Telestream

Storage Policy Manager

User Guide

Release: 8.2

Revision: 1.2

September 26, 2022

Copyrights and Trademark Notices

Contents

Telestream Contact Information

To obtain product information, technical support, or provide comments on this guide, contact us using our web site, email, or phone number as listed below.

Table 1.

Resource	Contact Information
DIVA Core Technical Support	Web Site: https://www.telestream.net/telestream-support/diva/support.htm Depending on the problem severity, we will respond to your request within 24 business hours. For P1, we will respond within 1 hour. Please see the Maintenance & Support Guide for these definitions. • Support hours for customers are Monday - Friday, 7am - 6pm local time. • P1 issues for customers are 24/7.
Telestream, LLC	Web Site: www.telestream.net Sales and Marketing Email: info@telestream.net Telestream, LLC 848 Gold Flat Road, Suite 1 Nevada City, CA USA 95959
International Distributor Support	Web Site: www.telestream.net See the Telestream Web site for your regional authorized Telestream distributor.
Telestream Technical Writers	Email: techwriter@telestream.net Share comments about this or other Telestream documents.

Preface

This document describes the installation, configuration, and operations of the Core Storage Policy Manager for this Core release.

Audience

This document is intended for Installation, Administration and Operations personnel to follow all of the necessary steps to achieve full functionality of the SPM (Storage Policy Manager) component.

Documentation Accessibility

For information about Telestream's commitment to accessibility, visit the Telestream Support Portal located at:

https://www.telestream.net/telestream-support/diva/support.htm

Access to Telestream Support

Telestream customers that have purchased support have access to electronic support through the Telestream Support Portal located at:

https://www.telestream.net/telestream-support/diva/support.htm

Related Documents

For more information, see the Telestream Core documentation set for this release located at:

https://www.telestream.net/telestream-support/diva/support.htm

Document Updates

The following table identifies updates made to this document.

Date	Update
April 2022	Updated Copyright information. Updated book for release 8.2. Updated terminology to new standards (see the Overview for updated terms).
July 2022	Migrated book to Telestream format and styles.
September 2022	Updated terminology and title page graphic.

Overview

Caution: Misconfiguration of SPM may lead to unexpected and disastrous results! Minor edits to slots can lead to catastrophic consequences including the deletion of hundreds of thousands of instances on tape (for example, an incorrect value in a slot's end time), or database corruption (for example, creating a Storage slot while the SPM is running, and changing it to Restore type slot while copy and delete instance actions have already begun processing). Without special training and familiarity with the product, you should always contact Telestream Support before making any changes to SPM. Failure to do so may result in severe damage to the DIVA Core system or even permanent data loss.

This chapter describes an overview of the SPM (Storage Policy Manager), and includes the following information:

Topics:

DIVA Core Storage Policy Manager Overview

The SPM (Storage Policy Manager) software component provides Virtual Object lifecycle management (interacting with DIVA Core), and is typically installed on the same computer as DIVA Core. For example, an archived Virtual Object can reside on a specific medium the first day, and migrate (over time) to a different medium according to your established policies and rules. DIVA Core executes the Virtual Object lifecycle migration as a background activity by following the rules and policies defined in the corresponding Storage Plan.

New Terminology

The following terminology has been updated to reflect standardization efforts across all DIVA and Kumulate applications. There will be some variations in the documentation compared to the interface until everything is switched over to the new terminology; the documentation uses the new terms wherever possible.

- Running Requests are now called Jobs
- Request History is now called Job History
- Libraries are now called Managed Storage
- Datahub is now called Actor
- Proxyhub is now called Proxy Actor
- DIVA Core and DIVA Manager are now called DIVA Core / Core / Core Manager
- Category is now called Collection
- Source/Destination is now called Unmanaged Storage Repository
- Storage Repository is now called Managed Storage Repository
- Object is now called Virtual Object
- Group is now called Tape Group
- Link is now called Storage Link
- Storage Plan Manager is now called Storage Policy Manager
- Drop Folder Monitor (DFM) is now called Watch Folder Monitor (WFM)
- DIVA Command and Control Panel are now called System Management App
- DIVA Analytics and DIVAProtect are now called Analytics App

New and Enhanced SPM Features and Functionality

SPM configuration is now possible in System Management App. It supports the same functionality as the Configuration Utility with the following difference:

SPM, SPM filters, and SPM slots are now grouped into a single structured view shown in tabs. When you add a new SPM rule, you can add an SPM filter and slot for this SPM directly under this tab. You no longer need to select which SPM a filter or slot belongs to. In the Configuration Utility, filter and slots are displayed in their own independent tables. The user must select which SPM a filter or slot belongs to.

You can change the status of SPM Failed Actions to COMPLETED by right clicking the action, and then selecting Mark Action COMPLETED from the context menu. See *Marking Failed Actions Complete* for detailed information.

The SPM service is ported to 64-bit in Windows. The OracleDivaDB_3-0-0_12_1_0_2_0_SE2_Windows_64-bit.zip and later releases no longer include the 32-bit Oracle database client. DIVA Core 7.7 and later in a Windows environment only supports DIVAOracle database package OracleDivaDB_3-0-0_12_1_0_2_0_SE2_Windows_64-bit.zip and later. No previous database package will work with DIVA Core 7.7 and later.

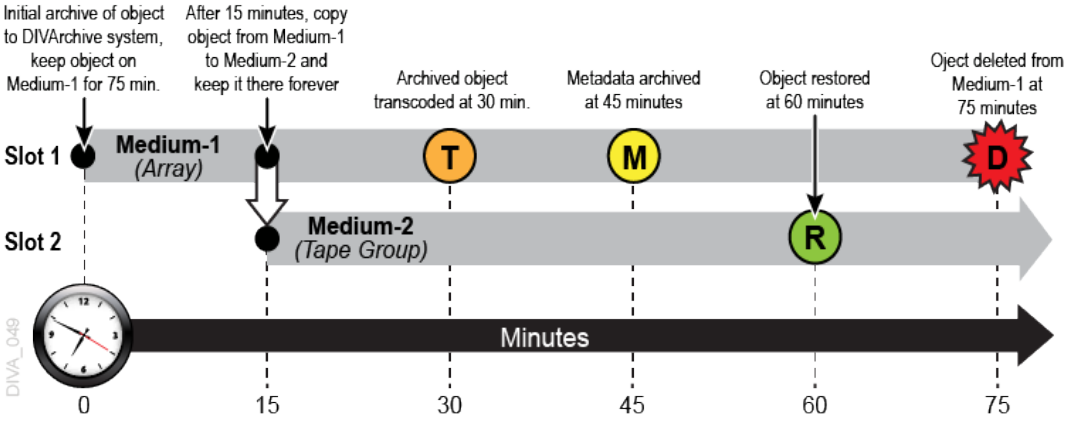
See the DIVA Core Installation and Configuration Guide for more information on running DIVA Core in a Linux environment, and DIVA Core new and enhanced features and functionality.

DIVA Core Virtual Object Lifecycle

This section describes the steps and processes encompassing the DIVA Core Virtual Object lifecycle. Along with the standard DIVA Core terminology, Storage Policy Manager has additional terms that require identification. Understanding these additional terms will assist in comprehending the DIVA Core SPM module, the processes involved, and the DIVA Core Virtual Object Lifecycle.

See the Glossary for SPM specific terminology.

There are several steps included in the Virtual Object lifecycle, depicted in the following example figure. You can use SPM to automate this generic Virtual Object lifecycle.



Managing the Virtual Object Lifecycle

DIVA Core can define a storage policy for each Virtual Object, or group of Virtual Objects, which enables describing the complete lifecycle of a Virtual Object, and managing the content migration throughout the lifecycle. Using the Storage Policy Manager, content lifecycle management becomes a background process that is performed automatically.

You use a Storage Plan to specify the different migrations, and copies of the specified Virtual Objects, from one media to another according to your defined storage rules and policies.

The SPM component in DIVA Core manages the Storage Plan. The Storage Plan is a way to select the best trade off between cost and performance of the different storage media technologies as a function of the age and access frequency of the Virtual Object. The Storage Plan feature enables the customization of content lifecycle management.

You can specify rules and policies that include parameters such as:

- The amount of time a Virtual Object is to be retained on a medium. This interval is called a Slot.
- Where to copy a Virtual Object when its time has expired on the medium (for example, copy from array to tape after two weeks).

  For example, you can schedule tasks that consume relatively large amounts of system resources during low use periods (for example, between midnight and five am). These tasks are executed within the interval specified in the corresponding slot configuration.
- Time of day (local time) when the action is executed.
- When and where to restore a Virtual Object.
- When to transcode a Virtual Object, and where to store the transcoded content.

SPM Slot Types

SPM has a logical workflow as follows:

- A Medium is a combination of defined medium (disk arrays or tape groups) available to SPM.
- Each Storage Plan is associated with one, or more, mediums and contains one, or more, slots.
- Each defined Filter is associated with a Storage Plan.
- Each slot defines one, or more, Actions to be performed.

To create a slot, navigate to the DIVA Configuration > Storage Plans > Slots tab in System Management App and click the + button on the top of the tab.

To edit an existing slot, navigate to the Slots tab in System Management App, and then click the Edit icon (looks like a pencil) for the slot you want to edit in the Slots panel.

The following is a list of Slot Types and Actions.

Storage Slot

A Storage Slot enables copying of a Virtual Object, and then deleting the Virtual Object (if the slot is configured for deletion). Virtual Object deletion refers to either a DeleteInstance or DeleteVirtual Object process, depending on the SPM configuration, and whether other instances still exist. You can only create and configure Storage Slots on System Management App Storage Plans tab.

See *Simple SPM Configuration Example* for information on Storage Slot workflows, and *Configuring Storage Slots* for information on configuring Storage Slots.

Transcode Archive Slot

A Transcode Archived Slot enables transcoding of an archived Virtual Object. You create and configure this type of slot using both the System Management App Storage Plans tab.

See *Action Slot Workflow* for information on Action Slot workflows, and *Configuring Transcode Archived Slots* for information on configuring Transcode Archived Slots.

Metadata Archive Slot

A Metadata Archive Slot enables archiving of the Virtual Object metadata. You create and configure this type of slot using both the System Management App Storage Plans tab.

See *Action Slot Workflow* for information on Action Slot workflows, and *Configuring Metadata Archive Slots* for information on configuring Metadata Archive Slots.

Restore Slot

A Restore Slot enables restoring of a Virtual Object. You create and configure this type of slot using both the System Management App Storage Plans tab.

See *Action Slot Workflow* for information on Action Slot workflows, and *Configuring Restore Slots* for information on configuring Restore Slots.

Actions, Action Steps, and Action States

A single Storage Plan can contain a single slot, or multiple slots. Each slot has one or two actions associated with it.

Every action has one Step, or two Steps associated with it as follows:

- Copy, Transcode Archived, Metadata Archive, and Restore

  These slots have a single Step associated with them.

- Delete (only valid for Storage Slots)

  The Delete Action has two Steps associated with it as follows:

  – POSTPONED

    Before SPM executes any Delete Actions, it checks whether the medium is watermarked. If the medium is marked Yes for Watermarking, and the level has not reached the High Watermark, SPM will not execute the Delete Action immediately. Instead, SPM marks the Virtual Object for deletion, and set the Virtual Object state to POSTPONED.

    If the medium is not watermarked, there will only be one step (Delete), and the Virtual Object will be deleted immediately upon expiration.

  – Delete

    When the medium is marked Yes for Watermarking, and the watermarked medium level reaches the High Watermark, then the Virtual Object will actually be deleted.

All slot details are managed from a single screen, and all Slot Types only require configuration from the System Management App Slots tab.

When a Virtual Object is added to DIVA Core, SPM checks for Virtual Object compliance with the filters specified in the system. When a Virtual Object conforms to one of the configured filters, the filter determines which configured Storage Plan to use for processing of the Virtual Object. A Virtual Object can be assigned to only one Storage Plan. If the Virtual Object conforms to multiple filters, the first filter the Virtual Object conforms to is the one that is considered.

The slots associated with the identified Storage Plan determine the actions performed on the Virtual Object. If a Virtual Object did not conform to any configured filters, the Virtual Object is assigned to the SP_DEFAULT Storage Plan. The SP_DEFAULT is the default Storage Plan and must have no slots associated with it.

Action States indicate the status of the action. Each action performed on a Virtual Object goes through different states as shown in the following example. Each state will finish processing before the status is updated to the next state.

Example:

1. A Virtual Object matches the filter for a specific Storage Plan.
2. The Storage Plan's slot schedules the associated actions for execution on the Virtual Object; the status is now SCHEDULED.
3. The action is then loaded into the Action Queue; the status is now LOADED.
4. The action is now executed; the status is now PROCESSING.
5. Upon successful execution of the action the status is updated to COMPLETED.

The additional possible action status states are POSTPONED, FAILED LONG, and REJECTED. The following table describes these additional statuses. See *Storage Policy Manager Workflows* for additional Slot Workflow information.

The following are the different SPM Action States:

SCHEDULED

This state is the initial state of an action. The action is scheduled, and will be loaded into the Action Queue for execution.

LOADED

The action is loaded into the Action Queue.

PROCESSING

The action is being executed. The action is loaded into the Action Queue, and SPM has started processing this action for execution.

POSTPONED

When a Delete Action is encountered on a watermarked array, SPM will mark the Virtual Object for deletion, and set the state to POSTPONED. SPM will not actually delete the Virtual Object until the High Watermark level is reached, and then the Virtual Object will be removed from the array. If the medium is not watermarked, the Virtual Object will be deleted immediately. This state only applies to Deleted Actions.

COMPLETED

The action has completed processing successfully.

FAILED LONG

The action failed, and will be retried according to the SPM Configuration. See the UPDATE_ACTIONS_RETRY_FAILED_DELAY parameter in *SPM Configuration File Parameters*.

Rejected

SPM sets an action to the Rejected state when it has reached the maximum number of retries and has failed. Rejected actions are never retried again.

SPM also sets Copy Actions of a storage slot to the Rejected state if the destination medium already has instances whose quantity is greater than the instances value configured in the Storage Slot.

Managing DIVA Core Watermarks and Arrays

You can manage deletions on disk mediums using watermarks. When watermarks are not used, deletions occur immediately after the slot expires. When watermarks are used, deletions are postponed until the disk array's occupied space hits a configured watermark.

The DSM (Disk Space Monitor) is a function of SPM, and monitors SPM identified arrays, not individual disks. The DSM process only starts if there is a SPM array that is configured for watermarking.

See *SPM Configuration File Parameters* and *SPM Default Configuration File (spm.conf.ini)* for configuration information.

When a Virtual Object is set for deletion by a Storage Slot, it is not actually deleted until the watermarked array reaches the High Watermark. After the array reaches the High Watermark, Virtual Objects marked for deletion will be deleted either by Last Access Time, or Largest Virtual Object Size. The method selected depends on how the watermark is configured. Virtual Objects are then deleted until the Low Watermark is reached.

Virtual Objects on watermarked arrays are deleted using one of the two following methods:

Last Access Time

This method deletes the oldest Virtual Objects, among the Virtual Objects marked for deletion (in non-Mixed Mode), first according to the last time the Virtual Object was accessed.

Largest Virtual Object Size

This method deletes the largest Virtual Objects, among the Virtual Objects marked for deletion (in non-Mixed Mode), first according to the Virtual Object size.

DSM has 2 methods for checking the Array watermark levels:

- Through the DIVA Core API

  This method is typically used. Requests are sent to the DIVA Core and the watermark levels are sent back to DSM.

- Directly from the array

  This method uses the Operating System's commands to retrieve the information.

SPM arrays have three possible Watermark settings that are configurable in System Management App:

- Yes

  This setting watermarks the identified array, and only considers Virtual Objects already marked for deletion.

- No

  This setting does not watermark the identified array.

- Mixed

  This setting incorporates a combination of watermarked and not watermarked. This setting is only valid for Storage Slots. The action taken depends on which of the following events occurs first:

  – The slot reaches its end time.

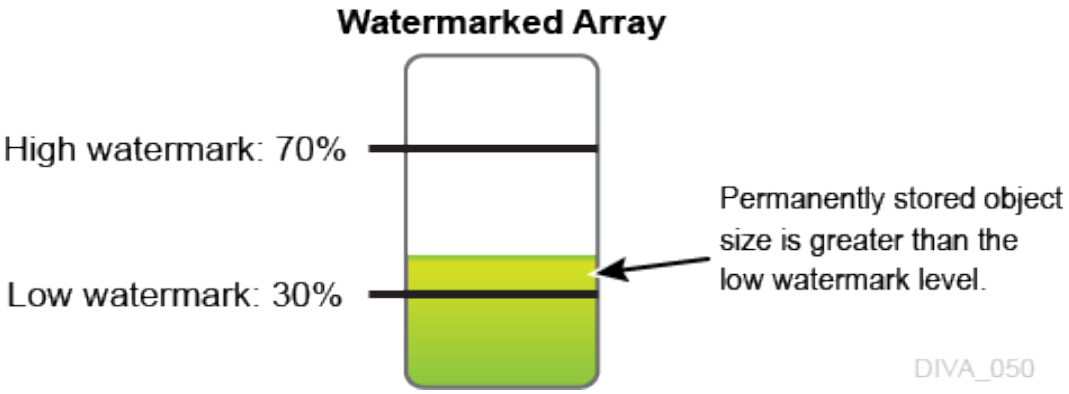
  – The High Watermark is reached. This considers both Virtual Objects marked for deletion, and Virtual Objects whose slots are still open.

Note: You must restart SPM if you change the watermark state of an array.

Recommended Best Practices for Watermarking

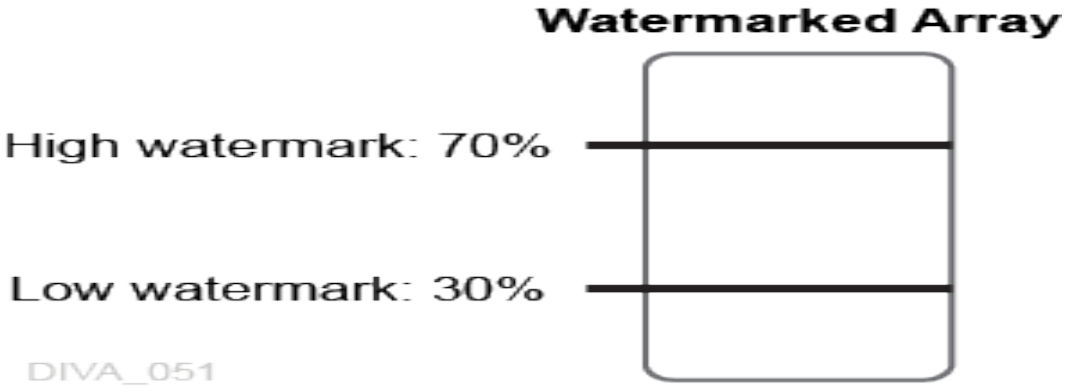
If the total size of all permanently stored Virtual Objects (that is, Virtual Objects with no specified expiration time) on an array is greater than the configured Low Watermark, Telestream recommends reconfiguring the Low Watermark setting to a value higher than the amount of permanently stored Virtual Objects.

The following figure represents this basic concept:

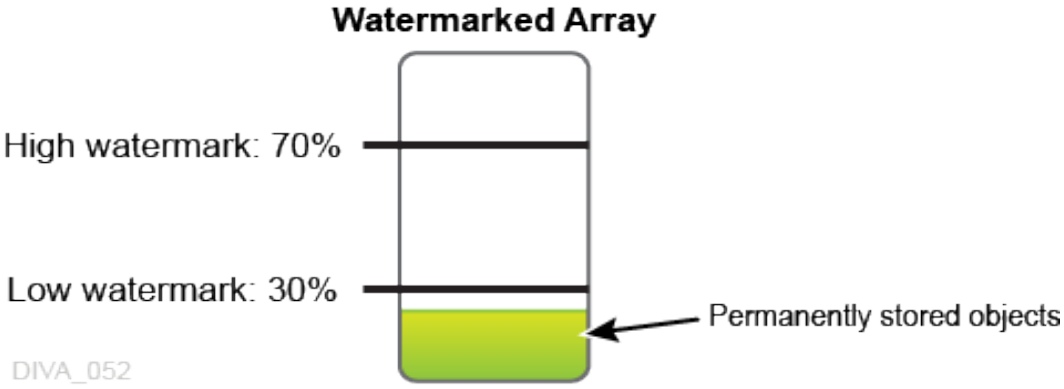


The following figures characterize a typical Watermarked Medium (watermarking set to Yes) and illustrate how watermarking ensures the medium does not reach full capacity.

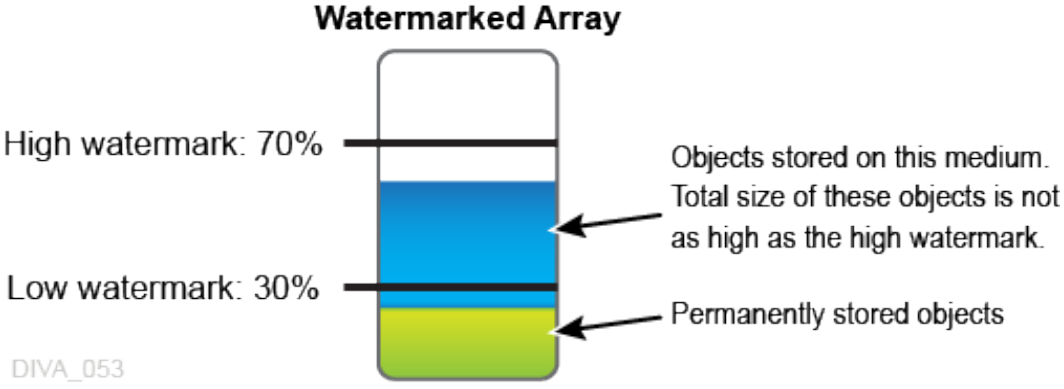
The following figure represents a watermarked disk array with no Virtual Objects, and shows the Low Watermark and High Watermark:



The following figure represents a watermarked disk array with permanently stored Virtual Objects with a total size that is less than the Low Watermark:

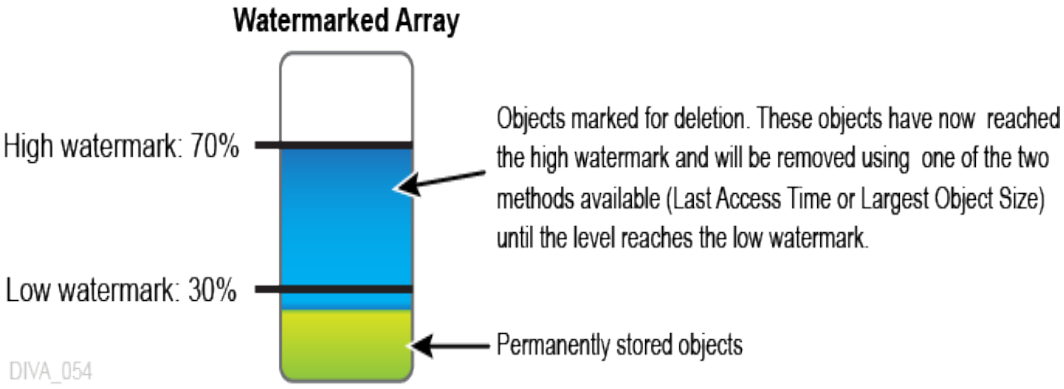


The following figure represents a watermarked disk array with permanently stored Virtual Objects with a total size that is less than the Low Watermark, and temporarily stored Virtual Objects with a total size that is higher than the Low Watermark:

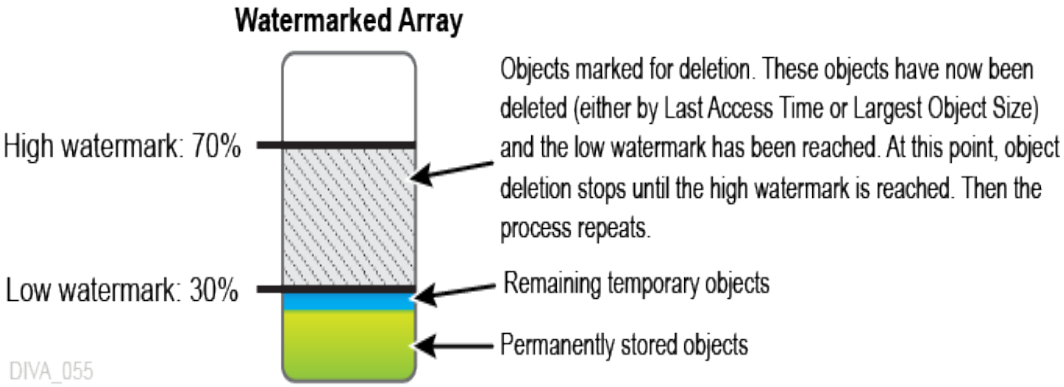


The following figure represents a watermarked disk array with permanently stored Virtual Objects with a total size that is less than the Low Watermark, and temporarily stored Virtual Objects with a total size that has reached the High Watermark level. In

this figure the temporary Virtual Objects have now reached the High Watermark, and will be removed using one of the two methods available (*Last Access Time or Largest Virtual Object Size*) until the level reaches the Low Watermark.



The following figure represents a watermarked disk array after the temporary Virtual Objects have been deleted down to the Low Watermark. Virtual Objects have now been deleted (*either by Last Access Time or Largest Virtual Object Size*) and the Low Watermark has been reached. At this point, Virtual Object deletion stops until the High Watermark is reached again; then the process repeats.



Storage Policy Manager Workflows

The following sections describe the Storage Policy Manager workflows.

Storage Policy Manager Tasks

The SPM module runs several processes, each in charge of a particular task, and uses the Core Database to process actions. There are always multiple tasks processing in parallel when SPM is operational. All actions currently being executed by SPM reside in the Action Queue until the execution is complete, and then they are deleted from the queue. You can view actions in the queue using the System Management App SPM Actions panel, which is located under the DIVA Core tab. You must at least be connected to the Core Database to access this view.

The following list describes internal tasks used by SPM:

Update

This task is responsible for generating, or updating, the actions based on the SPM configuration so the Load task can load the actions into the Action Queue.

Load

This task loads the actions into the Action Queue from the database for processing.

Execution

This task executes the actions loaded into the Action Queue, and submits requests to the DIVA Core as necessary.

DSM (Disk Space Monitor)

DSM monitors a particular array (not individual disks) to make sure it does not exceed the High Watermark value.

Recovery

When SPM starts, the Recovery task checks for the actions that are in an inconsistent state (LOADED or PROCESSING) and sets them to the SCHEDULED state. The action will be in an inconsistent state when SPM was terminated, or stopped during execution of the actions. If the rescheduled action has a DIVA Core Request ID associated with it, it will be loaded into the Action Queue as part of the Recovery task so that SPM can update the Action Status based on the status of the request submitted to DIVA Core.

An action will be in an inconsistent state and have a Request ID associated with it if SPM was terminated, or stopped, when SPM:

- Has executed the action
- Submitted the request to DIVA Core

- Was waiting for the status of the request

The Recovery task only runs during SPM startup. If there are no actions in an inconsistent state, the task will just end.

Simple Virtual Object Lifecycle Example

The following is a very simple Virtual Object lifecycle for a Virtual Object being processed through SPM. The process for this example lifecycle is as follows:

1. A Virtual Object exists on the Video server.
2. The Virtual Object is archived to a disk array that is known to DIVA Core, and is one of the SPM monitored mediums.
3. SPM processing now begins as follows:
   a. SPM checks if the Virtual Object matches any of the SPM Filters.
   b. If the Virtual Object matches one of the filters, processing by SPM continues by identifying which Storage Plan is associated with the corresponding filter.
   c. When the Storage Plan is identified, SPM detects the slots that are included in the managed Storage Plan.
   d. The slots contain the actions that will be performed on the Virtual Object.
   e. SPM processes the detected actions (included in the slots) on the Virtual Object until processing is complete, or the slot ends (*closes*).
4. According to the Storage Slot that was created for the Storage Plan in the example, 10 minutes after the Virtual Object is archived, it is copied to a target tape group.
5. SPM deletes the Virtual Object from the DIVA Core system and SPM 10 minutes later (20 minutes after initially being archived to DIVA Core).

Retrying SPM Failed Actions and Issue Resolution

When a requested action fails to execute properly, or a connection issue occurs, SPM will retry the action according to the following scenarios. The initial retry is automatic, and the requested action state remains as PROCESSING.

DIVA Core Connection Failures

SPM uses the DIVA Core API (C++) to connect to the DIVA Core. If the DIVA Core connection is down, SPM continues retrying to establish the connection to DIVA Core until a connection is established.

Database Connection Failures

SPM will continue retrying to connect to the Core Database every 20 seconds until a connection is established.

Missing Instances

The following example indicates what occurs if SPM is looking for an instance that does not exist:

A Storage Slot is configured with Once Only set to N. The slot starts five minutes after archiving, and ends 30 minutes after archiving.

- SPM finishes the Copy Action after 5 minutes.
- A user manually deletes the copy made by SPM after 10 minutes.

  SPM senses the deletion and creates the copy again, because the slot period has not ended yet.
- The copy made by SPM is manually deleted again after 35 minutes.

  SPM does not perform another Copy because the slot period has already ended.

If the Once Only parameter was set to Y, SPM will only make the copy one time. If the copy made by SPM is manually deleted, SPM will not make the copy again.

Action Retries

The following list describes when SPM will retry failed actions. See *Action Slot Workflow* and *Storage Slot Workflow* for Action Slot and Storage Slot workflows.

- DIVA Core is Busy

  If SPM could not execute the actions by submitting a request because DIVA Core is busy executing more requests than the value configured in DIVA_MANAGER_MONITOR_MAX_REQUESTS in the spm.conf configuration file, SPM will retry the same action after few seconds delay. The delay value is configured using the DIVA_MANAGER_MONITOR_ACTION_DELAY parameter in the spm.conf configuration file.
- Action Failed_Long State

  When SPM executes an action by submitting a request to DIVA Core and the Request fails, the action is marked as FAILED LONG. SPM will retry the failed action after the delay period configured in the UPDATE_ACTIONS_RETRY_FAILED_DELAY parameter in the spm.conf configuration file. SPM continues retrying this action 1,000 times using the configured delay. If the action continues to fail, it is marked as Rejected and is never retried again.

  After placed in a Failed_Long state, only Copy, Delete, and Restore Actions are retried. All other actions will not be retried.
- Action REJECT State

  SPM sets an action to the Rejected state when it has reached the maximum number of retries and has failed. Rejected actions are never retried again. SPM also sets Copy Actions of a storage slot to the rejected state if the destination medium already has instances whose quantity is greater than the instances value configured in the Storage Slot. **This is a permanent failure.**

- Once Only Slots

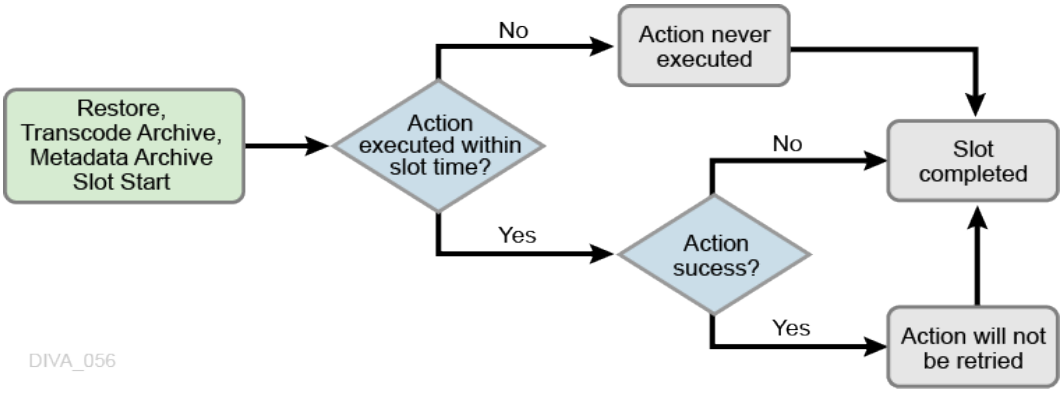
  If the Once Only parameter is set to false for a Storage Slot, all of its actions will be retried throughout the Slot Start Time and Slot End Time period.

Action Slot Workflow

Action Slots have a specific workflow that encompasses what DIVA Core will do in certain cases. The flowchart below displays the typical Action Slot workflow. There are three possible outcomes:

- Success
- Failed Long
- Not executed within the configured slot time

The following figure depicts the Action Slot workflow:

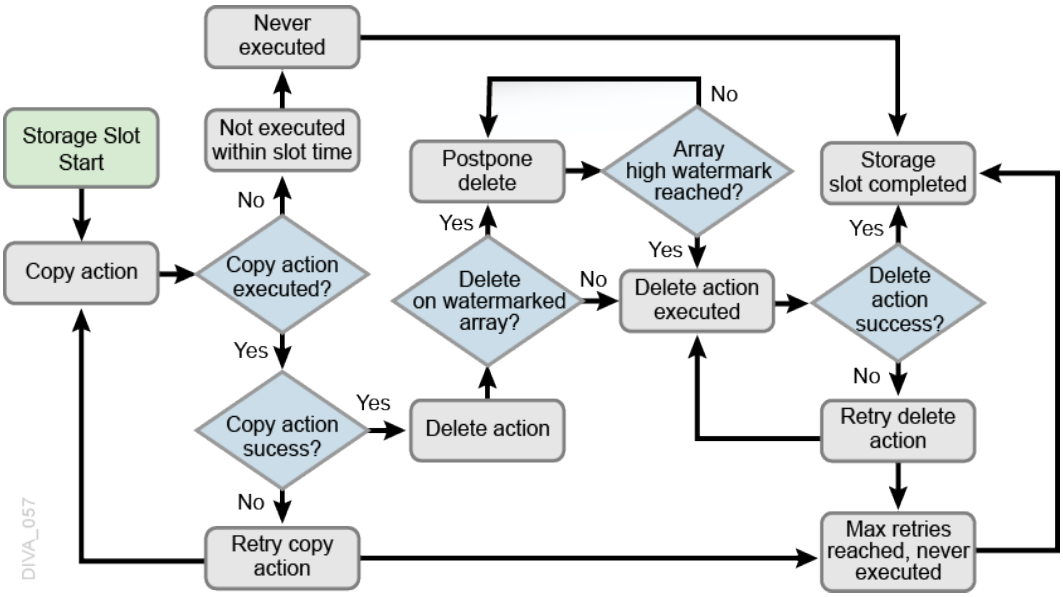


Storage Slot Workflow

Storage Slots also have a specific workflow that encompasses what DIVA Core will do in certain cases. The flowchart below displays the typical Storage Slot workflow. There are two possible outcomes:

- Success
- Failed Long

The following figure depicts the Storage Slot workflow:



Recommended Best Practices

The following sections describe Telestream recommended best practices for configuring and using the Storage Policy Manager.

SPM Configuration

This section identifies recommended best practices for SPM configuration.

Typical SPM Workflow

A typical SPM workflow consists of archiving Virtual Objects to disk, and then copying them to tape. In this type of workflow, the SPM is configured to keep the disk copy for a fixed time (non-watermarked mode), or until no more space is available (watermarked mode). Keeping a disk copy for some period allows faster access to the data.

See *Action Slot Workflow* and *Storage Slot Workflow* for Action Slot and Storage Slot workflows.

Non-Watermarked (Fixed Retention) Mode

In Non-Watermarked Mode (Fixed Retention) Mode, the disk copy is kept until the Slot End Time (for example, two 2 days from the Archive), at which time the disk instance is deleted; the corresponding DeleteInstance SPM action is sent to DIVA Core, and then updates the action to the Complete state.

See Managing DIVA Core Watermarks and Arrays for more information on watermarking arrays.

Watermarked Mode

In Watermarked Mode, the disk instance is not deleted when the Slot End Time is reached. Instead, it is flagged as expired by setting the corresponding DeleteInstance SPM action to the POSTPONED state. When the disk's space use reaches a configurable High Watermark (for example, 90%), SPM will remove as many expired instances from the disk as required to lower the space use down to the configured Low Watermark (for example, 60%).

In Watermarked Mode, Telestream advises not to set the disk's Slot End Time for an extended period. The reason for this is so that SPM does not run short of expired instances when a disk purge is triggered. However, the Slot End Time should be far enough in the future to allow the tape copy to complete. If a tape copy is not available at the time of the DeleteInstance, the action will update to a FAILED status instead of POSTPONED. SPM will refuse to expire an instance on disk when no alternate instance is available.

A common disk Slot End Time for Watermarked Mode is three hours (180 minutes), or 24 hours (1440 minutes) if the slot's schedule only allows copies to be executed at a particular time. Confirm that the array is large enough to accommodate additional archived instances for the slot's configured time. If 100 Gigabyte of data per hour is

archived at peak time, and the Slot End Time is configured to three hours, the array should be 500 Gigabyte (or larger).

See Managing DIVA Core Watermarks and Arrays for more information on watermarking arrays.

Choosing Appropriate Watermarks

You must choose your Watermarks carefully from the start. However, proper values are typically obtained after spending some time observing the system behavior. Telestream recommends starting with common values, and then fine-tuning them later. Typical common starting values are:

- HWM (High Watermark) = 90%
- LWM (Low Watermark) = 75%

The watermarks refer to the usage ratio of a particular array, not a disk. To compute an array's usage ratio, DIVA Core examines each disk in the array and divides the sum of each disk's used space by the sum of each disk's total capacity.

Example:

An array composed of two disks of the same size, one 100% full and one 50% full, is considered by DIVA Core to be 75% full. The same usage ratios with one disk of 2 Terabyte and one disk of 1 Terabyte will result in an 83% filling ratio for the array; 100% of 2 TB and 50% of 1 TB yields a total of 2.5 TB used, divided by a 3 TB total capacity.

See Managing DIVA Core Watermarks and Arrays for more information on watermarking arrays.

General Watermarking Rules

Telestream recommends that you adhere to the following general watermarking rules:

Do not set the High Watermark too high.

When the HWM is reached, SPM begins deleting expired instances. This process may take some time, especially if the instances are small, because the purge will require more DeleteInstance operations. Confirm that any archive activity will not store enough additional instances on the array during this process to fill it to 100%.

Example:

If the HWM is 90%, then the available space is 100 - 90 = 10%. If the array size is 2 Terabytes, this 10% represents 200 Gigabyte. If the SPM purge process encounters numerous small instances requiring deletion, the process will be slow and the archive activity may possibly store 200 Gigabytes (or more) during the purge's execution. This will fill the array to 100% during the process, and initiating Archive terminations.

If you experience this situation, try setting the HWM to a lower value.

Do not set the Low Watermark too low.

The lower you set the LWM, the shorter period disk instances will be kept on disk. The shorter period minimizes the chance to restore from disk, and the benefit of having disk instances.

If the array contains persistent data that cannot be purged (for example, Virtual Objects belonging to a Storage Plan that keeps disk instances for an unlimited time, or user files not belonging to DIVA Core), you must set the LWM accordingly. If the persistent data accounts for 40% of the array's capacity, and the LWM is set to 30%, the purges will never complete. However, SPM will still purge what it can.

See Managing DIVA Core Watermarks and Arrays for more information on watermarking arrays.

Setting Up Tape Groups in DIVA Core

Configuring tape groups is a different topic in DIVA Core. However, you must complete creating and configuring tape groups before configuring the Storage Plans. The number of Storage Plans required is usually the same as the number of tape groups you configured. Tape groups enable DIVA Core to physically separate archived content into different tapes, and typically creating one Storage Plan per tape group is necessary. All Storage Plans are typically setup to be the same, except that the copy goes to different a tape group. However, the more tape groups DIVA Core uses, the less efficiently content will be stored across tapes.

If complex Virtual Objects are going to be used in the system, you must setup tape groups containing tapes configured for AXF format. Complex Virtual Objects are not compatible with the Legacy formatted tapes or disks. You can store non-complex Virtual Objects using either Legacy or AXF format.

The following example illustrates how creating too many tape groups causes fragmentation of Virtual Objects across the Tape Group. Using fewer Tape Groups results in fewer Storage Plans to setup, less fragmentation across tapes, and is easier to maintain.

Having fewer tape groups will resolve the issue in the following example, and avoid fragmentation across the tape groups.

Example Configuration:

- 10 Tape Drives
- 30 Tape Groups with SetID=10
- 300 Total Tapes assigned to SetID=10

Results:

- Each Tape Group (in the worst case) will use at least ten tapes, and store files on each tape when a Virtual Object is archived.
- After Virtual Objects are archived to all 30 tape groups, all 60 tapes (total) are used.

- Over time, if any of the tape groups is heavily archived, and one of the tapes is 100% full, no more tapes are available.
- DIVA Core will run out of tapes, even though a lot of the tapes are still mostly empty (each containing only one Virtual Object), but cannot be used because it was assigned to a different tape group.

Valid Reasons for Creating Multiple Tape Groups

Generally, you should only create multiple tape groups for a good reason. Multiple tape groups will cause tape group fragmentation, resulting in some tapes not being filled, and storage space being wasted.

The following are several valid reasons for creating multiple tape groups:

- Long and short form materials should be stored on different tape groups. If small Virtual Objects are mixed with larger Virtual Objects on the same tape, access to the smaller Virtual Object will be delayed for an extended period until the larger Virtual Object restore is complete.
- Content that is deleted regularly from the archive should be stored in a different tape group than content that will never (or rarely) be deleted. Deleting from tape will cause tape fragmentation and the fragmented space cannot be used until the tape is repacked. If the two types of content are mixed together, deleting will cause more tapes to become fragmented, and repacking will be required more frequently and take longer.
- Online and backup copies of the same content should be on two different tape groups. Backup copies are meant to be removed from the tape library and stored in an Iron Mountain type of facility as a backup. However, if both copies are mixed in the same tape group, it is impossible to determine which tape contains the backup copy. The result is being unable to remove it from the library for offline storage.
- When requirements necessitate using tapes purchased by different departments and enforcing that each department uses only the tapes they purchase themselves, you must create tape groups for each department with a different set of tapes.
- Different storage formats must be assigned to different groups. For example, one group would be Legacy Format while another group is AXF Format. Complex Virtual Objects are not compatible with Legacy Format, and must be processed to and from AXF formatted medium. Non-complex Virtual Objects are compatible with both Legacy and AXF formats.

Invalid Reasons for Creating Multiple Tape Groups

Having too many tape groups will cause DIVA Core to work less efficiently, result in tape group fragmentation, and are more difficult to configure and maintain. Unless you have one of the previously described valid reasons, the recommended best practice for creating Tape groups is to use as few tape groups as possible. See Setting Up Tape Groups in DIVA Core for recommended best practices.

The following are several invalid reasons for creating multiple tape groups:

- You want to store different content in different tape groups because you think it is easier to manage. DIVA Core manages the tapes. When restoring a Virtual Object on a tape, DIVA Core automatically knows which tape the Virtual Object is on, and mounts that tape, or notifies you to insert the necessary tape into the library. It will not require you to figure out which tape is needed.
- Creating many tape groups because it is easier to search. DIVA Core performs searches using Virtual Object metadata stored in the database, not tape groups. The Tape Group only makes sure content is physically separated, and does not assist in searching functions.
- Creating multiple tape groups for cataloging. Users migrating from an analog tape environment tend to label what's recorded on each analog tape directly on the tape itself. A group of those tapes are then stored on different sections of a shelf. DIVA Core does not work efficiently this way, and this method should not be used.

Simple SPM Configuration Example

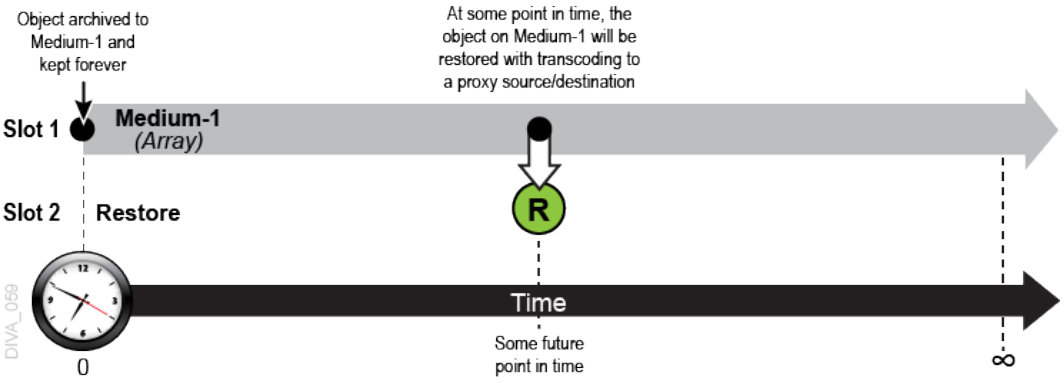
The following figure illustrates a simple SPM configuration example. The following occurs in this scenario:

1. The Virtual Object is archived to a watermarked array (Medium-1 with Watermark set to Yes) and held there for one day.
2. The Virtual Object is immediately copied to DIVA Core Medium-2, and stored there forever.
3. At the one day mark, the Virtual Object is marked for deletion from the Medium-1 array.

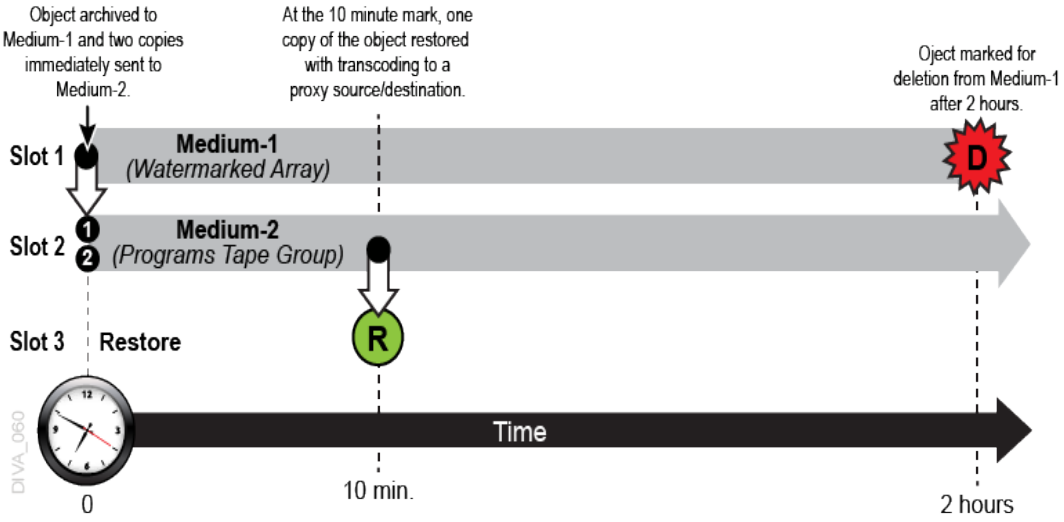
See *SPM Slot Types* and *Creating Slots* for more information about Slot Types and Slot configuration.

Additional SPM Configuration Examples

The following figure is an example of a more complex SPM configuration. In the first example, one copy is kept on the SPM monitored array forever. At some point in time, the Virtual Object will be restored with transcoding to a Proxy Server.

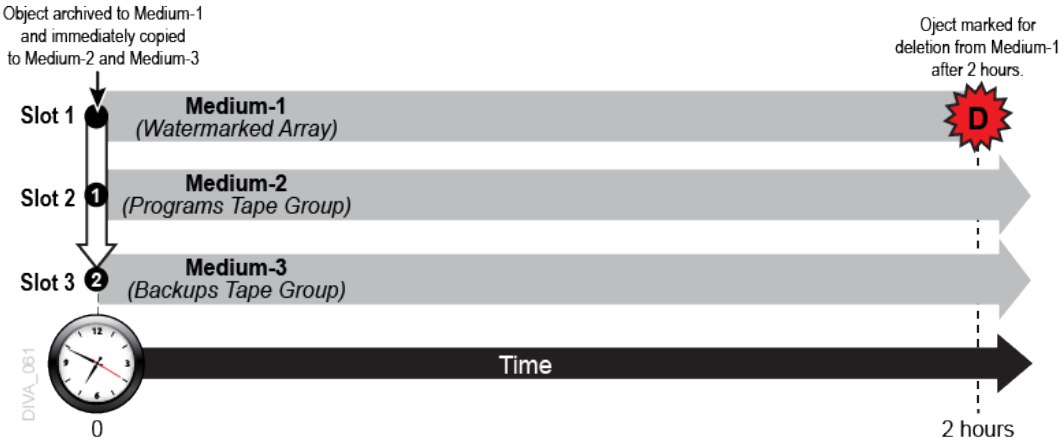


The following figure is an example of a more complex SPM configuration. In the second example, one copy is kept on the SPM watermarked array (*with Watermark set to Yes*) for two hours, and two copies are immediately made to a tape group named Programs. At the ten minute mark, one copy will be restored to a Proxy Server.



The following figure is an example of a SPM configuration with one copy left in the online tape group (Medium-2 named Programs), and a second backup copy made to a different tape group (Medium-3 named Backups) that is to be taken offline. In this example, one copy is kept on the SPM watermarked array (with Watermark set to Yes) for two hours, and two copies are made immediately. One copy is made to a tape group named Programs, the second copy is made to a different tape group named Backups.

The Backups Tape Group would typically be externalized from the DIVA Core system and sent to an Iron Mountain type of facility for offline storage.



Installation and Configuration

This chapter describes installation and configuration of SPM. References to installation and configuration procedures of additional software are included only to the extent required to allow interaction of those components with SPM. Refer to the specific additional software documentation in the Core Documentation library for full installation and configuration instructions for specific additional elements where necessary.

Topics:

Installation Prerequisites

There are several prerequisites required before installing SPM as follows:

- Core 7.7 and later only supports the following Core Database packages:
  - Windows only supports DIVAOracle Database Oracle 12c package OracleDivaDB_3-0-0_12_1_0_2_0_SE2_Windows_64-bit.zip and later. No previous Core Database package will work with Core 7.7 and later.

    The OracleDivaDB_3-0-0_12_1_0_2_0_SE2_Windows_64-bit.zip and later releases no longer include the 32-bit Oracle database client.
  - Linux only supports DIVAOracle Database package OracleDivaDB_3-0-0_12_1_0_2_0_SE2_OEL7_x86_64.sh and later.
- The Core Database user name and password must be the same ones used for the Core installation.
- Core must be installed and running.
- Core SPM Service must be installed and running.

SPM Configuration File Parameters

SPM uses a plain text file to store its configuration parameters. The configuration file, named spm.conf.ini, is supplied with SPM and is located in the %DIVA_HOME%\Program\SPM\bin folder. You must rename the file to spm.conf, and change the parameter values to meet your requirements using a plain text editor (for example, Notepad or Notepad++). Although all parameters are important, some are more important than others, and the following sections describe all parameters.

The following describes the parameters used in the SPM configuration file. You can also set some parameters using System Management App. However, the settings in the SPM configuration file called METADATA_ARCHIVE_TRANSFORMED_OBJECT_NAME and ALLOW_OBJECT_DELETION will always override any settings created through the System Management App.

The following parameter is for the SPM Windows Service Name:

SERVICE_NAME

This parameter identifies the Core SPM Service Name. You use this variable to specify the name of the Windows Service. This is useful when multiple instances of SPM are operational on the same server by giving a different name to each instance.

If you use this variable, the service name will be Core Spm - <SERVICE_NAME>. If you do not use this variable, the service name will be Core Spm.

The following parameters are for the Core:

DIVA_MANAGER_ADDRESS

This parameter identifies the IP address, or DNS Name, of the Core host computer. The default value is localhost.

DIVA_MANAGER_PORT

This parameter identifies the Core port number. The default value is 9000.

The following are connection parameters for the Core Oracle Database:

ORACLE_USER

This parameter identifies the Core Database user name. There is no default for this parameter, but the typical user name is diva.

ORACLE_PASSWORD

This parameter identifies the Core Database user password. There is no default for this parameter.

ORACLE_CONNECTION

This parameter identifies the Core Database connection string. This is usually the SQL*Net address name declared in the local Oracle client TNSNAMES.ORA file for the Core Oracle instance. The typical connect string is lib5.

ORACLE_POOL_SIZE

Caution: Do not change this value. Reducing the value can lead to database connection issues, and cause SPM to freeze.

This parameter identifies the maximum number of Core Database connections allowed by SPM. The default value is 5.

The following are parameters for the SPM Monitor:

DIVA_MANAGER_MONITOR_MAX_REQUESTS

SPM will check with the Core to find out how many requests are currently processing before it sends any requests. This parameter limits the number of requests (processing and pending from all sources) the SPM service will attempt to send to the Core. If the number of requests already processing on the Core is greater than this setting, SPM will not send any additional requests at that time. The valid range is 10 to 500, and the default value is 250.

DIVA_DELETE_MAX_REQUESTS

SPM will check with the Core to find out how many requests are currently being processed. If the number is less than the DIVA_MANAGER_MONITOR_MAX_REQUESTS value, SPM will verify how many Delete Virtual Object and Delete Instance requests are being processed. If the number is less than this parameter's value, SPM will send additional Delete requests equal to the difference between the number of Delete requests processing, and the value of this parameter.

Example:

```
DIVA_MANAGER_MONITOR_MAX_REQUESTS = 100
DIVA_DELETE_MAX_REQUESTS = 50
```

In this example, when SPM checks with the Core, if the Core is currently processing 75 requests total, and only 25 of them are Delete requests, SPM will send another 25 Delete requests to Core, because both numbers are less that the parameter's value in the configuration file. The valid range is 1 to 300, and the default value is 300.

This value identifies how many of this type of request are simultaneously processed from the Action Queue.

This value cannot exceed the value of DIVA_MANAGER_MONITOR_MAX_REQUESTS.

DIVA_RESTORE_MAX_REQUESTS, DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS, and DIVA_METADATA_ARCHIVE_MAX_REQUESTS

These parameters restrict the number of Restore, Transcode Archived and Metadata Archive requests SPM can submit to the Core simultaneously.

These parameters must all be enabled or disabled together.

The valid range is 0 to 300 and the default value is -1.

The value you use identifies how many of this type of request are allowed in the Action Queue. Using the default value of -1 disables these parameters. SPM will not execute these actions if the parameter is set to a value of 0.

Example:

```
DIVA_RESTORE_MAX_REQUESTS=3
DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS=52
DIVA_METADATA_ARCHIVE_MAX_REQUESTS=100
```

In this example, SPM will start normally using these configuration settings because they are all enabled, even though the values are different for each parameter (3, 52, and 100).

Example:

```
DIVA_RESTORE_MAX_REQUESTS=3
DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS= -1
DIVA_METADATA_ARCHIVE_MAX_REQUESTS= -1
```

In this example, SPM will fail to start using these configuration settings because the first one is enabled with a value of 3, but the next two parameters are disabled. They all must be either enabled or disabled, not a combination of both.

DIVA_MANAGER_MONITOR_DELAY

This parameter identifies the number of seconds between checks of the number of requests (processing and pending) on the Core. The valid range is 1 to 600, and the default value is 30.

DIVA_MANAGER_MONITOR_ACTION_DELAY

This parameter identifies the number of seconds to delay an action for execution if the Core is processing more than the value of the DIVA_MANAGER_MONITOR_MAX_REQUESTS parameter during its execution. The valid range is 1 to 3600, and the default value is 60.

The following are parameters for the SPM Controller:

ACTION_STEP_WATCH_DELAY

This parameter identifies the minimum delay (in seconds) before the next check on the request status. The valid range is 1 to 60, and the default value is 10.

ACTION_QUEUE_SIZE

This parameter identifies the maximum size of the Action Queue, including the number of actions that can be stored in the queue. The valid range is 1 to 300.

The following are rest time parameters for SPM working tasks:

SPM_EXECUTION_THREAD_REST

This parameter identifies the resting time for the Execution Task in seconds. The valid range is 0 to 600, and the default value is 5. Entering 0 disables the task completely.

SPM_UPDATE_THREAD_REST

This parameter identifies the resting time for the Update Task in seconds. The valid range is 0 to 600, and the default value is 20. Entering 0 disables the task completely.

SPM_LOAD_THREAD_REST

This parameter identifies the resting time for the Load Task in seconds. The valid range is 0 to 600, and the default value is 20. Entering 0 disables the task completely.

Note: The Recovery Load Task will never rest; it stops upon completion of all work.

The following are Row Chunk Size parameters that SPM will process simultaneously:

RECOVERY_ACTIONS_CHUNK

This parameter identifies the number of actions SPM will process (from the Core Database) simultaneously during each call of the Recovery Task. The valid range is 1 to 65535, and the default value is 500.

UPDATE_ACTIONS_CHUNK

This parameter identifies the number of actions SPM will process simultaneously during each call of the Update Task. The valid range is 1 to 500, and the default value is 50.

GET_ACTIONS_CHUNK

This parameter identifies the number of actions SPM will process simultaneously during each call of the Load Task. The valid range is 1 to 100, and the default value is 25.

UPDATE_ACTIONS_NEXT_UPDATE

This parameter identifies the time (in minutes) added to the action's date of next update after it has been updated by the Update Task. Current releases of Core SPM is more dynamic in nature than previous release, and constantly reexamines actions. The date of next update is used internally to sort the actions to be reexamined. The valid range is 10 to 65535, and the default value is 20160 (two weeks).

UPDATE_ACTIONS_RETRY_FAILED_DELAY

This parameter identifies the delay (in minutes) before a long retry of a failed action. After this value is reached, the Update Task will reschedule all actions in the FAILED LONG state to be executed again. This parameter configures the retry of SPM actions and will not retry the action if its corresponding slot is modified after the action is put to a FAILED_LONG state - it will wait for the UPDATE_ACTIONS_NEXT_UPDATE to update the actions according to the latest slot changes before executing the action. The valid range is 10 to 65535, and the default value is 720 minutes.

T**he following are SPM processing options:**

ALLOW_OBJECT_DELETION

Caution: This parameter causes complete Virtual Object deletion when enabled!

A Virtual Object can be deleted when a Storage Plan has no slot at a given time. You must manage this feature, and define the Storage Plans, carefully because you can lose many Virtual Objects according to the Storage Plan.

The valid settings are true or false, and the default value is false.

You set this field to true to enable Virtual Object deletion. The Storage Plan definition in the Core Database can either allow or deny deletion. If deletion is denied, configuring a Storage Plan to enable deletion will have no effect.

Note: The setting in the SPM configuration file for this parameter overrides any settings created through System Management App.

For example, if there is slot starting at day 0 and ending at day 30, and another slot starting at day 61 with an unlimited retention, Virtual Objects will be deleted after 30 days because there are no slots during the second month (days 31 to 60).

DELETE_OBJECT_ONLY_LAST_INSTANCE

This parameter works with the ALLOW_OBJECT_DELETION parameter, and forces SPM to Delete Virtual Object at the end of the Storage Slot. Deletion only occurs if this Virtual Object is the last instance in the entire Core system, and this last instance exists in the Storage Slot medium that has the highest Slot End Time among all Storage Slots of the Storage Plan. Otherwise, SPM will never perform a Delete Virtual Object at the end of the Storage Slot, it will always only do a Delete Instance instead. Refer to *SPM Delete Virtual Object Behavior* for more detailed information.

METADATA_ARCHIVE_TRANSFORMED_OBJECT_NAME

This parameter defines whether SPM will use the original, or transformed, Virtual Object name as the target Virtual Object name in a METADATA_ARCHIVE action. This setting overrides any configuration performed through System Management App.

The default setting is true. When this value is true, SPM uses the transformed Virtual Object name. When this value is false, SPM uses the original Virtual Object name.

The following is a DSM task option:

DSM_DIW_REQUEST_PRIORITY

This parameter identifies the priority of the requests produced by DELETE INSTANCE actions. The valid range is 1 to 100, and the default value is 15.

The following is a Rest Time parameter for DSM working tasks:

DSM_SPACE_MONITOR_THREAD_REST

This parameter identifies the resting time for the DSM Task in seconds. The valid range is 0 to 600, and the default value is 10. Entering 0 disables the task completely.

The following is a Row Chunk Size parameters that DSM will process simultaneously:

DSM_ACTIVATE_ACTIONS_CHUNK

This parameter identifies the number of actions DSM will process at a given time during each call of the procedure. The valid range is 1 to 65535, and the default value is 50.

The following are parameters for disk arrays being monitored:

ARRAY{number}

This parameter specifies watermark values for disk arrays. For each array, you must create an ARRAY{i} parameter, where {i} identifies the array number. The first array number is 1. The {i} must always be increased only by one.

The information for the Array is as follows:

```
ARRAY_NUMBER = "array name"; "watermark low-level percent";
"watermark high-level percent" ["mounted disk path1", "mounted
disk path2", and so on]
```

ARRAY_NUMBER

This parameter identifies the name known to SPM for this array (for example, ARRAY1).

“array name”

This parameter identifies the name of the disk array as it is named in Core. It is not the SPM medium name.

“watermark low-level percent”

This parameter identifies a number from 0 to 100 representing the lower array space setting use percent. DSM will attempt to reach this value when cleaning.

“watermark high-level percent”

This parameter identifies a number from 0 to 100 representing the higher array space use percent. DSM will start the cleaning process when the array space use is higher than this level.

[“mounted disk path1”, mounted disk path2”, and so on]

This parameter identifies the mounting points of the array disks. Either none, one, or many, separated by a comma.

- No spaces are allowed.
- If one or many, DSM will access arrays through the file system (old) interface.
- If none, DSM will access arrays through the Core API (new) interface (recommended).

If the monitored array is a password protected network share, the following syntax is allowed as a mount point:

```
cifs://user:pwd@\\nas\share
```

When the SPM Service is configured with a CIFS array, the log in of the SPM Service must be changed from the default Local System to a valid Windows user.

Core Actors in the Linux operating system do not support UNC paths for CIFS managed disks. However, you can define a local path to a mounted SMB share. UNC paths are supported for SMB managed disks if the UNC path is directly mounted on the Windows Core Actors.

Use the following steps to change the SPM Service login:

1. Run the Windows Services application (services.msc).
2. Select the SPM Service requiring the log in change.
3. Right-click the SPM Service, and then select Properties.
4. Select the Log On tab in the Properties window of the SPM Service.
5. Select the This account option.
6. Enter a valid Windows user name and password.
7. Click Apply (or OK) to commit the changes.

Examples:

ARRAY1 = ShortClips;75;90 (this is the recommended syntax)

ARRAY2 = Raid_001;60;85;X:\;Y:\;Z:\ (this is the legacy syntax)

The following is a Trace Log level parameter:

TRACE_LEVEL

This parameter identifies the level of tracing to use for the logs. The valid range is 1 to 2, and the default value is 2.

Caution: Setting this parameter to 1 will generate large volumes of trace information, and must be used for debug and validation purposes only!

- Setting this parameter to 1 will trace entry and exit points in all important functions.
- Setting this parameter to 2 will produce a normal, production level, trace.

Request Type Distribution

SPM contains three Request Type Distribution options that are defined in the SPM configuration file. These parameters restrict the number of Restore, Transcode Archived, and Metadata Archive actions that can be executed simultaneously. The default values for the following parameters are -1 in the SPM configuration file:

- DIVA_RESTORE_MAX_REQUESTS
- DIVA_TRANSCODE_ARCHIVED_MAX_REQUESTS
- DIVA_METADATA_ARCHIVE_MAX_REQUESTS

Acceptable values for these three parameters are -1, 0, and positive integers. Leaving the configuration file settings for these parameters to the default (-1) causes SPM to ignore the settings, and the defined behavior will not be changed.

You should set all, or none, of these parameters to a value higher than the default (-1) setting. For example, you cannot set DIVA_RESTORE_MAX_REQUESTS=3, but keep DIVA_TRANSCODE_ARCHIVED_MAX_REQUESTS= -1. If only one or two parameters are defined with other values, SPM will refuse to start because of the incorrect configuration.

Note: Whenever you change these parameter values, you must restart SPM for the new values to take effect.

There are several rules for Request Type Distribution as follows:

- SPM must not execute more Restore requests than the DIVA_RESTORE_MAX_REQUESTS value simultaneously. If the DIVA_RESTORE_MAX_REQUESTS=0, SPM will not execute any Restore requests.
- SPM must not execute more Transcode Archived requests than the DIVA_TRANSCODE_ARCHIVED_MAX_REQUESTS value simultaneously. If the DIVA_TRANSCODE_ARCHIVED_MAX_REQUESTS=0, SPM will not execute any Transcode Archived requests.
- SPM must not execute more Metadata Archive requests than the DIVA_METADATA_ARCHIVE_MAX_REQUESTS value simultaneously. If the DIVA_METADATA_ARCHIVE_MAX_REQUESTS=0, SPM will not execute any Metadata Archive requests.
- SPM will attempt to reserve the configured number of actions in the SPM queue to execute the corresponding number of the requests (if available). The number of simultaneous Restore requests may be less than DIVA_RESTORE_MAX_REQUESTS, even if some Restore actions are ready to be executed immediately. This configuration must be observed for less than 50 percent of the execution time.

Example:

```
DIVA_RESTORE_MAX_REQUESTS=3
DIVA_MANAGER_MONITOR_MAX_REQUESTS=100
```

Workflow:

- If Restore Actions exist and are scheduled, SPM will execute three Restore requests at the same time (most of the time).
- SPM will never execute more than three Restore requests.
- Sometimes (less than 50 percent of operational time) SPM will execute 0, 1 or 2 Restore requests when Restore Actions exist, and are scheduled.
- If no Restore Actions exist, and are scheduled, other actions will be executed so the total amount of requests executed will be 100 (not 97).

By default, the Execute process will perform Delete Actions first, and then other request types by Slot Priority. If this feature is configured, SPM will still perform Delete Actions first, and then decide the number of Copy, Restore, Metadata Archive, and Transcode Archived requests to send, and then for each type of request, sort the actions by Slot Priority.

Example:

- Two Copy Slots have the defined priorities 90 and 95, and 1000 Copy Actions are to be executed.
- There are two Restore Slots with priorities 30 and 35, and 100 Restore Actions to be executed.
- DIVA_RESTORE_MAX_REQUESTS=3
- DIVA_MANAGER_MONITOR_MAX_REQUESTS=100

Workflow:

- SPM will disregard the priority between the slot types, so 97 Copy requests will be scheduled, and 3 Restore requests will be scheduled.
- SPM will schedule Restore requests with the priority of 35 first.
- Delete Virtual Object and Delete Instance requests always have higher priority.

Storage Plan Definition

SPM requires configuration of Storage Plans before execution. You configure the SPM Storage Plans using System Management App.

There are two tabs in System Management App related to SPM configuration; Storage Plans and Slots.

System Management App Storage Plans Tab

The Storage Plans tab in System Management App has three sections:

- Storage Plans
- Media Groups
- Filters

Each of these sections is configured independently, but associates with other sections to complete the configuration.

Recommended Best Practices

Telestream recommends that you use the following order when configuring SPM through System Management App:

1. Create the Storage Plans.
2. Create the Media Groups.
3. Create the Filters.
4. Create the slots (using the Slots tab).

Creating the Storage Plan

The first step in configuring SPM is to create the Storage Plans. Use the following procedure to create a storage Plan:

1. Click the + in the Storage Plans section of the Storage Plans tab in System Management App.
2. When the Add New Row dialog box opens, begin configuring the new Storage Plan. Enter an name in the new Storage Plan Name field.
3. Use the lists to select the values for the parameter fields as follows:

   Storage Plan Name

   This parameter identifies the name of the Storage Plan.

Allow Last Instance Deletion

Caution: You must use this parameter with caution. Data loss can occur if set incorrectly.

This parameter determines whether SPM is allowed to delete the last instance of a Virtual Object when there is only one instance remaining on the Core system. Set to *Y* to allow, or *N* to deny, deletion of the last instance of a Virtual Object.

This parameter only applies to Storage Slots, and is overridden by the configuration of ALLOW_OBJECT_DELETION in the SPM configuration file.

Please specify origin (Internal/External)

Set to *I* for internal, or *E* for external origins. This parameter is typically used for all SPM actions. The internal (I) setting is for Virtual Objects contained within the Core system and is the typical selection. The external (E) setting is mainly used during data migration in combination with the SPM API. Only Telestream should use the external setting in this release. For example, for use with Core Automatic Data Migration from Avalon to Core.

Tape Group/Array Name

Select the default media from the list of media managed in the Core system. The Core uses the selected media for Virtual Object placement when a new Virtual Object is found, and the Storage Plan in use has an active Storage Slot. This placement occurs when an Archive Request is submitted to the Core without a destination Media, and only a Storage Plan selected in the request.

4. Click *OK* to save the Storage Plan. The new Storage Plan will be displayed in the Storage Plans section of System Management App.

Creating Mediums

The Media Groups Section has two purposes:

- Any medium that is defined in the Media Groups section of the System Management App Storage Plans tab is available for use by SPM. If a new Virtual Object is placed on a medium not included in the Media Groups list, it will not be managed by SPM and is assigned the SPM_DEFAULT Storage Plan.
- To select a medium as the target for actions when creating slots, it must be defined in the Media Groups list.

Use the following procedure to create a Media Group:

1. Navigate to the System Management App Storage Plans tab.
2. Click the + icon on the top of the Media Groups section.
3. When the Add New Row dialog box appears, begin configuring the new mediums.

4. Enter the information in the appropriate fields, and use the menu lists to select the remaining options as follows:

   Medium Name

   Enter the *Medium Name* in the Medium Name field.

   Storage Name

   Enter the desired *Storage Name* in the Storage Name field. The recommended Storage Name is *DIVA*. **This value is not currently in use.**

   Tape Group/Array Name

   Select the Tape Group or Disk Array to be associated with this medium from the list.

   The selections available in the list are determined by what media is defined in the Core system.

   Watermarked

   Monitors the Medium's usage (available space, used space, and so on). Select *Y*, *N*, or *M* to specify whether to apply watermarking to this medium. Refer to Sections *Choosing Appropriate Watermarks* and *General Watermarking Rules* for detailed information on Watermarking.

   In general it is recommended that slots are not deleted, but disabled instead. Deleting a slot does not remove the content from the Core system.

   In the case of a Storage Slot, setting the Slot End Time to 0 will force SPM to delete any contents created by this slot and once all the contents are removed, the Storage Slot can be deleted or disabled. If retaining the content in the Storage Slot is

desired, but no further processing of future actions is required, disabling the slot instead of deleting it is the safer way to do this.

The Watermark-Based Disk Cleaning Management options are as follows:

Yes (Y)

This selection applies watermarking.

No (N)

This selection does not apply watermarking.

Mixed (M)

This selection applies a combination of watermarking. The disk cleaning action taken depends on what event occurs first; either the slot reaches it's End Time, or the High Watermark is reached.

Mixed Mode only works if Once Only is set to *Y* (yes) for the slot.

Watermarks are only used for Storage Slots, and only for disk arrays configured as Mediums.

Disk Cleaning Strategy

Defines the cleaning strategy to perform on Virtual Objects marked for deletion on the Array when the high watermark is reached. See *Watermark Based Disk Cleaning Management* for detailed information on disk cleaning management.

There are two options available:

By Last Access Time

Older Virtual Objects are cleaned up before recently accessed Virtual Objects.

By Object Size

The largest Virtual Objects are cleaned up first until the Low Watermark is reached.

5. Click *OK* to complete the process.

   The new Media Group is now displayed in the Media Groups section of the Storage Plans tab.

Creating Filters

Filters determine whether an action must be performed on a Virtual Object, or if it should be ignored. Filters only functions for mediums defined in the Mediums list and monitored by SPM.

When a new Virtual Object is archived into Core, and it matches a defined filter, a Storage Plan is assigned to that Virtual Object as follows:

- If the new Virtual Object does not meet any filter criteria, the default Storage Plan will be assigned.
- If a Virtual Object satisfies multiple filters, it will be assigned the Storage Plan of the matching filter with the lowest Filter ID.

Use the following procedure to create a filter:

1. Navigate to the System Management App Storage Plans tab.
2. Click the + icon on the top of the Filters section.
3. When the Add New Row dialog box appears, begin configuring the new filter.

   Leaving the default * in Object Collection, Object Source, and Object Name fields causes no filtering to be identified for these parameters.
4. Enter the information in the appropriate fields, and use the menu lists to select the remaining options as follows:

   Filter Name

   Enter the *Filter Name* in the Filter Name field.

   Id

   The *Filter ID* is auto-generated by SPM starting with Filter 0.

   Media Filter

   Use the menu to select the Mediums you previously created.

   Object Collection Filter

   Enter any Virtual Object Collection filters you require. Leaving the default * will cause no filtering for this parameter.

   Object Source Filter

   Enter any Virtual Object Collection filters you require. Leaving the default * will cause no filtering for this parameter.

   Object Name Filter

   Enter any Virtual Object Collection filters you require. Leaving the default * will cause no filtering for this parameter.

   Min. object size

   Enter the minimum Virtual Object size.

   Max. object size

   Enter the maximum Virtual Object size.

   Size units

   Use the menu to select the size units (B, KB, MB, and GB). The initials refer to Byte, KiloByte, MegaByte, and GigaByte respectively.

   Storage Plan

   Use the menu to select the Storage Plan to associate with this filter.
5. Click *OK* to complete the process.

   The new Filter is now displayed in the Filters section of the Storage Plans tab.

Alternate Methods of Assigning Storage Plans to a Virtual Object

There are several alternate methods of assigning a Storage Plan to a Virtual Object including the following:

- Assigning a new Storage Plan to a Virtual Object using the Core GUI.
- Using a Core API; C++, Java, DIVA Enterprise Connect. See the appropriate API documentation in the Core Documentation library.
- Including a Storage Plan name when submitting an Archive request forcefully applies a Storage Plan to the Virtual Object and avoid the filters.

Media Mapping enables you to configure and alternate destination media, and to forcefully apply a Storage Plan, avoiding filters, when a Virtual Object is archived to a specific tape group or disk array.

You configure Media Mapping in System Management App under the Sets, Groups & Media Mapping tab, in the Media Mapping section as follows:

1. Open System Management App and navigate to the Media Mapping section under the DIVA Configuration > Storage Plans menu item.
2. Click the + icon to open the Add new row in Media Mapping dialog box.
3. Enter the appropriate information in the fields as follows:

   Id

   This field is automatically system generated and is not editable.

   Name

   Enter the name to use for this mapping in the Name field.

   From

   Use the menu list to select the media to use for this mapping.

   Map to Media

   Use the menu list to select the Destination Media for this mapping.

   Map to Storage Plan

   Use the menu list to select the Storage Plan to use for this mapping.

For example:

- Media Array1 is mapped to Array2. In this scenario, Virtual Objects archived to the media named Array1 will be transferred to the media named Array2 as its destination. If there is a filter configured with destination media as Array2, the Storage Plan configured in that filter will be assigned to the archived Virtual Object.
- Media Array1 is mapped to the Storage Plan named storagePlan1. In this scenario, Virtual Objects archived to the media named Array1 will be assigned the Storage Plan named storagePlan1, and avoid filters.
- Media Array1 is mapped to Array2, and also the Storage Plan named storagePlan1. In this scenario, Virtual Objects archived to the media named Array1 will be transferred to the media named Array2 as its destination, assigned the Storage Plan named storagePlan1, and avoid filters.

Creating Slots

The following sections describe the process for creating each different type of slot.

Configuring Storage Slots

The Storage Slot is the only slot that has two actions associated with it. It will create a copy of the Virtual Object when it starts and delete it when the slot ends, unless the medium used is watermarked. See *Watermark Based Disk Cleaning Management* for watermarking details.

The copy action will occur only if the slot destination medium does not already have the required number of instances identified in the slot configuration. The Delete Action will only occur if the slot is specifically configured to perform the Delete Action (slot has an ending time defined).

Use the following procedure to configure a Storage Slot:

1. Open System Management App, and navigate to the Slots tab under the DIVA Configuration > Storage Plans menu item.
2. Click the + icon on the top right of the Slots section.
3. When the Slot Configuration dialog box opens, configure the options as follows:

   Storage Plan

   Use the menu list to select the Storage Plan to be associated with this slot.

   Request Type

   Select the type of request for this slot (Storage, Transcode Archived, Metadata Archive, or Restore). Select *STORAGE* in this case.

   Slot Name

   Enter a name for the slot in the Slot Name field.

   Slot Begin Time (Minutes)

   This parameter identifies when the action will occur. If set to zero the action is initiated as soon as the Virtual Object is archived. If set to another number, for example 10 (minutes), the action will execute 10 minutes after the Virtual Object is archived.

Slot End Time (Minutes)

Caution: If the Slot Begin Time and Slot End Time are both set to zero, then the behavior is unknown, except for Storage Slots.

Setting this parameter to -1 indicates that the slot will never end (it is permanent). If set to another number, for example 10 (minutes), the Delete Action will execute 10 minutes after the Virtual Object is archived.

If a Storage Slot's Slot Begin Time and Slot End Time are the same (for example, 0, 0 or 10, 10), the slot is what may be referred to as an idle slot. SPM creates a Copy and a DeleteInstance Action and these actions get updated but NOT executed.

The only exception is if the medium is watermarked. If the medium is watermarked then the DeleteInstance Action will be executed when a disk clean-up is necessary and it will go through the usual POSTPONED through COMPLETED cycle.

In practice, creating an idle slot is useful for mediums where an external tool creates instances (not SPM), but use of the SPM's watermark-based clean-up mechanism is desired.

Example (typical slot):

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): 20

Workflow:

The original Virtual Object is archived onto the source medium.

Ten minutes later, the slot actions begin and it is copied to the target medium.

SPM deletes the copy from the target medium 10 minutes later.

Example (idle slot):

Slot End Time (Minutes): 0

Slot Begin Time (Minutes): 0

Workflow:

The Virtual Object is archived on medium Array1, and the idle slot is defined for medium Array2. An Array2 Copy Action is created, and the status is set to SCHEDULED. Then, an Array2 DeleteInstance Action is created, and the status is set to COMPLETED.

An external tool creates an instance on Array2 using the Copy function. The Array2 Copy Action status is set to COMPLETED, and the Array2 DeleteInstance Action status is set to POSTPONED.

The Array2 hits the High Watermark. The Array2 DeleteInstance Action is executed, and the status is set to COMPLETED.

The Array2 Copy Action status remains as COMPLETED.

Enabled

Determines whether the slot is active. Selecting *Y* indicates the slot is enabled, and *N* indicates it is disabled. When a slot is disabled its actions are not executed, how-

ever they are still created, and the status is updated. When the slot is enabled again, pending actions are executed immediately.

Examples:

A new Virtual Object that is archived and assigned to a Storage Plan with a Tape Storage Slot will have the Tape Slot's Copy Action created, but not executed. Therefore, no Copy request is sent to the Core.

A DeleteInstance Action reaching its execution time will not send the Delete Instance request to the Core, but if somebody manually deletes that instance using the System Management App, the action will be updated to the COMPLETED status.

Request Execution Begin Time (HH:MM)

This field is the slot execution window opening time in 24 hour time. If set to 00:00 the slot opens at the beginning of each day. If set to another time, for example 08:00, it will open at that time (in this case 8:00 am).

Request Execution End Time (HH:MM)

This field is the slot execution window closing time in 24 hour time. If set to 24:00 the slot closes after each day. If set to another time, for example 17:00, it will close at that time (in this case 5:00 pm).

Setting the Request Execution Begin Time and Request Execution End Time to something other than an entire 24 hour period will allow for scheduling of different functions through SPM at different times of the day (for example overnight Requests). If a Virtual Object is archived to the source medium outside of the designated time frame for the slot, even though the Virtual Object meets the Filter and Storage Plan criteria, no actions from this particular slot will occur.

If the original Virtual Object was archived to the source medium while the slot was inactive (closed) and has passed the Slot End Time, when the slot finally becomes active (open), no Delete Action will occur.

Note: This is not the case for Storage Slots. For Storage Slots the DeleteInstance Action will still be executed.

Example (non-Storage Slots):

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): 20

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium at 07:00 (while the slot is closed). If the slot had been open at 07:00 when the new Virtual Object was created it would be copied to the target at 07:10 and then deleted from the target at 07:20. Because the slot was not open when the new Virtual Object was created, and the

Slot End Time has passed when this slot opens at 08:00, no actions will be taken on the Virtual Object.

If a new Virtual Object is created on the source medium at 14:15 today, then tomorrow when this slot opens, no actions will be taken on the Virtual Object because the Slot End Time has passed.

If a Virtual Object is created on the source medium at 07:59 (before the slot opens), at 08:09 (10 minutes after the Virtual Object existed on the source) the Virtual Object will be copied to the target medium because the Slot End Time has not yet passed. Then at 08:19 (20 minutes after the Virtual Object existed on the source) the Virtual Object instance on the target will be deleted.

Example (Storage Slots):

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): 20

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium while the slot is closed. A Copy Action is created and the status is set to COMPLETED. A DeleteInstance Action is created and the status is set to SCHEDULED. No actions are executed because the slot is closed.

The disk instance expires after 20 minutes.

The slot opens the next day. The DeleteInstance Action is executed, and the status is set to COMPLETED. The Copy Action status remains as COMPLETED.

Request priority

This field identifies the priority of the request and order of execution when the action is submitted to the Core.

Medium Associated with Slot

Select the target medium to associate with the slot from the list.

Req. Number of Instances

The number of Virtual Object instances that should exist on the target medium. If the actual number of instances is less than this setting, additional copies will be created. This parameter is only applicable to Storage Slots.

Example:

Req. Number of Instances: 2

Workflow:

One Virtual Object instance already exists on the target medium.

Because there is only one instance existing, and this parameter is set to 2, another

copy will be made to another disk in the array, or to another tape in the Tape Group.

Once Only

Determines whether the slot can run multiple times. Setting this to N indicates it can run multiple times, while setting this to Y indicates it will only run once.

Note: This parameter is only applicable for Storage Slots and must be set to ***Y*** when using Mixed Mode Watermarking.

Example:

Once Only: N

Workflow:

A Virtual Object was copied to the target medium, but somebody manually deleted the copied instance. This slot will realize the Virtual Object does not exist on the target medium and run again, resulting in another copy on the target medium being created.

Example:

Once Only: Y

Workflow:

A Virtual Object was copied to the target medium, but somebody manually deleted the copied instance. This slot will not run again, resulting in no Virtual Object instance on the target medium (because it had been manually deleted).

If the Slot End Time for deletion has been set, when the Slot End Time is reached, SPM will automatically mark the Virtual Object instance on the target medium as being deleted because it was manually deleted before the Slot End Time was reached.

Click *OK* to complete the process. The new slot will be displayed in the Slots tab in System Management App.

Configuring Nearline Storage Slots

Starting with DIVAdirector 5.3, you can specify two new Quality of Service (QOS) parameters called NEARLINE_ONLY and NEARLINE_AND_DIRECT through the API or the System Management App in Restore and Multiple Restore requests. The default QOS for a Restore or Multiple Restore request is NEARLINE_AND_DIRECT. The default QOS for all other requests remains DIRECT_AND_CACHE.

When the Core receives either of the new QOS values, it will initiate a DIRECT restore from any available disk instance, regardless of whether the disk instance is located on a disk with a Core Actor-Disk connection that is configured for Nearline. If no disk instance is found, the Core will not terminate the request; instead it creates a disk instance and proceeds in the following sequence:

1. Check for NEARLINE disks during disk selection.

2. If no disks are available, then for QOS:

   NEARLINE_ONLY

   The workflow will terminate and an error is generated stating No AVAILABLE Actor-Disk connection is configured with Nearline storage.

   NEARLINE_AND_DIRECT

   DIRECT RESTORE will be used.
3. Reserve space for permanent storage during the restore.
4. Create a permanent disk instance in memory.

   If the disk instance can not be created, then for QOS:

   NEARLINE_ONLY

   The workflow will terminate.

   NEARLINE_AND_DIRECT

   DIRECT RESTORE will be used.

   If the disk instance can be created, the workflow will proceed to the next step.
5. Write the disk instance to the Core Database upon success or failure of the Restore.
6. Save component checksums for the new disk instance.
7. Update storage capacity.
8. Update last access date of Virtual Object instance, if not already updated.

If all disk instances are busy, a delayed solution will be generated.

The Core will terminate a Restore Instance request with a QOS of NEARLINE_ONLY, or a QOS of NEARLINE_AND_DIRECT, by sending an error message to the System Management App stating Nearline and Nearline & Direct QOS are not supported for Restore Instance, and sending an Invalid Parameter error code to the API.

Nearline storage is defined using the same technique for defining the disk use. The Nearline type is defined by specifying one of two uses under Actor-Disk Connections called STORAGE_AND_NEARLINE and CACHE_AND_STORAGE_AND_NEARLINE. Any disk defined with one of these uses can be used for any of these operations.

An SPM Storage Slot supports this workflow for purging disk instances created during Restore requests.

- If the disk instances are created on the same medium used in the Storage Slot, a Delete Instance action in the Storage Slot will be rescheduled.
- If the disk instance is created on a different medium, you must set the Storage Slot's Slot Begin Time and Slot End Time to 0 for the associated medium.

SPM supports purging of disk instances created by the Core during Restore requests in the following two work flows.

1. Nearline Instances are stored on the same medium, which is also used in an SPM Storage slot.
   - In this work flow it is assumed that the medium used for Nearline Storage must also be used in a SPM Storage Slot.
   - The Storage Slot must have Once Only set to N.
   - Nearline instances are not copied until the delete instance action is completed for the Storage Slot on the medium.
   - The medium used must be watermarked.
   - The disk cleaning strategy for the medium must be based on the Object Last Access Time.
   - After the action is put into the SCHEDULED state, SPM will start processing and put it into the POSTPONED state until the disk reaches its High Watermark.
   - After the disk reaches its High Watermark value, all of the Delete Instance Actions in the POSTPONED state will be executed based on the Disk Cleaning Strategy configured for the medium. In this case it will be the Object Last Access Time.
2. A separate medium is used for Nearline Storage.
   - In this workflow a Storage Slot is configured on the medium with the Slot Begin Time and Slot End Time is set to 0.
   - The Storage Slot must have Once Only set to N.
   - Setting the Slot Begin Time and Slot End Time to 0 causes SPM to generate a Storage Slot Action for this medium. However, it will be in the COMPLETED state by default.
   - The medium used must be watermarked.
   - The Disk Cleaning Strategy for the medium must be based on Object Last Access Time.
   - After the action is put into the SCHEDULED state, SPM will start processing and put it in the POSTPONED state until the disk reaches its High Watermark.
   - After the disk reaches its High Watermark value, all of the Delete Instance Actions in the POSTPONED state will be executed based on the Disk Cleaning Strategy configured for the medium. In this case it will be the Object Last Access Time.

A medium must not be configured for only one of previously mentioned workflows. Every medium that has a disk used for STORAGE_AND_NEARLINE or CACHE_AND_STORAGE_AND_NEARLINE must be configured with any one of the mentioned workflows for SPM to purge Nearline instances.

Configuring Transcode Archived Slots

Transcode Archived Slots require configuration on the Slots Tab of System Management App and contain only a single action. This slot will take an existing clip in a Core Virtual Object, transcode it to another format, and archive the resulting clip as a new Virtual Object.

Because a Transcode Archived request creates a new Virtual Object in the Core System, it requires directions as how to name the new Virtual Object (Transformation Rules).

Note: Each Core Actor defined as a Transcoder must also be defined as a ***LOCAL*** type Server. Linux based Core Actors only support Telestream Vantage transcoding operations.

Use the following procedure to configure a Transcode Archived Slot:

1. Open System Management App and navigate to the Slots tab under the DIVA Configuration > Storage Plans menu item.
2. Click the + icon in the Slots section of the screen.
3. When the Slot Configuration window opens begin configuring the new slot by selecting TRANSCODE_ARCHIVED from the Slot Request Type list, and then continue with the remaining configuration parameters as follows:

   Storage Plan

   Use the menu list to select the Storage Plan to be associated with this slot.

   Request Type

   Select the type of request for this slot (Storage, Transcode Archived, Metadata Archive, or Restore). Select TRANSCODE_ARCHIVED in this case.

   Slot Name

   Enter a name for the slot in the Slot Name field.

   Slot Begin Time (Minutes)

   This parameter identifies when the action will occur. If set to zero the action is initiated as soon as the Virtual Object is archived. If set to another number, for example 10 (minutes), the action will execute 10 minutes after the Virtual Object is archived.

   Slot End Time (Minutes)

Caution: If the Slot Begin Time and Slot End Time are both set to zero, then the behavior is unknown, except for Storage Slots.

   Setting this parameter to -1 indicates the slot will never end (it is permanent) and the Transcode Archived Action will occur regardless of whether the slot is open or closed.

   Example:

   Slot Begin Time (Minutes): 10

   Slot End Time (Minutes): -1

   Workflow:

   The original Virtual Object is archived onto the source medium. Ten minutes later, the slot actions begin and it is transcoded onto the target medium.

   Because the Slot End Time is set to -1, this slot will never end. If the slot is called during a period when the slot is closed, then as soon as the slot is open the transcode will take place. If this was set to another number (other than -1), the nor-

mal rules will apply as described later in the Request Execution End Time parameter.

Enabled

Determines whether the slot is active. Selecting *Y* indicates the slot is enabled, and *N* indicates it is disabled. When a slot is disabled its actions are not executed, however they are still created, and the status is updated. When the slot is enabled again, pending actions are executed immediately.

Request Execution Begin Time (HH:MM)

This field is the slot execution window opening time in 24 hour time. If set to 00:00 the slot opens at the beginning of each day. If set to another time, for example 08:00, it will open at that time (in this case 8:00 am).

Request Execution End Time (HH:MM)

This field is the slot execution window closing time in 24 hour time. If set to 24:00 the slot closes after each day. If set to another time, for example 17:00, it will close at that time (in this case 5:00 pm).

Setting the Request Execution Begin Time and Request Execution End Time to something other than an entire 24 hour period will allow for scheduling of different functions through SPM at different times of the day (for example overnight Requests). If a Virtual Object is archived to the source medium outside of the designated time frame for the slot, even though the Virtual Object meets the Filter and Storage Plan criteria, no actions from this particular slot will occur.

If the original Virtual Object was archived to the source medium while the slot was inactive (closed) and has passed the Slot End Time, when the slot finally becomes active (open), the action taken (or not taken) is determined by the Slot End Time setting.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): 20

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium at 07:00 (while the slot is closed). If the slot had been open at 07:00 when the new Virtual Object was created it would be transcoded to the target at 07:10. Because the slot was not open when the new Virtual Object was created, and the Slot End Time has passed, when this slot opens at 08:00, no action will be taken on the Virtual Object.

If a new Virtual Object is created on the source medium at 14:15 today, then tomorrow when this slot opens, no actions will be taken on the Virtual Object because the Slot End Time has passed.

If a Virtual Object is created on the source medium at 07:59 (before the slot opens), at 08:09 (10 minutes after the Virtual Object existed on the source) the Virtual Object will be transcoded to the target medium because the Slot End Time has not

yet passed.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): -1

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium while the slot is closed. If the slot had been open at 07:00 when the new Virtual Object was created, it would have been transcoded to the target at 07:10.

Since the slot was not open when the new Virtual Object was created, and the Slot End Time is set to never end (-1), when this slot opens at 08:00, the Transcode Action will be completed on the Virtual Object because the slot never ends.

If a new Virtual Object is created on the source medium at 14:15 today, then tomorrow when this slot opens, the Transcode Action will be completed on the Virtual Object because the slot never ends.

Request priority

This field identifies the priority of the request and order of execution when the action is submitted to the Core.

Medium Associated with Slot

Select the target medium to associate with the slot from the list.

Req. Number of Instances

The number of Virtual Object instances that should exist on the target medium. If the actual number of instances is less than this setting, additional copies will be created. **This parameter is only applicable to Copy Actions in Storage Slots.**

Once Only

This parameter is only applicable for Storage Slots and must be set to *Y* when using Mixed Mode Watermarking.

4. Click *OK* to complete the process. The new slot is displayed in the Slots tab under the DIVA Configuration > Storage Plans menu item.

   There is a second set of tabs at the bottom of the Slot Configuration screen that contains the following tabs:

   – Restore, Transcode & Metadata Archive

   – Storage

   The Transcode Archived Slot uses the Restore, Transcode & Metadata Archive tab to configure additional parameters including:

   – Quality Of Service

   – Request Options

   – Request Comments

   – Object Name Filter

- Object Source Filter
- Object Collection Filter
- Cascade Delete Links
- Transformation Rules (for renaming)

The Virtual Object filters allow restricting the Transcode process to particular Virtual Objects. You can use them if you do not want to transcode all Virtual Objects belonging to a Storage Plan.

The Transformation Rules are rules associated with the Virtual Object filters to transform the Virtual Object name and Collection for Transcode Archived and Metadata Archived.

5. Using the Restore, Transcode & Metadata Archive tab at the bottom of the Slot Configuration screen, configure the Virtual Object Filters and Transformation Rules as required:

Object Name Filter

Enter the Virtual Object Name filtering criteria you want to use for this filter. An asterisk is a wildcard character and the default entry in this field.

Object Source Filter

Enter the Virtual Object Source filtering criteria you want to use for this filter. An asterisk is a wildcard character and the default entry in this field.

Object Collection Filter

Enter the Virtual Object Collection filtering criteria you want to use for this filter. An asterisk is a wildcard character and the default entry in this field.

Target Name Template

Enter the name for the transcoded clip on the target medium. The caret (^) character indicates the system should use the original name.

Target Collection Template

Enter the Collection for the transcoded clip on the target. The caret (^) character indicates the system should use the original name.

If the Virtual Object matches the Object Filter criteria it will continue to be processed using the Transcode_Archived Slot. If the filter criteria are not matched then no further action is taken on the Virtual Object for this slot.

Transformation Rules transform the Virtual Object's name and Collection during the request, and the Transformation Name and Transformation Collection need to be added. At least one of these two values should be changed from the default.

Using the caret (^) in the Target Name Template, or Target Collection Template, or both, will cause SPM to use the original clip name, or the original clip Collection, or both. You can use the caret with additional naming conventions (for example, ^_wm9 or ^_cat).

Using only the caret with no additional characters causes the Core system to initiate an error stating that a Virtual Object with that name, or Collection, or both, already exists. Changing just one of these parameters avoids this error and creates the new transcoded clip on the target.

6. Similar to the Transformation Rules, the Request Templates parameters are located at the bottom of the Slot Configuration screen in the Restore, Transcode & Metadata Archive tab. Configure the additional parameters as required:

   Object Unmanaged Storage Repository

   This field is not accessible because the newly created Virtual Object is always archived from the Core Actor that performed the transcoding.

   Quality Of Service

   Select the appropriate Quality Of Service from the list for this request. The Quality Of Service parameter has seven different levels:

   DEFAULT

   Archiving is performed according to the default Quality Of Service (currently direct and cache for archive operations).

   CACHE-ONLY

   Use cache archive only.

   DIRECT-ONLY

   Use direct archive only.

   CACHE&DIRECT

   Cached transfers from, or to, a Server to, or from, Core are preferred, but direct transfers will occur if it is not possible to process the request in cache mode.

   DIRECT&CACHE

   Direct transfers from, or to, a Server to, or from, Core are preferred, but cache transfers will occur if it is not possible to process the request in direct mode.

   NEARLINE-ONLY

   If a Nearline disk instance exists, the data is transferred from Nearline disk to the Destination Server. Alternatively, the data is first transferred entirely to Nearline storage on disk from tape, and then transferred to the Destination Server. If no Nearline service is available, the request will be terminated.

   NEARLINE&DIRECT

   If Nearline transfer is not available (for example, no Core Actor with Nearline storage is available), a direct transfer is performed instead.

   Request Options

   Enter the transcoding options in this field based on the transcoders defined in System Management App. The available options are as follows:

   -tr_names

   This option specifies the transcoder to use as defined in the System Management App Transcoders section.

-tr_archive_format

This option specifies the factory and format to be transcoded to for Archive Slots.

-tr_restore_format

This option specifies the factory and format to be transcoded to for Restore Slots.

Request Comments

Enter any comments in this field to be sent to the Core with this action request.

Pathroot

This field is not accessible because the File Path Root for the archive of the transcoded Virtual Object is generated internally by the Transcode Archived request.

Cascade Delete Links

This parameter determines whether to perform a Delete Action on just the original Virtual Object, or on both the original and the transcoded Virtual Object. If set to Y, when the original is deleted, the transcoded copy will also be deleted. When set to N, when the original is deleted, the copy remains untouched.

Cascade Restore Links

This parameter determines whether to perform a Restore Action on just the original Virtual Object, or on both the original and the transcoded Virtual Object. If set to Y, when the original is restored, the transcoded copy will also be restored. When set to N, when the original is restored, the copy remains untouched.

Configuring Metadata Archive Slots

You configure the Metadata Archive Slots on the Slots tab of System Management App. These slots contain only a single action. This slot will archive a metadata file after a Virtual Object is archived. The metadata file's file name must be the Object Name with no extension, and must reside on a location (Server and Path) that is defined in the Slot Configuration window. The Object Name and Object Collection used for the metadata file archive are controlled by the Transformation Rule entries.

Use the following procedure to configure Metadata Archive Slots:

1. Open System Management App, and navigate to the Slots tab under the DIVA Configuration > Storage Plans menu item.
2. Click the + icon on the top right of the Slots section.

3. When the Slot Configuration dialog box opens, configure the options as follows:

Storage Plan

Use the menu list to select the Storage Plan to be associated with this slot.

Request Type

Select the type of request for this slot (Storage, Transcode Archived, Metadata Archive, or Restore). Select METADATA_ARCHIVE in this case.

Slot Name

Enter a name for the slot in the Slot Name field.

Slot Begin Time (Minutes)

This parameter identifies when the action will occur. If set to zero the action is initiated as soon as the Virtual Object is archived. If set to another number, for example 10 (minutes), the action will execute 10 minutes after the Virtual Object is archived.

Slot End Time (Minutes)

Caution: If the Slot Begin Time and Slot End Time are both set to zero, then the behavior is unknown, except for Storage Slots.

Setting this parameter to -1 indicates that the slot will never end (it is permanent), and the Metadata Archive Action will occur regardless of whether the slot is open or closed.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): -1

Workflow:

The original Virtual Object is archived onto the source medium.

Ten minutes later, the slot actions begin and the metadata file is archived onto the target medium.

Because the Slot End Time is set to -1, this slot will never end. If the slot is called during a period when the slot is closed, as soon as the slot is open the archive will take place. If this was set to another number (other than -1), the normal rules will apply as described later in the Request Execution End Time description.

Enabled

Determines whether the slot is active. Selecting Y indicates the slot is enabled, and N indicates it is disabled. When a slot is disabled its actions are not executed, however they are still created when a new Virtual Object is archived.

Request Execution Begin Time (HH:MM)

This field is the slot execution window opening time in 24 hour time. If set to 00:00 the slot opens at the beginning of each day. If set to another time, for example

08:00, it will open at that time (in this case 8:00 am).

Request Execution End Time (HH:MM)

This field is the slot execution window closing time in 24 hour time. If set to 24:00 the slot closes after each day. If set to another time, for example 17:00, it will close at that time (in this case 5:00 pm).

Setting the Request Execution Begin Time and Request Execution End Time to something other than an entire 24 hour period will allow for scheduling of different functions through SPM at different times of the day (for example overnight Requests). If a Virtual Object is archived to the source medium outside of the designated time frame for the slot, even though the Virtual Object meets the Filter and Storage Plan criteria, actions taken on the Virtual Object are determined but the setting of the Slot End Time parameter.

If the original Virtual Object was archived to the source medium while the slot was inactive (closed) and has passed the Slot End Time, when the slot finally becomes active (open), the action taken (or not taken) is determined by the Slot End Time setting.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): 20

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium at 07:00 (while the slot is closed). If the slot had been open at 07:00 when the new Virtual Object was created the metadata file would have been archived to the target at 07:10. Because the slot was not open when the new Virtual Object was created, and the Slot End Time has passed, when this slot opens at 08:00, no actions will be taken on the Virtual Object.

If a new Virtual Object is created on the source medium at 14:15 today, then tomorrow when this slot opens, no actions will be taken on the Virtual Object because the Slot End Time has passed.

If a Virtual Object is created on the source medium at 07:59 (before the slot opens), at 08:09 (10 minutes after the Virtual Object existed on the source) the Virtual Object will be copied to the target medium because the Slot End Time has not yet passed. Then at 08:19 (20 minutes after the Virtual Object existed on the source) the metadata file will be archived to the target medium because the Slot End Time has not yet passed.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): -1

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium at 07:00 (while the slot is closed). If the slot had been open at 07:00 when the new Virtual Object was created, the metadata file would have been archived to the target at 07:10.

Because the slot was not open when the new Virtual Object was created, and the Slot End Time is set to never end (-1), when this slot opens at 08:00, the Metadata Archive Action will be completed on the Virtual Object because the slot never ends.

If a new Virtual Object is created on the source medium at 14:15 today, then tomorrow when this slot opens, the Metadata Archive Action will be completed on the Virtual Object because the slot never ends.

Request priority

This field identifies the priority of the request and order of execution when the action is submitted to the Core.

Medium Associated with Slot

Select the target medium to associate with the slot from the list.

4. Similar to the Transformation Rules, the Request Templates parameters are located at the bottom of the Slot Configuration screen in the Restore, Transcode & Metadata Archive tab.
   - Virtual Object Filters allow restricting the Metadata Archive process to particular Virtual Objects. You can use them if archiving metadata files for all Virtual Objects belonging to a Storage Plan is not desired.
   - Transformation Rules are rules associated with the Virtual Object Filters to transform the name and Collection during the request.

   If the Virtual Object matches the Object Filter criteria it will continue to be processed using the Metadata_Archive request type. If the filter criteria are not matched then no further action is taken on the Virtual Object for this slot.

   Transformation Rules transform the Virtual Object's name and Collection during the request, and the Transformation Name and Transformation Collection must be added. At least one of these two values should be changed from the default.

   Using the caret (^) in the Target Name Template, or Target Collection Template, or both, will cause SPM to use the original clip name, or the original clip Collection, or both. You can use the caret with additional naming conventions (for example, ^_wm9 or ^_cat).

   Using only the caret with no additional characters causes the Core system to initiate an error stating that a Virtual Object with that name, or Collection, or both, already exists. Changing just one of these parameters avoids this error and creates the new transcoded clip on the target.

5. Configure the additional parameters as required:

Object Unmanaged Storage Repository

This field is not accessible because the newly created Virtual Object is always archived from the Core Actor that performed the transcoding.

Quality Of Service

Select the appropriate Quality Of Service from the list for this request. The Quality Of Service parameter has seven different levels:

DEFAULT

Archiving is performed according to the default Quality Of Service (currently direct and cache for archive operations).

CACHE-ONLY

Use cache archive only.

DIRECT-ONLY

Use direct archive only.

CACHE&DIRECT

Cached transfers from, or to, a Server to, or from, Core are preferred, but direct transfers will occur if it is not possible to process the request in cache mode.

DIRECT&CACHE

Direct transfers from, or to, a Server to, or from, Core are preferred, but cache transfers will occur if it is not possible to process the request in direct mode.

NEARLINE-ONLY

If a Nearline disk instance exists, the data is transferred from Nearline disk to the Destination Server. Alternatively, the data is first transferred entirely to Nearline storage on disk from tape, and then transferred to the Destination Server. If no Nearline service is available, the request will be terminated.

NEARLINE&DIRECT

If Nearline transfer is not available (for example, no Core Actor with Nearline storage is available), a direct transfer is performed instead.

Request Options

This is only used if a user name and password are required for the Server, otherwise leave this blank.

Request Comments

Enter any comments in this field to be sent to the Core with this action request.

Pathroot

Enter the path to the metadata files on the Server.

Object Name Filter

Enter the Virtual Object Name filtering criteria you want to use for this filter. An

asterisk is a wildcard character and the default entry in this field.

Object Source Filter

Enter the Virtual Object Source filtering criteria you want to use for this filter. An asterisk is a wildcard character and the default entry in this field.

Object Collection Filter

Enter the Virtual Object Collection filtering criteria you want to use for this filter. An asterisk is a wildcard character and the default entry in this field.

Cascade Delete Links

This parameter determines whether to perform a Delete Action on just the original Virtual Object or on both the original and the Metadata Virtual Object. If set to Y, when the original is deleted, the Metadata Virtual Object will also be deleted. When set to N, when the original is deleted, the Metadata Virtual Object remains untouched.

Cascade Restore Links

This parameter determines whether to perform a Restore Action on just the original Virtual Object, or on both the original and the metadata Virtual Object. If set to Y, when the original is restored, the metadata Virtual Object will also be restored. When set to N, when the original is restored the metadata Virtual Object remains untouched.

Target Name Template

For a Metadata Archive request, this parameter serves two purposes; it notifies SPM to look for a Virtual Object with this parameter value on the Server, and it notifies SPM to create a Metadata Archive file on the target using this parameter value. The caret (^) character indicates the system should use the original name.

Target Collection Template

Enter the Collection for the transcoded clip on the target. The caret (^) character indicates the system should use the original Collection.

6. Click *OK* to complete the process. The new slot will be displayed in the Slots tab under the DIVA Configuration > Storage Plans menu item.

Configuring Restore Slots

You configure Restore Slots on the Slots tab of System Management App. Restore Slots only contain a single action. This slot will take an existing Core Virtual Object and restore it onto the designated Server.

Use the following procedure to configure Restore Slots:

1. Open System Management App, and navigate to the Slots tab under the DIVA Configuration > Storage Plans menu item.
2. Click the + icon on the top right of the Slots section.

3. When the Slot Configuration dialog box opens, configure the options as follows:

   Storage Plan

   Use the menu list to select the Storage Plan to be associated with this slot.

   Request Type

   Select the type of request for this slot (Storage, Transcode Archived, Metadata Archive, or Restore). Select RESTORE in this case.

   Slot Name

   Enter a name for the slot in the Slot Name field.

   Slot Begin Time (Minutes)

   This parameter identifies when the action will occur. If set to zero the action is initiated as soon as the Virtual Object is archived. If set to another number, for example 10 (minutes), the action will execute 10 minutes after the Virtual Object is archived.

Slot End Time (Minutes)

Caution: If the Slot Begin Time and Slot End Time are both set to zero, then the behavior is unknown, except for Storage Slots.

Setting this parameter to -1 indicates that the slot will never end (it is permanent), and the Restore Action will occur regardless of whether the slot is open or closed.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): -1

Workflow:

The Virtual Object is found on the mediums monitored by SPM.

Ten minutes later, the slot actions begin and the Virtual Object is restored to the target Server.

If the Slot End Time was greater than zero, and if the Restore Action has not executed before the Slot End Time is reached (because the slot remained closed after the Slot End Time was reached) then the Restore Action will never be executed. If this was set to another number (other than -1), the normal rules will apply as described later in the Request Execution End Time description.

Enabled

Determines whether the slot is active. Selecting Y indicates the slot is enabled, and N indicates it is disabled. When a slot is disabled its actions are not executed, however they are still created when a new Virtual Object is archived.

Request Execution Begin Time (HH:MM)

This field is the slot execution window opening time in 24 hour time. If set to 00:00 the slot opens at the beginning of each day. If set to another time, for example 08:00, it will open at that time (in this case 8:00 am).

Request Execution End Time (HH:MM)

This field is the slot execution window closing time in 24 hour time. If set to 24:00

the slot closes after each day. If set to another time, for example 17:00, it will close at that time (in this case 5:00 pm).

Setting the Request Execution Begin Time and Request Execution End Time to something other than an entire 24 hour period will allow for scheduling of different functions through SPM at different times of the day (for example overnight Requests). If a Virtual Object is archived to the source medium outside of the designated time frame for the slot, even though the Virtual Object meets the Filter and Storage Plan criteria, actions taken on the Virtual Object are determined but the setting of the Slot End Time parameter.

If the original Virtual Object was archived to the medium while the slot was inactive (closed) and has passed the Slot End Time, when the slot finally becomes active (open), the action taken (or not taken) is determined by the Slot End Time setting.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): 20

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium at 07:00 (while the slot is closed). Because the slot was not open when the new Virtual Object was created, and the Slot End Time has passed, (20 minutes after the Virtual Object was placed on the medium, that is 7:20) when this slot opens at 08:00, no actions will be taken on the Virtual Object.

Note: If the slot had been opened at 07:00, then when the new Virtual Object was created it would have been restored to the target at 07:10.

If a new Virtual Object is created on the source medium at 14:15 today, then tomorrow when this slot opens, no actions will be taken on the Virtual Object because the Slot End Time has passed.

If a Virtual Object is created on the source medium at 07:59 (before the slot opens), at 08:09 (10 minutes after the Virtual Object existed on the source) the Virtual Object will be restored to the Server because the Slot End Time has not yet passed.

Example:

Slot Begin Time (Minutes): 10

Slot End Time (Minutes): -1

Request Execution Begin Time (HH:MM): 08:00

Request Execution End Time (HH:MM): 14:00

Workflow:

A new Virtual Object is placed onto the source medium at 07:00 (while the slot is closed). Because the slot is not open when the new Virtual Object was created, and

the Slot End Time is set to never end (-1), when this slot opens at 08:00, the Restore Action will be completed on the Virtual Object because the slot never ends.

Note: If the slot had been open at 07:00 when the new Virtual Object was created, it would have been restored to the Server at 07:10.

If a new Virtual Object is created on the source medium at 14:15 today, then tomorrow when this slot opens, the Restore Action will be completed on the Virtual Object because the slot never ends.

Request priority

This field identifies the priority of the request and order of execution when the action is submitted to the Core.

1. Similar to the Transformation Rules, the Request Templates parameters are located at the bottom of the Slot Configuration screen in the Restore, Transcode & Metadata Archive tab.
   - Virtual Object Filters allow restricting the Metadata Archive process to particular Virtual Objects. You can use them if archiving metadata files for all Virtual Objects belonging to a Storage Plan is not desired.
   - Transformation Rules are rules associated with the Virtual Object Filters to transform the name and Collection during the request.

   If the Virtual Object matches the Object Filter criteria it will continue to be processed using the Metadata_Archive request type. If the filter criteria are not matched then no further action is taken on the Virtual Object for this slot.

   Transformation Rules transform the Virtual Object's name and Collection during the request, and the Transformation Name and Transformation Collection must be added. At least one of these two values should be changed from the default.

   Using the caret (^) in the Target Name Template, or Target Collection Template, or both, will cause SPM to use the original clip name, or the original clip Collection, or both. You can use the caret with additional naming conventions (for example, ^_wm9 or ^_cat).

   Using only the caret with no additional characters causes the Core system to initiate an error stating that a Virtual Object with that name, or Collection, or both, already exists. Changing just one of these parameters avoids this error and creates the new transcoded clip on the target.

2. Configure the additional parameters as required:

Object Unmanaged Storage Repository

This field is not accessible because the newly created Virtual Object is always archived from the Core Actor that performed the transcoding.

Quality Of Service

Select the appropriate Quality Of Service from the list for this request. The Quality Of Service parameter has seven different levels:

DEFAULT

Archiving is performed according to the default Quality Of Service (currently direct and cache for archive operations).

CACHE-ONLY

Use cache archive only.

DIRECT-ONLY

Use direct archive only.

CACHE&DIRECT

Cached transfers from, or to, a Server to, or from, Core are preferred, but direct transfers will occur if it is not possible to process the request in cache mode.

DIRECT&CACHE

Direct transfers from, or to, a Server to, or from, Core are preferred, but cache transfers will occur if it is not possible to process the request in direct mode.

NEARLINE-ONLY

If a Nearline disk instance exists, the data is transferred from Nearline disk to the Destination Server. Alternatively, the data is first transferred entirely to Nearline storage on disk from tape, and then transferred to the Destination Server. If no Nearline service is available, the request will be terminated.

NEARLINE&DIRECT

If Nearline transfer is not available (for example, no Core Actor with Nearline storage is available), a direct transfer is performed instead.

Request Options

This is only used if a user name and password are required for the Server, otherwise leave this blank.

Request Comments

Enter any comments in this field to be sent to the Core with this action request.

Pathroot

Enter the path to the metadata files on the Server.

Object Name Filter

Enter the Virtual Object Name filtering criteria you want to use for this filter. An asterisk is a wildcard character and the default entry in this field.

Object Source Filter

Enter the Virtual Object Source filtering criteria you want to use for this filter. An

asterisk is a wildcard character and the default entry in this field.

Object Collection Filter

Enter the Virtual Object Collection filtering criteria you want to use for this filter. An asterisk is a wildcard character and the default entry in this field.

Cascade Delete Links

This parameter determines whether to perform a Delete Action on just the original Virtual Object or on both the original and the Metadata Virtual Object. If set to Y, when the original is deleted, the Metadata Virtual Object will also be deleted. When set to N, when the original is deleted, the Metadata Virtual Object remains untouched.

Cascade Restore Links

This parameter determines whether to perform a Restore Action on just the original Virtual Object, or on both the original and the metadata Virtual Object. If set to Y, when the original is restored, the metadata Virtual Object will also be restored. When set to N, when the original is restored the metadata Virtual Object remains untouched.

Target Name Template

For a Metadata Archive request, this parameter serves two purposes; it notifies SPM to look for a Virtual Object with this parameter value on the Server, and it notifies SPM to create a Metadata Archive file on the target using this parameter value. The caret (^) character indicates the system should use the original name.

Target Collection Template

Enter the Collection for the transcoded clip on the target. The caret (^) character indicates the system should use the original Collection.

3. Click *OK* to complete the process. The new slot will be displayed in the Slots tab under the DIVA Configuration > Storage Plans menu item.

SPM Delete Virtual Object Behavior

Caution: This feature must be managed carefully because you can lose Virtual Objects.

The DELETE_OBJECT_ONLY_LAST_INSTANCE parameter is a configuration parameter that works with the existing ALLOW_OBJECT_DELETION parameter.

ALLOW_OBJECT_DELETION Parameter

This parameter must be enabled for SPM to delete Virtual Objects. You must enable or disable Delete Virtual Object for each Storage Plan separately when configuring them in System Management App. This configuration method provides additional security and flexibility. Delete Virtual Object is enabled only when it is enabled in both the configuration file and also the Storage Plan configuration.

Note: If this parameter is disabled, allowing it for a Storage Plan will have no effect.

Delete Virtual Object is executed only when the following conditions are satisfied:

- The ALLOW_OBJECT_DELETION parameter must be enabled.
- Delete Virtual Object must be enabled for the Storage Plan level through System Management App.
- All of the Storage Slots in the Storage Plan associated with the Virtual Object must have a Slot End Time.
- Delete Virtual Object will be executed by the Storage Slot that has the highest end time in the Storage Plan.

  A Virtual Object can be associated with only one Storage Plan.

  If multiple Storage Slots have a maximum end time in a Storage Plan, the slot with the maximum Slot ID among the slots with maximum end time will perform the delete Virtual Object.
- Just enabling ALLOW_OBJECT_DELETION will not acknowledge a Storage Slot medium Watermark, and the Delete Virtual Object is executed at the end of the Storage Slot that has the highest end time in the Storage Plan.

The valid values for this parameter are true or false, and the default is false.

DELETE_OBJECT_ONLY_LAST_INSTANCE Parameter

This parameter adds additional checks and conditions to the SPM Delete Virtual Object feature.

Enabling this parameter forces SPM to delete a Virtual Object at the end of a Storage Slot only if it is the last instance in the entire Core system, and that last instance exists on the Storage Slot medium with the highest Slot End Time among all Storage Slots of the Storage Plan. Otherwise, SPM will never execute a Delete Virtual Object at the end of the Storage Slot, it will always only execute a Delete Instance at the end of a Storage Slot. The following conditions must be satisfied for this to execute correctly:

- This parameter requires ALLOW_OBJECT_DELETION to be enabled. Only enabling DELETE_OBJECT_ONLY_LAST_INSTANCE has no effect, and the SPM Delete Virtual Object feature will not be enabled.
- The parameter DELETE_OBJECT_ONLY_LAST_INSTANCE must be enabled.
- You must have Delete Virtual Object enabled for the Storage Plan level through System Management App.
- All of the Storage Slots in the Storage Plan associated with the Virtual Object must have a Slot End Time.

- Delete Virtual Object will be executed by the Storage Slot that has the highest Slot End Time in the Storage Plan.

  A Virtual Object can be associated with only one Storage Plan.

  If multiple Storage Slots have a maximum end time in a Storage Plan, the slot with maximum slot ID among the slots with maximum end time will perform the Delete Virtual Object function.
- At the end of the Storage Slot, if more than two instances exist in the Core system or on the medium, only Delete Instance is executed.
- Delete Virtual Object is executed only if one instance exists on the entire Core system, and that one instance exists on the medium of the Storage Slot that has the highest Slot End Time among all Storage Slots of the storage plan.
- If the medium of the Storage Slot that has the highest Slot End Time is Watermarked, the Delete Virtual Object is postponed until the Watermark condition is satisfied.

Changing a Storage Plan

Storage Plans contain Slots that you can altered if required. You can add slots, and edit, delete, and enable or disable existing slots. These adjustments to Storage Plans must be performed carefully to ensure the results that are desired. Below is an example of adding a slot to a Storage Plan, and changing an existing slot within the same Storage Plan.

Example:

- DISK1 = Local Core Managed Disk

  Storage Slot

  Slot Start Time = 0

  Slot End Time = -1
- NEARLINE = Local Tape Group

  Storage Slot

  Slot Start Time = 0

  Slot End Time = -1
- OFFLINE = Local Tape Group

  Storage Slot

  Slot Start Time = 0

  Slot End Time = -1
- DISK2 = New Core Managed Disk

  This is not an initial slot.

  This Storage Slot is added in Step 4 of the following workflow with a Slot Start Time = 0, and Slot End Time = -1.

The company has Storage Plan A, which includes three total instances; one instance on disk and two instances on tape. The following is the workflow for Storage Plan A:

1. Object1234 is archived to DISK1 under Storage Plan A.
2. Storage Plan A has slots for the following operations, and Object1234 is copied to both tape groups:

   Copy to NEARLINE

   Copy to OFFLINE
3. At some point in the future, the company changes its internal policy for content with Storage Plan A to three total instances; two instances on disks (DISK1 and DISK2), and one instance on the NEARLINE tape group.

4. The new slot change is added that will copy Object1234 to DISK2 but it is necessary to also delete the Virtual Object instance from the NEARLINE tape group.

   To delete the Virtual Object instance from the NEARLINE tape group, you must change the Slot End Time to 0 for the NEARLINE Storage Slot (see *Disabling and Deleting Slots* for more information).

   You must add a new Storage Slot on DISK2 with a Slot Start Time = 0 and Slot End Time = -1 (see *Adding a Slot* for more information).

Storage Plan A is now updated with the Nearline Slot configuration to delete the instance from the NEARLINE tape group. The Slot Begin Time and Slot End Time are both set equal to 0.

Disabling and Deleting Slots

Telestream generally recommends that slots are not deleted, but rather disabled instead. Deleting a slot does not remove the content from the Core system.

For a Storage Slot, setting the Slot End Time to 0 forces SPM to delete any contents created by this slot. You can delete, or disable, the Storage Slot after all the contents are removed. If you want to retain the content in a Storage Slot, but no further processing of future actions is required, you should disable the slot instead of deleting it.

Adding a Slot

You can add a slot to an existing Storage Plan. However, you must use caution to achieve the desired results.

When adding a slot to an existing Storage Plan, you must ensure that the Slot Start Time and Slot End Time are both valid (see *Configuring Storage Slots* for valid parameters). New slots are always immediately applicable to new Virtual Objects, but for existing Virtual Objects you must ensure that the Slot Start Time and Slot End Time are correct. Otherwise, the action specified in the new slot will not be executed on existing Virtual Objects.

Example:

A Storage Plan with numerous Virtual Objects that have been archived 48 hours before the current time. If a new Storage Slot is added, and is configured with a Slot Start Time = 0 and a Slot End Time = 24, the new slot's action will be added but not applied to existing Virtual Objects because the slot has already expired for all of the existing Virtual Objects:

```
((Archive Object Time + Slot End Time (24 hours)) < Current Time)
```

For existing Virtual Objects to be affected by the new Storage Slot, they must be immediately copied to the Storage Slot medium, but never deleted. To perform this function, you set the Slot Start Time to 0, and the Slot End Time to -1 (infinite). As a result, all instances will be copied to the Storage Slot medium:

```
((Archive Object Time + Slot End Time (infinite)) > Current Time)
```

For non-Storage Slots (for example, Restore Slots), you must configure the Slot Start Time = 0 and the Slot End Time = 100. In this case, SPM will try restoring Virtual Objects that have existed for less than the Slot End Time (100 hours); that is, the Virtual Objects whose Slot End Time has not expired:

```
((Archive Object Time + Slot End Time (100 hours)) > Current Time)
```

If SPM is not able to complete the restore of the Virtual Object within 100 hours from the Archive Time, that Virtual Object is never restored. If the Slot End Time = -1, SPM will try to restore all of the Virtual Objects.

Watermark Based Disk Cleaning Management

Disk Array cleaning can be managed by a fixed slot duration, where disk instances or Virtual Objects are deleted when they expire, or using a Watermark based housekeeping mechanism, where disk instances or Virtual Objects are only deleted when space must be recovered. This section discusses the details of the second option.

Based on configuration, SPM may attempt to delete instances, or Virtual Objects. When a Storage Slot's retention for a Virtual Object is expired, and the Virtual Object has no more open Storage Slots at the time, SPM will attempt to delete the Virtual Object if the configuration authorizes it. Otherwise, SPM will only attempt to delete the instance located on the medium referenced in the slot.

You can explicitly specify the order in which array instances, or Virtual Objects, or both, of content are removed during the Core Array Housekeeping activities. SPM includes an array monitoring component to manage array space usage called Watermarking.

Array housekeeping is based on a Watermark system monitored by the Disk Space Monitor component. When the array's Storage Slot retention period for a Virtual Object expires, the instance, or Virtual Object, or both, becomes eligible for deletion, but stays on the array until there not enough space available.

Note: Watermarking is only valid for Storage Slots.

High Watermark

Core initiates housekeeping activities and expired instances, or Virtual Objects, or both are deleted according to the defined policies when this threshold is reached. If no instances or Virtual Objects are eligible for deletion (for example, the retention period has not expired, or a dependency is not met), used array space can exceed the watermark. It is your responsibility to ensure that the array filling rate matches the system configuration, and to monitor how array use evolves.

Low Watermark

Core stops deleting instances and Virtual Objects at the Low Watermark. Supported policies and rules for housekeeping include deleting array instances based on Last Access Time or Largest Object Size.

Mixed Mode

When an array is watermarked in Mixed Mode, both instances and Virtual Objects marked for deletion, and Virtual Objects whose slots are still open, are considered. Instances and Virtual Objects in that array are deleted based on whichever of the following comes first:

Note: You must set Once Only to Y (Yes) in the associated slot's configuration for proper functionality.

- Array reached the High Watermark. Even if the Virtual Object's slot has not ended, instances and Virtual Objects will be deleted to bring the array to the Low Watermark level.
- Storage Slot for that Virtual Object has ended. Even though the High Watermark is not reached, because the slot ended, the instance, or Virtual Object, or both will be deleted.
- This does not mean the array will be cleaned to the level of Low Watermark.

For SPM to delete a Virtual Object belonging to a Storage Plan, the ALLOW_OBJECT_DELETION parameter must be set to Y in both the SPM Configuration File and System Management App.

External Storage Plans

External Storage Plan functionality allows management of the Core Virtual Object Lifecycle using third party external systems. You can use the SPM Database API to develop integration of third party components with Core and SPM.

Note: In this release of SPM, ***E*** (*External*) should only be used by Telestream personnel.

Configuration Validation

You can validate the accuracy of the SPM configuration using System Management App. Use the following procedure to validate the SPM configuration:

1. Open System Management App.
2. Select any of the tabs across the top of the screen.
3. From the menu bar, select Tools, and then Validate SPM configuration. Alternatively you can hold *Alt*+*t*, and then press the F3 function key.

   System Management App will verify the configuration of SPM and return a dialog box with one of two results:

   – System Management App verified the SPM configuration successfully and the dialog box indicates no errors were found.

   – System Management App verified the SPM configuration, found errors, and dialog box displays the errors that were discovered.

   For example:

ERROR: Slot Test must not be of type STORAGE, Please mention correct request type in Virtual Object Filters Configuration.

Operations

This chapter describes SPM operations, and includes the following information:

Topics:

Managing the SPM Service through the Command Line

You can manage the SPM Service through the Windows Command-line Interface (START, RUN, CMD), or using the Windows Services Panel. Telestream recommends you use the command-line interface.

```
spmservice.exe {command} [options]
```

The following are the commands:

install (or use -i)

This command installs SPM as a Windows Service.

uninstall (or use -u)

This command removes the SPM Windows Service.

debug (or use -d)

This command starts SPM in console mode for troubleshooting purposes.

version (or use -v)

This command displays the SPM release information, and then exits.

help (or use -h)

This command displays help information, and then exits.

The following are the options:

-conf (or use -f)

You use this option to load the settings from a specific configuration file.

For example, spmservice.exe install -conf C:\DIVA\conf\test.conf.

Accessing SPM Information through the System Management App

You can view SPM information for Virtual Objects, and reschedule failed actions using the DIVA Core System Management App. This section describes how to use the System Management App to access the information and what information is available through this interface.

Start the DIVA Core System Management App, and then follow through this section to discover where to view the SPM information required.

- You can see which Storage Plan is assigned to a Virtual Object on the Virtual Objects screen under the Manage tab.

  If a newly archived Virtual Object does not match any filters it is assigned the default Storage Plan. The default Storage Plan is SP_DEFAULT and has no slots (and therefore no actions) associated with it.

- You can view information concerning all actions established by SPM immediately through the SPM Actions panel under the Manage tab.

  The ID column on the screen is the automatically generated internal SPM ID.

  The Task Name column displays the Task Name for the action. The DEFAULT SPM SCHEDULER TASK name corresponds to internal SPM generated actions. External actions will display a different Task Name.

  The Req. ID column displays the Core Request ID. When an action is in the SCHEDULED state in SPM, the initial Request ID will be zero. When the action begins execution, Core receives a request and generates the Core Request ID.

  The remaining columns on this screen are self-explanatory.

  You can view the Request Properties by double-clicking on the action, or by right-clicking on the action and selecting Request Properties from the resulting context menu. If no Request Mapping exists for the Virtual Object, the Request Properties window will not appear.

Note: If the action is getting old, the Core Job History may have cycled and the Request ID may have been reused by a newer Core request. In this case, viewing the *Request Properties* will display a different request other than the one expected.

Filtering the SPM Actions Panel View

The SPM Actions panel enables filtering which actions you want to view. You use the menu lists and text boxes at the top of the screen to filter your view. The default asterisk displayed in the text boxes is a wildcard and signifies that no filtering restriction is being placed on the parameter where it is used (that is, the Virtual Object name, Collection, Task Name, and Slot name).

To filter by Scheduled Date (and time), select the check box next to enable, and then select the Begin and End dates and times to view. If you do not select the check box next to enable, this filtering function will be grayed out and unusable.

Additional filtering is possible by Action Status, or Action Type, or both. Click the button next to either one (or both) to display a list of options, and select the desired filters, then click Close to continue.

After changing any of the viewing filters, you must click Refresh to refresh the view.

Assigning a New Storage Plan

You can change Storage Plans only for a Virtual Object that is assigned the default Storage Plan (DEFAULT_SP). When a new Storage Plan is assigned to a Virtual Object, SPM sees that Virtual Object as being a new Virtual Object in the system because it has a new Storage Plan assigned.

You can select a list of multiple Virtual Objects by holding the SHIFT key and selecting the first, and then the last Virtual Object in the list. All Virtual Objects in between the first and last will be selected. You can also click and hold the mouse button on the first Virtual Object, and then drag the mouse pointer to the last Virtual Object and release the button.

To select individual Virtual Objects, hold the CTRL key and click each specific Virtual Object to highlight it, and then release the button.

Use the following procedure to assign a Storage Plan to the selected Virtual Objects:

1. Open the DIVA Core System Management App and connect to the Core Database.
2. Navigate to the Manage tab > click the *Virtual Objects* icon.
3. Select one or more Virtual Object to assign a new Storage Plan to using one of the methods previously described.
4. Right-click the Virtual Objects requiring the new Storage Plan.
5. Select *Assign Storage Plan* from the displayed context menu.
6. A new dialog box is displayed allowing selection of the Storage Plan to be assigned. Select the Storage Plan to assign from the list, and then click *Assign* to assign the new plan to the selected Virtual Objects.

Rescheduling Failed Actions

When an action fails to execute properly and does not complete correctly, SPM will automatically reschedule the action for re-execution at some time in the future. You can also manually reschedule the action to run immediately using the following procedure:

1. Open the DIVA Core System Management App.
2. Navigate to the Manage tab > click the *SPM Actions* icon.
3. Right-click the action that requires rescheduling (you can select multiple actions - see the previous section for multi-selection tips).

4. Click *Reschedule Action* from the resulting context menu. A confirmation dialog box will display.
5. On the confirmation dialog box, select *Yes* to confirm rescheduling the action to execute immediately, or select *No* to cancel the process.

Marking Failed Actions Complete

You can change the status of SPM Failed Actions to COMPLETED by right clicking the action, and then selecting *Mark Action Completed* from the context menu.

Normally, SPM will retry a completed Copy Action if the Once Only option is set to N, and a user manually (or accidentally) deletes the instance that SPM copied before the Storage Slot expires. Also, SPM will normally retry a completed Delete Action if a user manually (or accidentally) copies an instance to the Storage Slot medium after SPM deleted it.

Actions marked as complete by a user will never be retried by SPM. However, you can reschedule a user completed action by right clicking it, and the selecting Reschedule Action from the context menu. The Mark Action Completed option is only available using the Administrator profile.

Storage Policy Manager Action Result Codes

Result codes are displayed for every processing and completed action in the SPM Actions panel of the System Management App. The Action Result Code is a short text string reflecting the latest available result of action execution.

The following list describes the possible result codes:

Prefix: REQ

Description: DIVA_REQUEST_STATE code from the Core API.

Details: SPM always updates the status of an action after execution based on the status of the corresponding request submitted to the Manager. SPM uses the Core GET_REQUEST_INFO API call to find the status of the request, and updates the result code for the action based on the DIVA_REQUEST_STATE code returned by the Core GET_REQUEST_INFO API call. See the Core API documentation in the Core Documentation library for more details.

Sample RESULT_CODE: REQ-REQUEST_STATUS where REQUEST_STATUS is one of the following:

TRANSFERRING

MIGRATING

COMPLETED

ABORTED

CANCELLED

UNKNOWN

DELETING

WAITING FOR RESOURCES

ASSIGNING POOL

PARTIALLY ABORTED

RUNNING

PENDING

Prefix: API

Description: DIVA_STATUS of the Core API.

Details: SPM uses the Core API to execute actions and submit requests to the Manager. The result code of the action is updated with the DIVA_STATUS API return code if SPM failed to submit the request. See the Core API documentation in the Core Documentation library for more details.

Sample RESULT_CODE: API-RETURN_CODE where RETURN_CODE is one of the following:

DIVA_OK

DIVA_ERR_UNKNOWN

DIVA_ERR_INTERNAL

DIVA_ERR_NO_ARCHIVE_SYSTEM

DIVA_ERR_BROKEN_CONNECTION

DIVA_ERR_DISCONNECTING

DIVA_ERR_ALREADY_CONNECTED

DIVA_ERR_WRONG_VERSION

DIVA_ERR_INVALID_PARAMETER

DIVA_ERR_OBJECT_DOESNT_EXIST

DIVA_ERR_SEVERAL_OBJECTS

DIVA_ERR_NO_SUCH_REQUEST

DIVA_ERR_NOT_CANCELABLE

DIVA_ERR_SYSTEM_IDLE

DIVA_ERR_WRONG_LIST_SIZE

DIVA_ERR_LIST_NOT_INITIALIZED

DIVA_ERR_OBJECT_ALREADY_EXISTS

DIVA_ERR_GROUP_DOESNT_EXIST

DIVA_ERR_SOURCE_OR_DESTINATION_DOESNT_EXIST

DIVA_WARN_NO_MORE_OBJECTS

DIVA_ERR_NOT_CONNECTED

DIVA_ERR_GROUP_ALREADY_EXISTS

DIVA_ERR_GROUP_IN_USE

DIVA_ERR_OBJECT_OFFLINE

DIVA_ERR_TIMEOUT

DIVA_ERR_LAST_INSTANCE

DIVA_ERR_PATH_DESTINATION

DIVA_ERR_INSTANCE_DOESNT_EXIST

DIVA_ERR_INSTANCE_OFFLINE

DIVA_ERR_INSTANCE_MUST_BE_ON_TAPE

DIVA_ERR_NO_INSTANCE_TAPE_EXIST

DIVA_ERR_OBJECT_IN_USE

DIVA_ERR_CANNOT_ACCEPT_MORE_REQUESTS

DIVA_ERR_TAPE_DOESNT_EXIST

DIVA_ERR_INVALID_INSTANCE_TYPE

DIVA_ERR_ACCESS_DENIED

DIVA_ERR_OBJECT_PARTIALLY_DELETED

DIVA_ERR_LICENSE_DOES_NOT_SUPPORT_THIS_FEATURE

DIVA_ERR_COMPONENT_NOT_FOUND

DIVA_ERR_OBJECT_IS_LOCKED

DIVA_ERR_OBJECT_BEING_ARCHIVED

Prefix: SPM

Description: SPM business logic result code.

Details: The description for each result code is included with the sample codes in the following list.

Sample RESULT_CODE:

SPM-LR_CONFIG_PREVENTS_DELETE - ALLOW_OBJECT_DELETION parameter in the spm.conf file was set to false and prevented SPM from deleting the last instance of a Virtual Object.

SPM-LR_CANNOT_DELETE_LAST_INSTANCE - SPM refused to delete the last instance of a Virtual Object.

SPM-LR_NO_MORE_INSTANCES_ON_SLOT - SPM refused to create more instances on a slot than it was permitted.

SPM-LR_POSTPONED_AS_WATERMARKED - SPM postponed a Delete Instance Action because it was for a Watermarked media.

SPM-LR_NO_NEED_TO_LINK_OBJECTS - The Verify stage of the LINK OBJECTS step. If the METADATA_ARCHIVE Slot was configured without Cascade Objects for Delete or Restore (both set to N), there will be no link between the Virtual Objects, and this error will be produced.

SPM-LR_CANNOT_SEND_MORE_REQUESTS - the Manager Monitor does not allow sending more requests to the Manager.

SPM-LR_ACTION_SLOT_EXPIRED - An action will have this code either because the Virtual Object was deleted from Core, but is still under SPM, or the slot expired after loading the SPM action into memory.

Prefix: USR

Description: User action was executed.

Details: The user manually marked the action complete.

Sample RESULT_CODE: USR-ACTION where ACTION can be MARKED_COMPLETE.

Storage Policy Manager Logging

The SPM Logging configuration file assists with controlling the SPM log file size and file switching. It purges old SPM logs and identifies which specific module inside of SPM to enable or disable logging for. The description of global parameters is in the following sample spm.trace.ini:

```
#####################################################################
#
# Global parameters
#####################################################################
#
@timestep = How frequently SPM must switch log files; units in
minutes.
@sizelimit = How frequently SPM must switch log files after
reaching the log file size limit; units in KB.
@timetolive = How much history of the SPM logs must be preserved;
units in days.
@tracelevel = <Value>
Value 1 = Only error messages are logged in the SPM log files
Value 2 = Only Errors and Warnings are logged in SPM log files.
Value 3 = Errors, Warnings and Information will be logged in the
SPM log files.
@tracemanagerlog = <Value>
Value 1 = enable logging for trace manager module.
Value 0 = Disable trace manager logging.
```

You enable and disable SPM module logging as follows:

- If the module is prefixed with ! then it is enabled.
- If the module is prefixed with # then it is disabled.

For example:

!SPMService

Trace is enabled for the component SPMService.

#DbConnectionPool

Trace is disabled for the component DbConnectionPool.

Frequently Asked Questions

This chapter discusses some frequently asked questions and includes the following information:

Topics:

- A newly archived Virtual Object appears to have its DATELASTUPDATE and DATENEXTUPDATE fields set to the same value as DATEARCHIVED automatically by Core independent of SPM. Is this correct?

What should be done if no SPM actions are being generated?

- Confirm the SPM Service is operational.
- Check the SPM configuration file in the conf/spm folder and verify that the values for DIVA_MANAGER_ADDRESS, DIVA_MANAGER_PORT, ORACLE_USER, ORACLE_-PASSWORD, and ORACLE_CONNECTION parameters are set correctly.
- Confirm that slots are enabled.

What should be done if only Storage actions are being generated?

- Confirm that the non-Storage type slots are enabled and are included in the correct Storage Plan.
- Confirm that in the SPM Slot configuration on the Restore, Transcode & Metadata Archive tab that the Object Name, Object Source, and Object Collection filters are set so that they do not filter out the SPM actions you are expecting.
- Check the SPM configuration file and confirm that DIVA_RESTORE_MAX_RE-QUESTS, DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS, or DIVA_METADATA_AR-CHIVE_MAX_REQUESTS are not set to 0.

What should be done if SPM actions are generated but are never executed (they stay in the Scheduled state)?

Check the Archive Date of the Virtual Objects for which the actions were generated and compare it to the values configured in the Slot Begin Time and Slot End Time parameters, and Request Execution Begin Time and Request Execution End Time in System Management App to verify if one of the following cases is true:

- If the age of the Virtual Object has not met or exceeded the number of minutes specified in the Slot Begin Time. In this case, the actions may still be executed but only after the age of the Virtual Object has met or exceeded the number of minutes specified in the Slot Begin Time.
- If the age of the Virtual Object has met or exceeded the number of minutes specified in Slot Begin Time and has not exceeded the minutes specified in the Slot End Time, but the clock time the Virtual Object has existed (for example, 9:30 to the current time of 13:20) has not yet fallen within the Request Execution Begin Time and Request Execution End Time range (for example, 13:30 to 15:30), the actions may still be executed, but only after the clock time has met or exceeded the Request Execution Begin Time.

- If the age of the Virtual Object has exceeded the number of minutes specified in the Slot End Time without the SPM action being executed, then this SPM action will never execute and stay in the SCHEDULED state. For example, if the Slot End Time is 10 minutes, but the age of the Virtual Object is 20 minutes, the SPM action for this Virtual Object will not execute even if the clock time is within the slot Request Execution Begin Time and Request Execution End Time range because the Virtual Object has expired.

This can occur by design where the Virtual Object's lifetime is expected to expire before the slot Request Execution Begin Time is even reached. It can also occur if the Core Manager is frequently or constantly running more requests than specified in the DIVA_MANAGER_MONITOR_MAX_REQUESTS parameter in spm.conf. This, in turn, can cause the SPM Service to hold off submitting the SPM actions to the Core Manager to the point where the Virtual Object eventually expires before its associated SPM action is allowed to be submitted to the Core Manager.

What should be done if SPM actions always Fail Long?

Usually when an SPM action Fails Long it is due to the request submitted by SPM terminating in the Core Manager. Check the reason why the requests are terminating in the Core Manager and remedy the problem there.

Why are SPM actions removed from the SPM Actions List?

SPM actions are removed from the SPM Actions List when the Virtual Objects they are associated with have been deleted from Core.

Why does the Start At time not update when an SPM action is rescheduled?

The Start At time only represents the initial date and time when the action was initially executed.

Why are copies being recreated on the disk after a Delete Instance?

Check that the Once Only flag in the slot configuration is set to true. If Once Only is set to false, then the system will continue to make copies of deleted instances.

What are the different components of SPM?

DSM (Disk Space Monitor) Controller

This module monitors the Watermarked array and does the cleanup when the High Watermark is reached.

SPM Controller

This module is responsible for creating, updating, and executing actions based on the configuration.

It appears that sometimes not all components start as part of SPM. What, if anything, controls whether a given component starts, and how do I tell if the DSM Controller is started since it is now integrated with SPM?

The SPM Controller is always started. The DSM Controller is started only if there is at least one Watermarked array configured.

SPM appears to communicate only with the Oracle Database and the Core Manager (through Core API calls). Does SPM interact with any other processes?

SPM only interacts with the Oracle Database and the Core Manager using the Core API. SPM may interact with DIVA Connect instead of the Core Manager if configured to do so. Otherwise, SPM does not interact with any other processes.

Is SPM event driven, or does it purely repeat some looped processing of slots and rules or database query results?

Only the DSM Controller part of SPM is event driven. It waits for the Watermarked array to reach the High Watermark to trigger the cleanup of the array.

Is SPM stateful across restarts?

SPM is stateful across restarts. It saves the current state of the actions in the database. The actions that were running are recovered during the next SPM start.

An archived Virtual Object can only belong to a single Storage Plan. When a Virtual Object is first archived, Core appears to archive to the default media for the corresponding Storage Plan, or the next-best media if the default media is not available for some reason. Is this correct?

Yes, a Virtual Object can only belong to one Storage Plan. If a Storage Plan is specified in the request, that Virtual Object is always assigned to that Storage Plan.

If the Storage Plan is specified and no destination media is specified in the request, the Virtual Object is assigned the given Storage Plan and archived to the default media of the Storage Plan. However, if the default media is not available, the Virtual Object is then archived to the first available media of the Storage Slots in the given Storage Plan in ascending order of Slot Priority, Slot Window Begin Time, and Slot ID. If no media is available for the given Storage Plan the request is terminated.

What are the meanings of the DATARCHIVED, DATELASTUPDATE, and DATENEXTUPDATE properties for Virtual Objects under SPM?

DATEARCHIVED

This property is the date when Virtual Object was archived.

DATELASTUPDATE

This property is the date of the last time actions were created or updated for this Virtual Object.

DATENEXTUPDATE

This property is the date when SPM checks for any changes in the slot that will affect any actions of this Virtual Object, and updates them if any changes are identified.

A newly archived Virtual Object appears to have its DATELASTUPDATE and DATENEXTUPDATE fields set to the same value as DATEARCHIVED automatically by Core independent of SPM. Is this correct?

Yes, DATELASTUPDATE and DATENEXTUPDATE will change after SPM generates actions for that Virtual Object.

Default Configuration Files and Parameters

This appendix describes SPM configuration parameters, and sample configuration files. The following information is included:

Topics:

SPM Configuration Parameters

The following sections describe the default SPM parameters.

Storage Plan Definitions

Storage Plan Name

This parameter is the name given to the Storage Plan.

Allow Last Instance Deletion

- Y deletes the Virtual Object.
- N only deletes the instance.
- This parameter can be overridden in the configuration file.
- The Disc Cleaning strategy is not applicable when set to Y.

Mediums Definitions

Medium Name

This parameter is the given medium name (it is better if you provide the same name as Core Tape Group/Array Name).

Storage Name

This parameter is the name of the Storage Manager. This value is not used and Telestream recommends keeping the default value (DIVA).

Tape Group/Array Name

This parameter is the Core Tape Group or array name.

Watermarked

This parameter is only applicable to Storage slots, and only for disk arrays configured as mediums. It monitors the medium's use (available space, used space, and so on). Select Y (applies watermarking), N (does not apply watermarking), or M (a combination of watermarking that is event dependent) to specify whether to apply watermarking to this medium. See *Actions, Action Steps, and Action States*, and *Watermark Based Disk Cleaning Management* for detailed information on watermarking.

In general Telestream recommends that slots are not deleted, but rather disabled. Deleting a slot does not remove the content from the Core system.

For Storage slots, setting the Slot End Time to 0 forces SPM to delete any content created by this slot, and after all of the content is removed, the Storage slot can be

deleted or disabled. If you want to retain the content in a Storage slot, but no further processing of future actions is required, disabling the slot instead of deleting it is the safer way to do this.

See *Watermark Based Disk Cleaning Management* for details on watermarking.

Disc Cleaning Strategy

This parameter is the disc cleaning strategy to use:

- Last Access Time
- Size

Virtual Object Slot Definitions

Slot Name

This parameter is the name given to the slot.

Storage Plan

This parameter is the name given to the Storage Plan.

Id

This parameter is automatically generated by SPM and not editable.

Medium Associated With Slot

This parameter is the medium associated with the slot.

Slot Request Type (or Slot Type)

This parameter is the type of request as follows:

Storage

A Copy request is initiated at the beginning of the slot, and then a Delete request is initiated at the end of the slot.

Transcode Archived

This request is initiated at the beginning of the slot.

Metadata Archive

This request is initiated at the beginning of the slot.

Restore

This request is initiated at the beginning of the slot.

Slot Begin Time

This parameter identifies when the action will occur. If set to zero, the action is initiated as soon as the Virtual Object is archived. If set to another number, for example 10 (minutes), the action will execute 10 minutes after the original Virtual Object is archived.

Slot End Time

This parameter is the time (in minutes) from the Virtual Object creation, and points to the slot end.

Request Execution Begin Time

This parameter is the time in HH:MM format (00:00 to 24:00), and points to the time of the day when execution can start.

Request Execution End Time

This parameter is the Time in HH:MM format (00:00 to 24:00), and points to the time of the day after which execution cannot start.

If the Request Execution Begin Time and Request Execution End Time are 00:00 and 24:00 respectively, there are no limitations on execution time. If the begin time is greater than end time, then the execution interval is taken from the begin time of the first day to the end time of the second day (including midnight).

Request Priority

This parameter is the priority of operations for SPM and the Core.

Filter Definitions

Filter Name

This parameter is the name you want to call the filter.

Id

This parameter is automatically generated by SPM and not editable.

Medium Filter

This parameter is the name of the media to monitor.

Object Collection Filter

This parameter is the Virtual Object Collection to display. Leaving this field blank performs no filtering and all Virtual Objects are displayed regardless of the Collection. You can use any combination except abc*def.

Object Source Filter

This parameter is the source to display. Leaving this field blank performs no filtering and all Virtual Objects are displayed regardless of the source. You can use any combination except abc*def.

Object Name Filter

This parameter is the Virtual Object name to display. Leaving this field blank performs no filtering and all Virtual Objects are displayed regardless of the name. You can use any combination except abc*def.

Minimum Object Size

This parameter is the minimum Virtual Object size. A value of zero indicates no restrictions from the Maximum Object Size.

Maximum Object Size

This parameter is the minimum Virtual Object size. A value of zero indicates no restrictions from the Minimum Object Size.

Size Units

This parameter is the Virtual Object size units filter (KB, MB, GB).

Storage Plan

This parameter is the name of the Storage Plan to filter.

SPM Actions Definitions

The following sections describe action definitions that are only used with Transcode Archived, Metadata Archive, and Restore slots.

Object Filters

Filter Name

This parameter is the name you want to call the filter.

Storage Plan

This parameter is the name of the Storage Plan to filter.

ID

This parameter is automatically generated by SPM and not editable.

Request Type

This parameter is only used with Transcode Archived, Metadata Archive, and Restore slots.

Object Collection Filter

This parameter is the Virtual Object Collection to display. Leaving this field blank indicates this parameter is not usable. You can use any combination except abc*def.

Object Source Filter

This parameter is the source to display. Leaving this field blank indicates this parameter is not usable. You can use any combination except abc*def.

Object Name Filter

This parameter is the Virtual Object name to display. Leaving this field blank indicates this parameter is not usable. You can use any combination except abc*def.

Transformation Rules

The following rules are only used with Transcode Archived and Metadata Archive slots.

Transcode Archived Request

Target Name

This parameter is the name of the transcoded Virtual Object. The ^ character results in using the original Virtual Object name, and can be added to. For example, ^.txt.

Target Collection

This parameter is the Collection of the transcoded Virtual Object. The ^ character results in using the original Virtual Object Collection, and can be added to. For example, ^.txt.

Metadata Archive Request

Target Name

This parameter is the name of the transcoded Virtual Object. The ^ character results in using the original Virtual Object name, and can be added to. For example, ^.txt.

Target Collection

This parameter is the Collection of the transcoded Virtual Object. The ^ character results in using the original Virtual Object Collection, and can be added to. For example, ^.txt.

Restore Request

Target Name

Always leave this parameter set to the default ^ character.

Target Collection

Always leave this parameter set to the default ^ character.

Request Templates

Request Type

This parameter is only used with Transcode Archived, Metadata Archive, and Restore requests.

Transformation Rules

This parameter indicates to use an existing transformation rule.

Id

This parameter is automatically generated by SPM and not editable.

Object Unmanaged Storage Repository

For Metadata Archive this parameter is the Server to archive. For Restore this parameter is the Server to restore.

Pathroot

This parameter is the Pathroot to use for Metadata Archive or Restore.

Request Options

This parameter is any additional options necessary for Transcode Archived, Metadata Archive, or Restore.

Request Comments

This parameter is comments for Transcode Archived, Metadata Archive, or Restore.

QOS

This parameter is the Core Quality of Service.

Cascade Delete Links

This parameter is only used for Transcode Archived and Metadata Archive.

Cascade Restore Links

This parameter is only used for Transcode Archived and Metadata Archive.

SPM Default Configuration File (spm.conf.ini)

```
#####################################################################
###
# SPM4 Configuration File
#
# Front Porch Digital, Inc. (C) 2005, 2006, 2007
# All rights reserved.
#
# $Source: src/main/conf/spm.conf.ini $
# $Date: 2015/01/09 17:21:34EST $
# $Revision: 1.2 $
# $Author: Ramachandran, Prakash (PRamachandran) $
#####################################################################
###
# DIVA SPM service name
# This variable can be used to specify the name of the windows
service. This is
# useful when multiple SPM are running on the same server by giving
# different names for each SPM
# If this variable is used, the service name will be “DivaSpm-
<SERVICE_NAME>”.
# Default: If this variable is unset, the service name will be
"DivaSpm".

#SERVICE_NAME=

#####################################################################
###
# MANAGER: Parameters for Core
#####################################################################
###

# The ip address and port of DIVA Manager for SPM to connect to.
# Address default is “localhost”. Port valid range is 1..65535.
Default is 9000.

DIVA_MANAGER_ADDRESS = localhost
DIVA_MANAGER_PORT    = 9000

#####################################################################
###
# ORACLE: Connection parameters for the database
#####################################################################
###

# NB! For the following "ORACLE" parameters no defaults will be
assumed!

ORACLE_USER       = diva
ORACLE_CONNECTION = lib5

# Maximum number of simultaneous DB connections. Valid range is
1..20.
# Default is 5.

ORACLE_POOL_SIZE = 5
```

```
#####################################################################
###
# MANAGER MONITOR: Parameters
#####################################################################
###

# Limit of requests (running+pending) on DIVA Manager, entire SPM
service will
# try (see docs) no to surpass. Valid range is 10..500. Default is
250.

DIVA_MANAGER_MONITOR_MAX_REQUESTS = 250

# Maximum number of DIVA delete instance and delete Virtual Object
requests.This number is
# checked before sending new Delete Requests. This value is
internally calculated
# (it's not received by Request to Manager). Therefore it's
approximation and sometimes
# we can temporarily have Number of Delete requests grate than
total number of request
# (until the next refresh ).
# Valid range is 1..300. Default is 300.Disable (no any
limitations) is 0.

DIVA_DELETE_MAX_REQUESTS = 300

# The following parameters are used to restrict how many RESTORE,
TRANSCODE and METADATA requests SPM will
# submit to manager simultaneously.
# Kindly note that all the below three parameters must be enabled
or disabled together.
#   For instance DIVA_RESTORE_MAX_REQUESTS is enabled and
DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS
#   and DIVA_METADATA_ARCHIVE_MAX_REQUESTS is disabled.
#   Example:
#     DIVA_RESTORE_MAX_REQUESTS = 3
#     DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS = -1
#     DIVA_METADATA_ARCHIVE_MAX_REQUESTS = -1
# In the above case SPM service will fail to start.
# Valid range is 0..300. Value -1 will disable this feature.
Default value is -1.

DIVA_RESTORE_MAX_REQUESTS = -1
DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS = -1
DIVA_METADATA_ARCHIVE_MAX_REQUESTS = -1

# Minimum delay in seconds between two checks of number of requests
(running+
# pending) on DIVA Manager. Checks are opportunistic. Valid range
is 1..600.
# Default is 30.

DIVA_MANAGER_MONITOR_DELAY = 30
```

```
# Number of seconds that an action will be delay for, if the
Manager Monitor
# finds Manager at >= DIVA_MANAGER_MONITOR_MAX_REQUESTS during its
execution.
# Valid range is 1..3600. Default is 60.

DIVA_MANAGER_MONITOR_ACTION_DELAY = 60

##################################################################
###
# SPM: Parameters for the SPM Controller
##################################################################
###
# Total number of “short” failures per action. Next "short" failure
will be
# translated into a “long” failure. Valid range is 1..50. Default
is 5.

ACTION_SHORT_FAILURE_LIMIT = 5

# Total number of “short” failures per step. Next “short” failure
will be
# translated into a “long” failure. Valid range is 1..50. Default
is 2.

ACTION_STEP_SHORT_FAILURE_LIMIT = 2

# Minimum delay in seconds before next check on request status.
# Valid range is 1..60. Default is 10.
ACTION_STEP_WATCH_DELAY = 10

# Maximum size of the ACTION QUEUE. Valid range is 1..300. Default
is 100.

ACTION_QUEUE_SIZE = 100

##################################################################
###
# SPM: “Rest” time for working threads
##################################################################
###

# NB! The “RECOVERY” and “RECOVERY LOAD” threads never rest - they
stop upon
# completion of all work.

# Rest time for the “EXECUTION” thread in seconds. Valid range is
0..600. Zero
# “0” will disable the thread completely. Default is 5.

SPM_EXECUTION_THREAD_REST = 5

# Rest time for the "UPDATE" thread in seconds. Valid range is
0..600. Zero “0”
# will disable the thread completely. Default is 20.
SPM_UPDATE_THREAD_REST = 20
```

```
# Rest time for the "LOAD" thread in seconds. Valid range is
0..600. Zero "0"
# will disable the thread completely. Default is 20.

SPM_LOAD_THREAD_REST = 20

#####################################################################
###
# SPM: Sizes of chunks of rows it will process at a time
#####################################################################
###

# Number of rows SPM will process at a time in each call of the
RECOVER_ACTIONS
# procedure. Valid range is 1..65535. Default is 500.

RECOVER_ACTIONS_CHUNK = 500

# Number of rows SPM will process at a time in each call of
UPDATE_ACTION_LIST
# procedure. Valid range is 1..500. Default is 50.

UPDATE_ACTIONS_CHUNK = 50

# Number of rows SPM will process at a time in each call of the
GET_ACTION_LIST
# procedure. Valid range is 1..100. Default is 25.

GET_ACTIONS_CHUNK = 25

# Time before next action update inside the UPDATE_ACTION_LIST
procedure,
# in minutes. Valid range is 10..65535. Default is 20160 (two weeks
= 60*24*7*2)

UPDATE_ACTIONS_NEXT_UPDATE = 20160

# Delay before "long" retry of a failed action inside the
UPDATE_ACTION_LIST
# procedure, in minutes.
# Valid range is 10..65535. Default is 720 (one day = 60*24).

UPDATE_ACTIONS_RETRY_FAILED_DELAY = 720

#####################################################################
###
# SPM: Processing Options
#####################################################################
###

######### --> CAUTION! COMPLETE OBJECT DELETION! <-- #########

# This feature must be managed carefully, since we can lose Virtual
Objects. This parameter must be enabled for SPm to delete Virtual
Objects.
#
```

```
# Additionally delete Virtual Object must be enabled or disabled
for each storage plan separately during
# its configuration in configuration utility for additional
security & flexibility.
# Delete Virtual Object is enabled only when it is enabled both
here and also Storage plan configuration.
# Note that, if this parameter is disabled here, allowing it for a
storage plan will have no effect.

# Delete Virtual Object is executed only in the following condition
satisfy
# 1) This parameter ALLOW_OBJECT_DELETION must be enabled
# 2) Delete Virtual Object must be enabled for the storage-plan
level through configuration utility.
# 3) All The storage slot(s) in the storage-plan associated with
the Virtual Object must have an slot end time.
# 4) Delete Virtual Object will be executed by the storage slot
that has the highest end time in the storage plan.(Please note a
Virtual Object can be associated with only one storage plan)
#    If multiple storage slots have max end time in a storage-plan,
the slot with max slot id among the slot with max end time will do
the delete Virtual Object.
#    5) Just enabling ALLOW_OBJECT_DELETION will not acknowledge
Storage slot medium water mark and Delete Virtual Object is execute
at the end of storage slot that has the highest end time in the
storage plan.
# Valid range is [true,false]. Default is "false".

ALLOW_OBJECT_DELETION = false

# The below parameter DELETE_OBJECT_ONLY_LAST_INSTANCE added the
following additional conditions to the SPM delete Virtual Object
feature
# This parameter forces SPM to delete Virtual Object at the end of
a storage slot only it is the last instance in the entire Core
system and that last instance exist on the storage slot
# medium that has the highest slot end time among all storage slots
of the Storage plan otherwise SPM will never do a delete Virtual
Object at the end of the storage slot it will always only do delete
instance
# at the end of a storage slot.
# The following condition needs to satisfied for this feature to
work.
# 1) This parameter requires ALLOW_OBJECT_DELETION also be
enabled. Just enabling DELETE_OBJECT_ONLY_LAST_INSTANCE will have
no effect and SPM delete Virtual Object feature will not be
enabled.
# 2) This parameter DELETE_OBJECT_ONLY_LAST_INSTANCE must be
enabled.
# 3) Delete Virtual Object must be enabled for the storage-plan
level through configuration utility.
# 4) All The storage slot(s) in the storage-plan associated with
the Virtual Object must have an slot end time.
# 5) Delete Virtual Object will be executed by the storage slot
that has the highest end time in the storage plan.(Please note a
Virtual Object can be associated with only one storage plan)
```

```
#    If multiple storage slots have max end time in a storage-plan,
the slot with max slot id among the slot with max end time will do
the delete Virtual Object.
#    6) At the end of the storage slot if more than 2 instances
exist in the Core system or on the medium, only delete instance is
executed.
#   7) Delete Virtual Object is executed only if one instance exist
on the entire Core system and that one instance exist on the medium
of the storage slot that has the
#    Highest slot end time among all storage slots of the storage
plan.
# 8) If the medium of storage slot that has the highest end time is
water marked the delete Virtual Object is postponed until
watermark condition is satisfied.

DELETE_OBJECT_ONLY_LAST_INSTANCE = false

# Defines whether SPM will use original or transformed Virtual
Object name as “target”
# Virtual Object name in METADATA_ARCHIVE step (Default is "true):
#    - true: use transformed Virtual Object name
#    - false: use original Virtual Object name

METADATA_ARCHIVE_TRANSFORMED_OBJECT_NAME = true

##################################################################
###
# DSM: Processing Options
##################################################################
###

# Priority of the requests produced by DELETE INSTANCE (W) actions.
# Valid range is 1..100. Default is 15.

DSM_DIW_REQUEST_PRIORITY = 15

##################################################################
###
# DSM: "Rest" time for working threads
##################################################################
###

# Rest time for the "SPACE_MONITOR" thread in seconds. Valid range
is 0..600.
# Zero "0" will disable the thread completely. Default is 10.

DSM_SPACE_MONITOR_THREAD_REST = 10

##################################################################
###
# DSM: Sizes of chunks of rows it will process at a time
##################################################################
###

# Number of Actions DSM will process at a given time.
# procedure. Valid range is 1..65535. Default is 50.
```

```
DSM_ACTIVATE_ACTIONS_CHUNK = 50

#####################################################################
###
# DSM: Disk Arrays to be monitored
#####################################################################
###

# Description of disk arrays. For each array, an ARRAY<i> parameter
has to be
# created, with <i> indicating the array number. The first array
number is 1.
# <i> has to be increased always by one.
#
# The information on array is the following:
#
#    "array name";"watermark low-level percent";"watermark high-
level percent"
#          [[;"mounted disk path"] ...]
#
# where:
#   "array name"                   - name of the disk array, as it
named in
#                                  DIVA (It is not the SPM "medium"
name!)
#   "watermark low-level percent" - number from 0 to 100 - lower
disk space
#                                   usage percent: DSM will try to
reach this
#                                     percent when cleaning
#   "watermark high-level percent" - number from 0 to 100 - higher
disk space
#                                    usage percent: DSM will start
cleaning
#                                    process when disk space usage
is higher
#                                     than this level
#   ["mounted disk path"]...       - mount points of array disks.
None, one or
#                                     many, separated by ";". No
spaces allowed.
#                                  If one or many, DSM will access
arrays via
#                                   filesystem ("old") interface.
#                                  If none, DSM will access arrays
via DIVA
#                                     API ("new") interface
(recommended).
#
# Windows specific: If the monitored disk is a password protected
network
# share, the following syntax is allowed as a mount point:
#   cifs://user:pwd@\\nas\share

# When SPM service is configured with CIFS disk, The login of the
SPM service must be changed from the
# default "Local System" to a valid Windows User.
```

```
#Following are the steps to change the spm service login.

# Step 1: Start -> run
# Step 2: Type services.msc in the rum prompt and press enter.
# Step 3: This should open the Windows service Control Manager
# Step 4: Select the SPM service to which you want to change the
login.
# Step 5: Right click on the SPM service and select properties.
# Step 6: Select the Log on tab in the properties window of the SPM
service.
# Step 7: Select the radio button next to "This account".
# Step 8: Enter a valid windows user and password
# Step 9: Click apply or ok button to commit the changes.

#ARRAY1 = FastDisk;75;90;/home/diva;/usr/bin
#ARRAY2 = SlowDisk;60;75;/tmp
#ARRAY3 = SlowDisk2;60;99
#ARRAY4 = FullDisk;50;80
#####################################################################
###
# SPM/DSM: Additional parameters
#####################################################################
###

# Level of tracing. Valid range is 1..2. Default is 2.
#     "1" - will trace "entry" and "exit" points in all important
functions.
#          WARNING! Will generate big volume of trace! Must be used
for 'debug'
#           and 'validation' purposes only!
#     "2" - will produce normal, 'production' trace.
#
TRACE_LEVEL = 2
```

SPM Trace Configuration File (spm.trace.ini)

```
#####################################################################
#
# SPM Trace Configuration File
#
# Front Porch Digital, Inc. (C) 2005, 2006, 2007
# All rights reserved.
#
# $Source: spm.trace.ini $
# $Date: 2007/12/13 15:43:35EST $
# $Revision: 1.8 $
# $Author: Ramachandran, Prakash (PRamachandran) $
#####################################################################
#
# uncomment the following lines to trace
#####################################################################
#

#####################################################################
#
# Global parameters
#####################################################################
#
@timestep=60
@sizelimit=200
@timetolive=10
@tracelevel=3
@tracemanagerlog=0

#####################################################################
#
# Daemon and Managers
#####################################################################
#

!SPMService

!Config

!DBManager
!CommManager
!FSManager
!ManagerMonitor
!TraceWrapper
?
#####################################################################
#
# DivaOracle Interface
#####################################################################
#

!Oracle
#DbConnectionPool
#DbConnection
```

```
#####################################################################
#
# SPM Controller
#####################################################################
#

!RecoveryThread
!RecoveryLoadThread
!LoadThread
!UpdateThread
!ExecutionThread

!ActionQueue

#####################################################################
#
# DSM Controller
#####################################################################
#

!SpaceMonitorThread

!ArrayContainer

#####################################################################
#
# Action Processing Level
#####################################################################
#

!RestoreAction
!MetaDataArchiveAction
!TranscodeArchiveAction
!ArchiveRequestAction
!DeleteObjectAction
!DeleteInstanceAction
!CopyAction
!RequestAction
!Action

!ActionRecord

#####################################################################
#
# Action Step Processing Level
#####################################################################
#

!RESTORE_STEP
!ARCHIVE_STEP
!TRANSCODE_ARCHIVE_STEP
!LINK_OBJECTS_STEP
!DELETE_OBJECT_STEP
!DELETE_INSTANCE_W2_STEP
!DELETE_INSTANCE_W1_STEP
!DELETE_INSTANCE_STEP
!COPY_STEP
```

```
!RequestActionStep
!ActionStep

#####################################################################
#
# Method Tracer
#####################################################################
#

#MethodTracer
```

Glossary

Actions

An action is associated with a slot and executed during the slot's open period.

AXF Format

The AXF (Archive Exchange Format) is based on a file and storage media encapsulation approach which abstracts the underlying file system, operating system, and storage technology making the format truly open and non-proprietary. AXF helps ensure long term accessibility to valued assets, and keeps up with evolving storage technologies.

DSM (Disk Space Monitor)

A module in SPM that assists in array cleanup when the Virtual Object level reaches the High Watermark.

Filter

Determines what Virtual Objects are affected by what Storage Plan.

Legacy Format

Core proprietary storage format used in Core releases 1.0 through 6.5.1.

Medium

Storage media accessible to SPM (Disk Arrays and Tape Groups).

Slot

Contains the action to be applied when the associated Storage Plan is executed.

Step

The current state of progression of an action or process.

Storage Plan

Actions to execute when new content arrives.