CA Vtape™ Virtual Tape System

Best Practices Guide

Release 12.6.00

CA Technologies Product References

The CA Vtape™ Virtual Tape System guides refer to the following CA products and components:

- CA 1® Tape Management (CA 1)
- CA Allocate™ DASD Space and Placement (CA Allocate)
- CA Compress™ Data Compression (CA Compress)
- CA Earl® (CA Earl)
- CA Graphical Management Interface (CA GMI)
- CA Chorus Software Manager™ (CA CSM)
- CA MIM™ Resource Sharing (CA MIM)
- CA Sort® (CA Sort)
- CA Tape Encryption
- CA TLMS® Tape Management (CA TLMS)
- CA Vantage™ Storage Resource Manager (CA Vantage)
- CA Vtape™ Virtual Tape System (CA Vtape)
- CA Vtape™ Virtual Tape System Peer-To-Peer Option (CA Vtape P2P)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Best Practices Guide Process

These best practices are based on customer experience reported through interviews with development, technical support, and technical services. Therefore, many of these best practices are a collaborative effort stemming from customer feedback.

To continue to build on this process, we encourage you to share common themes of product use that might benefit other users. Please consider sharing your best practices with us.

To share your best *practices, c*ontact us at techpubs@ca.com and preface your email subject line with "Best Practices for product name" so that we can easily identify and categorize them.

Contents

Chapter 1: Introduction 7

Purpose of this Guide 7
Audience 7
Mainframe 2.0 Overview 7
Mainframe 2.0 Features 8

Chapter 2: Installation and Configuration Best Practices 11

Installation 11
Configuration for Optimal Performance 11
- Protect Control Data Sets 12
- Enable CA 1 or CA TLMS Tape Management System Interface 13
- Use Dynamic Cache Management 13
- Exploit the zIIP Specialty Processor 14
- Use the Backstore Engine in Primary and Failover Roles 15
- Set Up System Logger for Data Collection 17
- Share a Common Parmlib Data Set 18
- Set Up Enqueues and Reserves 19
- Use Virtual Volume Compression 22
- Use Parmlib Member VTPOOL to Define Virtual Volume Pools 24
- Set Concurrent Read Access to Virtual Volumes Created by CA Disk 25
- Use SYSSTC as the Service Class in Work Load Manager (WLM) 26
- Use the Graphical Management Interface (GMI) 26
- Monitor the CA Vtape Health Checks 27
- USS Backstore Considerations 29

Index 31

Chapter 1: Introduction

This section contains the following topics:

Purpose of this Guide (see page 7)
Audience (see page 7)
Mainframe 2.0 Overview (see page 7)
Mainframe 2.0 Features (see page 8)

Purpose of this Guide

The guide provides a brief introduction to the CA Technologies mainframe management strategy and features, and describes the best practices for installing and configuring CA Vtape.

Audience

The intended audience of this guide is systems programmers and administrators who install, configure, deploy, and maintain CA Vtape.

Mainframe 2.0 Overview

Mainframe 2.0 is our strategy for providing leadership in the mainframe operating environment. We intend to lead the mainframe marketplace for customer experience, Out-Tasking solutions, and solution innovation. After listening to customer needs and requirements to keep the mainframe operating environment viable and cost-effective, we are providing new tools to simplify usage and to energize this operating environment for years to come.

CA Mainframe Software Manager™ (CA MSM) is an important step in realizing the Mainframe 2.0 strategy. CA MSM simplifies and standardizes the delivery, installation, and maintenance of mainframe products on z/OS systems. CA MSM has a browser-based user interface (UI) with a modern look and feel for managing those solutions. As products adopt Mainframe 2.0 features and CA MSM services, you can acquire, install, and manage your software in a common way.

CA MSM provides software acquisition and installation that make it easier for you to obtain and install CA mainframe products, and apply the recommended maintenance. The services within CA MSM enable you to manage your software easily based on industry accepted best practices. The common browser-based UI makes the look and feel of the environment friendly and familiar.

We follow the IBM z/OS packaging standards using SMP/E, with some additional CA qualities of service added, to make installation simple and consistent. Additionally, through the synchronization of product releases and the use of common test environments, we will declare a yearly mainframe software stack that includes many new releases with enhanced functionality. This stack is certified for interoperability across the CA mainframe product portfolio and the base IBM z/OS product stack.

Mainframe 2.0 Features

Mainframe 2.0 has the following main features:

CA Mainframe Software Manager (CA MSM)

Delivers simplified acquisition, installation, and deployment capabilities using a common z/OS-based web application delivered through a browser-based UI. CA MSM includes the following services:

Product Acquisition Service (PAS)

Facilitates the acquisition of our mainframe products and services, including product base installation packages and program temporary fixes (PTFs). This service integrates the inventory of products available on your system with CA Support, providing a seamless environment for managing and downloading software and fixes onto your system.

Software Installation Service (SIS)

Facilitates the installation and maintenance of our mainframe products in the software inventory of the driving system. This service enables you to browse and manage the software inventory using a web interface, and automates tasks for products that use SMP/E to manage installation. You can browse downloaded software packages, and browse and manage one or more consolidated software inventories (CSIs) on the driving system.

Software Deployment Service (SDS)

Facilitates the deployment of CA Technologies mainframe products from the software inventory of the driving system. This service enables you to deploy installed products that are policy-driven with a set of appropriate transport mechanisms across a known topology. The enterprise system topology can include shared DASD environments, networked environments, and z/OS systems. Policies represent a combination of metadata input and user-supplied input. Metadata input identifies the component parts of a product. User-supplied input identifies the deployment criteria, such as where it goes and what it is named.

Electronic Software Delivery (ESD)

Enables you to get our products from an FTP server. We have improved this process so that you no longer need to build a tape to install the product.

Best Practices Management

Integrates with IBM Health Checker for z/OS to verify that deployed software follows our best practices. The health checks continually monitor the system and software to provide feedback on whether the software continues to be configured optimally.

Best Practices Guide

Provides best practices for product installation and configuration.

Note: For additional information about the CA Mainframe 2.0 initiative, see http://ca.com//mainframe2.

Chapter 2: Installation and Configuration Best Practices

This section contains the following topics:

Installation (see page 11)
Configuration for Optimal Performance (see page 11)

Installation

Use CA MSM to acquire, install, and maintain CA Vtape.

Business Value:

CA MSM provides a web interface, which works with ESD and standardized installation, to provide a common way to manage CA mainframe products. You can use it to download and install CA Vtape.

CA MSM lets you download product and maintenance releases over the Internet directly to your system from the CA Support website. After you use CA MSM to download your product or maintenance, you use the same interface to install the downloaded software packages using SMP/E.

Additional Considerations:

After you install CA Vtape using CA MSM, use the *Configuration Guide* to set it up. CA MSM can continue to help you maintain your product.

More Information:

For more information about CA MSM, see the *CA Mainframe Software Manager Guide*. For more information about product setup, see the *Configuration Guide* and the *Administration Guide*.

Configuration for Optimal Performance

This section describes best practices configurations to optimize performance of CA Vtape.

Protect Control Data Sets

Protect critical system control data sets.

Business Value:

Keeping the Global VCAT and Boot Strap Data Set on separate high performance, reliable media such as RAID-compliant DASD will improve overall performance by eliminating I/O contention between these two critical files. You minimize potential system outages by maintaining current backups of the Boot Strap data set.

Additional Considerations:

CA Vtape uses a set of files or control data sets (CDS) to record system and configuration information. The CA Vtape control data sets consist of the following:

Global VCAT

This file records multi-system control information such as Virtual Volume status and DASD buffer data sets. This file is shared by all of the CA Vtape Subsystems participating in a CA Vtape complex.

Boot Strap Data Set (BSDS)

This file records information necessary to recover the Global VCAT. Like the Global VCAT, this file is shared by all of the CA Vtape Subsystems participating in a CA Vtape complex.

CA Vtape control data sets should be considered critical system files. Keeping these files on high performance, reliable media, such as a RAID-compliant DASD, will help to improve overall performance and minimize potential system outages. CA Vtape uses Hardware Reserve processing to serialize access to the Global VCAT. The need to perform volume Hardware Reserve may require you to isolate the Global VCAT on a dedicated DASD volume. Maintaining current backups of the BSDS and keeping the Global VCAT and BSDS on separate DASD volumes is critical to insuring recoverability if the DASD hardware unexpectedly fails.

More Information:

See the section Control Data Sets in the chapter “Operational Considerations” in the *Configuration Guide*.

Enable CA 1 or CA TLMS Tape Management System Interface

Enable the CA 1 or CA TLMS tape management system interface to provide you with the advantage of a fully integrated support system that provides you with the Best of Class support for your tape management system.

Business Value:

Both CA 1 and CA TLMS provide expanded audit, reporting, and stacking capability for the Virtual and Physical tapes used by CA Vtape.

Additional Considerations:

CA Vtape supports an internal interface between CA 1 and CA TLMS. This interface retrieves Virtual Volume expiration date information and updates the tape management system records for the Virtual Volumes with their physical tape location (where they were externalized or recycled to). The TapeManagementSystem= attribute in the <DynamicOptions> section of the parmlib determines if the interface is activated.

More Information:

See the various sections regarding CA 1 or CA TLMS in the chapter “Operational Considerations” in the *Configuration Guide*.

Use Dynamic Cache Management

Use Dynamic Cache Management to create, access, and manage the VSAM Linear Data Sets (LDSs) CA Vtape uses to represent Virtual Volumes. When Dynamic Cache Management is active, CA Vtape dynamically defines an LDS when it is needed.

Business Value:

Provides greater cache capacity, reduces I/O utilization, and significantly improves overall throughput compared to using Static Cache.

Up to 50% reduction in CPU utilization and a 50% increase in throughput is encountered when using Dynamic Cache. Cache management is facilitated because there is a one-to-one relationship between the Linear Data Set (LDS) and the Virtual Volume.

The following additional cache features are available only if Dynamic Cache is implemented:

- Idle Space Released
- Secondary Extents
- DASD Volumes Spanning
- Maintains free space in the Storage Group to eliminate mount and recall delays

- Virtual VOLSER in the LDS name
- Reduction of allocation request to fit on a DASD volume
- Setting Concurrent Read Access to Virtual Volumes Created by CA Disk

Additional Considerations:

Dynamic Cache Management requires the use of DFSMS. The setup requires only the definition of a few constructs and minor changes to the ACS routines. Unique DFSMS constructs should be defined for use by CA Vtape. Defining unique constructs will provide better performance and greater control of the DASD cache volumes.

More Information:

See the section Dynamic Cache Management in the chapter “Cache Management” in the *Configuration Guide*. For information on how to migrate from Static Cache to Dynamic Cache, see the chapter “Conversion Procedures” in the *Administration Guide*.

Exploit the zIIP Specialty Processor

If IBM System z9 Integrated Information Processors (zIIPs) are available, elect to use zIIPs when you set up your regions.

Business Value:

Off loading CA Vtape processing from the main processor to the zIIP provides the following benefits:

- Saves billable CPU time by reducing the execution time on the normal central processing unit (CPU).
- Frees up processing cycles from the CPU for other work.

Additional Considerations:

CA Vtape is designed to exploit IBM zIIP specialty processors when they are installed and enabled on the system where CA Vtape will run and the appropriate parmlib attributes are specified. By default, CA Vtape will not attempt to make work eligible to run on the zIIP. You must activate zIIP processing using the zIIPExploitation= and PercentRunOnZIIP= attributes in the CA Vtape parmlib.

The Virtualization Engine for CA Vtape is designed to run in Service Request Block (SRB) mode, a prerequisite of zIIP processing. When the zIIP feature is activated, CA Vtape creates the environment necessary to schedule work on the zIIP. CA Vtape allows you to specify a percentage of work that is to be made eligible to run on the zIIP. For example, when you specify:

```
zIIPExploitation=Y
PercentRunOnZIIP=20
```

Every fifth SRB is directed to the zIIP. All Virtual Engine processing, including compression and decompression processing, runs in SRB mode; therefore the most processor intensive functions of CA Vtape can be made eligible to run on the zIIP if the zIIP capacity permits it.

More Information:

See the section Exploitation of the zIIP Specialty Processor in the chapter “Operational Considerations” in the *Configuration Guide*.

Note: In its documentation, IBM states that work directed to the zIIP processor might not be processed on the zIIP for a number of reasons. Refer to the appropriate IBM documentation for more information about the zIIP.

Use the Backstore Engine in Primary and Failover Roles

Configure the Backstore Engine in Primary and Failover roles.

Business Value:

Configuring a Primary Server for all Externalization and Recall activities minimizes the number of tapes and physical tape drives being used for stacking and recalls. Configuring a Failover Server allows for simple failover in the event the Primary Server system encounters a problem. CA Vtape physical tape activity is centralized, which simplifies management and monitoring.

Additional Considerations:

Based on the values set for some CA Vtape parmlib attributes and settings, a Backstore Engine can be defined as a Primary Server for all Externalization and Recall activities or a Failover Server. The Primary Server will take care of all Externalization and Recall tasks for a CA Vtape Complex installed on one or more LPARs. The Failover Servers will be dormant, waiting to take over this work if needed. With this configuration, you can centrally manage the Externalization and Recall processes from just one LPAR and CA Vtape subsystem without having to track tasks, alerts, commands, and messages from multiple LPARs and subsystems. The configuration of Primary and Failover Servers in a CA Vtape Complex is very simple. All the CA Vtape Subsystems in the same CA Vtape Complex should share the same Group Definitions and the same Externalization and Recall related attributes in the <DynamicOptions> section of parmlib. All subsystems should also have FullMaxdrivesEnforcement set to a value of Y. These parmlib attributes are:

- CacheAutoHoldLowThreshold
- CacheAutomationSchedule
- CacheAutoReleaseHighThreshold
- CacheWarningThreshold
- FullMaxdrivesEnforcement (must be set to Y)
- RecallNotificationEvent
- RecallServer (must be set to SERVER)
- RecallServerTimeout

Setting these parmlib attributes to the same values will ensure that each Backstore Engine reacts in the same way once MAXDRIVES is set to a value greater than zero. A Backstore Engine is the Primary Server when its MAXDRIVES is set to a number greater than zero. A Backstore Engine is a Failover Server when its MAXDRIVES is set to zero. To accomplish this the SVT*n* SET MAXDRIVES=0 console command is issued for the CA Vtape Subsystems that are the Failover Servers and SVT*n* SET MAXDRIVES=*nn* where *nn* is a value greater than zero for the CA Vtape Subsystem that is the Primary Server. The <StartupCommands> and the <ShutdownCommands> sections in the CA Vtape parmlib or your own automation product can be used to automate the issuing of these commands for the Primary and Failover Servers. When a problem occurs that prevents the Primary Server from processing Recall or Externalization requests, issuing the SVT*n* SET MAXDRIVES=*nn* console command for one of the Failover Servers will make it the Primary Server. If you do not want the original Primary Server to perform any physical tape mounts until the problem is resolved, then you should change its MAXDRIVES setting to zero.

More Information:

See the section Backstore Management in the chapter “Operational Considerations” in the *Configuration Guide*.

Set Up System Logger for Data Collection

Set up the System Logger to facilitate statistical reporting and problem analysis.

Business Value:

The System Logger allows you to centrally collect data generated by transactional applications and databases. CA Vtape creates log records for internal events. You can copy these log records to an operating system log stream and use them for statistical reports and problem analysis.

Additional Information:

The CA Vtape Internal Logger uses a dataspace created at startup as a repository for the logger data. The size of the dataspace is determined by a parmlib attribute in the VTPARMS member. The default value is 8 MB. Any dump automatically generated by CA Vtape, or any manually requested dump generated by executing the CA Vtape DUMP console command will include the Internal Logger data.

Note: The Logger dataspace can be included in a z/OS console dump command or SLIP command by specifying DSPNAME=('SVT*'.VCAT,'SVT*'.LOG).

The CA Vtape External Logger lets you optionally offload the internal logger data to an operating system log stream. By implementing the External Logger, logged events can be kept for several days in operating system-managed storage and then copied to physical sequential data sets for archival. This feature is required for statistical reports and we strongly recommend using it to help determine problems.

The LogDetailLevel= attribute in the <DynamicOptions> section of the parmlib determines if the CA Vtape Logger is activated and defines the level of detail of internal processes that is recorded by it. The best practice is to use the default value of 1.

The following are the valid values for the LogDetailLevel= attribute:

- 0

  Indicates that no logging should be done.

- 1

  Indicates that basic events like messages issued, console commands issued, batch utilities executed, recovery errors, subtask attach/detach, and major routines entered should be logged. This setting is sufficient for reports and most problems unless additional log detail is needed to diagnose a problem.

- 2

  Indicates that in addition to level 1 details, subchannel commands, interrupt commands, and low-level routines entered should be logged.

- 3

  Indicates that in addition to level 2 details, detailed I/O information should be logged. LogDetailLevel=2 and 3 may generate a large volume of records on an LPAR with a very active CA Vtape subsystem.

HLQ.CCUUJCL(LOGEVENT) contains complete descriptions of what events are logged and the detail level they are logged at.

More Information:

See the section System Logger in the chapter "Using the Basic Components" in the *Administration Guide*.

Share a Common Parmlib Data Set

Share a common parmlib data set across a CA Vtape Complex.

Business Value:

There is less administrative work maintaining a common parmlib across a complex. Maintaining separate parmlib members or a separate parmlib for each subsystem can lead to administrative inconsistencies.

Additional Considerations:

For subsystems within the same CA Vtape complex, a single version of the following items can be used by sharing a common parmlib data set:

- The Data Set Name Filter Lists
- The Data Class Filter Lists
- The group definitions

These definitions are loaded from the common parmlib data set to the system-specific Local VCAT at CA Vtape startup or when those definitions, which are dynamic, are updated and then refreshed for each of the subsystems.

If the above definitions are not shared, then separate parmlib members or separate parmlibs must be maintained for each subsystem.

More Information:

See the section General Description in the chapter “The Parameter Library (PARMLIB)” in the *Configuration Guide*.

Set Up Enqueues and Reserves

Set up enqueues and reserves to ensure data integrity and control over critical data and control files.

Business Value:

Correct utilization of enqueues and reserves will ensure that your data is protected and your control files are not compromised.

Additional Considerations:

In this section, one reserve and two enqueues are discussed:

SVTS Reserve

CA Vtape keeps its control information in a dataspace and writes it out to the Global VCAT and BSDS. To ensure the integrity of these updates, a hardware reserve is issued with QNAME=SVTS against the volume on which the Global VCAT resides. The best practice is to dedicate DASD volumes to the Global VCAT and BSDS data sets.

Different RNAME values are used depending on the GlobalReserve attribute located in the <DynamicOptions> section of the VTPARMS parmlib member, as follows:

- If the attribute value is ENHANCED, the RNAME includes part of the Global VCAT and BSDS data set names. This mode of operation is known as ENHANCED mode. This mode allows multiple CA Vtape Complexes, each using its own Global VCAT and BSDS, to reserve their own control data sets with a unique RNAME value. Complex A can update its own control data sets without having to wait on Complex B while it is updating its control data sets.
- If the attribute value is COMPATIBILITY, the RNAME is GLOBAL. This mode of operation is known as COMPATIBILITY mode. In a multiple CA Vtape Complex environment, a reserve issued with the generic RNAME of GLOBAL will force one complex to wait before updating its control data sets while another complex is updating its own control data sets.
- If all your subsystems are running CA Vtape at release 11 SP0 or any subsequent service pack or release, you should setup ENHANCED mode to avoid contention between multiple independent CA Vtape complexes.
- If you are still running releases older than CA Vtape release 11 SP0, you should run in COMPATIBILITY mode until those releases are upgraded.

Note: CA Vtape can and will dynamically switch from ENHANCED to COMPATIBILITY mode if necessary. If one subsystem in a complex is started in COMPATIBILITY mode, the other subsystems in the complex will detect this, issue a warning message, and dynamically switch to COMPATIBILITY mode.

You can also dynamically switch between modes by updating the GlobalReserve attribute and issuing the SVT*n* REFRESH=OPTIONS console command. If you change to COMPATIBILITY mode, any other subsystem in the complex that is running in ENHANCED mode automatically switches to COMPATIBILITY mode as soon as your change is detected. If you change to ENHANCED mode and any other subsystem is running in COMPATIBILITY mode, the subsystem you just changed will switch back to COMPATIBILITY mode as soon as your change is detected.

If you have a DASD resource serialization manager such as CA MIM RS or IBM Global Resource Serialization (GRS), you may want to convert QNAME=SVTS hardware reserves to SCOPE=SYSTEMS enqueues. However, in some cases this approach can lead to integrity exposures and corruption of the Global VCAT and the BSDS.

The following describes two different implementations, CA's recommended implementation, and the implementation that is needed if you are using IBM's GDPS/PPRC HyperSwap product.

Note: If you are not using IBM's GDPS/PPRC HyperSwap product, do not convert the QNAME=SVTS hardware reserve with CA MIM RS or GRS to an enqueue.

- To run CA MIM RS in ALLSYSTEMS mode, add the following statement to the CA MIM RS MIMQNAME parmlib member to tell CA MIM RS not to convert the QNAME=SVTS hardware reserves:

  ```
  SVTS GDIF=NO
  ```

- To run CA MIM RS in SELECT mode, you do not need any CA MIM RS MIMQNAME or GDIEXMPT parmlib member updates, because by default no hardware reserves are converted by CA MIM RS in this mode. In other words, there should be no QNAME=SVTS statements in these members.

- To run IBM GRS, you do not need any IBM GRS GRSRNL00 parmlib member updates, because, by default, no hardware reserves are converted by IBM GRS. In other words, there should be no QNAME=SVTS statement in this member.

Implementation with IBM's GDPS/PPRC HyperSwap

If you are using IBM's GDPS/PPRC HyperSwap or you are sure you will never share a Global VCAT and BSDS between more than one sysplex, you can safely have CA MIM RS or IBM GRS convert all instances of QNAME=SVTS hardware reserves to a SCOPE=SYSTEMS enqueue.

- To run CA MIM RS in ALLSYSTEMS mode, you do not need any CA MIM RS MIMQNAME or GDIEXMPT parmlib member updates, because by default all hardware reserves are converted by CA MIM RS in this mode. In other words there should be no QNAME=SVTS statements in these members.
- To run CA MIM RS in SELECT mode, add the following statement to the MIM MIMQNAME parmlib member to tell CA MIM RS to convert all QNAME=SVTS hardware reserves to a SCOPE=SYSTEMS enqueue:

  ```
  SVTS GDIF=YES,
  SCOPE=RESERVES,
  EXEMPT=NO,
  ECMF=YES,
  RPTAFTER=30,
  RPTCYCLE=60
  ```

- To run IBM GRS, add the following statement to the IBM GRS GRSRNL00 parmlib member to tell IBM GRS to convert the QNAME=SVTS hardware reserves to a SCOPE=SYSTEMS enqueue:

  ```
  RNLDEF RNL(CON) TYPE(GENERIC) QNAME(SVTS)
  ```

  Note: The F MIM, DISPLAY INIT command can be used to determine your GDIF PROCESS=SELECT/ALLSYSTEMS operating mode value.

Important! The QNAME=SVTS can be issued with different RNAME values depending on the CA Vtape parmlib settings. All instances of the QNAME=SVTS hardware reserve, regardless of the RNAME value, must be managed the same way on all systems. Converting some QNAME=SVTS reserves and not others will create an integrity exposure that could result in data loss. If you decide to convert the QNAME=SVTS reserve, all instances of the reserve must be converted with your multisystem DASD resource serialization manager, CA MIM RS or IBM GRS.

Also, if the QNAME=SVTS reserve is converted to an enqueue for a CA Vtape Complex that is running in multiple sysplexes and you do not use IBM's GDPS/PPRC HyperSwap, the resulting integrity exposure could cause a data loss.

SVTSX and SVTRCYCL Enqueues

Physical tapes containing Externalized Virtual Volumes can be enqueued by the Backstore Engine Externalization, Recall, and Recycle functions, all running on different systems. These three functions can try to mount the same physical tape at the same time and experience the standard wait for an in-use tape volume. You can eliminate this delay by always propagating the QNAMES SVTSX and SVTRCYCL as SCOPE=SYSTEMS enqueues to all systems. This allows CA Vtape automatic tape take-away to occur.

More Information:

See the section Enqueues and Reserves in the chapter "System Setup" in the *Configuration Guide*.

Use Virtual Volume Compression

Use Virtual Volume compression.

Business Value:

Virtual Volume compression lets you simulate Improved Data Recording Capability (IDRC) in CA Vtape Virtual Volumes by compressing the data as soon as the Virtual Device Engine receives it from the application.

The advantages of using compression are:

- Compressed Virtual Volumes occupy less DASD buffer space.
- The effective increase in Virtual Volume capacity allows CA Vtape to manage more application data.
- Compressed data requires fewer I/Os to read or write at the expense of some additional CPU overhead needed to perform the actual compression or expansion of that data.
- All Virtual Engine processing, including compression and decompression processing, runs in SRB mode; therefore, the most processor intensive functions of CA Vtape can be made eligible to run on the zIIP if the zIIP capacity permits it.
- The Backstore Engine and the RECYCLE function do not expand or compress the copied data. Their runtimes and resource consumption are reduced correspondingly by the percentage of compression achieved.

Note: The additional CPU overhead required as well as the compression rate achieved is a function of the data. While it was common during CA's own benchmarkings to see two-to-one compression rates, you may not get the same amount of compression.

Additional Considerations:

The CA Vtape compression routines use a combination of techniques including the run length limited (RLL) processing facility and the hardware compression facility. For more information, see the IBM publication *Enterprise Systems Architecture/390 Data Compression, SA22-7208*. The compression facility provides LZ compression capabilities through hardware instructions, software instructions, or both. When Virtual Volume Compression is enabled, the Virtual Device Engine constantly performs dynamic analysis as data is written to a Virtual Volume, allowing the Virtual Device Engine to select the most efficient compression technique.

Virtual Volume Compression reduces the need to code volume count parameters in application JCL. A volume count parameter is required by MVS when more than five tape volumes are mounted for a single Data Definition (DD) name. An application that writes a full 3490E tape that was compressed with IDRC will require a volume count parameter when it is switched to write to uncompressed 400 MB or 800 MB Virtual Volumes.

Using 2000 MB Virtual Volumes reduces both the need to use Virtual Volume Compression and CPU usage, while also reducing the need to code a volume count parameter in application JCL.

More Information:

See the section Virtual Volume Compression in the chapter "Operational Considerations" in the *Administration Guide*.

Use Parmlib Member VTPOOL to Define Virtual Volume Pools

Use the CA Vtape parmlib member VTPOOL to define the CA Vtape VOLSER range rather than add the VOLSER range with the SVTS ADD VVP= console command for better utilization of your scratch pools.

Volume Pooling is required when using control data sets that allow more than 510,800 Virtual Volumes to be defined.

Business Value:

By using the CA Vtape parmlib member VTPOOL to define your VOLSER ranges, scratch VOLSERs are chosen in a round-robin pattern rather than always reusing the lowest available VOLSER in the range. When a VOLSER is scratched, that VOLSER is immediately returned to the scratch pool and assigned a five day grace period before reusing the VOLSER. This provides the following benefits:

- The five day grace period allows access to the VOLSER in emergency situations.
- The scratch pool statistics are more accurate because the VOLSER is immediately returned to the scratch pool and not delayed by execution of the next Scratch Synchronization job.

Additional Considerations:

A CA Vtape complex is comprised of one or more CA Vtape subsystems accessing a common set of control files.

Defining a new CA Vtape complex or after running RECOVER=GLOBAL the Volume Pools are formatted in the original format.

Exploiting Volume Pools within a CA Vtape complex requires little more than making sure Virtual Volume ranges do not overlap with another CA Vtape complex sharing the same Tape Management Database.

When using the CA Vtape Peer-To-Peer Option (CA Vtape P2P) more planning is required. CA recommends dedicating pools to be used by either local or remote groups but not mixed. Using pools for both local and remote use is supported but it is discouraged because:

- Scratch counts are difficult to reconcile.
- The scratch volumes created locally waste volume range capacity on the Remote.
- Greater overhead for scratch negotiation.

Isolating the volume ranges when configuring CA Vtape P2P to share between more than two CA Vtape Complexes is even more important. We do not recommend sharing overlapping volume ranges between CA Vtape Complexes or sharing local and remote groups with the same pools.

More Information:

See the sections Scratch Pool Processing and VTPOOLS Parmlib Member in the *Configuration Guide*.

Set Concurrent Read Access to Virtual Volumes Created by CA Disk

Use Concurrent Read Access to Virtual Volumes created by CA Disk.

Business Value:

Storing CA Disk ARCHVOLS on CA Vtape Virtual Volumes give you tape utilization efficiencies and operational performance improvements during disaster recovery and day-to-day storage management processes. The performance improvements are realized because the CA Disk DMSAR restore jobs are not serialized with concurrent read access.

Additional Considerations:

Most tape media can only be mounted on one device at a time forcing applications like CA Disk to serialize tape access when accessing multiple files that reside on the ARCHVOL. For example, multiple DMSAR subtasks or batch Restore jobs directed to restore different user disk files residing on the same tape volume, ARCHVOL must wait and access the tape volume one at a time. If you are running with CacheManagement=Dynamic on a JES2 system, you can have CA Vtape enable concurrent read access to CA Disk ARCHVOLS that reside on Virtual Volumes for any LPAR that shares the CA Vtape Global VCAT.

To activate concurrent Read Access, set the AllowConcurrentVVEReadAccess= attribute in the <StartupOptions> to CADISK.

Note: CA Disk ARCHVOL Restore processing fully exploits AllowConcurrentVVERead Access. Other CA Disk utilities exploit AllowConcurrentVVERead but may require your automation software to respond WAIT to message IEF690I if allocation is run simultaneously for the same Virtual Volume.

More Information:

See the section Concurrent Read Access to Virtual Volumes Created by CA Disk in the chapter “Virtual Device Engines” in the *Configuration Guide*.

Use SYSSTC as the Service Class in Work Load Manager (WLM)

Use SYSSTC as the WLM Service Class for the CA Vtape Started Tasks or at least a Service Class with dispatching priority above TSO and Batch workloads.

Business Value:

Because CA Vtape is a started task, it must have high dispatching priority. SYSSTC is used to assign a higher priority to system started classes such as the CATALOG. Using SYSSTC ensures that the Virtual Tape I/O has precedence over other work.

More Information:

See the section Work Load Manager (WLM) Considerations in the chapter "System Setup" in the *Configuration Guide*.

Use the Graphical Management Interface (GMI)

Use the CA Graphical Management Interface (CA GMI) to view and monitor CA Vtape activity.

Business Value:

CA GMI is CA's graphical management interface product that allows you to view and manage CA Vtape activity from a Windows PC. CA GMI's structure is object oriented and provides a common layout consisting of an object tree, and consistent menu options and icons. This common layout makes it easy to remember how to navigate and use features. It also supports having multiple windows open at the same time (not hierarchical like the 3270), which allows you to view and compare information simultaneously.

This point-and-click interface provides a common and consistent method for viewing and managing multiple CA products, which can save considerable cost and time on training and learning.

Additional Considerations:

CA GMI consists of PC clients which interface with a z/OS server component to allow access to basic z/OS server functions.

The following are the available PC clients:

Windows-based Client

This client provides full functionality. That is, you can manually perform view and analysis functions, filter and sort desired entries, zoom (drill-down) to related objects, and take actions upon selected entries. You can create customized colored reports in different formats, for example, tables and graphs. These reports can be printed and exported to your PC directory, servers, intranet, and so on. You can create, manage, and view Summary objects. This client also provides designer wizards to create scripts to monitor and respond to any condition, exceptional or routine, in automatic ways. These automation services let you replace many if not all of the manual processes of managing your system.

Web-based Client

This client can be used from any PC with internet access to the CA GMI application server. The current version of the Web-based Client provides the user-driven functionality of view and analysis, filtering and sorting, zooming, and the ability to take actions on selected entries. You can create customized colored reports in different formats, for example, tables and graphs, and you can also view Summary objects.

CA GMI is included free of charge with many CA products, including CA Vtape.

Note: For more information about CA GMI for CA Vtape, see the *CA Vtape CA GMI Guide*.

Monitor the CA Vtape Health Checks

Monitor the health checks generated for CA Vtape.

Business Value:

Health checks alert you of conditions that could prevent CA Vtape from running properly, if left uncorrected, and they guide you in addressing the problem. These health checks provide best practices for running CA Vtape.

Additional Information:

The following health checks are provided for CA Vtape:

VTAPE_TMS_COMPATIBILITY

CA Vtape must work with a supported version of the TMS so the TMS interface can be established.

VTAPE_CDS_SEPARATION

The Global VCAT, BSDS and ICF Catalog used by CA Vtape should all be allocated on different DASD volumes. Allocating any combination of these data sets on the same volume creates a single point of failure, inhibits disaster recovery, and degrades performance due to contention.

VTAPE_MODULE_CONSISTENCY

The main address space and the sub address spaces are running with the same load modules, but at different maintenance levels.

VTAPE_DATAQ_ALLOCATION

The Data Queue (VDATAQ) environment failed to initialize the VDATAQ data set.

VTAPE_INACTIVE_RECALLSERVER

A Recall queued for a Virtual Volume has not started.

VTAPE_PARM_CACHE_MGMT

Dynamic Cache Management offers superior performance and reduced CPU usage and is supported by all new CA Vtape features. Static Cache Management will sunset in a future release (undetermined) of CA Vtape.

VTAPE_PARM_RECALL_SERVER

CA Vtape is not running with the recommended configuration of a Primary Backstore Server and a Failover Backstore Server. The RecallServer attribute in the Dynamic Options section of parmlib is not set to a value of SERVER.

VTAPE_PARM_FULL_MAXDRVS

The Backstore Engine is not configured for Failover Backstore Server operation.

VTAPE_PARM_zIIP_STATUS

IBM zIIP specialty processors are installed and available for use yet CA Vtape is not currently configured to take advantage of these specialty processors.

VTAPE_PARM_zIIP_CONFLICT

CA Vtape has detected an inconsistency between the settings of the zIIPExploitation and the PercentRunOnzIIP attributes.

VTAPE_PARM_TAPE_MGMT_SYSTEM

The VTPARMS Dynamic Options verification check has detected the presence of a CA 1 or CA TLMS tape management subsystem. However, the parmlib TapeManagementSystem attribute is defined as NONE. The recommended value for this attribute is AUTOMATIC.

Note: For more information about CA Vtape health checks, see the *CA Vtape Administration Guide* and *Message Reference Guide*.

USS Backstore Considerations

The USS Backstore feature utilizes z/OS UNIX System Services (USS) for reading and writing Virtual Volumes during Recall and Externalization.

Note: For more information about USS Backstore, see the chapter "USS Backstore" in the *Administration Guide*.

Planning for Mount Points

At a minimum, the Triplex directory is required to be a mount point. Without further customization, USS files written to the 32 different group directories are part of the Triplex file system.

However, to facilitate distributing file space and server capacity, each of the 32 different group directories can also be mount points to different NFS servers.

A common configuration involves defaulting most group directories to the Triplex file system, and customizing a few groups to other NFS servers.

The reasons for creating separate mount points are to accommodate dasd throughput and space. By using separate mount points you can spread the Virtual Volume workload across multiple devices. In addition, the amount of Virtual Volume data written could exceed the capacity of a single mount point. Utilizing multiple mount points allows you to overcome the physical limitations of a single mount point.

USS Mount Points in a CA Vtape Complex

A CA Vtape Complex consists of a number of CA Vtape Subsystems all sharing the same BSDS1 and Global VCAT control files. All of the CA Vtape Subsystems of a Complex must share the same mount point which contains the USS mount point definitions. These definitions are used to dynamically mount USS file systems used by an CA Vtape Subsystem. The default parmlib member name is VUSSMNTS.

While CA Vtape Subsystems within a Complex must share mount points, multiple CA Vtape Complexes must never share mount points. If more than one CA Vtape Complex is defined, they must be defined with different mount points.

Note: For more information about setting up USS Backstore, see the chapter "USS Backstore" in the *Administration Guide*.

Index

2

2000 MB Virtual Volumes • 22

A

ALLSYSTEMS mode • 19

B

Backstore Engine in Primary and Failover roles • 15

C

CA 1, use CA tape management systems • 13
CA Disk, concurrent Read Access to Virtual Volumes • 25
CA Graphical Management Interface (CA GMI) • 26
CA MIM RS • 19
COMPATIBILITY mode • 19
compressing data • 22
concurrent Read Access to Virtual Volumes created by CA Disk • 25
control data files, protect system critical • 12

D

Dynamic Cache Management • 13

E

ENHANCED mode • 19
enqueues and reserves • 19

F

Failover and Primary roles, use the Backstore Engine • 15

G

GlobalReserve attribute • 19

H

Health Checker • 27

I

IBM GRS • 19
Improved Data Recording Capability (IDRC) • 22

P

parmlib, sharing a common parmlib data set • 18
Primary and Failover roles, use the Backstore Engine • 15

Q

QNAME • 19

R

Read Access to Virtual Volumes created by CA Disk • 25
reserves and enqueues • 19
RNAME • 19

S

SELECT mode • 19
SVTRCYCL and SVTSX enqueues • 19
SVTS reserves • 19
SVTSX and SVTRCYCL enqueues • 19
system control data files, protect • 12
System Logger • 17

T

tape management systems, use CA's • 13

U

USS Backstore • 29

V

Virtual Volume compression • 22
Volume Pool planning • 24

W

Web-based user interface • 26
Windows-based user interface • 26
Work Load Manager (WLM) considerations • 26

Z

zIIP, exploit the zIIP specialty processors • 14