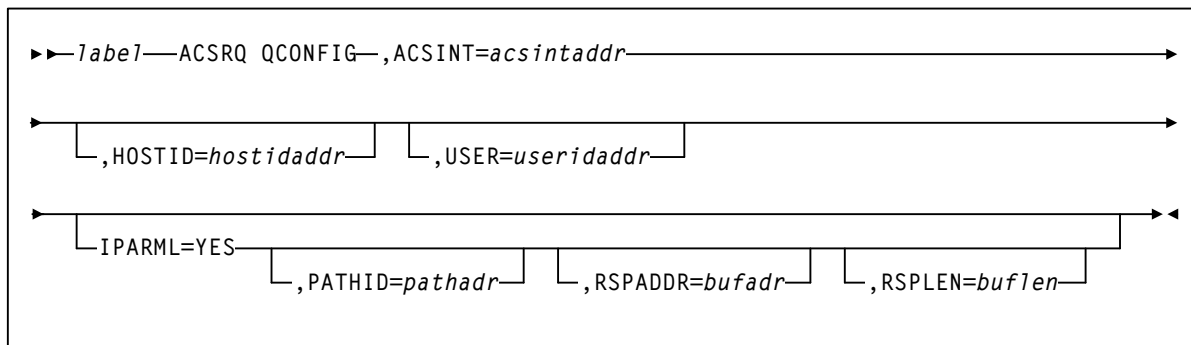
## Considerations

The QCONFIG request should be the first request issued after an IUCV connection has been established because its response contains a recommended answer buffer length (response length) for each type of TMS interface request.

The length of the QCONFIG response may change from release to release. A TMS should use the following technique to obtain the recommended answer buffer length for a QCONFIG request:

1. Issue a QCONFIG request with answer buffer length of decimal 16. The response from this request consists of a Reply Header, truncated to 16 bytes. The return code in the header is 4, which indicates that the answer buffer was too small to contain the entire response. The word at offset decimal 12, SLXCRLN, contains the recommended answer buffer length for a QCONFIG request.
2. Reissue the QCONFIG request using the recommended answer buffer length.

## Syntax



**FIGURE 7-10** ACSRQ QCONFIG request syntax

## Parameters

### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the data address. ACSRQ references the ACSINT at this address when filling in the data.

### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

#### *hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

### IPARML=YES

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

### PATHID

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

#### *pathadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

### RSPADDR

optionally, specifies the address of the IUCV answer buffer.

#### *bufadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

### RSPLN

optionally, specifies the address of the length of the IUCV answer buffer.

#### *buflen*

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLN statement.

RSPLN is valid only if IPARML=YES is specified. If RSPLN is not specified, the subsequent IUCV SEND must specify it. The response length must be determined by the technique described in the QCONFIG Considerations section above.

**USER**

specifies an 8-byte userid associated with a console ID for the request.

***useridaddr***

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

**Request Response**

The QCONFIG response contains a Reply Header and a Configuration Summary Element. Note that the length of each type of response element (e.g., Volume Element) is returned in the Reply Header by QCONFIG.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

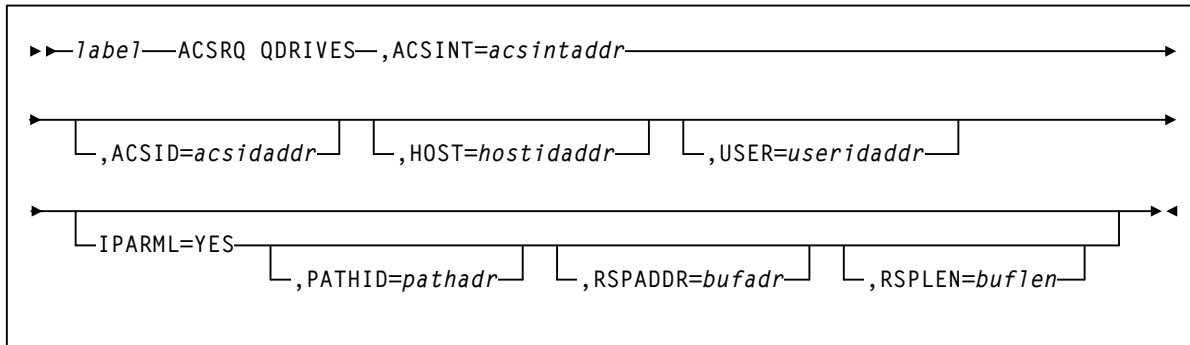
## QDRIVES

The QDRIVES request is used to obtain detailed information about all transports and LSMs associated with the library, or with a particular ACS.

### Considerations

None.

### Syntax



**FIGURE 7-11** ACSRQ QDRIVES request syntax

### Parameters

#### ACSID

optionally, specifies the address of the ACS used to satisfy the request.

*acsidaddr*

Format for the ACSid is AA, where AA is a decimal number. For example, 01, designates ACS 01. Specify either an RX-type address of the data or the number of the register containing the address of the data.

If ACSID is specified, the data is returned for the specific ACS. If ACSID is not specified, data is returned for all ACSs.

#### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

*acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data.

**HOSTID**

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

***hostidaddr***

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

**IPARML=YES**

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

**PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

**RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

***bufadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

**RSPLN**

optionally, specifies the address of the length of the IUCV answer buffer.

***buflen***

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLEN statement.

RSPLN is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZQDRV by QCONFIG.

## USER

specifies an 8-byte userid associated with a console ID for the request.

### *useridaddr*

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

## Request Response

The QDRIVES response consists of a Reply Header, a Drive Information section, and an LSM Information section. The Drive Information section contains one Drive Information Element for each transport in the library or ACS. The LSM Information section contains one LSM Information Element for each LSM in the library or ACS.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.



## QDRLIST

The QDRLIST request is used to obtain the TapePlex recommendation for a library transport to be specified for a subsequent MOUNT request.

### Considerations

The VM Client makes its recommendation by returning a list of Drive Information Elements which are ordered so that the first element describes the best transport to use, the second element describes the second best transport to use, etc.

When the QDRLIST request specifies a particular cartridge (i.e., VOLSER is specified), the Drive Information Elements are arranged so that the first transport listed is in the LSM that is closest to (or the same as) the LSM containing the cartridge. The last transport listed is in the LSM that is farthest from the LSM containing the cartridge. Only transports in the same ACS as the cartridge are represented in the Drive Information section.

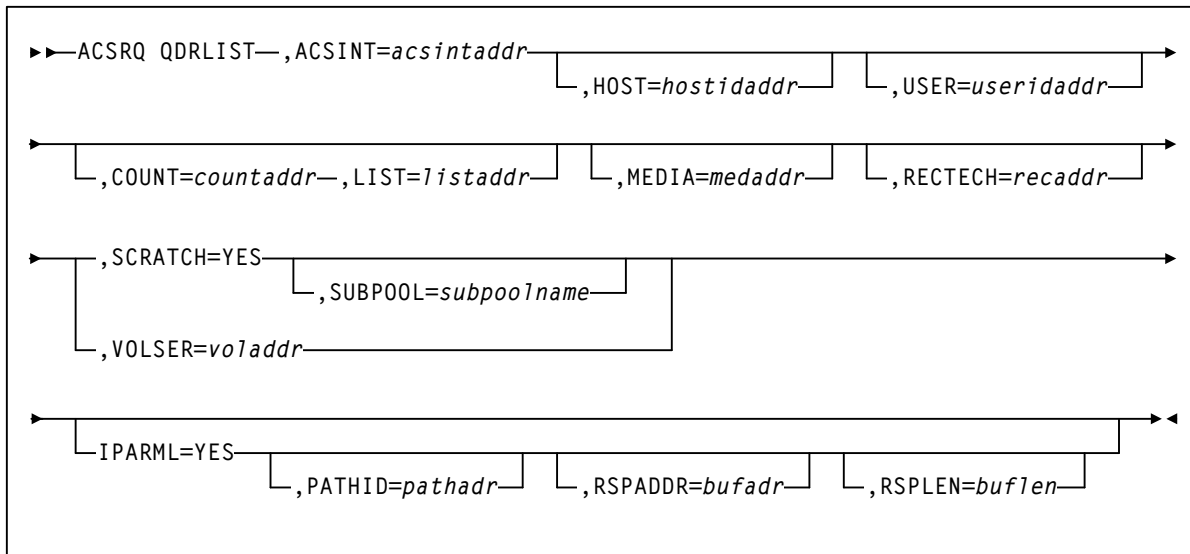
When the QDRLIST request specifies a scratch volume (i.e., SCRATCH=YES is specified), the Drive Information Elements are arranged so the first transport listed is in the LSM containing the most scratch volumes. The last transport listed is in the LSM containing the fewest scratch volumes. All transports in all ACSs are represented in the Drive Information section.

For requests for a specific virtual volume, either the drives in the VTSS where the volume is resident or can be recalled are returned. For requests for a virtual scratch volume, drives in VTSSs that support the requested management class are returned.

The VM Client ignores whether a transport already has a volume mounted, or is in an offline LSM or ACS when it orders the Drive Information Elements.

An optional list of transport addresses may be supplied with the QDRLIST request. If a list is provided, the VM Client uses it as a screen while building its response. A Drive Information Element is included in the response only when its transport address is present in the list.

## Syntax



**FIGURE 7-12** ACSRQ QDRLIST request syntax

## Parameters

### ACSINT

specifies the address of the data area being sent to the library service machine.

*acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data. This parameter is required.

### COUNT

optionally, specifies the address of a 2-byte field containing the number of tape drive device addresses in the list designated by the LIST parameter.

*countaddr*

an RX address of the data or the number of the register that contains the address of the data.

COUNT is required with the LIST parameter.

### MEDIA

optionally, specifies the address of an 8-byte character field containing the media type of the selected transport.

If MEDia is not specified, transports are selected without regard to media type.

*medaddr*

an RX-type address of the data or the register (2) - (12) containing the address of the data.

See [“Media Type \(MEDia\)” on page 226](#) for a list of valid *media-type* values.

**RECTECH**

optionally, specifies the address of an 8-byte field containing the recording technique used to record data tracks on the tape surface.

If RECTECH is not specified, transports are selected depending on the MEDIA type that has been specified.

***recaddr***

an RX-type address of the data or the register (2) - (12) containing the address of the data.

See [“Recording Technique \(RECTECH\)” on page 229](#) for a list of valid *recording-technique* values.

**HOSTID**

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

***hostidaddr***

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

**IPARML=YES**

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

**LIST**

optionally, specifies the address of the list of the elements.

***listaddr***

an RX-type address of the data or the number of a register that contains the address of the data.

Each element in this list is a 2-byte drive address (ccua).

A special form of this parameter, LIST=\* indicates to ACSRQ that the list is already appended to the ACSINT data area, and does not need to be moved.

**PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

## **RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

### *bufadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

## **RSPLN**

optionally, specifies the address of the length of the IUCV answer buffer.

### *buflen*

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLEN statement.

RSPLN is valid only if IPARML=YES is specified. If RSPLN is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZQDRL by QCONFIG.

## **SCRATCH=YES**

optionally, specifies that the request is for a nonspecific (scratch) volume.

When this option is specified, the order of the returned list of drive elements is determined by the number of scratch volumes in the LSM containing the transport. Transports in the LSM with highest scratch volume count appear first, followed by transports in the LSM with the second highest scratch volume count. Transports in the LSM with the lowest scratch volume count appear last.

Either SCRATCH=YES or VOLSER must be specified.

## **SUBPOOL**

optionally, specifies the address of a 13-character field containing the name of the scratch subpool.

### *subpoolname*

an RX-type address of the data or the number of the register containing the address of the data. Note that SCRPOOL (subpool index) is no longer supported; you must use the SUBPOOL parameter to select a scratch pool.

SUBPOOL is valid only if SCRATCH=YES is specified.

## **USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

### *useridaddr*

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

**VOLSER**

optionally, specifies the address of a 6-character volume label.

***voladdr***

an RX-type address of the data or the number of the register containing the address of the data.

Either VOLSER or SCRATCH=YES must be specified. When this option is specified, the order of the returned list of drive elements is determined by the proximity of the specified volume to a transport. Transports in the LSM which holds the specified volume appear first, followed by transports in adjacent LSMs. Transports in the LSM farthest from the specified volume appear last.

**Request Response**

The QDRLIST response always contains a Reply Header and a Drive Information section. If VOLSER was specified, the response also contains a Volume Information Element.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

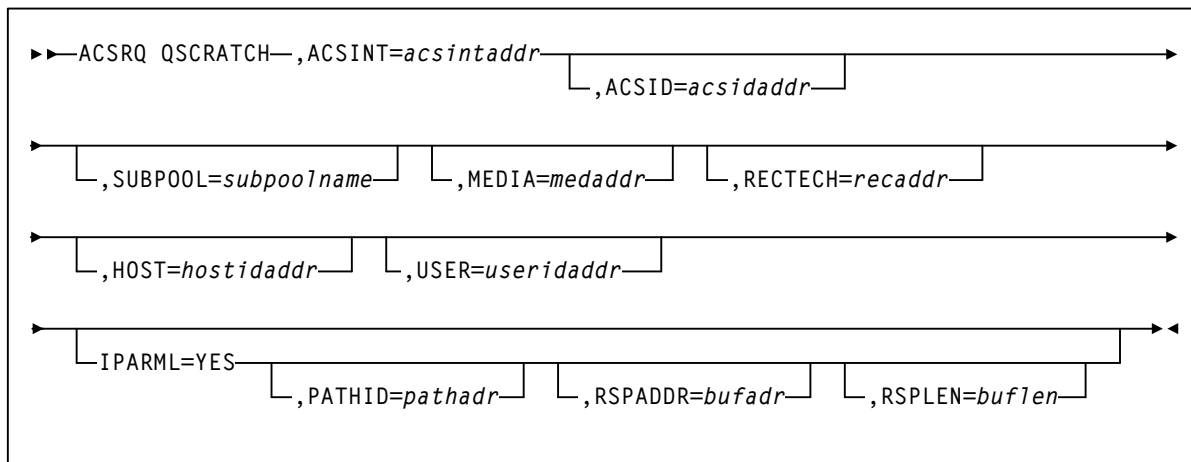
## QSCRATCH

The QSCRATCH request is used to obtain detailed information about all LSMs associated with the library, or with a particular ACS. This information includes the number of scratch volumes in each LSM.

### Considerations

When the library and its associated control data set are shared by more than one HSC, then the reported scratch totals may differ from the true totals because they may not account for recent scratch volume activity on other processors. However, each HSC refreshes its scratch volume totals from the control data set every five minutes so the variance should be slight.

### Syntax



**FIGURE 7-13** ACSRQ QSCRATCH request syntax

## Parameters

### ACSID

optionally, specifies the address of the ACS used to satisfy the request.

#### *acsidaddr*

Format for the ACSid is AA, where AA is a decimal number. For example, 01, designates ACS 01.

an RX-type address of the data or the number of the register containing the address of the data.

If ACSID is specified, the data is returned for the specific ACS. If ACSID is not specified, data is returned for all ACSs.

### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data.

### MEDIA

optionally, specifies the address of an 8-byte character field containing the media type of the requested scratch cartridge.

If MEDia is not specified, scratch cartridges are selected without regard to media type.

#### *medaddr*

an RX-type address of the data or the register (2) - (12) containing the address of the data.

See [“Media Type \(MEDia\)” on page 226](#) for a list of valid *media-type* values.

### RECTECH

optionally, specifies the address of an 8-byte field containing the recording technique used to record data tracks on the tape surface.

**This parameter is optional.** If RECtech is not specified, scratch cartridges are selected depending on the MEDia type that has been specified.

#### *recaddr*

an RX-type address of the data or the register (2) - (12) containing the address of the data.

See [“Recording Technique \(RECtech\)” on page 229](#) for a list of valid *recording-technique* values.

## HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

### *hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

## IPARML=YES

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

## PATHID

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

### *pathadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

## RSPADDR

optionally, specifies the address of the IUCV answer buffer.

### *bufadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

## RSPLEN

optionally, specifies the address of the length of the IUCV answer buffer.

### *buflen*

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLEN statement.

RSPLEN is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZQSCR.



**SUBPOOL**

optionally, specifies the address of a 13-character field containing the name of the scratch subpool.

***subpoolname***

Specify either an RX-type address of the data or the number of the register containing the address of the data. Note that SCRPOOL (subpool index) is no longer supported; you must use the SUBPOOL parameter to select a scratch pool.

SUBPOOL is valid only if SCRATCH=YES is specified.

**USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

***useridaddr***

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

**Request Response**

The QSCRATCH response contains a Reply Header and an LSM Information Element for each LSM.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

## QVOLUME

The QVOLUME request is used to obtain the current library status of one or more (up to 500) cartridges.

### Considerations

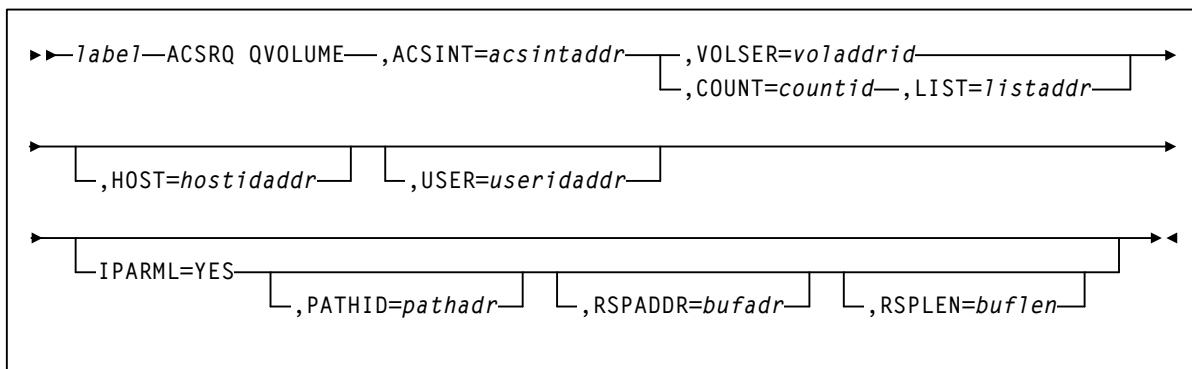
The length of the response may vary considerably, depending on the number of volumes specified in the request. Several values are available in the response from a QCONFIG request for use in determining the appropriate answer buffer length for a particular QVOLUME request. These values include:

SLXZQVOL contains the length of a QVOLUME response for a single volume. Use this value for the answer buffer length when a QVOLUME request specifies either VOLSER or COUNT=1.

SLXXVOLL contains the length of a single Volume Information Element. When a QVOLUME request specifies COUNT=n, then the answer buffer length is computed using the formula:  $((n-1)*SLXXVOLL)+SLXZQVOL$ .

SLXZVOL contains the length of a QVOLUME response when the maximum number of VOLSERs (500) is specified in the request list. Use this value for the answer buffer length when the above formula cannot be used, and when the requestor can afford to commit a large amount of storage (approximately 16KB) to the request.

### Syntax



**FIGURE 7-14** ACSRQ QVOLUME request syntax

## Parameters

### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data.

### COUNT

optionally, specifies the address of a 2-byte field containing the number of VOLSERs in the list designated by the LIST parameter.

#### *countaddr*

an RX address of the data or the number of the register that contains the address of the data.

COUNT is required with the LIST parameter and is mutually exclusive with the VOLSER parameter.

### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

#### *hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

### IPARML=YES

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

### LIST

optionally, specifies the address of the list of the elements.

#### *listaddr*

Specify either an RX-type address of the data or the number of a register that contains the address of the data.

Each element in this list is a 6-byte VOLSER.

A special form of this parameter, LIST=\* indicates to ACSRQ that the list is already appended to the ACSINT data area, and does not need to be moved.

LIST is required along with the COUNT parameter and is mutually exclusive with the VOLSER=parameter.

## **PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

### ***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

## **RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

### ***bufadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

## **RSPLN**

optionally, specifies the address of the length of the IUCV answer buffer.

### ***buflen***

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLN statement.

RSPLN is valid only if IPARML=YES is specified.

If RSPLN is not specified, the subsequent IUCV SEND must specify it. The response length must be determined by one of the methods described in the QVOLUME Considerations section, above.

## **USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

### ***useridaddr***

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

## **VOLSER**

optionally, specifies the address of a 6-character volume label.

### ***voladdr***

an RX-type address of the data or the number of the register containing the address of the data.

Either VOLSER or LIST must be specified. VOLSER specifies the volume for which status is requested. VOLSER is mutually exclusive with the COUNT parameter.

## Request Response

The QVOLUME response consists of a Reply Header and a Volume Information Element for each VOLSER that was specified in the request. Volume Information Elements appear in the same order as the VOLSERs in the request.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

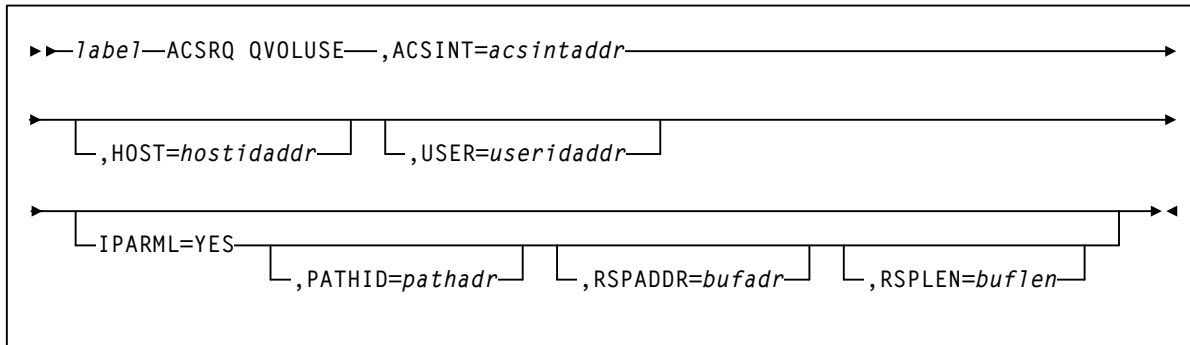
## QVOLUME

The QVOLUME request is used to obtain the current status of mounted volumes.

### Considerations

The length of the response may vary considerably, depending on the number of volumes returned by the request.

### Syntax



**FIGURE 7-15** ACSRQ QVOLUME request syntax

### Parameters

#### ACSINT

specifies the address of the data area being sent to the library service machine.

*acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data. This parameter is required.

#### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

*hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

#### IPARML=YES

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

**PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

*pathadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

**RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

*bufadr*

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

**RSPLN**

optionally, specifies the address of the length of the IUCV answer buffer.

*buflen*

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLN statement.

RSPLN is valid only if IPARML=YES is specified. If RSPLN is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZQVOL.

**USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

*useridaddr*

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

**Request Response**

The QVOLUSE response consists of a Reply Header and a Volume Information Element for each VOLSER that was returned by the request. Volume Information Elements appear in the same order as the VOLSERS returned by the request.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

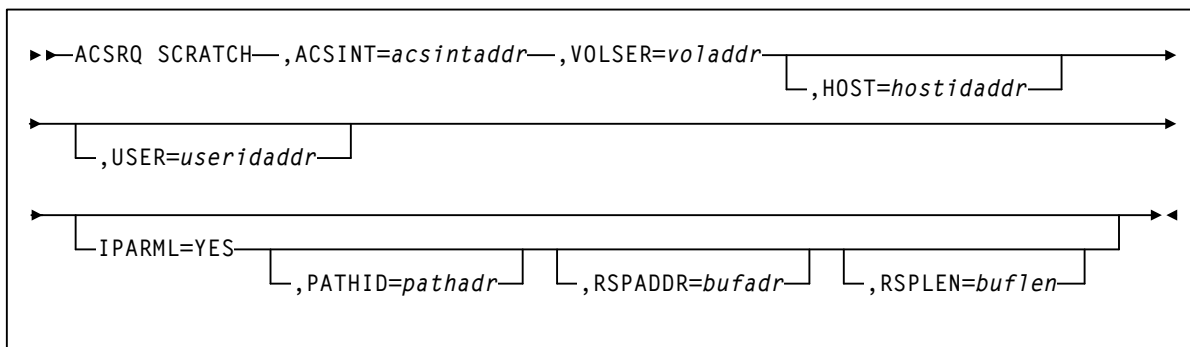
## SCRATCH

The SCRATCH request causes a volume to be placed in scratch status in the server control data set.

### Considerations

The specified cartridge must already be in the library. No cartridge movement occurs as the result of a SCRATCH request. However, SCRATCH request processing must select the cartridge (i.e., must acquire exclusive use of the cartridge) in order to change its state. This means that a SCRATCH request will fail if the cartridge is mounted on a drive (see "MOUNT Considerations" on page 49).

### Syntax



**FIGURE 7-16** ACSRQ SCRATCH request syntax

### Parameters

#### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

##### *acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data.

#### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

##### *hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If the parameter is not specified, the host ID executing the request is used.



**IPARML=YES**

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

**PATHID**

specifies the address of a 2-byte IUCV path ID of the library service machine.

***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

**RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

***bufadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

**RSPLN**

optionally, specifies the address of the length of the IUCV answer buffer.

***buflen***

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLEN statement.

RSPLN is valid only if IPARML=YES is specified. If RSPLN is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZSCR by QCONFIG.

**USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

***useridaddr***

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

**VOLSER**

specifies the address of a 6-character volume label.

***voladdr***

an RX-type address of the data or the number of the register containing the address of the data.

VOLSER is required. It specifies the volume which is to be marked as scratch.

## **Request Response**

The response to SCRATCH contains only a Reply Header.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

## SELSCR

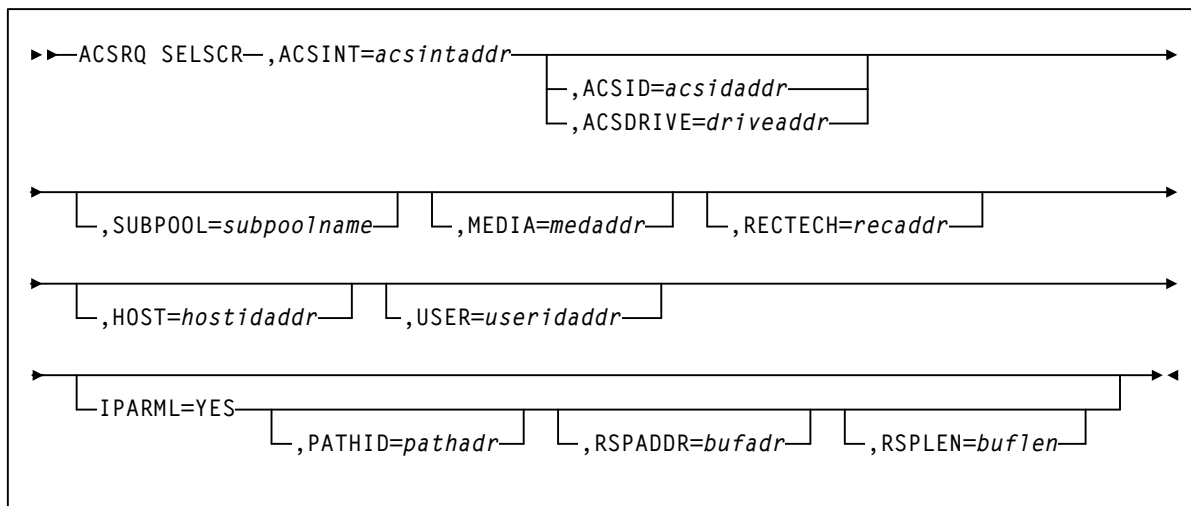
The SELSCR request causes the server to select a library scratch volume and remove it from scratch status in the control data set.

### Considerations

No volume movement occurs.

When neither ACSID nor DRIVE is specified, the HSC searches through all library LSMs and chooses a scratch volume from the LSM containing the most scratch volumes. When ACSID is specified, the HSC chooses a scratch volume from the LSM in the specified ACS that holds the most cartridges. When DRIVE is specified the HSC chooses a scratch volume from the closest LSM, if the drive is in an automatic mode LSM.

### Syntax



**FIGURE 7-17** ACSRQ SELSCR request syntax

## Parameters

### ACSID

optionally, specifies the address of the ACS used to satisfy the request. Format for the ACSid is AA, where AA is a hexadecimal number 00-FF. For example, 01, designates ACS 01.

#### *acsidaddr*

an RX-type address of the data or the number of the register containing the address of the data.

If ACSID is specified, the data is returned for the specific ACS. If ACSID is not specified, data is returned for all ACSs. This parameter is mutually exclusive with DRIVE.

### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

#### *acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data.

### DRIVE

optionally, specifies the address of the 2-byte drive specification (ccua).

#### *driveaddr*

an RX-type address of the data or the number of the register containing the address of the data.

DRIVE is mutually exclusive with ACSID. It specifies the drive which the scratch volume should be near.

### MEDIA

optionally, specifies the address of an 8-byte character field containing the media type of the selected scratch cartridge. If MEDia is not specified, scratch cartridges are selected without regard to media type.

#### *medaddr*

an RX-type address of the data or the register (2) - (12) containing the address of the data.

See [“Media Type \(MEDia\)” on page 226](#) for a list of valid *media-type* values.

**RECTECH**

optionally, specifies the address of an 8-byte field containing the recording technique used to record data tracks on the tape surface.

If RECtech is not specified, scratch cartridges are selected depending on the MEDia type that has been specified.

***recaddr***

an RX-type address of the data or the register (2) - (12) containing the address of the data.

See [“Recording Technique \(RECtech\)” on page 229](#) for a list of valid *recording-technique* values.

**HOSTID**

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

***hostidaddr***

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

**IPARML=YES**

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

**PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If PATHID is not specified, the subsequent IUCV SEND must specify it.

**RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

***bufadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If RSPADDR is not specified, the subsequent IUCV SEND must specify it.

## RSPLEN

optionally, specifies the address of the length of the IUCV answer buffer.

### *buflen*

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLEN statement.

RSPLEN is valid only if IPARML=YES is specified. If RSPLEN is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZGSCR by QCONFIG.

## SUBPOOL

optionally, specifies the address of a 13-character field containing the name of the scratch subpool.

### *subpoolname*

an RX-type address of the data or the number of the register containing the address of the data. Note that SCRPOOL (subpool index) is no longer supported; you must use the SUBPOOL parameter to select a scratch pool.

SUBPOOL is valid only if SCRATCH=YES is specified.

## USER

optionally, specifies an 8-byte userid associated with a console ID for the request.

### *useridaddr*

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.

## Request Response

The response to SELSCR contains a Reply Header and a Volume Information Element. The Reply Header contains a return code (SLXCMDRC) indicating the success of the operation. The Volume Information Element describes the selected volume.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

## UNSCRATCH

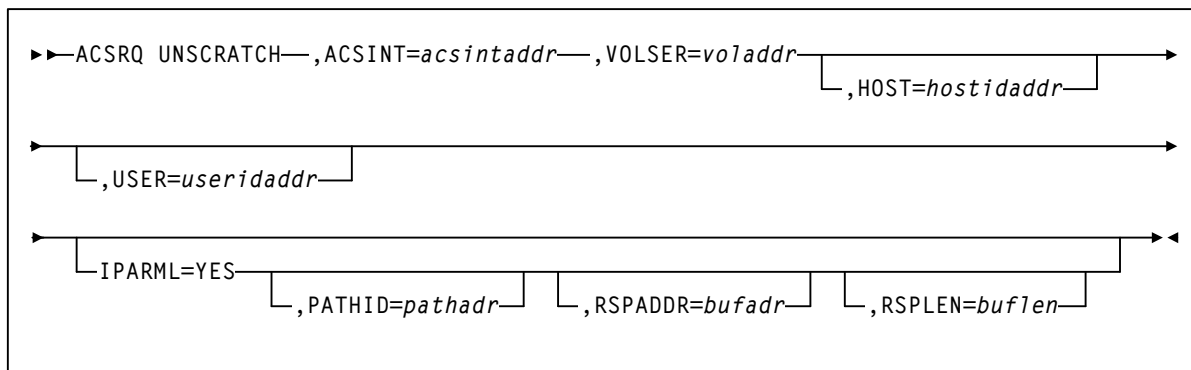
The UNSCRATCH request causes a volume to be removed from scratch status in the control data set.

### Considerations

A request to unscratch of volume not defined in the control data set is considered to be invalid.

For real volumes no cartridge movement occurs as a result of the UNSCRATCH request. However, UNSCRATCH request processing must select the volume (i.e. must acquire exclusive use of the volume) in order to change its state. This means that an UNSCRATCH request will fail if the volume is mounted on a real or virtual drive (see [“Considerations” on page 151](#)).

### Syntax



**FIGURE 7-18** ACSRQ UNSCRATCH request syntax

### Parameters

#### ACSINT

specifies the address of the data area being sent to the library service machine. This parameter is required.

##### *acsintaddr*

an RX-type address of the data or the number of the register containing the address of the data. ACSRQ references the ACSINT at this address when filling in the data.

#### HOSTID

optionally, specifies the address of an eight character host ID, left justified and padded with blanks.

##### *hostidaddr*

an RX-type address of the host ID or the register (2) - (12) containing the address of the host ID.

If HOSTID is not specified, the host ID executing the request is used.

### **IPARML=YES**

optionally, specifies that the requestor has established addressability to an IUCV IPARML, and that the IPARML is initialized.

### **PATHID**

optionally, specifies the address of a 2-byte IUCV path ID of the library service machine.

#### ***pathadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND PATHID statement.

PATHID is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

### **RSPADDR**

optionally, specifies the address of the IUCV answer buffer.

#### ***bufadr***

an RX-type address of the data or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSBUF statement.

RSPADDR is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it.

### **RSPLN**

optionally, specifies the address of the length of the IUCV answer buffer.

#### ***buflen***

an RX-type address of the 2-byte field or the number of the register containing the address of the data. This data is used in the IUCV SEND ANSLEN statement.

RSPLN is valid only if IPARML=YES is specified. If this parameter is not specified, the subsequent IUCV SEND must specify it. The response length must be at least the value returned in SLXZSCR by QCONFIG.

### **USER**

optionally, specifies an 8-byte userid associated with a console ID for the request.

#### ***useridaddr***

an RX-type address of the data or the number of the register containing the address of the data.

If USER is not specified, the userid executing the request is used.



**VOLSER**

specifies the address of a 6-character volume label. This parameter is required. It specifies the volume which is to be removed from scratch status.

***voladdr***

an RX-type address of the data or the number of the register containing the address of the data.

**Request Response**

The response to UNSCRATCH contains only a Reply Header.

See [“SLX Macro Mapping” on page 196](#) for information on the SLX macro.

# Interface Data Areas

## SLX Macro

A reply always begins with a header. The header may be followed by one or more “sections”. Each section is a table of “elements” of a particular type (e.g. Volume Information Element). If the Reply Header is aligned on a doubleword boundary, then all subsequent sections and elements are guaranteed to also begin on doubleword boundaries.

A section directory (number/offset/length) is defined in the Reply Header for each possible section type, even though no reply will ever contain all types of sections. The order in which the section directories appear within the header has no relationship to the order in which sections are physically arranged after the header.

A section directory’s number specifies how many elements of that type are actually present in the reply. If a section directory’s number is nonzero, then that section directory’s offset specifies the offset, from the start of the reply header, to the first (or only) element of that type. If a section directory’s number is greater than one, then that section directory’s length, which specifies the length of a single element of that type, must be used to access the second and subsequent elements of that type. For example, add the length to the offset to get the offset to the second element; add in the length again to get the offset to the third element; and so on. The number of elements in each section is variable. The following tables denote which sections of the reply will be returned for a given request.

**TABLE 7-1** SLX Macro - Parameter Matrix

Reply Section	Header	Config	CAP	Volume	Drive	LSM	Msg Text
DISMOUNT	1						1
EJECT	1			*			*
MOUNT	1			1 if scratch			1
MOVE	1			1 if successful			1
QCAP	1		1+				
QCONFIG	1	1					
QDRIVES	1				1+	1+	
QDRLIST	1			1 if not scratch	1+		
QEJECT	1						
QREQUEST	1						
QSCRATCH	1						
QVOLUME	1			*			
QVOLUME	1			*			
QVOLUME	1			*			
SCRATCH	1					1+	
SELSCR	1			1			
UNSCRATCH	1						

**Note –**

- 1 = 1 and only 1.
- \* = EJECT and QVOLUME are limited by the number of Volume Information Elements contained in the SLX reply area (maximum of 500).
- 1+ = From 1 to “n” depending on library configuration.

## SLX Macro Mapping

**TABLE 7-2** SLX Record Format

Dec	Hex	Type	Length	Label	Description
-----	-----	------	--------	-------	-------------

SLX - VM CLIENT EXTERNAL INTERFACE REPLY  
FUNCTION:

MAPS A REPLY AREA RETURNED BY ONE OF THE FOLLOWING VM CLIENT REQUESTS:

DISMOUNT - DISMOUNT A VOLUME  
EJECT - EJECT A VOLUME FROM THE LIBRARY  
MOUNT - MOUNT A VOLUME  
MOVE - MOVE A VOLUME  
QCAP - RETURN CAP SUMMARY  
QCONFIG - RETURN CONFIGURATION SUMMARY  
QDRIVES - RETURN DRIVE AND LSM INFORMATION  
QDRLIST - RETURN DRIVE INFORMATION, ORDERED BY PREFERENCE  
QSCRATCH - RETURN LSM INFORMATION, ORDERED BY PREFERENCE  
QVOLUME - RETURN VOLUME INFORMATION  
SCRATCH - CHANGE A VOLUME'S STATUS TO 'SCRATCH'  
SELSCR - SELECT A SCRATCH VOLUME

SPECIAL CONSIDERATIONS:

A REPLY ALWAYS BEGINS WITH A HEADER. THE HEADER MAY BE FOLLOWED BY ONE OR MORE "SECTIONS". EACH SECTION IS A TABLE OF "ELEMENTS" OF A PARTICULAR TYPE (E.G. VOLUME INFORMATION ELEMENT). IF THE REPLY HEADER IS ALIGNED ON A DOUBLEWORD BOUNDARY, THEN ALL SUBSEQUENT SECTIONS AND ELEMENTS ARE GUARANTEED TO ALSO BEGIN ON DOUBLEWORD BOUNDARIES.

A SECTION DIRECTORY (NUMBER/OFFSET/LENGTH) IS DEFINED IN THE REPLY HEADER FOR EACH POSSIBLE SECTION TYPE, EVEN THOUGH NO REPLY WILL EVER CONTAIN ALL TYPES OF SECTIONS. THE ORDER IN WHICH THE SECTION DIRECTORIES APPEAR WITHIN THE HEADER HAS NO RELATIONSHIP TO THE ORDER IN WHICH SECTIONS ARE PHYSICALLY ARRANGED AFTER THE HEADER.

A SECTION DIRECTORY'S NUMBER SPECIFIES HOW MANY ELEMENTS OF THAT TYPE ARE ACTUALLY PRESENT IN THE REPLY. IF A SECTION DIRECTORY'S NUMBER IS NONZERO, THEN THAT SECTION DIRECTORY'S OFFSET SPECIFIES THE OFFSET, FROM THE START OF THE REPLY HEADER, TO THE FIRST (OR ONLY) ELEMENT OF THAT TYPE. IF A SECTION DIRECTORY'S NUMBER IS GREATER THAN ONE, THEN THAT SECTION DIRECTORY'S LENGTH, WHICH SPECIFIES THE LENGTH OF A SINGLE ELEMENT OF THAT TYPE, MUST BE USED TO ACCESS THE SECOND AND SUBSEQUENT ELEMENTS OF THAT TYPE: ADD THE LENGTH TO THE OFFSET TO GET THE OFFSET TO THE SECOND ELEMENT; ADD IN THE LENGTH AGAIN TO GET THE OFFSET TO THE THIRD ELEMENT; AND SO ON.

HEADER					
0	(0)	STRUCTURE		SLX	
0	(0)	AREA	1	SLXRPLY	REPLY HEADER
0	(0)	CHARACTER	3	SLXHID	HEADER IDENTIFIER
3	(3)	A-ADDR	1	SLXCMDRC	RETURN CODE:
0	(00)	CONST		SLXROK	REQUEST PROCESSED SUCCESSFULLY

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
4	(04)	CONST		SLXRWARN	REQUEST SUCCESSFUL WITH WARNING SLXSRC WILL PROVIDE THE SPECIFIC REASON FOR THE WARNING
8	(08)	CONST		SLXRBADP	REQUEST FAILED; THE REQUEST BLOCK (MAPPED BY ACSINT) CONTAINED INVALID DATA (E.G., INCOMPATIBLE OPTIONS); SLXSRC (REASON CODE) WILL PROVIDE THE OFFSET OF THE ACSINT FIELD FOUND TO BE IN ERROR.
12	(0C)	CONST		SLXRIERR	REQUEST FAILED; AN UNRECOVERABLE INTERNAL ERROR OCCURRED WHILE PROCESSING THE REQUEST.
16	(10)	CONST		SLXRFAIL	REQUEST FAILED; SLXSRC WILL PROVIDE THE SPECIFIC REASON FOR THE FAILURE.
20	(14)	CONST		SLXRNHSC	REQUEST FAILED - HSC NOT AVAILABLE
44	(2C)	CONST		SLXRBADL	REQUEST FAILED; REPLY AREA PROVIDED BY REQUESTOR WAS TOO SMALL TO CONTAIN ALL REPLY DATA ASSOCIATED WITH THE REQUEST. IF FIELD SLXCRLN IS NON-ZERO, IT CONTAINS THE LENGTH VALUE THAT SHOULD BE SPECIFIED FOR THE REPLY AREA FOR THIS REQUEST.
48	(30)	CONST		SLXRNVCI	VCIRQST AND VCIRESPT NOT SUPPORTED. EITHER VTCS IS NOT INSTALLED - OR - IS NOT AT THE REQUIRED LEVEL TO SUPPORT THE PGMI VCI RESPONSES.
4	(04)	A-ADDR	1	SLXVERS	REPLY VERSION CODE:
7	(07)	CONST	3	SLXVCODE	THIS IS VERSION 9 OF THE REPLY AREA.
5	(5)	HEXSTRING	4	-RESERVED-	RESERVED.
8	(8)	SIGNED-FWORD		SLXSRC	REASON CODE FOR FAILED OPERATION.
32818	(8032)	CONST		SLXTINTR	PGMI TASK INTERRUPTED.
32822	(8036)	CONST		SLXSANF	SEARCH ARGUMENT NOT FOUND.
32826	(803A)	CONST		SLXMSTT	MISMATCHED TOKEN TYPES.
32832	(8040)	CONST		SLXTRNF	TOKEN AREA NOT FOUND.
32848	(8050)	CONST		SLXSFUL	REPLY AREA FULL.
32849	(8051)	CONST		SLXDVMM	MEDIA INCOMPATIBLE WITH DEVICE TYPE.

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
12	(C)	SIGNED-FWORD	4	SLXCRLN	IF RETURN CODE (SLXCMDRC) IS 2C (SLXRBADL), THEN THIS FIELD CONTAINS EITHER THE MINIMUM ACCEPTABLE REPLY AREA LENGTH FOR THE REQUEST, OR 0 IF THE MINIMUM LENGTH COULDN'T BE DETERMINED. OTHERWISE (I.E., RETURN CODE ISN'T 2C), THIS FIELD CONTAINS THE ACTUAL LENGTH OF THIS REPLY.
16	(10)	SIGNED-FWORD	4	SLXPEOFF	PARAMETER ERROR OFFSET IF SLSXRC <> 0 THEN THIS POINTS TO AN ELEMENT IN A LIST WHERE PROCESSING STOPPED WHEN THE REQUEST WAS "QCAP".
20	(14)	LENGTH		SLXHL	TO MAKE COMPATIBLE WITH MVS CODE
CONFIGURATION SUMMARY SECTION DIRECTORY					
20	(14)	SIGNED-FWORD	4	SLXXCFGN	NUMBER OF CONFIGURATION ELEMENTS PRESENT IN THIS REPLY.
24	(18)	SIGNED-FWORD	4	SLXXCFGO	OFFSET TO CONFIGURATION SECTION, FROM START OF REPLY, OR 0 IF REPLY DOESN'T CONTAIN ANY CONFIGURATION ELEMENTS.
28	(1C)	SIGNED-FWORD	4	SLXXCFGL	LENGTH OF A CONFIGURATION ELEMENT.
VOLUME INFORMATION SECTION DIRECTORY					
32	(20)	SIGNED-FWORD	4	SLXXVOLN	NUMBER OF VOLUME ELEMENTS PRESENT IN THIS REPLY.
36	(24)	SIGNED-FWORD	4	SLXXVOLO	OFFSET TO VOLUME SECTION, FROM START OF REPLY, OR 0 IF REPLY DOESN'T CONTAIN ANY VOLUME ELEMENTS.
40	(28)	SIGNED-FWORD	4	SLXXVOLL	LENGTH OF A VOLUME ELEMENT.
DRIVE INFORMATION SECTION DIRECTORY					
44	(2C)	SIGNED-FWORD	4	SLXXDRVN	NUMBER OF DRIVE ELEMENTS PRESENT IN THIS REPLY.
48	(30)	SIGNED-FWORD	4	SLXXDRVO	OFFSET TO DRIVE SECTION, FROM START OF REPLY, OR 0 IF REPLY DOESN'T CONTAIN ANY DRIVE ELEMENTS.
52	(34)	SIGNED-FWORD	4	SLXXDRVL	LENGTH OF A DRIVE ELEMENT.
LSM INFORMATION SECTION DIRECTORY					
56	(38)	SIGNED-FWORD	4	SLXXLSMN	NUMBER OF LSM ELEMENTS PRESENT IN THIS REPLY.

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
60	(3C)	SIGNED-FWORD	4	SLXXLSMO	OFFSET TO LSM SECTION, FROM START OF REPLY, OR 0 IF REPLY DOESN'T CONTAIN ANY LSM ELEMENTS.
64	(40)	SIGNED-FWORD	4	SLXXLSML	LENGTH OF AN LSM ELEMENT.
MESSAGE TEXT SECTION DIRECTORY WARNING: THIS DIRECTORY DOES NOT EXIST WHEN THE VALUE IN THE REPLY VERSION NUMBER FIELD, SLXVERS, IS LESS THAN 2.					
68	(44)	SIGNED-FWORD	4	SLXXMSGN	NUMBER OF MESSAGE ELEMENTS PRESENT IN THIS REPLY.
72	(48)	SIGNED-FWORD	4	SLXXMSGO	OFFSET TO MESSAGE SECTION, FROM START OF REPLY, OR 0 IF REPLY DOESN'T CONTAIN ANY MESSAGE ELEMENTS.
76	(4C)	SIGNED-FWORD	4	SLXXMSGL	LENGTH OF A MESSAGE ELEMENT.
QCAP INFORMATION SECTION DIRECTORY					
80	(50)	SIGNED-FWORD	4	SLXXCAPN	NUMBER OF CAP ELEMENTS PRESENT IN THIS REPLY.
84	(54)	SIGNED-FWORD	4	SLXXCAPO	OFFSET TO CAP SECTION FROM START OF REPLY.
88	(58)	SIGNED-FWORD	4	SLXXCAPL	LENGTH OF A CAP ELEMENT.
CONFIGURATION SUMMARY ELEMENT THIS ELEMENT APPEARS IN THE REPLY TO A QCONFIG REQUEST AND SUPPLIES SUMMARY INFORMATION ABOUT THE LIBRARY AND ABOUT REPLY LENGTHS NECESSARY FOR OTHER TYPES OF REQUESTS.					
0	(0)	STRUCTURE		SLXSCFG	CONFIGURATION SUMMARY ELEMENT.
0	(0)	CHARACTER	3	SLXLID	ELEMENT IDENTIFIER.
3	(3)	HEXSTRING	1	-RESERVED-	RESERVED.
4	(4)	CHARACTER	8	SLXLHNAM	HOST NAME.
12	(C)	SIGNED-FWORD	4	SLXLHHBT	HOST PULSE VALUE.
16	(10)	SIGNED-FWORD	4	SLXLRSTM	RESERVE TIMEOUT LIMIT.
20	(14)	CHARACTER	8	SLXLQNAM	ENQ MAJOR NAME.
28	(1C)	CHARACTER	8	SLXLEJPS	EJECT COMMAND PASSWORD (ENCRYPTED)
36	(24)	CHARACTER	1	SLXLCMPF	COMMAND PREFIX CHARACTER.
37	(25)	A-ADDR	1	SLXLSCLB	LIBRARY DEFAULT SCRATCH LABEL TYPE CODE:
1	(01)	CONST		SLXLLBSL	STANDARD (SL).
2	(02)	CONST		SLXLLBAL	ASCII (AL).
3	(03)	CONST		SLXLLBNL	NON-LABELED (NL).
4	(04)	CONST		SLXLLBNS	NON-STANDARD LABEL (NSL)

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
38	(26)	A-ADDR	1	SLXLSMF	SMF RECORD TYPE
39	(27)	HEXSTRING	1	-RESERVED-	RESERVED.
40	(28)	SIGNED-FWORD	4	SLXQMDR	LARGEST NUMBER OF DRIVES IN ANY ACS.
44	(2C)	SIGNED-FWORD	4	SLXQDRCT	NUMBER OF DRIVES IN THE LIBRARY.
48	(30)	SIGNED-FWORD	4	SLXQACNT	NUMBER OF ACSS IN THE LIBRARY.
52	(34)	SIGNED-FWORD	4	SLXQLCNT	NUMBER OF LSMS IN THE LIBRARY.
56	(38)	SIGNED-FWORD	4	SLXZVOL	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A QVOLUME REQUEST THAT SPECIFIES THE LARGEST SUPPORTED NUMBER OF VOLUMES (500).
60	(3C)	SIGNED-FWORD	4	SLXZQDRV	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A QDRIVES REQUEST.
64	(40)	SIGNED-FWORD	4	SLXZQDRL	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A QDRLIST REQUEST.
68	(44)	SIGNED-FWORD	4	SLXQVOL	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A QVOLUME REQUEST THAT SPECIFIES ONLY 1 VOLUME.
72	(48)	SIGNED-FWORD	4	SLXZGSCR	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A SELSCR REQUEST.
76	(4C)	SIGNED-FWORD	4	SLXZMDM	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A MOUNT REQUEST.
80	(50)	SIGNED-FWORD	4	SLXZQSCR	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A QSCRATCH REQUEST.
84	(54)	SIGNED-FWORD	4	SLXZDISM	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A DISMOUNT REQUEST.
88	(58)	SIGNED-FWORD	4	SLXZEJCT	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO AN EJECT REQUEST THAT SPECIFIES THE LARGEST SUPPORTED NUMBER OF VOLUMES (500).
92	(5C)	SIGNED-FWORD	4	SLXZSCR	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A SCRATCH REQUEST.



**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
WARNING: THE REMAINING FIELDS OF THIS ELEMENT ARE AVAILABLE FOR VERSION(S) 3 AND ABOVE.					
108	(6C)	SIGNED-FWORD	4	SLXZMOVE	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO AN MOVE REQUEST.
112	(70)	SIGNED-FWORD	4	SLXZEJC1	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO AN EJECT REQUEST FOR ONLY 1 VOLUME.
WARNING: THE REMAINING FIELDS OF THIS ELEMENT ARE AVAILABLE FOR VERSION(S) 6 AND ABOVE.					
116	(74)	A-ADDR	4	SLXQUCSA	MVS -- ADDRESS OF SLSUXCSA.
120	(78)	SIGNED-FWORD	4	-RESERVED-	RESERVED.
124	(7C)	SIGNED-FWORD	4	SLXQLCAP	NUMBER OF CAPS IN LIBRARY.
128	(80)	SIGNED-FWORD	4	SLXEXLM0	ExLM R15
132	(84)	SIGNED-FWORD	4	SLXEXLM1	ExLM R1
136	(88)	SIGNED-FWORD	4	SLXEXLM2	ExLM R2
140	(8C)	SIGNED-FWORD	4	SLXZQCAP	MAXIMUM LENGTH OF REPLY DATA RETURNED IN RESPONSE TO A QCAP REQUEST.
156	(9C)	SIGNED-FWORD	4	-RESERVED-	RESERVED FUTURE USE.
160	(A0)	SIGNED-HWORD	2	SLXHSCV	HSC VERSION NUMBER
162	(A2)	HEXSTRING	6	-RESERVED-	
168	(A8)	CONST		SLXSCFGL	LENGTH OF A CONFIGURATION ELEMENT.
QDSN INFORMATION ELEMENT THIS ELEMENT APPEARS IN THE REPLY TO A QDSN REQUEST AND SUPPLIES SUMMARY INFORMATION ABOUT THE CURRENT REFERENCED DATASETS USED BY THE HSC.					
0	(0)	STRUCTURE		SLXDSNIM	DATASET INFORMATION MAP.
0	(0)	CHARACTER	3	SLXQDID	SECTION IDENTIFIER.
3	(03)	BITSTRING	1	SLXDSFLG	DATASET TYPE.
1	(01)	CONST		SLXDSPRM	CDS PRIMARY.
2	(02)	CONST		SLXDSSEC	CDS SECONDARY.
3	(03)	CONST		SLXDSSBY	CDS STANDBY.
4	(04)	CONST		SLXDVSAT	VOLUME ATTRIBUTES.
5	(05)	CONST		SLXDSUAT	UNIT ATTRIBUTES.
6	(06)	CONST		SLXDSTRQ	TAPEREQS.
7	(07)	CONST		SLXDSPLB	PARMLIB.
8	(08)	CONST		SLXDSJNP	PRIMARY JOURNAL.
9	(09)	CONST		SLXDSJNA	ALTERNATE JOURNAL.
9	(09)	CONST		SLXDSEMAX	MAX NUMBER OF QDS RETURNED.

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
4	(04)	CHARACTER	44	SLXDSNAM	DATASET NAME.
48	(30)	CHARACTER	8	SLXDSEMBR	MEMBER NAME.
56	(38)	CHARACTER	6	SLXDSEVOL	VOLUME NAME.
62	(3E)	CHARACTER	8	SLXDSEUNT	UNIT NAME.
70	(46)	CHARACTER	2	-RESERVED-	RESERVED.
72	(48)	CHARACTER	96	-RESERVED-	RESERVED.
168	(A8)	AREA	8	-RESERVED-	ALIGN
168	(A8)	LENGTH		SLXDSENEL	LENGTH OF ONE DATASET ENTRY.
CAP INFORMATION ELEMENT THIS ELEMENT SUPPLIES INFORMATION ABOUT A SINGLE LIBRARY CAP					
0	(0)	STRUCTURE		SLXSCAP	
0	(0)	CHARACTER	4	SLXCID	SECTION IDENTIFIER.
4	(4)	HEXSTRING	1	SLXCACS	ACS ADDRESS.
5	(5)	HEXSTRING	1	SLXCLSM	LSM ADDRESS.
6	(6)	HEXSTRING	1	SLXCCAP	CAP NUMBER.
7	(7)	HEXSTRING	1	-RESERVED-	RESERVED.
8	(8)	AREA	2	SLXCSTAT	CAP STATUS.
8	(8)	BITSTRING	1	SLXCSTB1	CAP STATUS.
		1... .... X'80'		SLXCSTCA	CAP IS ACTIVE.
		.1.. .... X'40'		SLXCSTNR	CAP NEEDS RECOVERY.
		..1. .... X'20'		SLXCSTAM	CAP IS IN AUTOMATIC MODE.
		...1 .... X'10'		SLXCSTCL	CAP IS LINKED.
		.... 1... X'08'		SLXCSTCO	CAP IS ONLINE.
9	(9)	BITSTRING	1	SLXCSTB2	CAP MODE.
		1... .... X'80'		SLXCSTIE	CAP IS ENTERING.
		.1.. .... X'40'		SLXCSTID	CAP IS DRAINING.
		..1. .... X'20'		SLXCSTIJ	CAP IS EJECTING.
		...1 .... X'10'		SLXCSTIC	CAP IS CLEANING.
		.... 1... X'08'		SLXCSTII	CAP IS IDLE .
10	(A)	BITSTRING	1	SLXTYPE	TYPE OF CAP.
		1... .... X'80'		SLXCTPC	PRIORITY CAP
		.... ..1 X'01'		SLXCTCIM	CIMARRON
		.... ..1. X'02'		SLXCTCLP	CLIPPER.
		.... ..11 X'03'		SLXCTTWS	STANDARD CLIPPER
		.... ..1.. X'04'		SLXCTTWO	OPTIONAL CLIPPER
		.... ..1.1 X'05'		SLXCTTIM	(9740/TimberWolf)
12	(C)	SIGNED-HWORD	2	SLXCCELL	CELLS IN CAP.

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
14	(E)	HEXSTRING	1	SLXCNROW	ROWS.
15	(F)	HEXSTRING	1	SLXCNCOL	COLUMNS .
16	(10)	HEXSTRING	1	SLXCCMAG	MAGAZINES.
17	(11)	HEXSTRING	1	SLXCCMGC	CELLS IN MAGAZINE.
18	(12)	CHARACTER	8	SLXCJOB	JOBNAME OF OWNER.
26	(1A)	HEXSTRING	6	-RESERVED-	ALIGN TO DOUBLE WORD
32	(20)	CONST		SLXSCAPL	LENGTH OF A CAP ELEMENT.
VOLUME INFORMATION ELEMENT THIS ELEMENT SUPPLIES INFORMATION ABOUT A SINGLE VOLUME AND IS REPEATED FOR EACH VOLUME ASSOCIATED WITH A REQUEST. THIS ELEMENT MAY APPEAR IN REPLIES TO THE FOLLOWING REQUESTS: QDRLIST - RETURN DRIVE INFORMATION, ORDERED BY PREFERENCE QVOLUME - RETURN VOLUME INFORMATION SELSCR - SELECT A SCRATCH VOLUME EJECT - EJECT VOLUMES					
0	(0)	STRUCTURE		SLXSVOL	VOLUME INFORMATION ELEMENT.
0	(0)	CHARACTER	3	SLXVID	ELEMENT IDENTIFIER.
3	(3)	BITSTRING	1	SLXVSTA	VOLUME STATUS:
		1... .... X'80'		SLXVILB	VOLUME IS IN LIBRARY
		. .1.. .... X'40'		SLXVOHST	VOLUME IS IN USE BY ANOTHER HOST
		. ..1. .... X'20'		SLXVSCR	VOLUME IS CONSIDERED SCRATCH
		. ...1 .... X'10'		SLXVMAL	VOLUME IS IN MANUAL-MODE LSM.
		.... 1... X'08'		SLXVDSC	VOLUME IS IN DISCONNECTED ACS.
		.... .1.. X'04'		SLXVMNT	VOLUME IS MOUNTED ON A DRIVE.
		.... ..1. X'02'		SLXVERR	VOLUME IS 'ERRANT' (I.E., ITS LOCATION WITHIN THE LIBRARY IS UNCERTAIN).
		.... ...1 X'01'		SLXVTV	VOLUME IS A VTCS VIRTUAL VOLUME
4	(4)	CHARACTER	6	SLXVSER	VOLUME SERIAL.
10	(A)	A-ADDR	1	SLXVLC	VOLUME LOCATION CODE:
0	(0)	CONST		SLXVUNK	LOCATION DATA UNAVAILABLE (SLXVLOC IS 0).
1	(1)	CONST		SLXVCEL	LOCATION DATA DESCRIBES A CELL.
2	(02)	CONST		SLXVDRV	LOCATION DATA DESCRIBES A DRIVE.
11	(B)	AREA	5	SLXVLOC	VOLUME LOCATION DATA:

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
11	(B)	A-ADDR	1	SLXVACS	ACS ID.
12	(C)	A-ADDR	1	SLXVLSM	LSM ID.
13	(D)	A-ADDR	3	SLXVPNL	CELL'S PANEL ID, ROW ID, COLUMN ID.
13	(D)	A-ADDR	2	SLXVDRIV	DRIVE ADDRESS (0CUU).
15	(F)	BITSTRING	1	SLXVSTA2	MORE VOLUME STATUS: EQU X'E0' RESERVED.
		...1 .... X'10'		SLXVMLMU	VOLUME MEDIA TYPE CAME FROM LMU. EQU X'08' RESERVED.
		.... .1.. X'04'		SLXVMUNR	VOLUME MEDIA TYPE UNREADABLE.
		.... ..1. X'02'		SLXVMVLA	VOLUME MEDIA TYPE CAME FROM VOLATTR.
		.... ...1 X'01'		SLXVMDFL	VOLUME MEDIA TYPE DEFAULT ASSIGNED.
16	(10)	AREA	8	SLXVTSSN	VTSS NAME
16	(10)	SIGNED-FWORD	4	SLXVDATI	HI-WORD OF TOD AT INSERTION.
20	(14)	SIGNED-FWORD	4	SLXVDATL	HI-WORD OF TOD LAST SELECTION.
24	(18)	SIGNED-FWORD	4	SLXVSCNT	SELECTION COUNT.
28	(1C)	SIGNED-FWORD	4	SLXVDATD	HI-WORD OF TOD LAST MOUNT.
32	(20)	CHARACTER	8	SLXVMED	TYPE OF MEDIA.
40	(28)	CONST		SLXSVOLN	LENGTH OF A VOLUME ELEMENT.
40	(28)	CONST		SLXSVOLL	LENGTH OF A VOLUME ELEMENT.
DRIVE INFORMATION ELEMENT THIS ELEMENT SUPPLIES INFORMATION ABOUT A SINGLE LIBRARY TAPE DRIVE AND IS REPEATED FOR EACH DRIVE ASSOCIATED WITH A REQUEST. THIS ELEMENT MAY APPEAR IN REPLIES TO THE FOLLOWING REQUESTS: QDRIVES - RETURN DRIVE AND LSM INFORMATION QDRLIST - RETURN DRIVE INFORMATION, ORDEREDBY PREFERENCE					
0	(0)	STRUCTURE		SLXSDRV	DRIVE INFORMATION ELEMENT.
0	(0)	CHARACTER	3	SLXDID	ELEMENT IDENTIFIER
3	(3)	BITSTRING	1	SLXDSTA	LIBRARY STATUS:
		...1 .... X'10'		SLXDMANU	LSM IS IN MANUAL MODE.
		.... 1... X'08'		SLXDDISC	ACS IS DISCONNECTED.
4	(4)	A-ADDR	1	SLXQDEAC	ACS ID.
5	(5)	A-ADDR	1	SLXQDELS	LSM ID.
6	(6)	A-ADDR	2	SLXQDECU	DRIVE ADDRESS (0CUU).
8	(8)	CHARACTER	8	SLXQDRT	RECORDING TECHNIQUE OF DRIVE
16	(10)	CONST		SLXSDRVL	LENGTH OF A DRIVE ELEMENT.

**TABLE 7-2** SLX Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
LSM INFORMATION ELEMENT THIS ELEMENT SUPPLIES INFORMATION ABOUT A SINGLE LSM (LIBRARY STORAGE MODULE) AND IS REPEATED FOR EACH LSM ASSOCIATED WITH A REQUEST. THIS ELEMENT MAY APPEAR IN REPLIES TO THE FOLLOWING REQUESTS: QDRIVES - RETURN DRIVE AND LSM INFORMATION QSCRATCH - RETURN LSM INFORMATION, ORDERED BY PREFERENCE					
0	(0)	STRUCTURE		SLXSLSM	LSM INFORMATION ELEMENT
0	(0)	CHARACTER	3	SLXMID	ELEMENT IDENTIFIER.
3	(3)	BITSTRING	1	SLXMSTAT	LIBRARY STATUS:
		...1 .... X'10'		SLXMANUL	LSM IS IN MANUAL MODE
		.... 1... X'08'		SLXMDISC	ACS IS DISCONNECTED.
4	(4)	A-ADDR	1	SLXMACS	ACS ID.
5	(5)	A-ADDR	1	SLXMLSM	LSM ID.
6	(6)	SIGNED-FWORD	1	SLXMADJN	NUMBER OF ADJACENT LSMS.
7	(7)	A-ADDR	1	SLXMADJI(4)	LIST OF LSM IDS OF ADJACENT LSMS (ONLY THE FIRST N IDS ARE VALID, WHERE N IS THE VALUE IN SLXMADJN).
11	(B)	HEXSTRING	1	-RESERVED-	RESERVED.
12	(C)	SIGNED-FWORD	4	SLXMNSCR	NUMBER OF SCRATCH VOLUMES IN THIS LSM.
16	(10)	SIGNED-FWORD	4	SLXMTCEL	TOTAL CELLS IN LSM.
20	(14)	SIGNED-FWORD	4	SLXMFCEL	FREE CELLS IN LSM.
24	(18)	CONST		SLXSLSML	LENGTH OF AN LSM ELEMENT.
MESSAGE TEXT ELEMENT THIS ELEMENT SUPPLIES THE COMPLETE TEXT OF THE MESSAGE WHOSE BINARY MESSAGE ID NUMBER IS REPORTED IN HEADER FIELD SLXSRC. THIS ELEMENT MAY APPEAR IN REPLIES TO THE FOLLOWING REQUESTS: DISMOUNT - DISMOUNT A VOLUME MOUNT - MOUNT A VOLUME MOVE - MOVE A VOLUME EJECT - EJECT VOLUMES					
0	(0)	STRUCTURE		SLXMSG	MESSAGE TEXT ELEMENT.
0	(0)	CHARACTER	3	SLXGID	ELEMENT IDENTIFIER.
3	(3)	CHARACTER	125	SLXGTEXT	TEXT OF THE MESSAGE WHOSE NUMBER IS IN FIELD SLXSRC OF THE REPLY HEADER.
0	(0)	STRUCTURE		SLXMSG	
128	(80)	CONST		SLXSMSGL	LENGTH OF A MESSAGE ELEMENT.

**Cross Reference**

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
SLXCACS	000001	04
SLXCCAP	000001	06
SLXCCELL	000002	0C
SLXCCMAG	000001	10
SLXCCMGC	000001	11
SLXCID	000004	00
SLXCJOB	000008	12
SLXCLSM	000001	05
SLXCMDRC	000001	03
SLXCNCOL	000001	0F
SLXCNROW	000001	0E
SLXCRLN	000004	0C
SLXCSTAM	-	20
SLXCSTAT	000002	08
SLXCSTB1	000001	08
SLXCSTB2	000001	09
SLXCSTCA	-	80
SLXCSTCL	-	10
SLXCSTCO	-	08
SLXCSTIC	-	10
SLXCSTID	-	40
SLXCSTIE	-	80
SLXCSTII	-	08
SLXCSTIJ	-	20
SLXCSTNR	-	40
SLXCTCIM	-	01
SLXCTCLP	-	02
SLXCTPC	-	80
SLXCTTIM	-	05
SLXCTTWO	-	04
SLXCTTWS	-	03
SLXCTYPE	000001	0A
SLXDDISC	-	08

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
SLXDID	000003	00
SLXDMANU	-	10
SLXDSFLG	000001	03
SLXDSJNA	-	09
SLXDSJNP	-	08
SLXDSDMAX	-	09
SLXDSDMBR	000008	30
SLXDSDNAM	000044	04
SLXDSDNEL	-	A8
SLXDSDPLB	-	07
SLXDSDPRM	-	01
SLXDSSBY	-	03
SLXDSSSEC	-	02
SLXDSTA	000001	03
SLXDSTRQ	-	06
SLXDSUAT	-	05
SLXDSUNT	000008	3E
SLXDSDVAT	-	04
SLXDSDVOL	000006	38
SLXDSDVMM	-	8051
SLXEND	000008	B0
SLXEXLM0	000004	80
SLXEXLM1	000004	84
SLXEXLM2	000004	88
SLXGID	000003	00
SLXGTEXT	000125	03
SLXHID	000003	00
SLXHHL	-	14
SLXHSCV	000002	A0
SLXL	-	B0
SLXLCMPF	000001	24
SLXLEJPS	000008	1C
SLXLHHBT	000004	0C
SLXLHNAM	000008	04

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
SLXLID	000003	00
SLXLLBAL	-	02
SLXLLBNL	-	03
SLXLLBNS	-	04
SLXLLBSL	-	01
SLXLOCKD	-	20
SLXLQNAM	000008	14
SLXLRSTM	000004	10
SLXLSCLB	000001	25
SLXLSMF	000001	26
SLXMACS	000001	04
SLXMADJI	000001	07
SLXMADJN	000001	06
SLXMANUL	-	10
SLXMDISC	-	08
SLXMFCEL	000004	14
SLXMID	000003	00
SLXMLSM	000001	05
SLXMNSCR	000004	0C
SLXMSTAT	000001	03
SLXMSTT	-	803A
SLXMTCEL	000004	10
SLXNORSP	-	28
SLXNTCB	-	1C
SLXPEOFF	000004	10
SLXQACNT	000004	30
SLXQDEAC	000001	04
SLXQDECU	000002	06
SLXQDELS	000001	05
SLXQDID	000003	00
SLXQDRCT	000004	2C
SLXQDRT	000008	08
SLXQID	000004	00
SLXQJTC	000004	08



<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
SLXQJTD	-	00
SLXQJTL	-	18
SLXQJTN	-	FFFF
SLXQJTS	000002	0C
SLXQJTT	000004	04
SLXQJTV	000006	0E
SLXQLCAP	000004	7C
SLXQLCNT	000004	34
SLXQMDR	000004	28
SLXQUCSA	000004	74
SLXRBADL	-	2C
SLXRBADP	-	08
SLXRBOK	-	3C
SLXREOV	-	34
SLXRFAIL	-	10
SLXRIERR	-	0C
SLXRNAUT	-	18
SLXRNHSC	-	14
SLXRNVC	-	30
SLXROK	-	00
SLXRPLY	000001	00
SLXRVNV	-	38
SLXRWARN	-	04
SLXSANF	-	8036
SLXSCAPL	-	20
SLXSCFGL	-	A8
SLXSDRVL	-	10
SLXSFUL	-	8050
SLXSID	000004	00
SLXSLSML	-	18
SLXSMSG	-	80
SLXSRC	000004	08
SLXSTPE	-	EE
SLXSTPK	-	00

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
SLXSTPL	-	10
SLXSTPN	-	FF
SLXSTPS	000001	09
SLXSTPT	000004	04
SLXSTPY	000001	08
SLXSVOLL	-	28
SLXSVOLN	-	28
SLXTINTR	-	8032
SLXTPROT	-	24
SLXTRNF	-	8040
SLXVACS	000001	0B
SLXVCEL	-	01
SLXVCODE	-	07
SLXVDATD	000004	1C
SLXVDATI	000004	10
SLXVDATL	000004	14
SLXVDRIV	000002	0D
SLXVDRV	-	02
SLXVDSC	-	08
SLXVERR	-	02
SLXVERS	000001	04
SLXVID	000003	00
SLXVILB	-	80
SLXVLC	000001	0A
SLXVLOC	000005	0B
SLXVLSM	000001	0C
SLXVMAL	-	10
SLXVMDFL	-	01
SLXVMED	000008	20
SLXVMLMU	-	10
SLXVMNT	-	04
SLXVMUNR	-	04
SLXVMVLA	-	02
SLXVOHST	-	40

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
SLXVPNL	000003	0D
SLXVSCNT	000004	18
SLXVSCR	-	20
SLXVSER	000006	04
SLXVSTA	000001	03
SLXVSTA2	000001	0F
SLXVTSSN	000008	10
SLXVTV	-	01
SLXVUNK	-	00
SLXXCAPL	000004	58
SLXXCAPN	000004	50
SLXXCAPO	000004	54
SLXXCFGL	000004	1C
SLXXCFGN	000004	14
SLXXCFG0	000004	18
SLXXDRVL	000004	34
SLXXDRVN	000004	2C
SLXXDRVO	000004	30
SLXXLSML	000004	40
SLXXLSMN	000004	38
SLXXLSMO	000004	3C
SLXXMSG1	000004	4C
SLXXMSGN	000004	44
SLXXMSG0	000004	48
SLXXQDSL	000004	7C
SLXXQDSN	000004	74
SLXXQDSO	000004	78
SLXXQJTL	000004	64
SLXXQJTN	000004	5C
SLXXQJTO	000004	60
SLXXSDL	-	B0
SLXXSTPL	000004	70
SLXXSTPN	000004	68
SLXXSTPO	000004	6C

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
SLXXVCIL	000004	88
SLXXVCIN	000004	80
SLXXVCIO	000004	84
SLXXVOLL	000004	28
SLXXVOLN	000004	20
SLXXVOLO	000004	24
SLXZDEFP	000004	60
SLXZDEFS	000004	64
SLXZDISM	000004	54
SLXZEJCT	000004	58
SLXZEJC1	000004	70
SLXZGSCR	000004	48
SLXZMDM	000004	4C
SLXZMOVE	000004	6C
SLXZQCAP	000004	8C
SLXZQDRL	000004	40
SLXZQDRV	000004	3C
SLXZQDSN	000004	98
SLXZQEJT	000004	90
SLXZQSCR	000004	50
SLXZQVOL	000004	44
SLXZSCR	000004	5C
SLXZSETO	000004	68
SLXZSTOP	000004	94
SLXZVOL	000004	38

## ACSINT Request DSECT

The ACSINT is used to pass information between the TMS and the ACS service machine. It is built by the ACSRQ macro when DSECT=YES is specified and is sent using IUCV.

**Note** – The following DSECT includes some parameters that are not supported by the VM Client. They are included for completeness and compatibility only.

**TABLE 7-3** ACSINT Record Format

Dec	Hex	Type	Length	Label	Description
ACSINT - TMS INTERFACE REQUEST PARAMETER LIST					
FUNCTION: THIS DEFINES THE DATA PASSED TO THE TMS VIA IUCV IT DEFINES THE VARIOUS FUNCTIONS THAT CAN BE REQUESTED AND THE STRUCTURE PASSED.					
0	(0)	STRUCTURE		ACSINT	TMS INTERFACE PARAMETER LIST:
0	(0)	CHARACTER	4	ACSIHDR	PARAMETER LIST IDENTIFIER.
'ACSI'	(C1C3E2C9)	CHAR CONST		ACSIID	
4	(4)	SIGNED-FWORD	4	ACSILEN	PARAMETER LIST LENGTH.
8	(8)	A-ADDR	1	ACSIVER	PARAMETER LIST VERSION NUMBER.
7	(07)	CONST		ACSIVN	CURRENT VERSION.
9	(9)	A-ADDR	1	ACSIRT	FUNCTION CODE:
0	(00)	CONST		ACSINOOP	NOOP - NO OPERATION.
1	(01)	CONST		ASCIRS01	RESERVED - MVS ONLY.
2	(02)	CONST		ASCIRS02	RESERVED - MVS ONLY.
3	(03)	CONST		ASCIRS03	RESERVED - MVS ONLY.
4	(04)	CONST		ASCISTOP	STOP - STOP AN INTERRUPTABLE PGMI TASK.
5	(05)	CONST		ACSIOPR	SETOPER - SET OPERATOR INTERACTION MODE.
20	(14)	CONST		ACSIQCNF	QCONFIG - RETURN CONFIGURATION SUMMARY.
21	(15)	CONST		ACSIQDRV	QDRIVES - RETURN DRIVE AND LSM INFO.
22	(16)	CONST		ACSIQDRL	QDRLIST - RETURN DRIVE INFORMATION, X .
23	(17)	CONST		ACSIQSCR	QSCRATCH - RETURN SCRATCH COUNT INFO.
24	(18)	CONST		ACSIQVOL	QVOLUME - RETURN VOLUME INFORMATION.
25	(19)	CONST		ACSIQJCT	QEJECT - RETURN EJECT STATUS.

**TABLE 7-3** ACSINT Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
26	(1A)	CONST		ACSIQCAP	QCAP - QUERY CAP STATUS.
27	(1B)	CONST		ACSIQVLU	QVOLUME - RETURN MOUNTED VOLUMES.
28	(1C)	CONST		ACSIQRQS	QREQUEST - RETURN PENDING REQUESTS.
29	(1D)	CONST		ACSIQDSN	QDSN - QUERY DATASET.
40	(29)	CONST		ACSIMNT	MOUNT - MOUNT VOLUME.
41	(29)	CONST		ACSIMOVE	MOVE - MOVE A CARTRIDGE TO AN LSMID X.
42	(2A)	CONST		ACSIDSM	DISMOUNT - DISMOUNT VOLUME.
43	(2B)	CONST		ACSIEJCT	EJECT - EJECT A VOL FROM THE LIBRARY.
60	(3C)	CONST		ACSISSCR	SELSCR - SELECT A SCRATCH VOLUME.
61	(3D)	CONST		ACSISCRA	SCRATCH - CHANGE VOL STATUS TO 'SCRATCH'.
62	(3E)	CONST		ACSIUNSC	UNSCRATCH- CHANGE VOLUME STATUS TO NOT X.
63	(3F)	CONST		ACSIDSCR	DEFSCR - SPECIFY NO. OF SCRATCH POOLS.
64	(40)	CONST		ACSIDPOL	DEFPOOL - SPECIFY A SCRATCH POOL'S VOLSER RANGE.
10	(A)	BITSTRING	1	ACSIFLG1	FLAG BYTE 1: (PGMI CONTROL 1) 1
				ACSIF180	RESERVED MVS OPTION=SYNC
				ACSIVUSR	USER= SPECIFIED; ACSUSER CONTAINS NAME
				ACSIMANO	DIALOG=OFF SPECIFIED.
				ACSIWTOR	DIALOG=ON SPECIFIED.
				ACSINDEL	NOTIFY=INDEL SPECIFIED.
				ACSINNDL	NOTIFY=NOINDEL SPECIFIED.
				ACSIACC1	ACCT1= SPECIFIED ACCOUNTING TOKEN.
				ACSIACC2	ACCT2= SPECIFIED ACCOUNTING TOKEN.
11	(B)	HEXSTRING	1	ACSIFLG2	FLAG BYTE 2: (PGMI CONTROL 2).
12	(C)	BITSTRING	1	ACSIFLG3	FLAG BYTE 3: (MOVEMENT CONTROL 1).

**TABLE 7-3** ACSINT Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
13	(D)	BITSTRING	1		1... .... X'80'
				ACSIVHST	ACSIHOST CONTAINS HOST_ID.
					.1.. .... X'40'
				ACSIVACS	ACSIACS CONTAINS AN ACSID.
					..1. .... X'20'
				ACSIVLSM	ACSILSMI CONTAINS LSMID.
					...1 .... X'10'
				ACSIVCAP	ACSICAP CONTAINS CAP_ID.
					.... 1... X'08'
				ACSIF308	RESERVED FUTURE USE.
					.... .1.. X'04'
				ACSIVVOL	ACSIVOLS CONTAINS A VOLSER.
					.... .1. X'02'
14	(E)	HEXSTRING	1	ACSIVLST	ACSILOFF CONTAINS LIST OFFSET.
					.... ...1 X'01'
				ACSIVCNT	ACSICNT CONTAINS LIST COUNT.
					1... .... X'80'
				ACSIVTLM	ACSITLSM CONTAINS TO LSM_ID.
					.1.. .... X'40'
				ACSIVTPN	ACSITPAN CONTAINS TO PANEL.
					..1. .... X'20'
				ACSIF420	RESERVED FUTURE USE.
					...1 .... X'10'
				ACSIF410	RESERVED FUTURE USE.
					.... 1... X'08'
				ACSIVPAN	ACSIPAN CONTAINS PANEL NUMBER
15	(F)	BITSTRING	1		.... .1.. X'04'
				ACSIVROW	ACSIROW CONTAINS ROW NUMBER.
					.... .1. X'02'
				ACSIVCOL	ACSICOL CONTAINS COLUMN NUMBER.
					.... ...1 X'01'
				ACSIVDRV	ACSIDRV CONTAINS DRIVE DEVICE NUMBER.
					1... .... X'80'
				ACSIFLG5	FLAG BYTE 5: (MOVEMENT CONTROL 3).
					.1.. .... X'40'
				ACSIPROT	PROTECT=YES SPECIFIED.
				ACSISCR	SCRATCH=YES SPECIFIED.
					..1. .... X'20'
				ACSIVSCP	ACSIPOOL CONTAINS SCRATCH POOL NUMBER.
					...1 .... X'10'
				ACSISUBN	ACSISUBP CONTAINS SUBPOOL NAME.
					.... 1... X'08'
				ACSIVTKN	ACSITKNO CONTAINS TOKEN NUMBER.
					.... .1.. X'04'
				ACSIVTXT	ACSITEXT CONTAINS TEXT STRING.

**TABLE 7-3** ACSINT Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
		.... ..1. X'02'		ACSIF602	RESERVED FUTURE USE.
		.... ..1 X'01'		ACSIF601	RESERVED FUTURE USE.
16	(10)	HEXSTRING	1	ACSIFLG7	FLAG BYTE 7: (MISCELLANOUS CONTROL 2).
17	(11)	A-ADDR	1	ACSILABT	SCRATCH LABEL TYPE LTYPE= PARAMETER.
		.... .... X'00'		ACSILLDT	LDT (LIBRARY DEFAULT TYPE).
		.... ..1 X'01'		ACSILSL	SL (STANDARD LABEL).
		.... ..1. X'02'		ACSILAL	AL (ANSI LABEL).
		.... ..11 X'03'		ACSILNL	NL (NON-LABELED).
		.... ..1.. X'04'		ACSILNS	NSL (NON-STANDARD LABEL).
18	(12)	CHARACTER	8	ACSIUSER	USER NAME USED TO ASSOCIATE CONSOLE ID.
26	(1A)	CHARACTER	8	ACSIACT1	ACCOUNTING TOKEN 1.
34	(22)	CHARACTER	8	ACSIACT2	ACCOUNTING TOKEN 2.
42	(2A)	HEXSTRING	2	-RESERVED-	RESERVE SLACK BYTES.
44	(2C)	SIGNED-FWORD	4	-RESERVED-	RESERVED MVS.
48	(30)	HEXSTRING	4	ACSITKNO	PASS THROUGH TOKEN NUMBER.
52	(34)	A-ADDR	4	-RESERVED-	RESERVED MVS.
56	(38)	SIGNED-HWORD	2	ACSICNT	COUNT FROM COUNT= PARAMETER.
58	(3A)	SIGNED-HWORD	2	ACSILOFF	OFFSET, FROM START OF PARAMETER LIST, TO START OF THE ELEMENT LIST AREA.
60	(3C)	CHARACTER			CHARACTER 6 ACSIVOLS VOLSER FROM VOLSER= PARAMETER.
66	(42)	CHARACTER	8	ACSIHOST	ASSOCIATED HOST FROM HOSTID= PARAMETER.
74	(4A)	AREA	4	ACSIALC	ACS / LSM / CAP IDENTIFICATION.
74	(4A)	HEXSTRING	1	ACSIACS	ACS ID NUMBER (AA).
74	(4A)	HEXSTRING	2	ACSILSMI	LSM ID NUMBER (AA0L).
74	(4A)	HEXSTRING	4	ACSICAP	CAP ID NUMBER (AA0LCC00).
74	(4A)	HEXSTRING	3	-RESERVED-	
77	(4D)	HEXSTRING	1	-RESERVED-	NOT IMPLEMENTED (ALWAYS X'00') .



**TABLE 7-3 ACSINT Record Format (Continued)**

Dec	Hex	Type	Length	Label	Description
78	(4E)	HEXSTRING	1	ACSIPAN	PANEL FROM PAN= PARAMETER.
79	(4F)	HEXSTRING	1	ACSIROW	ROW FROM ROW= PARAMETER.
80	(50)	HEXSTRING	1	ACSICOL	COLUMN FROM COL= PARAMETER.
81	(51)	HEXSTRING	1	ACSITPAN	TO PANEL FROM TOPAN= PARAMETER.
82	(52)	HEXSTRING	2	ACSITLSM	TO LSM FROM TOLSM= PARAMETER.
84	(54)	A-ADDR	2	ACSIDRIV	DRIVE DEVICE NUMBER FROM DRIVE= PARAMETER.
86	(56)	A-ADDR	1	ACSIPOOL	SCRATCH POOL NUMBER, SCRPOOL= PARAMETER.
87	(57)	CHARACTER	32	ACSITEXT	TEXT ASSOCIATED WITH REQUEST.
119	(77)	CHARACTER	13	ACSISUBP	NAME FROM SUBPOOL= KEYWORD.
TAPEREQ INPUT KEY VALUES. THE FOLLOWING VALUES ARE USED AS THE KEY IN SEARCHING THE CURRENT TAPEREQ PARAMETER FILE.					
132	(84)	BITSTRING	1	ACSIFLG8	FLAG BYTE 8: (TAPEREQ CONTROL 1).
		1... .... X'80'		ACSIFJOB	ACSIJOBN PRESENT.
		.1.. .... X'40'		ACSIFSTP	ACSISTEP PRESENT.
		..1. .... X'20'		ACSIFPGM	ACSIPGMN PRESENT.
		...1 .... X'10'		ACSIFDSN	ACSIDSN PRESENT.
		.... 1... X'08'		ACSIFEXP	ACSIEXPD PRESENT.
		.... .1.. X'04'		ACSIFRET	ACSIRETP PRESENT.
		.... ..1. X'02'		ACSIFVOL	ACSIVOLT PRESENT.
133	(85)	OFFSET		ACSITRI	TAPEREQ INPUT VALUES.
133	(85)	CHARACTER	8	ACSIJOBN	STRING TO MATCH TAPEREQ JOBNAME VALUE.
141	(8D)	CHARACTER	8	ACSISTEP	STRING TO MATCH TAPEREQ STEPNAME VALUE.
149	(95)	CHARACTER	8	ACSIPGMN	STRING TO MATCH TAPEREQ PROGNAME VALUE.
157	(9D)	CHARACTER	44	ACSIDSN	STRING TO MATCH TAPEREQ DSN VALUE.
201	(C0)	HEXSTRING	3	ACSIEXPD	VALUE TO MATCH TAPEREQ EXPDT VALUE

**TABLE 7-3** ACSINT Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
204	(CC)	HEXSTRING	2	ACSIRETP	VALUE TO MATCH TAPEREQ RETPD VALUE.
206	(CE)	CHARACTER	1	ACSIVOLT	STRING TO MATCH TAPEREQ VOLTYPE VALUE .
74	(4A)	LENGTH		ACSITRIL	
TAPEREQ OUTPUT VALUES. THE FOLLOWING VALUES ARE USED AS OVERRIDE (OR SPECIFIC) VALUES TO THE VALUES FOUND IN THE CURRENT TAPEREQ PARAMETER FILE.					
207	(CF)	BITSTRING	1	ACSIFLG9	FLAG BYTE 9: (TAPEREQ CONTROL 2).
		1... .... X'80'		ACSIFREC	ACSIRECT PRESENT.
		.1.. .... X'40'		ACSIFMED	ACSIMED PRESENT.
208	(D0)	CHARACTER	8	ACSIRECT	RECORDING TECHNIQUE.
216	(D8)	CHARACTER	8	ACSIMED	MEDIA.
224	(E0)	HEXSTRING	256	-RESERVED-	RESERVED FOR FUTURE PARM EXPANSION.
480	(1E0)	AREA	8	-RESERVED-	ALIGNMENT.
480	(1E0)	LENGTH		ACSIHLN	LENGTH OF FIXED AREA.
480	(1E0)	AREA	1	ACSILIST	ELEMENT LIST DESIGNATED BY LIST= PARAMETER BEGINS HERE (FIELD ACSICNT CONTAINS THE NUMBER OF ELEMENTS IN THIS LIST).
2	(02)	CONST		ACSILDRL	LENGTH OF EACH ELEMENT (DRIVE DEVICE NUMBER) IN THE LIST USED BY THE QDRLIST FUNCTION.
1500	(5DC)	CONST		ACSIMDRL	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE QDRLIST FUNCTION.
6	(06)	CONST		ACSILVSL	LENGTH OF EACH ELEMENT (VOLSER) IN THE LIST USED BY THE QVOLUME FUNCTION.
500	(1F4)	CONST		ACSIMVSL	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE QVOLUME AND EJECT.
12	(0C)	CONST		ACSILPOL	LENGTH OF EACH ELEMENT (VOLSER RANGE PAIR) IN THE LIST USED BY THE DEFPOOL FUNCTION.

**TABLE 7-3** ACSINT Record Format (Continued)

Dec	Hex	Type	Length	Label	Description
250	(FA)	CONST		ACSIMPOL	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE DEFPOOL FUNCTION.
4	(04)	CONST		ACSILCAP	LENGTH OF EACH ELEMENT (CAP IDENTIFIER) IN THE LIST USED BY THE QCAP FUNCTION.
500	(1F4)	CONST		ACSIMCAP	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE QCAP FUNCTION.
4	(04)	CONST		ACSILTOK	LENGTH OF EACH ELEMENT (UNIQUE TOKEN) IN THE LIST USED BY THE QCAP FUNCTION.
500	(1F4)	CONST		ACSIMTOK	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE QJECT/STOP FUNCTIONS.

**Cross Reference**

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>	<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
ACSIACC1	-	02	ACSIFRET	-	04
ACSIACC2	-	01	ACSIFSTP	-	40
ACSIACS	000001	4A	ACSIFVOL	-	02
ACSIACT1	000008	1A	ACSIF180	-	80
ACSIACT2	000008	22	ACSIF308	-	08
ACSIALC	000004	4A	ACSIF410	-	10
ACSICAP	000004	4A	ACSIF420	-	20
ACSICNT	000002	38	ACSIF601	-	01
ACSICOL	000001	50	ACSIF602	-	02
ACSIDPOL	-	40	ACSIHDR	000004	00
ACSIDRIV	000002	54	ACSIHLN	-	1E0
ACSIDSCR	-	3F	ACSIHOST	000008	42
ACSIDSM	-	2A	ACSIID	-	'CVAL'
ACSIDSN	000044	9D	ACSIJOB	000008	85
ACSIEJCT	-	2B	ACSILABT	000001	11
ACSIEXPD	000003	C9	ACSILAL	-	02
ACSIFDSN	-	10	ACSILCAP	-	04
ACSIFEXP	-	08	ACSILDRL	-	02
ACSIFJOB	-	80	ACSILIST	000001	1E0
ACSIFLG1	000001	0A	ACSILLDT	-	00
ACSIFLG2	000001	0B	ACSILNL	-	03
ACSIFLG3	000001	0C	ACSILNS	-	04
ACSIFLG4	000001	0D	ACSILOFF	000002	3A
ACSIFLG5	000001	0E	ACSILPOL	-	0C
ACSIFLG6	000001	0F	ACSILSL	-	01
ACSIFLG7	000001	10	ACSILSMI	000002	4A
ACSIFLG8	000001	84	ACSILTOK	-	04
ACSIFLG9	000001	CF	ACSILVSL	-	06
ACSIFMED	-	40	ACSIMANO	-	20
ACSIFPGM	-	20	ACSIMCAP	-	1F4
ACSIFREC	-	80	ACSIMDRL	-	5DC
ACSIMED	000008	D8	ACSIVTPN	-	40
ACSIMNT	-	28	ACSIVTXT	-	04
ACSIMOVE	-	29	ACSIVUSR	-	40
ACSIMPOL	-	FA	ACSIVVOL	-	04
ACSIMTOK	-	1F4	ACSIWTOR	-	10

Name	Len	Offset Value	Name	Len	Offset Value
ACSIMVSL	-	1F4	ACSIROW	000001	4F
ACSINDEL	-	08	ACSIRS01	-	01
ACSINNDL	-	04	ACSIRS02	-	02
ACSINOOP	-	00	ACSIRS03	-	03
ACSIPAN	000001	4E	ACSIRT	000001	09
ACSIPGMN	000008	95	ACSISCR	-	40
ACSIPOOL	000001	56	ACSISCRA	-	3D
ACSIPROT	-	80	ACSISOPR	-	05
ACSIQCAP	-	1A	ACSISSCR	-	3C
ACSIQCNF	-	14	ACSISTEP	000008	8D
ACSIQDRL	-	16	ACSISTOP	-	04
ACSIQDRV	-	15	ACSISUBN	-	10
ACSIQDSN	-	1D	ACSISUBP	000013	77
ACSIQEJT	-	19	ACSITEXT	000032	57
ACSIQRQS	-	1C	ACSITKNO	000004	30
ACSIVDRV	-	01	ACSITLSM	000002	52
ACSIVER	000001	08	ACSITPAN	000001	51
ACSIVHST	-	80	ACSITRI	-	85
ACSIVLSM	-	20	ACSITRIL	-	4A
ACSIVLST	-	02	ACSIUNSC	-	3E
ACSIVN	-	07	ACSIUSER	000008	12
ACSIVOLS	000006	3C	ACSIVACS	-	40
ACSIVOLT	000001	CE	ACSIVCAP	-	10
ACSIVPAN	-	08	ACSIVCNT	-	01
ACSIVROW	-	04	ACSIVCOL	-	02
ACSIVSCP	-	20			
ACSIVTKN	-	08			
ACSIVTLM	-	80			

**IUB Record Format*****IUB - IUCV Request Block***

The IUB describes an outstanding IUCV request resulting from an IUCV operation. Both the request and the final status are contained in the IUB data structure. The IUB is an IUCV counterpart to the 'IOBLOK' structure used by device management.

**TABLE 7-4** IUB Record Format

Name	Len	Offset Value	Name	Len	Offset Value
ACSIACC1	-	02	ACSIFRET	-	04
ACSIACC2	-	01	ACSIFSTP	-	40
ACSIACS	000001	4A	ACSIFVOL	-	02
ACSIACT1	000008	1A	ACSIF180	-	80
ACSIACT2	000008	22	ACSIF308	-	08
ACSIALC	000004	4A	ACSIF410	-	10
ACSICAP	000004	4A	ACSIF420	-	20
ACSICNT	000002	38	ACSIF601	-	01
ACSICOL	000001	50	ACSIF602	-	02
ACSIDPOL	-	40	ACSIHDR	000004	00
ACSIDRIV	000002	54	ACSIHLN	-	1E0
ACSIDSCR	-	3F	ACSIHOST	000008	42
ACSIDSM	-	2A	ACSIID	-	'CVAL'
ACSIDSN	000044	9D	ACSIJOB	000008	85
ACSIJECT	-	2B	ACSILABT	000001	11
ACSIEXPD	000003	C9	ACSILAL	-	02
ACSIFDSN	-	10	ACSILCAP	-	04
ACSIFEXP	-	08	ACSILDRL	-	02
ACSIFJOB	-	80	ACSILIST	000001	1E0
ACSIFLG1	000001	0A	ACSILLDT	-	00
ACSIFLG2	000001	0B	ACSILNL	-	03
ACSIFLG3	000001	0C	ACSILNS	-	04
ACSIFLG4	000001	0D	ACSILOFF	000002	3A
ACSIFLG5	000001	0E	ACSILPOL	-	0C
ACSIFLG6	000001	0F	ACSILSL	-	01
ACSIFLG7	000001	10	ACSILSMI	000002	4A
ACSIFLG8	000001	84	ACSILTOK	-	04
ACSIFLG9	000001	CF	ACSILVSL	-	06
ACSIFMED	-	40	ACSIMANO	-	20
ACSIFPGM	-	20	ACSIMCAP	-	1F4

**TABLE 7-4** IUB Record Format (Continued)

Name	Len	Offset Value	Name	Len	Offset Value
ACSIFREC	-	80	ACSIMDRL	-	5DC
ACSIMED	000008	D8	ACSIVTPN	-	40
ACSIMNT	-	28	ACSIVTXT	-	04
ACSIMOVE	-	29	ACSIVUSR	-	40
ACSIMPOL	-	FA	ACSIVVOL	-	04
ACSIMTOK	-	1F4	ACSIWTOR	-	10
ACSIMVSL	-	1F4	ACSIROW	000001	4F
ACSINDEL	-	08	ACSIRS01	-	01
ACSINNDL	-	04	ACSIRS02	-	02
ACSINOOP	-	00	ACSIRS03	-	03
ACSIPAN	000001	4E	ACSIRT	000001	09
ACSIPGMN	000008	95	ACSISCR	-	40
ACSIPOOL	000001	56	ACSISCRA	-	3D
ACSIPROT	-	80	ACSISOPR	-	05
ACSIQCAP	-	1A	ACSISSCR	-	3C
ACSIQCNF	-	14	ACSISTEP	000008	8D
ACSIQDRL	-	16	ACSISTOP	-	04
ACSIQDRV	-	15	ACSISUBN	-	10
ACSIQDSN	-	1D	ACSISUBP	000013	77
ACSIQEJT	-	19	ACSITEXT	000032	57
ACSIQRQS	-	1C	ACSITKNO	000004	30
ACSIVDRV	-	01	ACSITLSM	000002	52
ACSIVER	000001	08	ACSITPAN	000001	51
ACSIVHST	-	80	ACSITRI	-	85
ACSIVLSM	-	20	ACSITRIL	-	4A
ACSIVLST	-	02	ACSIUNSC	-	3E
ACSIVN	-	07	ACSIUSER	000008	12
ACSIVOLS	000006	3C	ACSIVACS	-	40
ACSIVOLT	000001	CE	ACSIVCAP	-	10
ACSIVPAN	-	08	ACSIVCNT	-	01
ACSIVROW	-	04	ACSIVCOL	-	02
ACSIVSCP	-	20			
ACSIVTKN	-	08			
ACSIVTLM	-	80			

**Cross Reference**

<b>Name</b>	<b>Len</b>	<b>Offset Value</b>	<b>Name</b>	<b>Len</b>	<b>Offset Value</b>
IUBCC	000001	31	IUBPARM5	000008	70
IUBCONN	-	10	IUBQ	000016	00
IUBDABQ	000016	10	IUBREAD	-	80
IUBDAVL	000001	20	IUBREJECT	-	08
IUBDCBPT	000004	44	IUBREPLY	-	20
IUBECBKY	000001	21	IUBSEND	-	40
IUBBECBT	000004	34	IUBSENT	-	80
IUBEXT	000040	78	IUBTASK	000004	28
IUBEXT1	000008	78			
IUBEXT2	000008	80			
IUBEXT3	000008	88			
IUBEXT4	000008	90			
IUBEXT5	000008	98			
IUBEYE	000004	0C			
IUBFLG1	000001	30			
IUBFLG2	000001	32			
IUBHCOMM	-	20			
IUBIRT	000004	2C			
IUBIUBPT	-	24			
IUBLEN	-	A0			
IUBLOK	-	08			
IUBNPOST	-	40			
IUBORGID	000004	3C			
IUBPARML	000040	50			
IUBPARM1	000008	50			
IUBPARM2	000008	58			
IUBPARM3	000008	60			
IUBPARM4	000008	68			



---

## MEDia, RECtech, and MODel Values

### Overview

This appendix provides values for MEDia, RECtech, and MODel parameters. These parameters are used to specify transport and media characteristics. They are specified in various VM Client commands and TMI requests.

**Note –**

- The SL8500 library supports only the T9840A/B/C/D, T9940B, LTO, SDLT, and T10000A/B/C/D media types and recording techniques.
- The SL3000 library supports only the T9840C/D, LTO, SDLT, and T10000A/B/C/D media types and recording techniques.
- LTO and SDLT drives are not supported in an MVS environment. These drives are recognized by the HSC but are accessible only to open systems clients using LibraryStation.

## Media Type (MEDia)

The Media type is used to specify the desired type of media to be used for a data set. It is specified in the following VM Client commands and TMI requests:

- MOUNT command
- QDRLIST TMI request
- QSCRATCH TMI request
- SELSCR TMI request

The following table describes valid MEDia types:

**TABLE A-1** Media Types

Media Type	Description
LONGitud	Indicates any standard or enhanced (ECART) capacity cartridges.
ZLONGI	Indicates standard, enhanced (ECART), or extended-enhanced (ZCART) capacity cartridges.
Standard	Indicates a standard length, 3480 cartridge. It can be read on any longitudinal drive (4480, 4490, 9490, or 9490EE). Data can be written in 36-track mode on a 4490, 9490, or 9490EE drive but cannot be read on an 18-track (4480) drive. Synonyms include CST, MEDIA1, STD, 1, and 3480.
ECART	Indicates a 3490E, extended capacity cartridge. It can be used only on a 36-track drive (4490, 9490, or 9490EE). Synonyms include E, ECCST, ETAPE, Long, MEDIA2, and 3490E.
ZCART	Indicates a 3490E, extended-enhanced capacity cartridge. It can be used only on a 9490EE drive. ZCART can be abbreviated as Z.
Virtual	Indicates a VTV (Virtual Tape Volume) mounted on a VTD (Virtual Tape Drive).
HELical	Indicates a helical cartridge. A helical cartridge can be used only on RedWood drives. The following subtypes and abbreviations specify a helical cartridge:  <b>DD3</b> indicates any DD3A, DD3B, or DD3C helical cartridge. <b>DD3A</b> or <b>A</b> indicates a helical cartridge with a 10GB media capacity. <b>DD3B</b> or <b>B</b> indicates a helical cartridge with a 25GB media capacity. <b>DD3C</b> or <b>C</b> indicates a helical cartridge with a 50GB media capacity.  The seventh position in the external label is encoded with the cartridge type (i.e., A, B, or C).
STK1	Indicates any T9840 cartridge.
STK1R	Indicates a T9840 data cartridge. The media indicator in the external label is encoded with the cartridge type (R). STK1R can be abbreviated as R.  T9840 cartridge media capacities are 20GB (T9840A and T9840B), 40GB (T9840C), or 75GB (T9840D).
STK1U	Indicates a T9840A, T9840B, or T9840C cleaning cartridge. STK1U can be abbreviated as U.
STK1Y	Indicates a T9840D cleaning cartridge. STK1Y can be abbreviated to Y.
STK2	Indicates any T9940 cartridge.

**TABLE A-1** Media Types (Continued)

<b>Media Type</b>	<b>Description</b>
STK2P	Indicates a T9940 data cartridge. STK2P can be abbreviated as P. T9940 cartridge media capacities are 60GB (T9940A) or 200GB (T9940B).
STK2W	Indicates a T9940 cleaning cartridge. STK2W can be abbreviated as W.
LTO-10G	Indicates an LTO 10GB capacity cartridge.
LTO-35G	Indicates an LTO 35GB capacity cartridge.
LTO-50G	Indicates an LTO 50GB capacity cartridge.
LTO-100G	Indicates an LTO 100GB capacity cartridge.
LTO-200G	Indicates an LTO 200GB capacity cartridge.
LTO-400G	Indicates an LTO 400GB capacity cartridge.
LTO-400W	Indicates an LTO Generation 3 WORM cartridge.
LTO-800G	Indicates an LTO 800GB capacity cartridge.
LTO-800W	Indicates an LTO Generation 4 WORM cartridge.
LTO-1.5T	Indicates an LTO 1.5TB capacity cartridge.
LTO-1.5W	Indicates an LTO Generation 5 WORM cartridge.
LTO-2.5T	Indicates an LTO 2.5TB capacity cartridge.
LTO-2.5W	Indicates an LTO Generation 6 WORM cartridge.
LTO-CLN1	Indicates an LTO type 1 cleaning cartridge.
LTO-CLN2	Indicates an LTO type 2 cleaning cartridge.
LTO-CLNU	Indicates an LTO universal cleaning cartridge.
SDLT	Indicates an SDLT 160GB cartridge.
SDLT-2	Indicates an SDLT 125GB cartridge.
SDLT-S1	Indicates an SDLT Generation S1 cartridge.
SDLT-S2	Indicates an SDLT Generation S2 cartridge.
SDLT-S3	Indicates an SDLT Generation S3 cartridge.
SDLT-S4	Indicates an SDLT Generation S4 cartridge.
SDLT-4	Indicates an SDLT Generation 4 cartridge.
T10000T1	Indicates a full-capacity 500GB T10000A or 1TB T10000B cartridge. T10000T1 can be abbreviated as T1.
T10000TS	Indicates a smaller-capacity 120GB T10000A or 240GB T10000B cartridge. T10000TS can be abbreviated as TS.
T10000CT	Indicates a T10000A or T10000B cleaning cartridge. T10000CT can be abbreviated as CT.
T10000T2	Indicates a full-capacity 5TB T10000C or 8TB T10000D cartridge. T10000T2 can be abbreviated as T2.
T10000TT	Indicates a smaller-capacity 1TB T10000C or 1.6TB T10000D cartridge. T10000TT can be abbreviated as TT.
T10000CL	Indicates a T10000A, T10000B, T10000C, or T10000D cleaning cartridge. T10000CL can be abbreviated as CL.

**Note** – T10000C and T10000D drives can read T10000T1 or T10000TS media but cannot write to that media. T10000C and T10000D drives can only write to T10000T2 or T10000TT media.

When the MEDia parameter is not specified, a default is chosen based on the value of the RECtech parameter. The following table shows default values used if MEDia is omitted:

**TABLE A-2** Media Type Defaults

RECtech Entered	MEDia Default
18track	Standard
36track, 36Atrack, 36Btrack	LONGItud
36Ctrack	ZLONGI
LONGItud	LONGItud
DD3, Helical	DD3A
STK1R, STK1R34, STK1R35 STK1RA, STK1RA34, STK1RA35 STK1RB, STK1RB34, STK1RB35 STK1RAB, STK1RAB34, STK1RAB35 STK1RC, STK1RC34, STK1RC35 STK1RD, STK1RDE, STK1RDN, STK1RD34, STK1RD35, STK1RDE4, STK1RDE5	STK1R
STK2P, STK2P34, STK2P35 STK2PA, STK2PA34, STK2PA35 STK2PB, STK2PB34, STK2PB35	STK2P
T10K, T10KN, T10KE T10KA, T10KAN, T1A34, T1A35, T10KAE, T1AE34, T1AE35 T10KB, T10KBN, T1B34, T1B35, T10KBE, T1BE34, T1BE35 T10KC, T10KCN, T1C34, T1C35, T10KCE, T1CE34, T1CE35 T10KD, T10KDN, T1D34, T1D35, T10KDE, T1DE34, T1DE35	T10000T1*

\* T10000C and T10000D drives can read T10000T1 or T10000TS media but cannot write to that media. T10000C and T10000D drives can only write to T10000T2 or T10000TT media.

# Recording Technique (RECtech)

The recording technique, or RECtech, is used to specify the method used to record data tracks on the tape surface for the desired data set. It is specified in the following VM Client TMI requests:

- QDRLIST
- QSCRATCH
- SELSCR

The following table describes valid recording techniques:

**TABLE A-3** Recording Techniques

Recording Technique	Description
LONGItud	Indicates any device that uses longitudinal recording. Devices include 4480, 4490, 9490, and 9490EE drives.
18track	Indicates a 4480 drive.
36track	Indicates a 4490, 9490, or 9490EE drive (any device that records in 36-track mode).
36Atrack	Indicates a 4490 (Silverton) drive.
36Btrack	Indicates a 9490 (Timberline) drive.
36Ctrack	Indicates a 9490EE drive.
HELical	Indicates a device using helical recording.
DD3	Indicates a device using helical recording.
LTO2	Indicates any HP or IBM LTO Generation 2 drive.
LTO3	Indicates any HP or IIBM LTO Generation 3 drive.
LTO4	Indicates any HP or IIBM LTO Generation 4 drive.
LTO5	Indicates any HP or IIBM LTO Generation 5 drive.
LTO6	Indicates any HP or IIBM LTO Generation 6 drive.
STK1R	Indicates any T9840 drive.
STK1R34	Indicates a 3490E-image T9840 drive.
STK1R35	Indicates a 3590-image T9840 drive.
STK1RA	Indicates any T9840A drive.
STK1RA34	Indicates a 3490E-image T9840A drive.
STK1RA35	Indicates a 3590-image T9840A drive.
STK1RB	Indicates any T9840B drive.
STK1RB34	Indicates a 3490E-image T9840B drive.
STK1RB35	Indicates a 3590-image T9840B drive.
STK1RAB	Indicates any T9840A or T9840B drive.
STK1RAB4	Indicates a 3490E-image T9840A or T9840B drive.
STK1RAB5	Indicates a 3590-image T9840A or T9840B drive.
STK1RC	Indicates any T9840C drive.

**TABLE A-3** Recording Techniques (Continued)

<b>Recording Technique</b>	<b>Description</b>
STK1RC34	Indicates a 3490E-image T9840C drive.
STK1RC35	Indicates a 3590-image T9840C drive.
STK1RD	Indicates any T9840D drive.
STK1RDE	Indicates an encryption-enabled T9840D drive.
STK1RDN	Indicates a non-encryption-enabled T9840D drive.
STK1RD34	Indicates a non-encryption-enabled 3490E-image T9840D drive.
STK1RD35	Indicates a non-encryption-enabled 3590-image T9840D drive.
STK1RDE4	Indicates an encryption-enabled 3490E-image T9840D drive.
STK1RDE5	Indicates an encryption-enabled 3590-image T9840D drive.
STK2P	Indicates any T9940 drive.
STK2P34	Indicates a 3490E-image T9940 drive.
STK2P35	Indicates a 3590-image T9940 drive.
STK2PA	Indicates any T9940A drive.
STK2PA34	Indicates a 3490E-image T9940A drive.
STK2PA35	Indicates a 3590-image T9940A drive.
STK2PB	Indicates any T9940B drive.
STK2PB34	Indicates a 3490E-image T9940B drive.
STK2PB35	Indicates a 3590-image T9940B drive.
T10K	Indicates any T10000 drives.
T10KN	Indicates all non-encrypted T10000 drives.
T10KE	Indicates all encrypted T10000 drives.
T10KA	Indicates any T10000A drive.
T10KAN	Indicates a non-encryption enabled 3490E- or 3590-image T10000A drive.
T1A34	Indicates a non-encryption enabled 3490E-image T10000A drive.
T1A35	Indicates a non-encryption enabled 3590-image T10000A drive.
T10KAE	Indicates an encryption-enabled 3490E- or 3590-image T10000A drive.
T1AE34	Indicates an encryption-enabled 3490E-image T10000A drive.
T1AE35	Indicates an encryption-enabled 3590-image T10000A drive.
T10KB	Indicates any T10000B drive.
T10KBN	Indicates a non-encryption enabled 3490E- or 3590-image T10000B drive.
T1B34	Indicates a non-encryption enabled 3490E-image T10000B drive.
T1B35	Indicates a non-encryption enabled 3590-image T10000B drive.
T10KBE	Indicates an encryption-enabled 3490E- or 3590-image T10000B drive.
T1BE34	Indicates an encryption-enabled 3490E-image T10000B drive.
T1BE35	Indicates an encryption-enabled 3590-image T10000B drive.
T10KC	Indicates any T10000C drive.

**TABLE A-3** Recording Techniques (Continued)

<b>Recording Technique</b>	<b>Description</b>
T10KCN	Indicates a non-encrypted 3490E- or 3590-image T10000C drive.
T1C34	Indicates a non-encrypted 3490E-image T10000C drive.
T1C35	Indicates a non-encrypted 3590-image T10000C drive.
T10KCE	Indicates an encryption-enabled 3490E- or 3590-image T10000C drive.
T1CE34	Indicates an encryption-enabled 3490E-image T10000C drive.
T1CE35	Indicates an encryption-enabled 3590-image T10000C drive.
T10KD	Indicates any T10000D drive.
T10KDN	Indicates a non-encrypted 3490E- or 3590-image T10000D drive
T1D34	Indicates a non-encrypted 3490E-image T10000D drive.
T1D35	Indicates a non-encrypted 3590-image T10000D drive.
T10KDE	Indicates an encryption-enabled 3490E- or 3590-image T10000D drive.
T1DE34	Indicates an encryption-enabled 3490E-image T10000D drive.
T1DE35	Indicates an encryption-enabled 3590-image T10000D drive.
Virtual	Indicates a VTV (Virtual Tape Volume) mounted on a VTD (Virtual Tape Drive).

When the RECtech parameter is not specified, a default is chosen based on the value of the MEDia parameter. The following table shows default values used if RECtech is omitted.

**TABLE A-4** Recording Technique Defaults

<b>MEDia Entered</b>	<b>RECtech Default</b>
LONGItud	LONGItud
ZLONGI	LONGitude
Standard	LONGItud
ECART	36track
ZCART	36Ctrack
DD3A, DD3B, DD3C, DD3D	DD3
STKR, STK1U, STKY	STK1R
STK2P, STK2W	STK2P
T10000T1, T10000TS, T10000CL	T10K
T10000CT	T10KA + T10KB
T10000T2, T10000TT	T10KC + T10KD
Virtual	Virtual

# MODeI Type (MODeI)

The model type is used to specify the model number of a transport (drive), or drive. MODeI provides the same type of information as RECTech, but a user may find it more convenient to specify a transport model rather than a recording technique.

**Note –**

- MODeI and RECTech are mutually exclusive.
- The SL8500 library supports **only** model types associated with T9840, T9940, and T10000 series drives.
- You can specify multiple values for this parameter; separate each value with a comma.

The following table describes valid MODeI types:

**TABLE A-5** Model Types

Model Type	Description
4480	Indicates a 4480 (18-track) drive.
4490	Indicates a 4490 (36-track Silverton) drive.
9490	Indicates a 9490 (36-track Timberline) drive.
9490EE	Indicates a 9490EE (36-track Timberline EE) drive.
SD3	Indicates an SD-3 (RedWood) drive.
9840	Indicates a 3490E-image T9840A drive.
984035	Indicates a 3590-image T9840A drive.
T9840B	Indicates a 3490E-image T9840B drive.
T9840B35	Indicates a 3590-image T9840B drive.
T9840C	Indicates a 3490E-image T9840C drive.
T9840C35	Indicates a 3590-image T9840C drive.
T9840D	Indicates a non encryption-enabled 3490E-image T9840D drive.
T9840D35	Indicates a non encryption-enabled 3590E-image T9840D drive.
T9840DE	Indicates an encryption-enabled 3490E-image T9840D drive.
T9840DE5	Indicates an encryption-enabled 3590E-image T9840D drive.
T9940A	Indicates a 3490E-image T9940A drive.
T9940A35	Indicates a 3590-image T9940A drive.
T9940B	Indicates a 3490E-image T9940B drive.
T9940B35	Indicates a 3590-image T9940B drive.
HP-LTO2	Indicates an HP LTO Generation 2 drive.
HP-LTO3	Indicates an HP LTO Generation 3 drive.
HP-LTO4	Indicates an HP LTO Generation 4 drive.
HP-LTO5	Indicates an HP LTO Generation 5 drive.
HP-LTO6	Indicates an HP LTO Generation 6 drive.
IBM-LTO2	Indicates an IBM LTO Generation 2 drive.



**TABLE A-5** Model Types (Continued)

Model Type	Description
IBM-LTO3	Indicates an IBM LTO Generation 3 drive.
IBM-LTO4	Indicates an IBM LTO Generation 4 drive.
IBM-LTO5	Indicates an IBM LTO Generation 5 drive.
IBM-LTO6	Indicates an IBM LTO Generation 6 drive.
SDLT-600	Indicates a SuperDLT model 600 drive.
DLT-S4	Indicates a Quantum DLT-S4 drive.
T1A34	indicates a non-encryption enabled 3490E-image T10000A drive.
T1A35	Indicates a non-encryption enabled 3590-image T10000A drive.
T1AE34	Indicates an encryption-enabled 3490E-image T10000A drive.
T1AE35	Indicates an encryption-enabled 3590-image T10000A drive.
T1B34	Indicates a non-encryption enabled 3490E-image T10000B drive.
T1B35	Indicates a non-encryption enabled 3590E-image T10000B drive.
T1BE34	Indicates an encryption-enabled 3490E-image T10000B drive.
T1BE35	Indicates an encryption-enabled 3590-image T10000B drive.
T1C34	indicates a non-encryption enabled 3490E-image T10000C drive.
T1C35	Indicates a non-encryption enabled 3590-image T10000C drive.
T1CE34	Indicates an encryption-enabled 3490E-image T10000C drive.
T1CE35	Indicates an encryption-enabled 3590-image T10000C drive.
T1D34	Indicates a non-encryption enabled 3490E-image T10000D drive.
T1D35	Indicates a non-encryption enabled 3590-image T10000D drive.
T1DE34	Indicates an encryption-enabled 3490E-image T10000D drive.
T1DE35	Indicates an encryption-enabled 3590-image T10000D drive.
Virtual	Indicates a VTV (Virtual Tape Volume) mounted on a VTD (Virtual Tape Drive).

MODeI Type (MODeI)

---

# Index

## A

---

ACSINT DSECT 213  
ACSRQ Macro  
    overview 139  
    requests  
        DISMOUNT 143  
        EJECT 146  
        MOUNT 151  
        MOVE 155  
        QCAP 159  
        QCONFIG 163  
        QDRIVES 166  
        QDRLIST 169  
        QSCRATCH 174  
        QVOLUME 178  
        QVOLUSE 182  
        SCRATCH 184  
        SELSCR 187  
        UNSCRATCH 191  
    syntax 140  
AUTHorize command 40

## C

---

CMS command 42  
command files 36  
commands  
    AUTHorize 40  
    CMS 42  
    COMMtest 43  
    CP 45  
    DISMOUNT 46  
    Display DRive 47  
    Display RC 49  
    Display Volume 50  
    DRIVemap 51  
    DUMP 53  
    DUMPOpts 54  
    EXIT 55  
    Help 56  
    List 57  
    LOGdisk 59

MOunt 61  
MSGDef 63  
OPERator 66  
OPTion 67  
POOLmap 68  
READ 71  
RESYNChronize 72  
Route 73  
SERVer 74  
SMCUDBX 78  
SMCUSCR 81  
TAPEPlex 82  
TCPip 85  
TRace 88  
COMMtest command 43  
conventions  
    control statements 16  
    syntax flow diagrams 12  
    typographic 12  
corrective service, installing 31  
CP command 45  
CP DETACH support 38

## D

---

DASD requirements 23  
determining VM Client resource requirements 26  
DISMOUNT command 46  
DISMOUNT request 143  
Display DRive command 47  
Display RC command 49  
Display Volume command 50  
DRIVemap command 51  
DUMP command 53  
DUMPOpts command 54

## E

---

EJECT request 146  
ELS 23  
EXIT command 55  
exits 37

## H

---

hardware requirements 22  
Help command 56

## I

---

installation  
    building VM Client executable code 29  
    creating a PPF override file 27  
    creating VM Client service machine 29  
    DASD requirements 23  
    IBM VSES/E 21, 25, 31  
    installing corrective service 31  
    installing VM Client product files 28  
    MVS requirements 23  
    placing VM Client into production 30  
    software and hardware requirements 22  
    summary of steps 25  
    testing VM Client 30  
interface data areas 194  
introduction 19  
IUB record format 222  
IUCV considerations 129

## L

---

Llist command 57  
LOGdisk command 59

## M

---

maintenance, installing 31  
MAXRC keyword value pair 36  
Media Type (MEDIA) values 226  
MODEl Type (MODEl) values 232  
MOunt command 61  
MOUNT request 151  
MOVE request 155  
MSGDef command 63  
MVS requirements 23

## O

---

OPERator command 66  
OPERATOR keyword value pair 36  
OPTion command 67  
overview 19

## P

---

POOLmap command 68  
PPF override file, creating 27

## Q

---

QCAP request 159  
QCONFIG request 163  
QDRIVES request 166  
QDRLIST request 169  
QSCRATCH request 174  
QVOLUME request 178  
QVOLUSE request 182

## R

---

READ command 71  
Recording Technique (RECtech) values 229  
RESYNChronize command 72  
Route command 73

## S

---

SCRATCH request 184  
SELSCR request 187  
SERVer command 74  
service machine, creating 29  
SLX macro 194  
SMCBINT module parameters 35  
SMCCMDS file 36  
SMCPARMS file 36  
SMCUDBX command 78  
SMCUSCR command 81  
SMCXIT00 exit 37  
SMCXIT01 exit 37  
software requirements 22  
syntax  
    ACSRQ macro 140  
    AUTHorize command 40  
    CMS command 42  
    COMMtest command 43  
    CP command 45  
    DISMount command 46  
    DISMOUNT request 143  
    Display DRive command 47  
    Display RC command 49  
    Display Volume command 50  
    DRIVemap command 51  
    DUMP command 53  
    DUMPOpts command 54  
    EJECT request 147  
    EXIT command 55  
    Help command 56  
    Llist command 57  
    LOGdisk command 59  
    MOunt command 61  
    MOUNT request 151  
    MOVE request 155

- MSGDef command 63
- OPERator command 66
- OPTion command 67
- POOLmap command 68
- QCAP request 159
- QCONFIG request 163
- QDRIVES request 166
- QDRLIST request 170
- QSCRATCH request 174
- QVOLUME request 178
- QVOLUME request 182
- READ command 71
- RESYNChronize command 72
- Route command 73
- SCRATCH request 184
- SELSCR request 187
- SERVer command 74
- SMCUSCR command 81
- TAPEPlex command 82
- TCPip command 85
- TRace command 88
- UNSCRATCH request 191

## **T**

---

- TAPEPlex command 82
- TCPip command 85
- testing VM Client 30
- TMS
  - decision points 126
  - interaction with VM Client 132
  - overview 125
- TRace command 88
- TRACE keyword value pair 36

## **U**

---

- UNSCRATCH request 191

## **V**

---

- VM Client overview 19
- VMSES/E 21, 25, 31

