Oracle® DIVArchive

Storage Plan Manager (SPM) Guide

Release 7.3

E64045-01

January 2016

ORACLE®

Oracle DIVArchive Storage Plan Manager (SPM) Guide, Release 7.3

E64045-01

Primary Author: Lou Bonaventura

Table of Contents

1 INTRODUCTION 1
1.1 Document Purpose and Scope 1
1.2 Document Conventions 1
1.3 Definitions, Acronyms, and Special Terms 2
1.4 Storage Plan Manager Overview 3
1.5 DIVArchive Object Lifecycle 3
1.6 Object Lifecycle Management 9
1.7 SPM Slot Types 10
1.7.1 Actions, Action Steps and Action States 11
1.8 DIVArchive Watermark Array Management 13
1.9 Storage Plan Manager Workflows 18
1.9.1 Storage Plan Manager Tasks 18
1.9.2 Simple Object Lifecycle Example 19
1.9.3 How SPM Retries Failed Actions and Connection Issues 20
1.9.4 Action Slot Workflow 22
1.9.5 Storage Slot Workflow 23
1.10 Recommended Practices 24
1.10.1 Best Practices for SPM Configuration 24
1.10.2 Best Practices to Setup Tape Groups in DIVArchive 26
1.10.3 Valid Reasons to Create Multiple Tape Groups 26
1.10.4 Invalid Reasons to Create Multiple Tape Groups 27
1.10.5 Simple SPM Configuration Example 28
1.10.6 Additional SPM Configuration Examples 29
2 INSTALLATION AND CONFIGURATION 32
2.1 Installation Prerequisites 32
2.2 SPM Configuration File Parameters 32
2.3 Request Type Distribution 42
2.4 Storage Plan Definition 44
2.4.1 The DIVArchive Configuration Utility Storage Plans Tab 44
2.4.2 Recommended Practice 45
2.4.3 Creating the new Storage Plan 45
2.4.4 Creating Mediums 48
2.4.5 Creating Filters 51
2.4.6 Alternate Methods of Assigning Storage Plans to an Object 53
2.5 Creating Slots 57
2.5.1 Storage Slot Configuration 57

2.5.1.1 Nearline Storage Slot Configuration 65
2.5.2 Transcode Archived Slot Configuration 67
2.5.3 Metadata Archive Slot Configuration 77
2.5.4 Restore Slot Configuration 85
2.5.5 SPM Behavior with Delete Object 94
2.5.5.1 `ALLOW_OBJECT_DELETION` Parameter 94
2.5.5.2 `DELETE_OBJECT_ONLY_LAST_INSTANCE` Parameter 94
2.6 CHANGING A STORAGE PLAN 96
2.6.1 Disabling and Deleting Slots 99
2.6.2 Adding a Slot 99
2.7 WATERMARK-BASED DISK CLEANING MANAGEMENT 100
2.8 EXTERNAL STORAGE PLANS 101
2.9 CONFIGURATION VALIDATION 101
3 OPERATIONS 104
3.1 SERVICE MANAGEMENT – SPM COMMAND LINE INTERFACE 104
3.2 ACCESSING SPM INFORMATION THROUGH THE DIVARCHIVE CONTROL GUI 105
3.2.1 SPM Actions Panel View Filtering 108
3.2.2 Assigning a New Storage Plan 109
3.2.3 Rescheduling Failed Actions 112
3.3 SPM ACTION RESULT CODES 114
3.4 SPM LOGGING 117
4 FREQUENTLY ASKED QUESTIONS 118
4.1 WHAT SHOULD BE DONE IF I SEE THAT NO SPM ACTIONS ARE BEING GENERATED? 118
4.2 WHAT SHOULD BE DONE IF ONLY STORAGE ACTIONS ARE BEING GENERATED? 118
4.3 WHAT SHOULD BE DONE IF SPM ACTIONS ARE GENERATED BUT ARE NEVER EXECUTED (THEY STAY IN THE SCHEDULED STATE)? 118
4.4 WHAT SHOULD BE DONE IF SPM ACTIONS ALWAYS FAIL LONG? 119
4.5 WHY ARE SPM ACTIONS REMOVED FROM THE SPM ACTIONS LIST? 119
4.6 WHY DOES THE START AT TIME NOT UPDATE WHEN AN SPM ACTION IS RESCHEDULED? 119
4.7 WHY ARE COPIES BEING RECREATED ON THE DISK AFTER A DELETE INSTANCE? 119
4.8 WHAT ARE THE DIFFERENT COMPONENTS OF SPM? 120
4.9 IT APPEARS THAT SOMETIMES NOT ALL COMPONENTS START AS PART OF SPM. WHAT, IF ANYTHING, CONTROLS WHETHER A GIVEN COMPONENT STARTS OR NOT AND HOW DO WE TELL IF THE DSM CONTROLLER IS STARTED SINCE IT IS NOW INTEGRATED WITH SPM? 120
4.10 SPM APPEARS TO COMMUNICATE ONLY WITH THE ORACLE DATABASE AND THE MANAGER (BY WAY OF DIVARCHIVE API CALLS). DOES SPM INTERACT WITH ANY OTHER PROCESSES? 120
4.11 IS SPM EVENT DRIVEN OR DOES IT PURELY REPEAT SOME LOOPED PROCESSING OF SLOTS/RULES OR DATABASE QUERY RESULTS? 120
4.12 IS SPM STATEFUL ACROSS RESTARTS? 120

4.13 An archived Object can only belong to a single Storage Plan. When an Object is first archived, DIVArchive Manager 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? *121*

4.14 What are the meanings of the DATARCHIVED, DATELASTUPDATE, and DATENEXTUPDATE properties for objects under SPM? *121*

4.15 A newly archived Object appears to have its DATELASTUPDATE and DATENEXTUPDATE fields set to the same value as DATEARCHIVED automatically by DIVArchive independent of SPM. Is this correct? *121*

APPENDIX **122**

A1 Default Configuration (spm.conf.ini) *122*

A2 SPM Trace Configuration File (spm.trace.ini) *131*

A3 Configuration Parameter Summary *134*

A3.1 Storage Plan Definitions *134*

A3.2 Mediums Definitions *134*

A3.3 Object Slots Definitions *136*

A3.4 Filter Definitions *138*

A3.5 SPM Actions Definitions (only used with Transcode Archived, Metadata Archive, and Restore Slots) *139*

A3.5.1 Object Filters 139

A3.5.2 Transformation Rules (only used for Transcode Archived and Metadata Archive) 139

A3.5.3 Request Templates 141

Tables Index

Table 1: Definitions, Acronyms, and Special Terms 2

Table 2: SPM-Specific Terminology 3

Table 3 SPM Slot Types and Actions 10

Table 4: SPM Action States 12

Table 5: SPM Internal Tasks 18

Table 6: DIVArchive SPM Service Name Parameters 33

Table 7: Parameters for DIVArchive Manager 33

Table 8: Connection Parameters for the Oracle Database 33

Table 9: SPM Manager Monitor Parameters 34

Table 10: SPM Controller Parameters 36

Table 11: Rest Time Parameters for SPM Working Tasks 36

Table 12: Sizes of Chunks of Rows SPM will Process at One Time 37

Table 13: SPM Processing Options 38

Table 14: DSM Task Options 39

Table 15: Rest Time Parameters for DSM Working Tasks 39
Table 16: Sizes of Chunks of Rows DSM will Process at One Time....40
Table 17: Disk Arrays to be Monitored 40
Table 18: Additional Parameters....41
Table 19: DIVArchive Configuration GUI Storage Plan Parameters 47
Table 20: DIVArchive Configuration GUI Mediums Parameters 49
Table 21: DIVArchive Configuration GUI Filters Parameters....52
Table 22: DIVArchive Configuration Utility Parameters for Storage Slots....58
Table 23: DIVArchive Configuration Utility Parameters for Transcode Archived Slots....69
Table 24: Object Filter Parameters for Transcode Archived Slots 73
Table 25: Transcode Archived Request Template Parameters 75
Table 26: DIVArchive Configuration Utility Parameters for Metadata Archive Slots....78
Table 27: Object Filter Parameters for Metadata Archive Slots 81
Table 28: Request Template Parameters for Metadata Archive Slots 84
Table 29: DIVArchive Configuration Utility Parameters for Restore Slots....87
Table 30: Object Filter Parameters for Restore Slots 91
Table 31: Request Template Parameters for Restore Slots 93
Table 32: SPM Command Line Interface Commands 104
Table 33: SPM Command Line Interface Options 104
Table 34: SPM Action Result Codes....114
Table 35: SPM Result Code Descriptions 115

Figures Index

Figure 1: Keep Object on Medium-1 for 75 Minutes (0 to 75)....4
Figure 2: Copy the Object to Medium-2 and Store Forever 5
Figure 3: Transcode Archived Object at 30 Minutes....6
Figure 4: Archive Metadata at 45 Minutes....7
Figure 5: The Object is Restored at 60 Minutes and Deleted from Medium-1 at 75 Minutes. 8
Figure 6: Creating a New Slot....11
Figure 7: Permanently Stored Object Size is Larger than Low Watermark Level....14
Figure 8: Watermarked Disk Array: No Objects....14
Figure 9: Watermarked Disk Array: Permanently Stored Objects....15
Figure 10: Watermarked Disk Array: Temporary Storage Lower than High Watermark....15
Figure 11: Watermarked Disk Array: Temporary Storage Reaches High Watermark 16
Figure 12: Watermarked Disk Array: Objects Deleted Down to Low Watermark 16
Figure 13: Control GUI – SPM Actions View....18

Figure 14: Action Slot Workflow 22
Figure 15: Storage Slot Workflow 23
Figure 16: Simple SPM Configuration Example 28
Figure 17: Additional SPM Configuration Example – 1 29
Figure 18: Additional SPM Configuration Example – 2 30
Figure 19: Additional SPM Configuration Example – 3 31
Figure 20: DIVArchive Configuration Utility Storage Plans Tab 44
Figure 21: Creating a New Storage Plan 45
Figure 22: Add New Row Window for Storage Plans 46
Figure 23: New Storage Plan Showing in the Configuration Utility Storage Plan Section 48
Figure 24: Creating a New Media Group 48
Figure 25: Add New Row Window for Media Groups 49
Figure 26: New Medium Showing in the Configuration Utility Mediums Section 51
Figure 27: Creating a New Filter 51
Figure 28: Add New Row Window for Filters 52
Figure 29: New Filter Showing in the Configuration Utility Filters Section 53
Figure 30: Sending an Archive Request to Mapped Media 54
Figure 31: Configuration Utility Showing Media Mapping 55
Figure 32: Adding a New Media Map Configuration 55
Figure 33: Click the Plus Sign to Begin Creating a New Slot 57
Figure 34: Slot Configuration Window for Storage Slots 58
Figure 35: New Storage Slot Showing in the Configuration Utility Slots Section 64
Figure 36: Creating a New Slot 67
Figure 37: Select **TRANSCODE_ARCHIVED** from the Pull-down Box 68
Figure 38: Add New Row Window for Transcode Archived Slots 69
Figure 39: New Transcode Archived Slot Showing in the Configuration Utility Slots Section 72
Figure 40: Slot Configuration Window – Additional Configuration Bottom Tabs 73
Figure 41: Slot Configuration Window – Additional Configuration Bottom Tabs 75
Figure 42: Add New Row Window for Metadata Archive Slots 77
Figure 43: Slot Configuration Window – Additional Configuration Bottom Tabs 81
Figure 44: Slot Configuration Window – Additional Configuration Bottom Tabs 83
Figure 45: Slot Configuration Window – Restore Slot 85
Figure 46: Slot Configuration Window for Restore Slots 86
Figure 47: Slot Configuration Window – Additional Configuration Bottom Tabs 91
Figure 48: Slot Configuration Window – Additional Configuration Bottom Tabs 93
Figure 49: Storage Plan A Configuration 96
Figure 50: Initial Slot Configuration for Storage Plan A 97

Figure 51: New Slot Configured for Storage Plan A 97
Figure 52: Updated Final Slot Configuration 98
Figure 53: Open the Configuration Utility and Select Any Tab....101
Figure 54: Validating the SPM Configuration using the DIVArchive Configuration Utility....101
Figure 55: Successful SPM Validation Completed 102
Figure 56: SPM Validation Found an Error in the SPM Configuration 103
Figure 57: Control GUI Archive Objects Window Showing Object Storage Plans....105
Figure 58: DIVArchive Control GUI SPM Actions Panel – 1 106
Figure 59: DIVArchive Control GUI SPM Actions Panel – 2 107
Figure 60: Request Properties Pop-up Window 107
Figure 61: SPM Actions Panel Filtering....108
Figure 62: SPM Actions Panel Filtering by Action Status 108
Figure 63: SPM Actions Panel Filtering by Action Type 109
Figure 64: Selecting Multiple Objects for Assignment 109
Figure 65: Assigning a New Storage Plan for an Object....110
Figure 66: Selecting the Object's New Storage Plan....111
Figure 67: Reschedule Pop-up Menu....112
Figure 68: Action Rescheduling Confirmation Window....113
Figure 69: SPM Actions Tab Showing Action Result Codes....114

1 Introduction

1.1 Document Purpose and Scope

The purpose of this document is to describe the installation, configuration, and operations of the Oracle DIVArchive Storage Plan Manager for Oracle DIVArchive 7.3. Installation, Administration and Operations personnel should use this guide to follow all of the necessary steps to provide full functionality of the Storage Plan Manager component.

WARNING: Misconfiguring the SPM may lead to unexpected and disastrous results! Minor editions to slots can lead to catastrophic consequences, for example the deletion of hundreds of thousands of instances on tape (*e.g. wrong value in a slot's end time*), or database corruption (*e.g. creating a STORAGE slot while the SPM is running and changing it to RESTORE type while* `copy` *and* `deleteinstance` *actions have already begun processing*). Without special training and familiarity with the product, the Oracle representative should always be contacted before making any changes to SPM. Failure to do so may result in severe damage to the DIVArchive System or even permanent data loss.

1.2 Document Conventions

The following conventions are used with respect to text:

Normal Standard Text.

Italic Used to emphasize a term or variable.

Bold Used to emphasize critical information.

6.1 Refers to a section or sub-section in the document.

`Courier New` Used for system screen output and system commands.

The following conventions are used with respect to file paths or variables:

- **`DIVA_HOME`**: The Root Path on the file system where DIVArchive is installed.
- **`$db_user`**: DIVArchive Database user name
- **`$db_password`**: DIVArchive Database user password
- **`$sid`**: Oracle instance name (*configured in* ***`tnsnames.ora`***)

The following conventions are used with respect to figures and drawings:

Red outlined boxes pointing to specific areas in a figure indicate procedural steps, or point out specific parameters being discussed in the section text.

Red outlined boxes that surround specific areas in a figure indicate specific areas of the figure being discussed in the section text.

1.3 Definitions, Acronyms, and Special Terms

Table 1: Definitions, Acronyms, and Special Terms

Term	Definition
AXF (*or AXF Media Format*)	The Archive Exchange Format (*AXF*) 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.
Complex Object	An Object is defined as a Complex Object when it contains 1,000 (*configurable*) or more components. Complex Object handling may differ from non-Complex Objects as noted throughout this document.
DSM	A module in SPM that assists in Array clean-up when the object level reaches the High Watermark.
Filter	Determines what objects are affected by what Storage Plan.
FTP	File Transfer Protocol
Medium	Storage media accessible to SPM (*Disk Arrays and Tape Groups*).
Legacy Format	DIVArchive proprietary storage format used in DIVArchive version 1.0 through 6.5.1.
Middleware	Software consisting of a set of services allowing multiple application processes running on one or more machines to interact.
Non-Complex Object	By default, DIVArchive Object with 1,000 files or less are considered Non-Complex Object. The Maximum number of files a Non-Complex Object can hold is configurable.
Process	A sequence of instructions that may execute in parallel with each other, or in a specified order from beginning to end.
Slot	Contains the action to be applied when the associated Storage Plan is executed.
SPM	Storage Plan Manager: Typically installed on the same machine as the Oracle DIVArchive Manager.
Storage Plan	Actions to execute when new content arrives.

1.4 Storage Plan Manager Overview

The DIVArchive **Storage Plan Manager** (*SPM*) software component provides the capability to manage object lifecycle (*interacting with the Oracle DIVArchive Manager*) and is typically installed on the same machine as the DIVArchive Manager. For example, an archived object can reside on a specific medium the first day, and migrate (*over time*) to a different medium according to the policies and rules established by the user. DIVArchive executes the object lifecycle migration as a background activity by following the rules and policies defined in the corresponding Storage Plan.

1.5 DIVArchive Object Lifecycle

The following sections describe the steps and processes encompassing the DIVArchive Object Lifecycle.

Along with the standard DIVArchive terminology, Storage Plan Manager has additional terms that require identification. Understanding these additional terms will assist in comprehending the DIVArchive SPM module, the processes involved, and the DIVArchive Object Lifecycle.

Table 2: SPM-Specific Terminology

Term	Definition
Storage Plan	Set of rules defining the object lifecycle within the DIVArchive System. • Each object is assigned 1 and only 1 Storage Plan. • Storage Plans are assigned to objects using Filters. • The default Storage Plan contains no Filter, and no actions are assigned to be executed on the object. The default Storage Plan is **`SP_DEFAULT`**. If an Object did not correspond with any filter parameters, SPM will assign the **`SP_DEFAULT`** Storage Plan. **`SP_DEFAULT`** is defined only for compatibility purposes and must not have any slots assigned to it.
Slot	A rule with a set of Actions to be taken against the objects associated with a Storage Plan. • Storage Plans may contain a single Slot or multiple Slots. • A Slot is generally analogous to a *window of opportunity* for when an Action may be performed. This is the open time left for the Slot, minus the daily schedule of the Slot.
Action	Actions are contained within the Slots and define the activity to be taken on objects assigned to a Storage Plan (*Copy, Delete, Transcode Archived, Metadata Archive, or Restore*).

Term	Definition
Filter	A set of parameters used by SPM to identify if any Storage Plan needs to be assigned to the object. • If an object corresponds with a Filter's parameters, the Filter will assign a Storage Plan to the object and the Storage Plan's associated Slots will execute the necessary actions on the object.
Media	A Tape Group or Disk Array as defined in the DIVArchive System.
Medium	The alias from media (*Groups or Arrays*) used in SPM configuration and workflows.

There are several steps included in the object lifecycle, which are depicted in the general example below. This example illustrates a generic object lifecycle which could be automated using SPM.

Figure 1: Keep Object on Medium-1 for 75 Minutes (0 to 75)

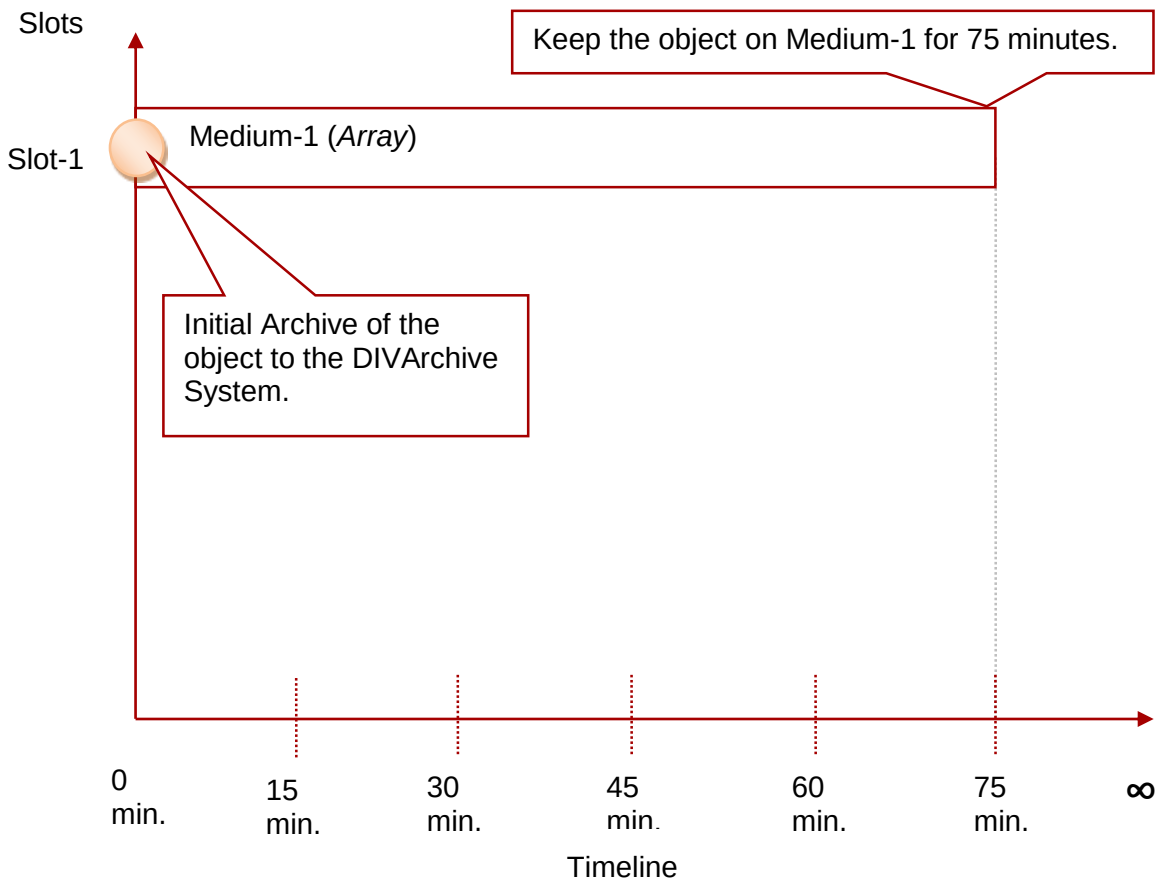


Figure 2: Copy the Object to Medium-2 and Store Forever

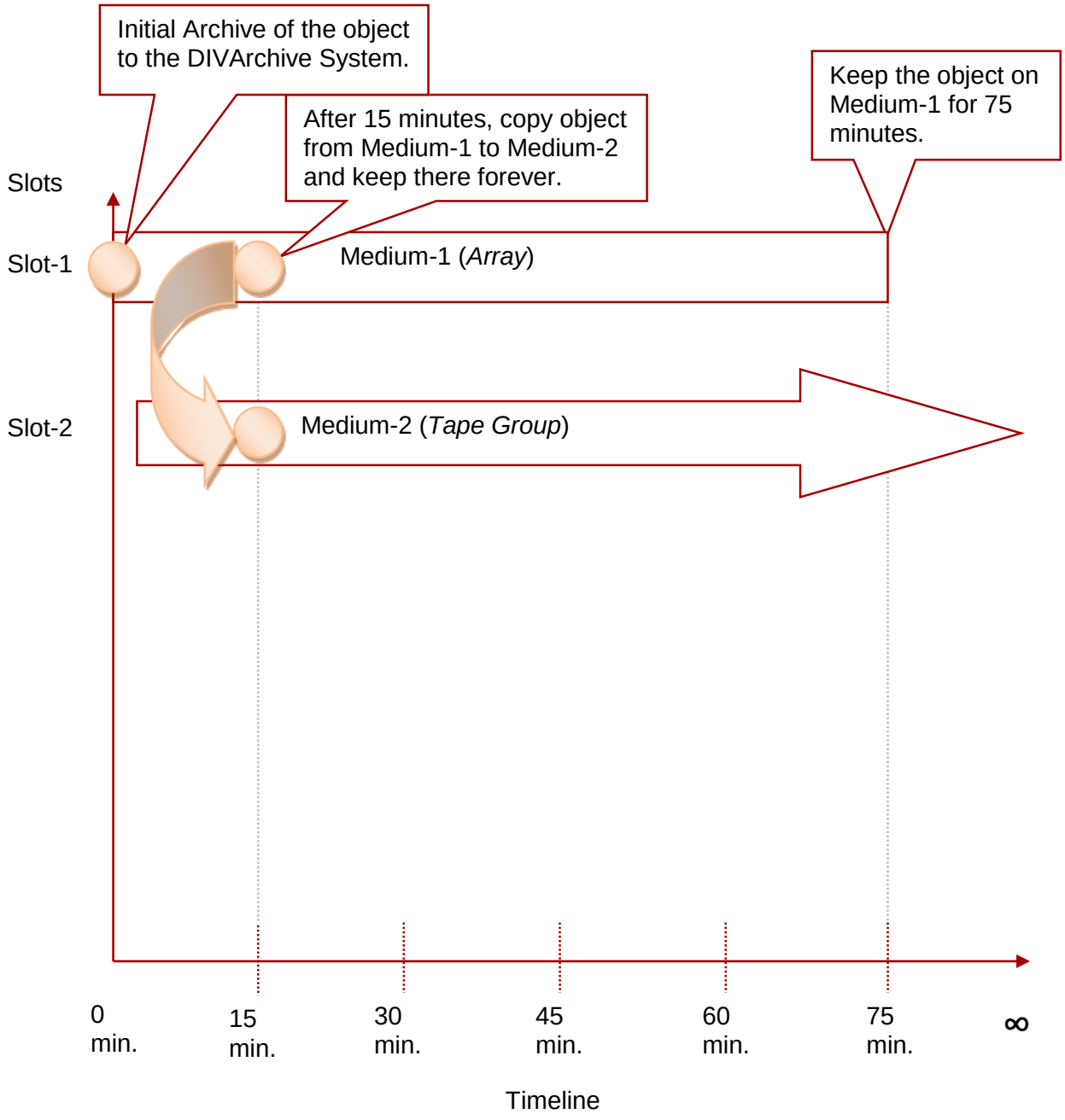


Figure 3: Transcode Archived Object at 30 Minutes

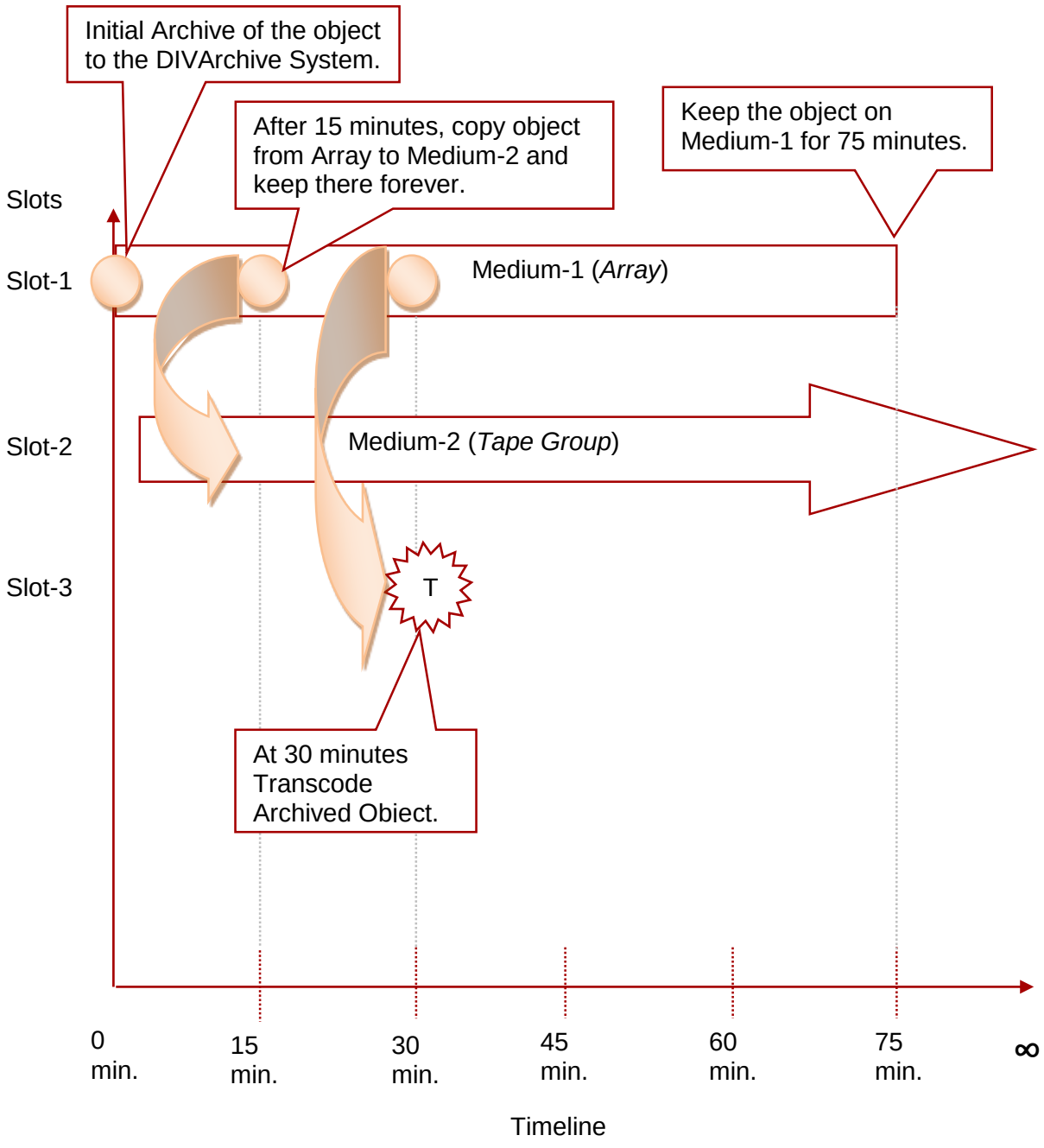


Figure 4: Archive Metadata at 45 Minutes

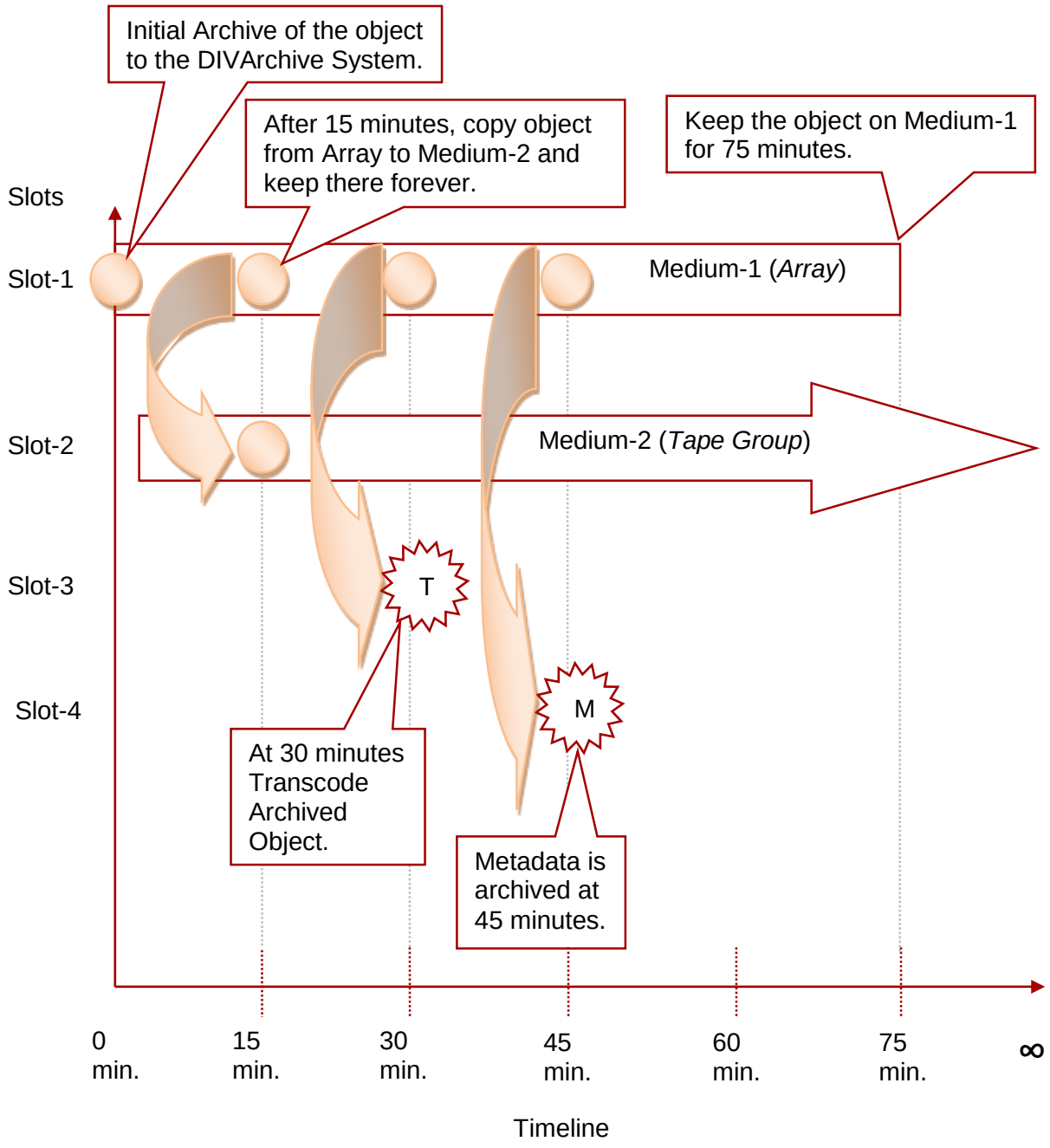
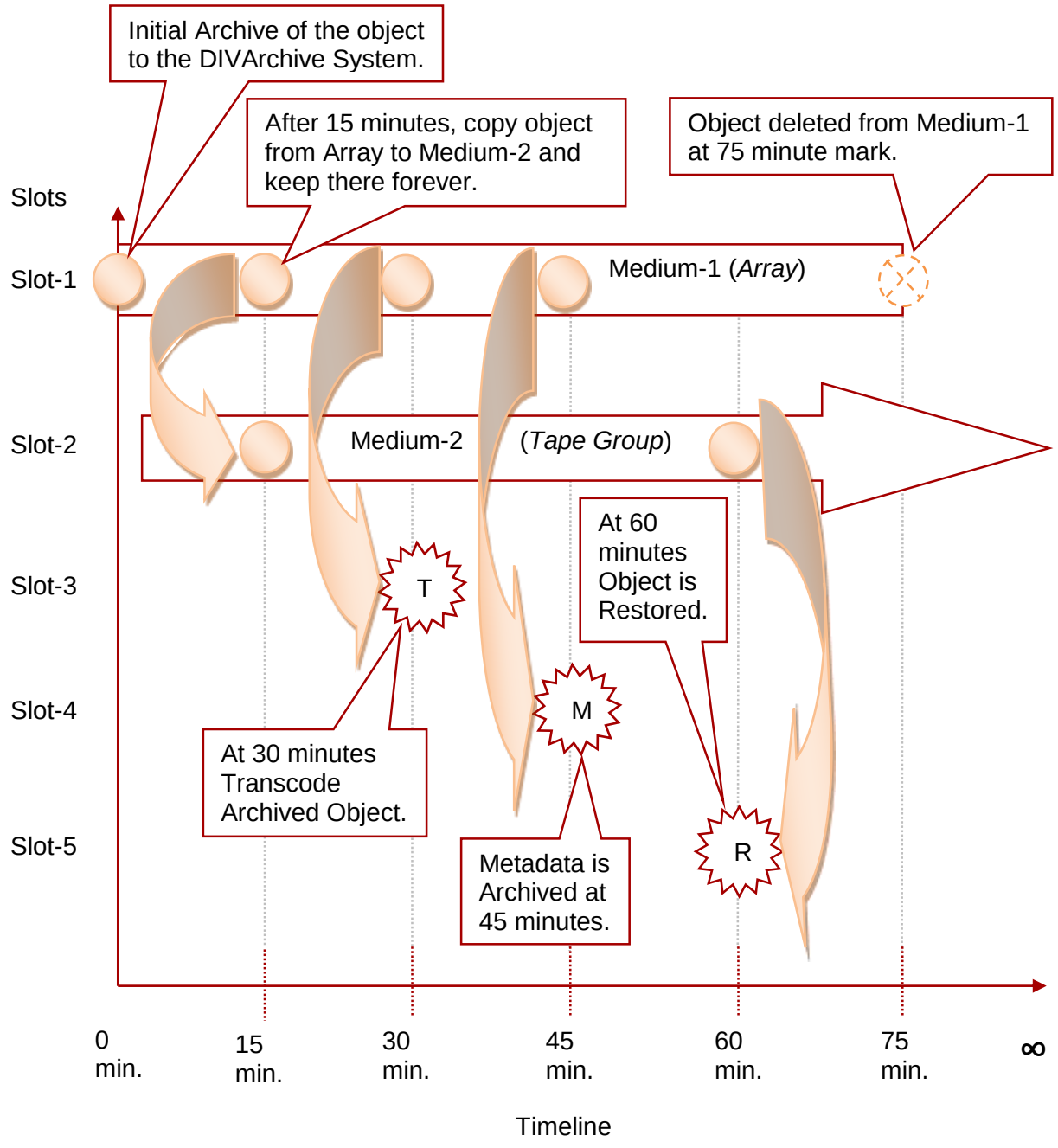


Figure 5: The Object is Restored at 60 Minutes and Deleted from Medium-1 at 75 Minutes.



1.6 Object Lifecycle Management

DIVArchive can define a storage policy for each object or group of objects. This allows the user to describe the complete lifecycle of an object and to manage the content migration throughout the lifecycle. By using the Storage Plan Manager, content lifecycle management becomes a background process that is performed automatically.

The Storage Plan is used to specify the different migrations and copies of the specified objects from one media to another, according to storage rules and policies defined by the user. These rules and policies include parameters such as:

- The amount of time an object is to be retained on a medium.
  - This time interval is known as a **slot**.
- Where to copy an object when its time has expired on the medium (*e.g., copy from Array to tape after 2 weeks*).
- Time-of-day (*local time*) when the action is executed.
  - For example, tasks that consume relatively large amounts of system resources can be scheduled during low-usage periods (*e.g. between midnight and 5 am*). Note that these tasks are executed within the interval specified in the corresponding slot configuration.
- When and where to restore an object.
- When to transcode an object and where to store the transcoded content.

The Storage Plan is managed by the Storage Plan Manager (*SPM*) component in DIVArchive. 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 object. The Storage Plan feature enables the customization of content lifecycle management.

1.7 SPM Slot Types

There is a logical workflow to the DIVArchive SPM component:

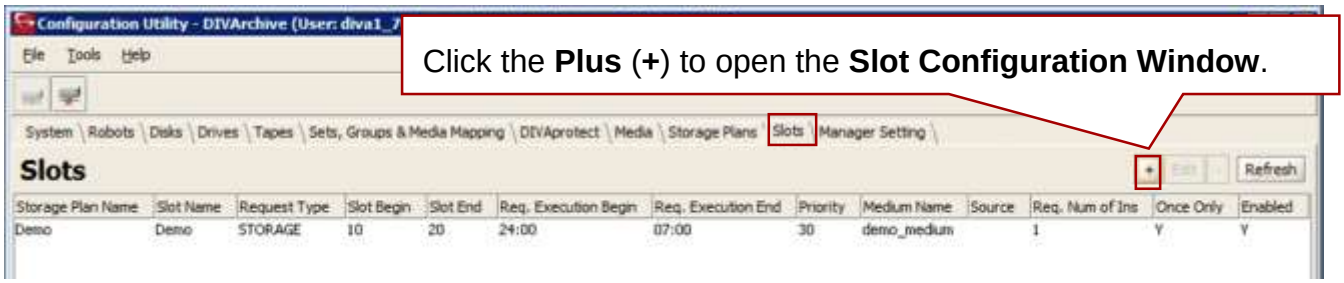
- **Mediums** are a combination of defined medium (*Disk Arrays or Tape Groups*) available to SPM.
- Each **Storage Plan** is associated with one or more **Mediums** and contains one or more Slots.
- Each defined **Filter** is associated with a **Storage Plan**.
- Each **Slot** defines 1 or more actions to be performed:

Table 3 SPM Slot Types and Actions

Slot Type	Actions
Storage	Allows for an object to be copied and then deleted (*if the slot is configured for deletion*). Object deletion refers to either a `DeleteInstance` or `DeleteObject` process, depending upon the SPM configuration and whether or not other instances still exist. Uses only the **Storage Plans Tab** in the DIVArchive Configuration Utility. **Refer to Section 1.10.5 for information on Storage Slot workflows, and Section 2.5.1 for information on configuring Storage Slots.**
Transcode Archive	Allows for Transcoding an archived object. Uses both the **Storage Plans Tab** and **SPM Actions Tab** in the DIVArchive Configuration Utility. **Refer to Section 1.10.4 for information on Action Slot workflows and Section 2.5.2 for information on configuring Transcode Archive Slots.**
Metadata Archive	Allows for archiving of the object's Metadata File. Uses both the **Storage Plans Tab** and **SPM Actions Tab** in the DIVArchive Configuration Utility. **Refer to Section 1.9.4 for information on Action Slot workflows and Section 2.5.3 for information on configuring Metadata Archive Slots.**
Restore	Allows for Restoring of an object. Uses both the **Storage Plans Tab** and **SPM Actions Tab** in the DIVArchive Configuration Utility. **Refer to Section 1.9.4 for information on Action Slot workflows and Section 2.5.3 for information on configuring Restore Slots.**

To create a new Slot, click the **Plus** (+) in the **Slots** section of the Configuration Utility.

Figure 6: Creating a New Slot



1.7.1 Actions, Action Steps and Action States

A single **Storage Plan** may have a single **Slot** or multiple **Slots**. Each **Slot** has one or two **Actions** associated with it.

Every **Action** has one or two **Steps** associated with it:

- Copy, Transcode Archived, Metadata Archive and Restore.
  - One step (Copy, Transcode Archived, Metadata Archive, or Restore).
- Delete (valid for Storage Slots only)
  - Two steps:
    i. Postponed
       1. Before SPM executes any Delete Actions, it checks whether or not 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; it will mark the object for deletion and set the object's state to Postponed.
       2. If the medium is **not** watermarked, there will only be one step (*Delete)* and the object will be deleted immediately upon expiration.

    ii. Delete
       1 When the (***Yes***) watermarked medium level reaches the High Watermark, then the object will actually be deleted.

With the new **Unified Slot View (*Slots Tab*)**, all Slot details are managed from a single Window and all Slot Types only require configuration from the **Slots Tab** of the DIVArchive Configuration Utility.

When a new object is added to DIVArchive, SPM checks for object compliance with the filters specified in the system. When an object conforms to one of the configured filters, the filter determines which configured Storage Plan to use for processing of the object. An object can be assigned to only one Storage Plan; if the object conforms to multiple filters, the first filter the object conforms to is the one that is considered.

The Slots associated with the identified Storage Plan determine the actions performed on the object. If an object did not conform to any configured filters, the object is assigned to the **SP_DEFAULT** Storage Plan. ***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 an object goes through different states as shown in the table below. Each state will finish processing before the status is updated to the next state.

Example:

1. An object matches the filter for a specific Storage Plan.
2. The Storage Plan's Slot schedules the associated Action(s) for execution on the object (*Status* = *Scheduled*).
3. The Action is then loaded into the Action Queue (*Status* = *Loaded*).
4. The Action is now executed (*Status* = *Processing*).
5. Upon successful execution of the Action, the status is updated to Completed (*Status* = *Completed*).

The additional statuses that an Action could be in are Postponed, Failed Long, and Rejected. Refer to the table below for the definitions of these additional statuses and Sections 1.10.4 and 1.9.5 for additional Slot Workflow information.

The table below identifies the different SPM Action States:

Table 4: SPM Action States

State	Definition
Scheduled	The initial state of an Action. The Action is scheduled for execution 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 object for deletion and set the state to **Postponed**. SPM will not actually delete the object until the **High Watermark** level is reached and then the object will be removed from the Array. If the medium is not watermarked, the object will be deleted immediately. **Only Deleted Actions can be in this state.**
Completed	The Action has completed processing successfully.
Failed Long	The Action failed and will be retried according to the SPM Configuration. Refer to parameter **UPDATE_ACTIONS_RETRY_FAILED_DELAY** in Table 12.

State	Definition
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 a number of instances greater than the number of instances configured in the storage slot.

1.8 DIVArchive Watermark Array Management

Deletions on disk mediums can be managed by watermarks. When watermarks aren't used, deletions occur immediately after the slot expires. When watermarks are used, deletions will be postponed until the Disk Array's occupied space hits a configured watermark.

The **Disk Space Monitor** (*DSM*), a function of the SPM, monitors SPM identified Arrays, not individual disks. The **DSM** process only starts if there is a SPM Array that is configured for Watermarking. Refer to the

SPM Configuration File Parameters or **A1 Default Configuration** for configuration information.

When an object is set for deletion by a Storage Slot, it is not actually deleted until the watermarked Array reaches the High Watermark. Once the Array reaches the High Watermark, objects marked for deletion will be deleted either by Last Access Time or Largest Object Size, depending on how the Watermark is configured. Objects are then deleted until the Low Watermark is reached.

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

- **Last Access Time**: This will delete the oldest objects, amongst the objects marked for deletion (*in non-Mixed Mode*), first according to the last time the object was accessed.
- **Largest Object Size**: This will delete the largest objects, amongst the objects marked for deletion (*in non-Mixed Mode*), first according to the object size.

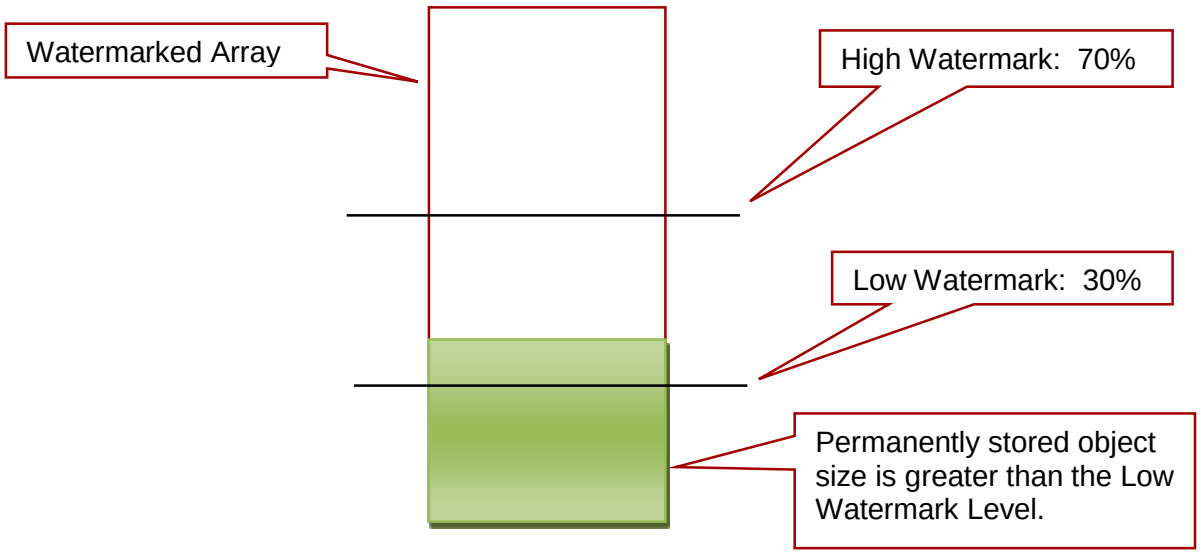
DSM has 2 methods for checking the Array watermark levels:

- Through the DIVArchive API.
  - Requests are sent to the Manager and the watermark levels are sent back to DSM.
  - Typically this method is used.
- Directly from the Array.
  - This method uses the Operating System's commands to retrieve the information.

Recommended Practice:

If the total size of all permanently stored objects (*objects with no specified expiration time*) on an Array is greater than the configured Low Watermark, it is recommended to reconfigure the Low Watermark setting to a value higher than the amount of permanently stored objects.

Figure 7: Permanently Stored Object Size is Larger than Low Watermark Level



The following figures characterize a typical Watermarked Medium (*watermarking set to **Y**es*) and illustrate how Watermarking is used to ensure the medium does not reach full capacity.

Figure 8: Watermarked Disk Array: No Objects

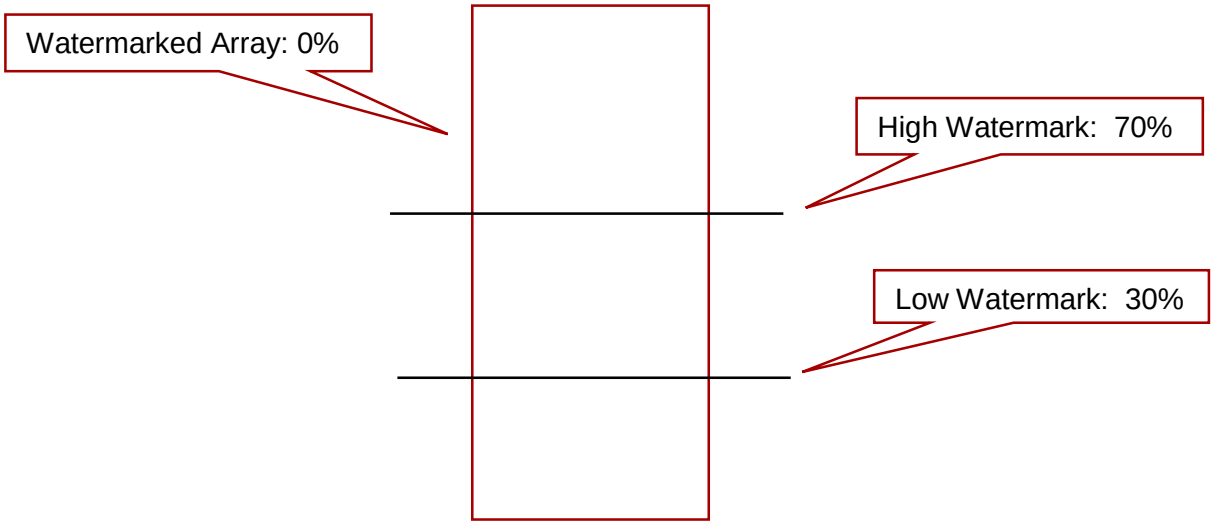


Figure 9: Watermarked Disk Array: Permanently Stored Objects

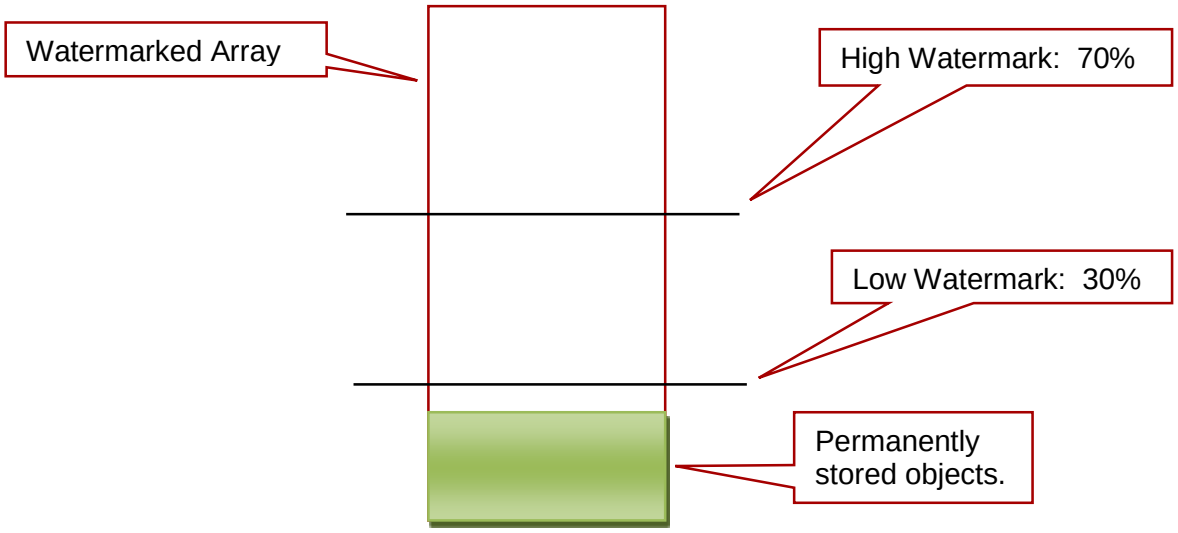


Figure 10: Watermarked Disk Array: Temporary Storage Lower than High Watermark

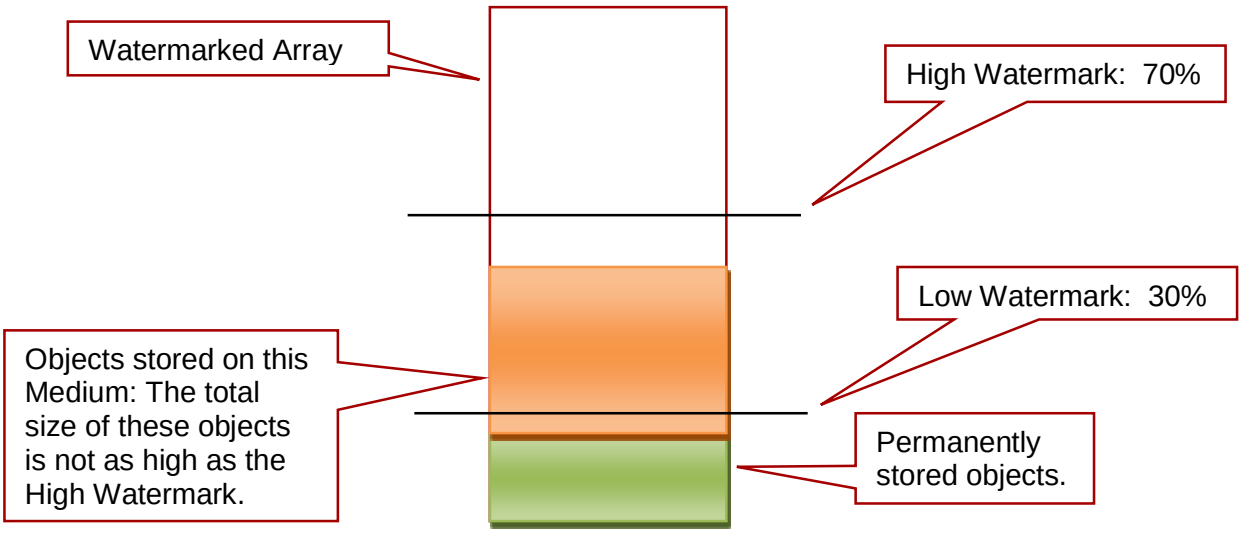


Figure 11: Watermarked Disk Array: Temporary Storage Reaches High Watermark

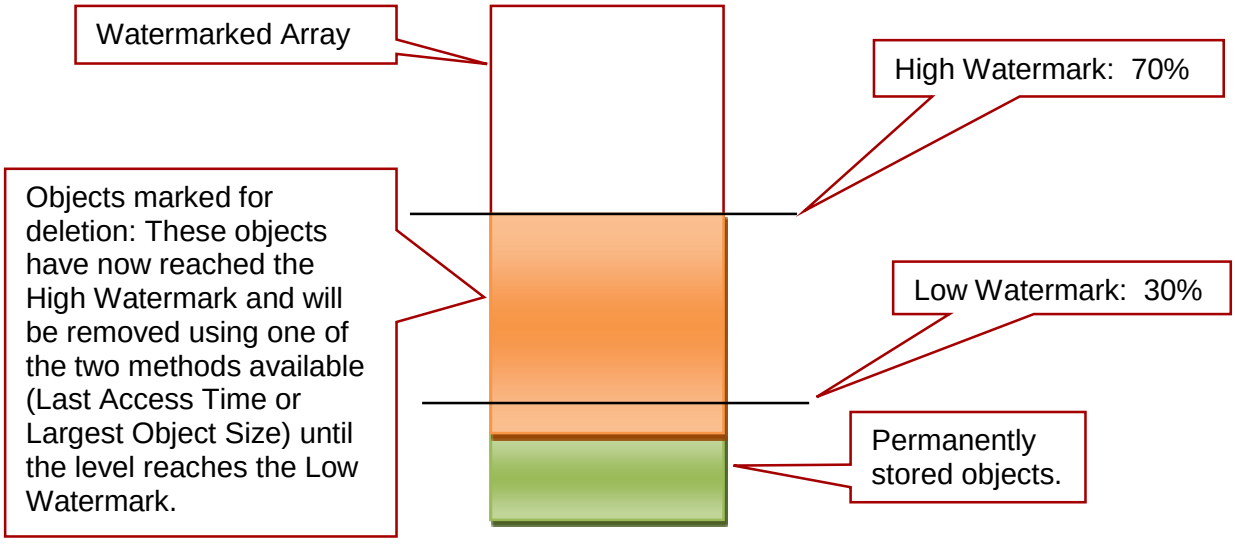
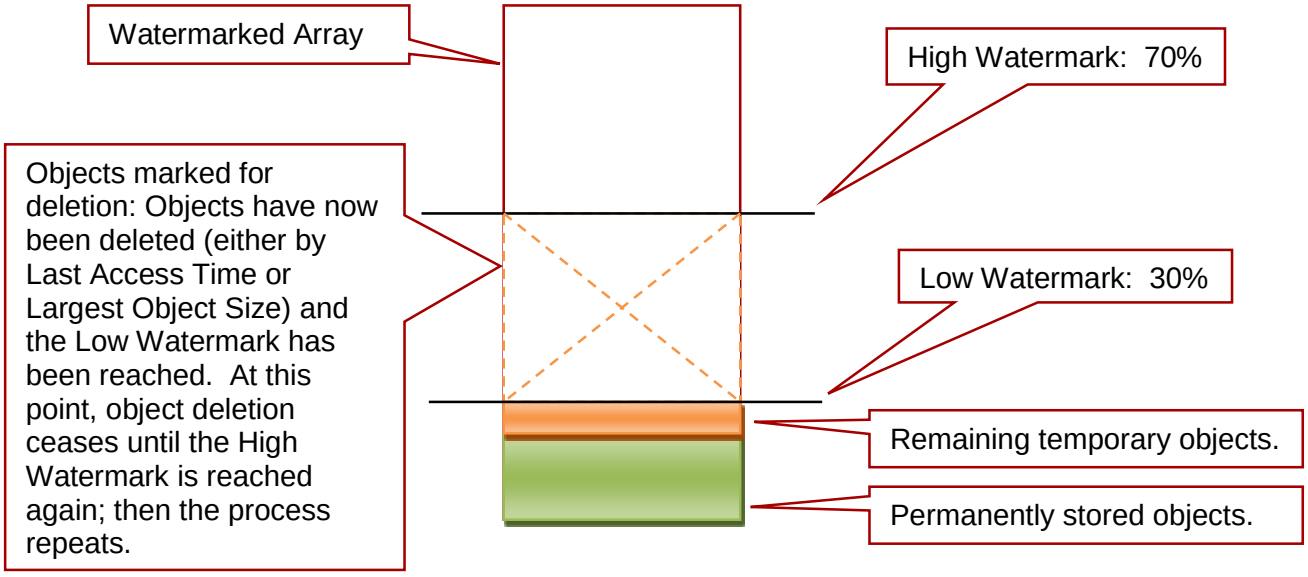


Figure 12: Watermarked Disk Array: Objects Deleted Down to Low Watermark



SPM Arrays have three possible Watermark settings that are configurable in the DIVArchive Configuration Utility:

- **Y**es: Watermark this Array (*as in the examples*).
  - Only considers objects already marked for deletion.
- **N**o: Do not watermark this Array.
- **M**ixed: This is a combination of watermarked and non-watermarked. The action taken is dependent upon which event happens first:
  - The Slot reaches its end time.
  - The High Watermark is reached.
  - Considers both objects marked for deletion and objects whose slots are still open.
  - Only valid for **Storage Slots**.

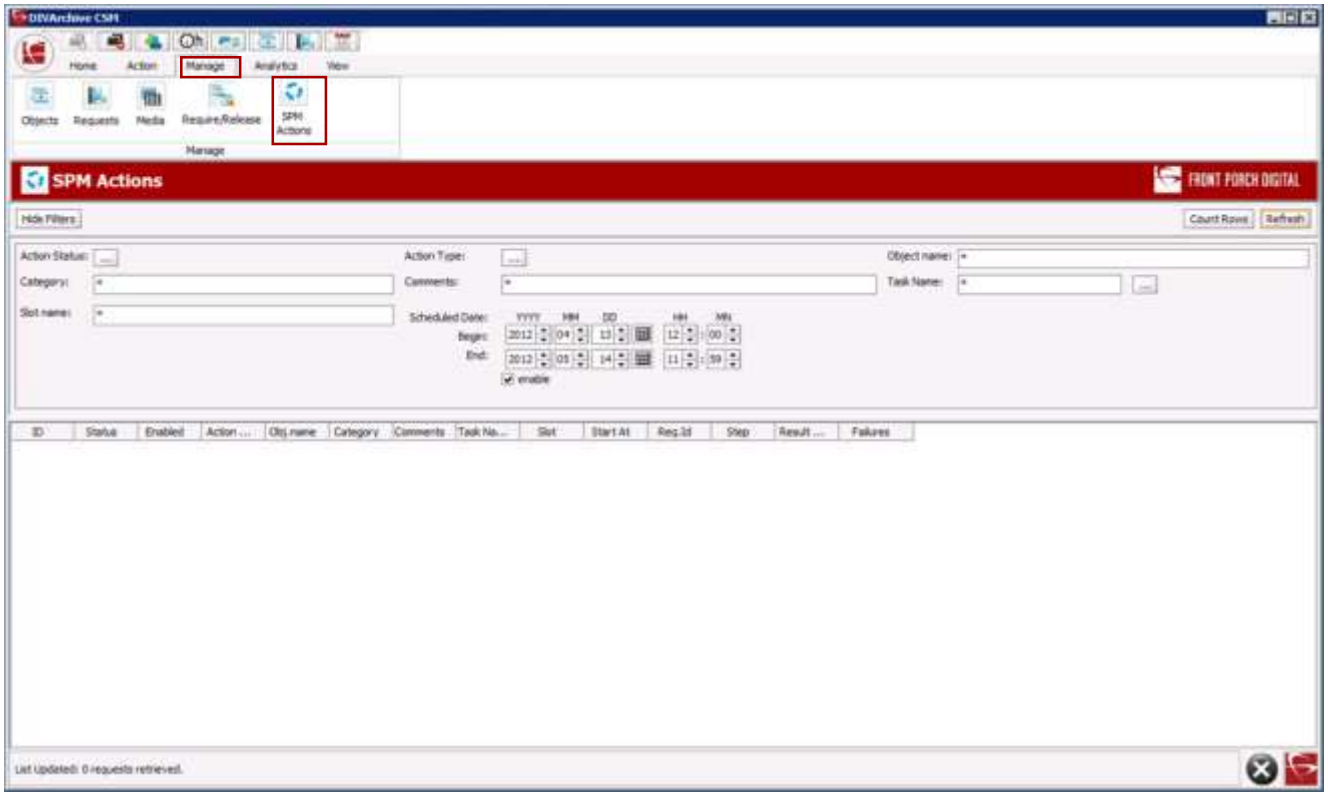
Note: To change the watermark state of an Array, SPM must be restarted.

1.9 Storage Plan Manager Workflows

1.9.1 Storage Plan Manager Tasks

The SPM module runs several processes, each in charge of a particular task, and uses the DIVArchive Database to process Actions. There are multiple tasks processing in parallel at all times when SPM is operational. All **Actions** currently being executed by SPM reside in the **Action Queue** until the execution is completed and then they are deleted from the queue. The Control GUI now has an **SPM Actions** panel located in the **Manager Tab** and Actions within the queue may be viewed using the GUI's **SPM Actions Panel**. You must at least be connected to the DIVArchive Database to access this view.

Figure 13: Control GUI – SPM Actions View



The tasks utilized by SPM are:

Table 5: SPM Internal Tasks

Task	Function
Update	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	Loads the Actions into the Action Queue from the Database for processing.

Task	Function
Execution	Executes the Actions that are loaded into the Action Queue and submits requests to the Manager as necessary.
Disk Space Monitor (*DSM*)	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 killed or stopped during execution of the Actions. If the rescheduled Action has a **DIVArchive 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 the DIVArchive Manager. An action will be in an inconsistent state and have a **Request ID** associated with it if SPM was killed or stopped when SPM: 1. Has executed the action. 2. Submitted the request to the DIVArchive Manager. 3. Was waiting to for the status of the request. The **Recovery Task** only runs during start-up of SPM. If there are no actions in an inconsistent state, the task will just end.

1.9.2 Simple Object Lifecycle Example

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

1. An object exists on the Video Server.
2. The object is archived to a Disk Array that is:
    - Known to the DIVArchive System.
    - On one of the monitored Mediums in SPM.
3. SPM processing then begins:
    a. SPM checks to see if the object matches any of the SPM Filters.
    b. If the object matches one of the filters, processing by SPM continues by identifying which Storage Plan is associated with the corresponding filter.
    c. Once 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 object.
    e. SPM processes the detected Actions (*included in the Slots*) on the 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 object is archived, it is copied to a target Tape Group.
5. Ten minutes later (*20 minutes after initially being archived to DIVArchive*) it is deleted from the DIVArchive System and SPM.

1.9.3 How SPM Retries Failed Actions and Connection Issues

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`**.

Manager Connection Failures

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

Database Connection Failures

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

Missing Instances

Below is an example of what will occur if the SPM is looking for an instance that does not exist.

Example:

A Storage Slot is configured with **`Once Only`** set to **`N`**, the Slot starts at 5 minutes after archiving and ends 30 minutes after archiving.

- After 5 minutes, SPM has finished the Copy Action.
- After 10 minutes the user manually deletes the copy made by SPM
  - SPM will sense the deletion made by the user and creates the copy again because the Slot period has not ended yet.
- The user manually deletes the copy made by SPM 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 would make the copy only one time. If the user manually deletes the copy made by SPM, SPM won't make the copy again.

Action Retries

- **Manager is Busy:**
  - If SPM could not execute the Actions by submitting a request because the Manager is busy executing more request 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 the Manager 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 1000 times using the configured delay. If the Action continues to fail, it is marked as **`REJECTED`** and is never retried again.
  - Once 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 a number of instances greater than the number of instances 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.

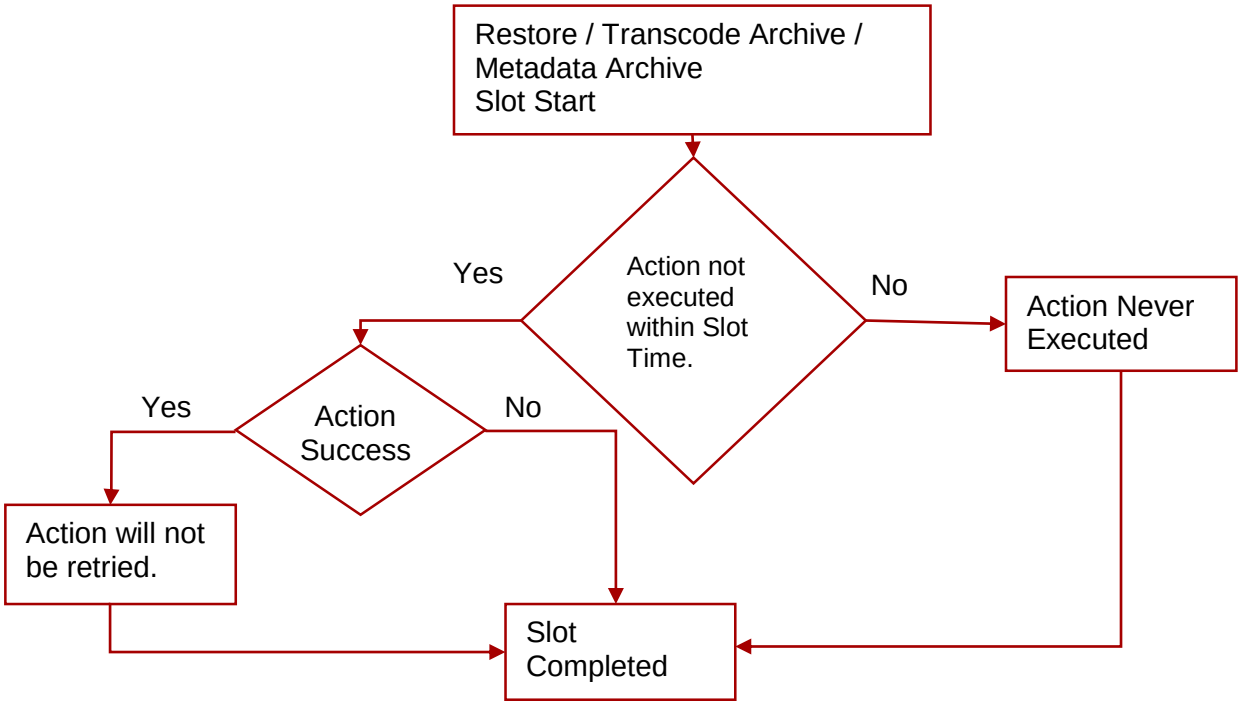
Note: Refer to Sections 1.10.4 and 1.10.5 for Action Slot and Storage Slot Workflows.

1.9.4 Action Slot Workflow

Action Slots have a specific workflow that encompasses what DIVArchive 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 configured Slot Time.

Figure 14: Action Slot Workflow

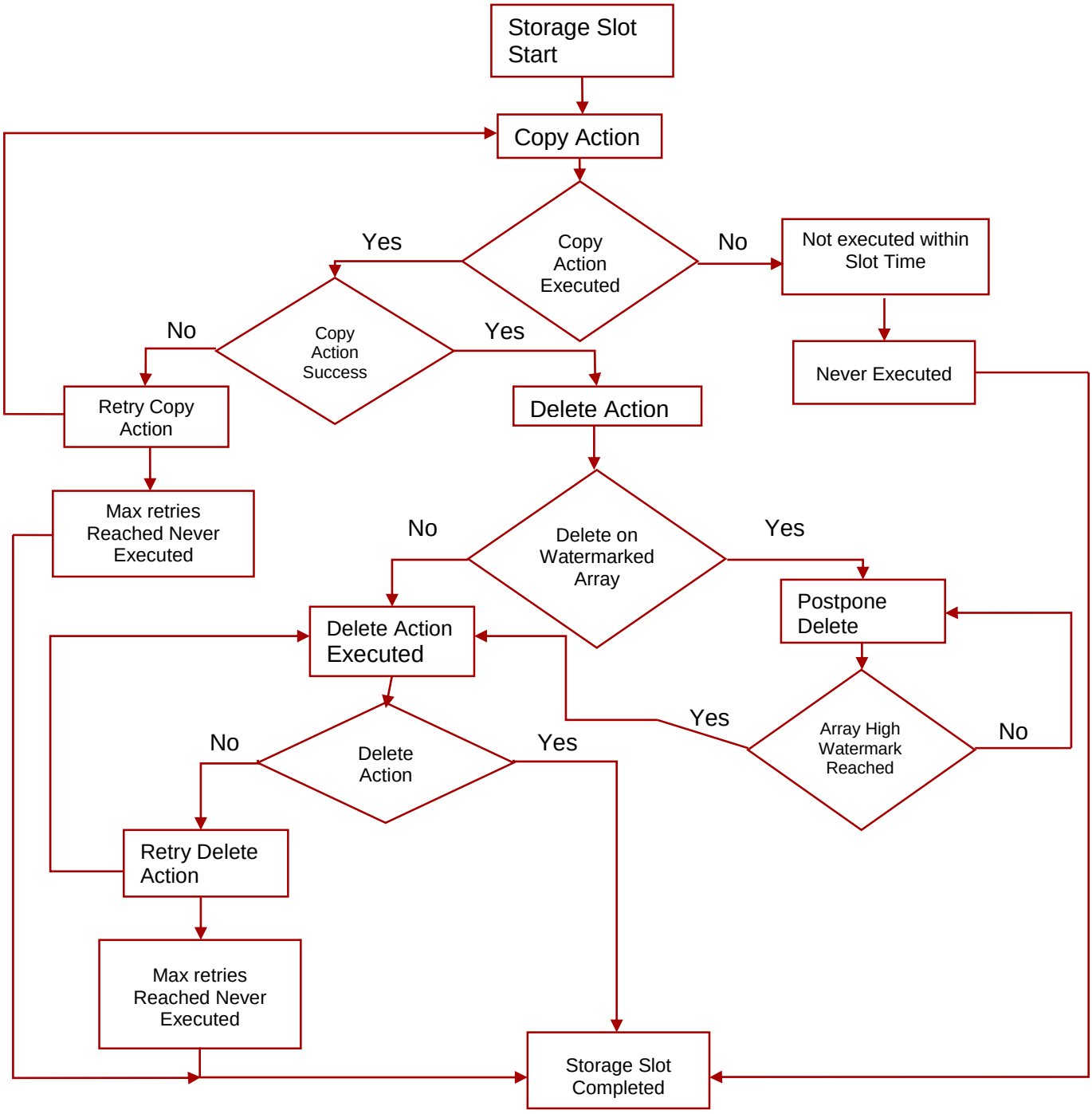


1.9.5 Storage Slot Workflow

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

- Success
- Failed Long

Figure 15: Storage Slot Workflow



1.10 Recommended Practices

1.10.1 Best Practices for SPM Configuration

Typical SPM Workflow

A typical SPM workflow consists of archiving objects to disk, 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 as long as space allows (*watermarked mode*). Keeping a disk copy for some period of time allows faster access to the data.

Note: Refer to Sections 1.10.4 and 1.10.5 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 end time of the slot (*e.g. 2 days from the Archive*), at which time the disk instance is deleted (*the corresponding* **`DeleteInstance`** *SPM Action is sent to the Manager then updates to the Complete state*).

Note: Refer to Section 1.8 for more information on Watermarking.

Watermarked Mode

In Watermarked Mode, the disk instance is not deleted when the end time of the slot is reached; it is simply flagged as expired (*by setting the corresponding* **`DeleteInstance`** *SPM Action to the Postponed state*). When the disk's space usage hits a configurable High Watermark (*e.g. 90%*), SPM will remove as many expired instances from the disk as required to get the space usage down to the configured Low Watermark (*e.g. 60%*).

In Watermarked Mode, it is advisable not to set the disk slot's end time for an extended time period. The reason for this is so that SPM does not run short of expired instances when a disk purge is in order. However, the slot's end time should be far enough in the future to allow the tape copy to complete; if a tape copy isn't 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 3 hours (*180 minutes*), or 24 hours (*1440 minutes*) if the slot's schedule only allows copies to be executed at a particular time of day. Make sure the array is large enough to accommodate newly archived instances for the slot's configured time. If 100 gigabyte of data per hour is archived at peak time and the slot's end time is configured to 3 hours, the array should be 500 gigabyte or more.

Note: Refer to Section 1.8 for more information on Watermarking.

Choosing Appropriate Watermarks

Watermarks should be carefully chosen from the start, but experience dictates that proper values are typically obtained after spending some time observing the system's behavior. It is recommended to start with common values and fine-tune them later. Typical common starting values are:

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

Remember the watermarks refer to the usage ratio of a particular Array, not a Disk. To compute an Array's usage ratio, the Manager examines each disk composing the Array and divides the sum of each disk's space usage 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% will be considered by the Manager as 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 gives a total of 2.5 TB used, divided by a 3 TB total capacity*).

Note: Refer to Section 1.8 for more information on Watermarking.

General Watermarking Rules:

- Do not set the High Watermark too high.
  - When the HWM is reached, SPM will begin deleting expired instances. This process may take some time, especially if the instances are small because the purge will require more **`DeleteInstance`** operations. Make sure that the archive activity won't store enough new instances on the array in the meantime to fill it to 100%.

    Example:

    If the HWM is 90%, then the available space is 100 - 90 = 10%. If the Array is 2 terabyte large, this 10% represents 200 gigabyte. If the SPM purge process encounters numerous small instances requiring deletion, the process will be slow and the possibility exists that archive activity may store 200 gigabyte (*and more*) during the purge's execution, filling the array to 100% during the process and initiating Archive aborts.

    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 disk instances will be kept on disk, minimizing the chance to restore from disk and the benefit of having disk instances.

If the Array contains persistent data that can't be purged (e.g. objects belonging to a Storage Plan that keeps disk instances for an unlimited time, or user files not belonging to DIVArchive), the LWM will need to be set 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).

Note: Refer to Section 1.8 for more information on Watermarking.

1.10.2 Best Practices to Setup Tape Groups in DIVArchive

Although Tape Group setup is a different topic in DIVArchive, setting up Tape Groups needs to be completed before configuring the Storage Plans. The number of Storage Plans required is usually the same as the number of Tape Groups configured. Tape Groups allow DIVArchive to physically separate archived content into different tapes, and typically creating one Storage Plan per Tape Group is necessary. All SPs are typically setup to be exactly the same, except that the copy goes to different a Tape Group. However, the more Tape Groups DIVArchive uses, the less efficiently content will be stored across tapes.

Note: If Complex Objects are to be used in the system, be certain to setup Tape Groups having tapes formatted to use the AXF format. Complex objects are <u>not</u> compatible with the Legacy formatted tapes or disks. Non-Complex Objects may be stored using either Legacy or AXF format.

The example below illustrates how creating too many Tape Groups will cause fragmentation of objects across the groups. Using fewer groups results in fewer SPs to setup, less fragmentation across tapes, and is easier to maintain.

Example:

Configuration:

- 10 Tape Drives
- 30 Tape Groups with SET-ID 10
- 300 Total Tapes assigned to SET-ID 10

Results:

- Each Tape Group (*in the worst case*) will use at least 10 tapes and store files on each tape when an Object is archived.
- After 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% filled, no more tapes are available.
- DIVArchive will run out of tapes even though lot of the tapes are still mostly empty (*each containing only 1 object*) but cannot be used as it was assigned to a different Tape Group.

In this case having fewer Tape Groups will resolve this issue. This method avoids fragmentation across the Tape Groups.

1.10.3 Valid Reasons to Create Multiple Tape Groups

There are several valid reasons to create multiple Tape Groups:

- Long and short form materials should be stored on different Tape Groups because if small objects are mixed with larger objects on the same tape, access to the smaller object will be delayed for an extended period of time until the larger 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 get fragmented and repack 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.
- When requirements necessitate using tapes purchased by different departments and enforcing that each department uses only the tapes they purchase themselves, Tape Groups need to be created for each department with a different set of tapes.
- Different storage formats should be assigned to different groups. For example, one group would be Legacy Format while another group is AXF Format. Complex Objects are **not** compatible with Legacy format and **must** be processed to/from AXF formatted medium. Non-Complex Objects are compatible with both Legacy and AXF formats.

In general creating multiple tape groups should only be done for a good reason; otherwise Tape Groups will cause Tape Group Fragmentation, resulting in some of the tapes not getting filled and storage space being wasted.

1.10.4 Invalid Reasons to Create Multiple Tape Groups

There are several invalid reasons for creating multiple Tape Groups:

- Different content is to be stored on different Tape Groups because it is thought to be easier to manage. DIVArchive manages the tapes; when restoring an object on a tape, DIVArchive automatically knows which tape the object is on and mounts that tape, or notifies the user to insert the necessary tape into the library. It will not require the user to figure out which tape is needed.
- Creating many Tape Groups because it is easier to search. DIVArchive does not search by Tape Groups but by Object Metadata stored in the database. The 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. DIVArchive does not work efficiently this way and this method should not be utilized.

Excessive Tape Groups will cause DIVArchive to work less efficiently, result in fragmentation across the Tape Groups, and are more difficult to configure and maintain. So unless for the above valid reasons, the recommended best practice for creating Tape Groups is to use as few Tape Groups as possible (*refer to Section 1.10.2 for recommended best practices*).

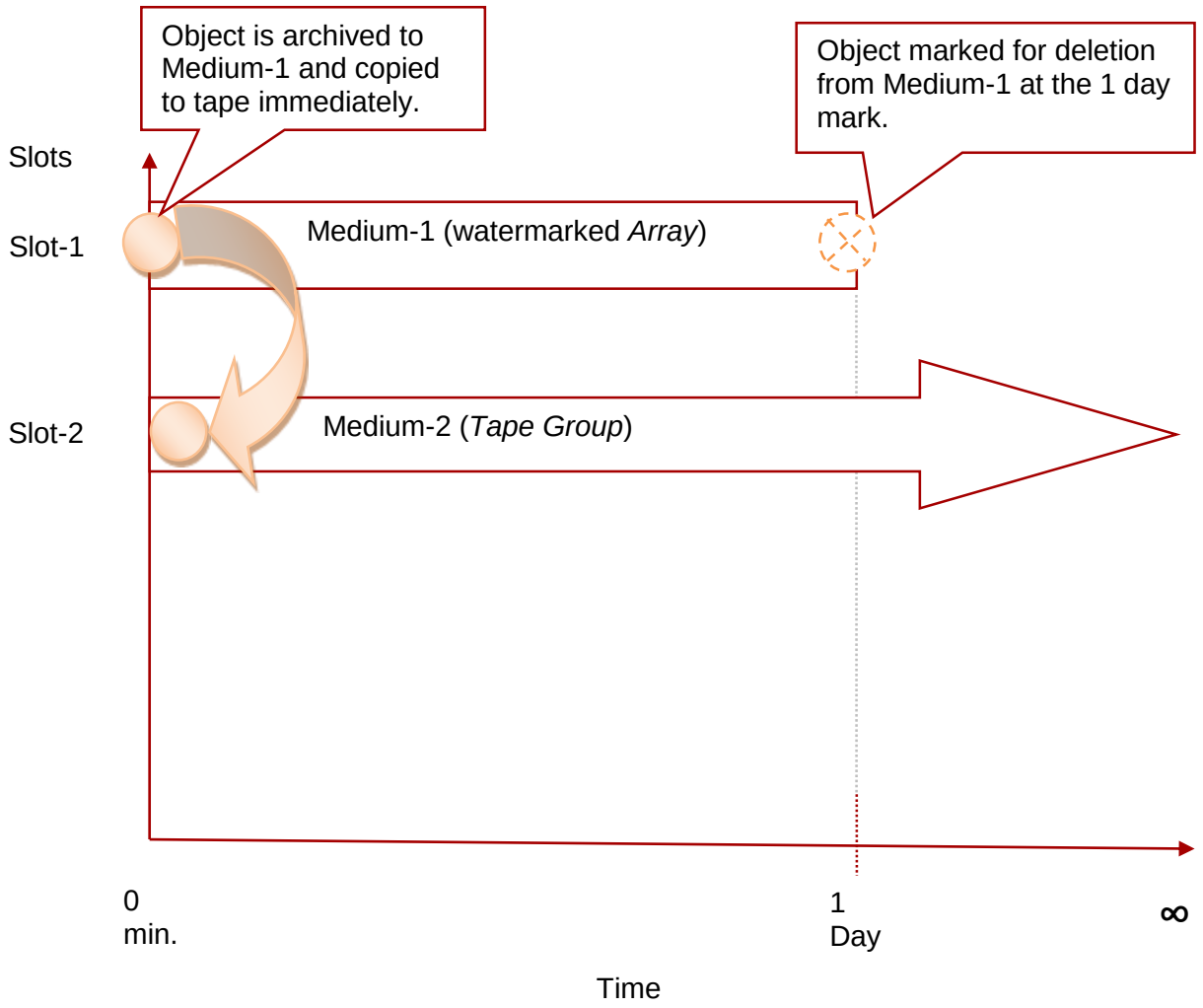
1.10.5 Simple SPM Configuration Example

The diagram below illustrates a simple SPM configuration example.

In this scenario:

1. The object is archived to a watermarked Array (*Medium-1 with watermark set to **Y**es*) and held there for 1 day.
2. The object is immediately copied to DIVArchive Medium-2 and stored there forever.
3. At the 1 day mark, the object is marked for deletion from the Array (*Medium-1*).

Figure 16: Simple SPM Configuration Example

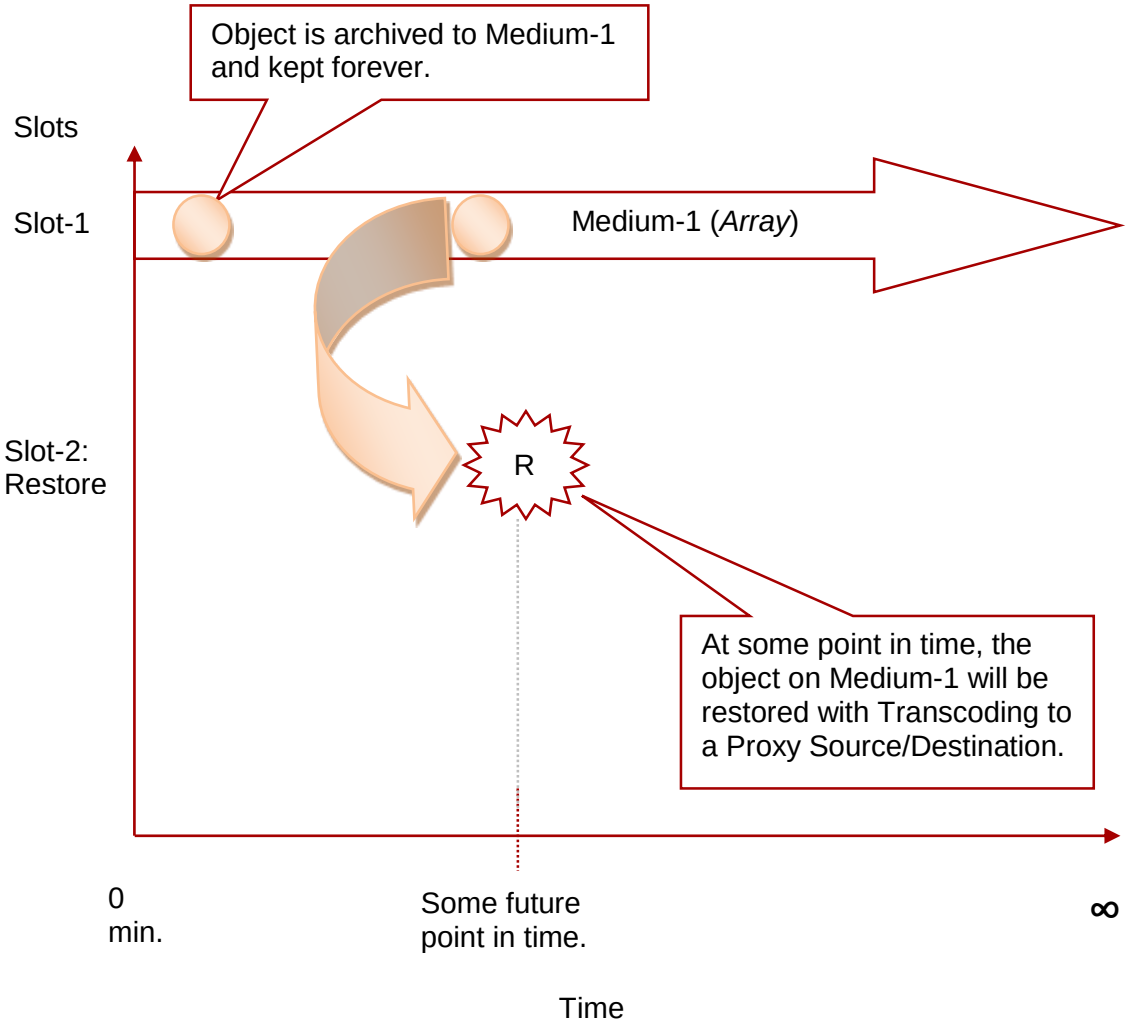


Note: Refer to Sections 1.7 and 2.5 for more information about Slot Types and Slot configuration.

1.10.6 Additional SPM Configuration Examples

The first diagram below 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 object will be restored with transcoding to a Proxy Source/Destination.

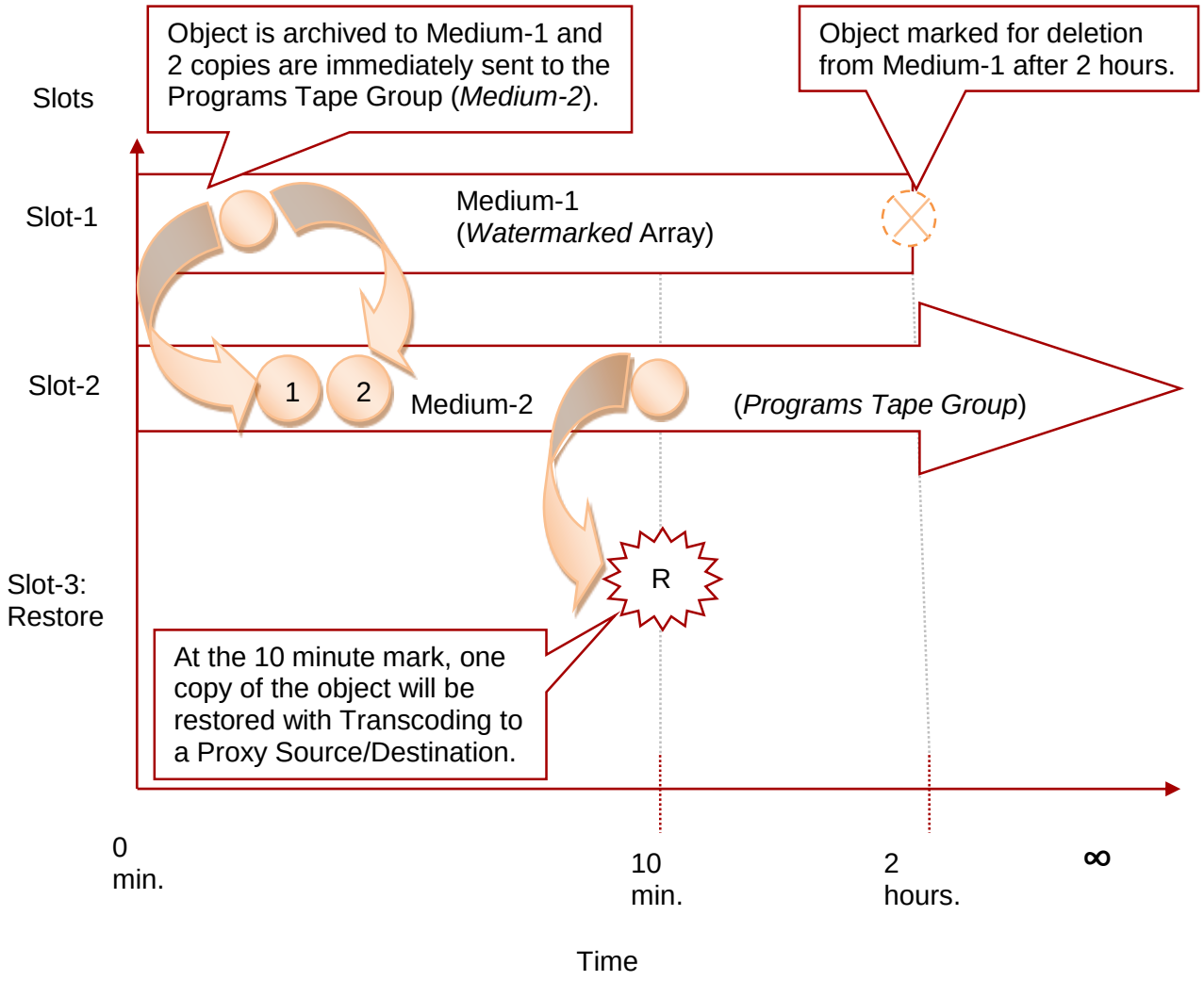
Figure 17: Additional SPM Configuration Example – 1



Note: Refer to Sections 1.7 and 2.5 for more information about Slot Types and Slot configuration.

The second diagram below 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 **Y**es*) for 2 hours and two copies are immediately made to a Tape Group named *Programs*. At the 10 minute mark, one copy will be restored to a Proxy Source/Destination.

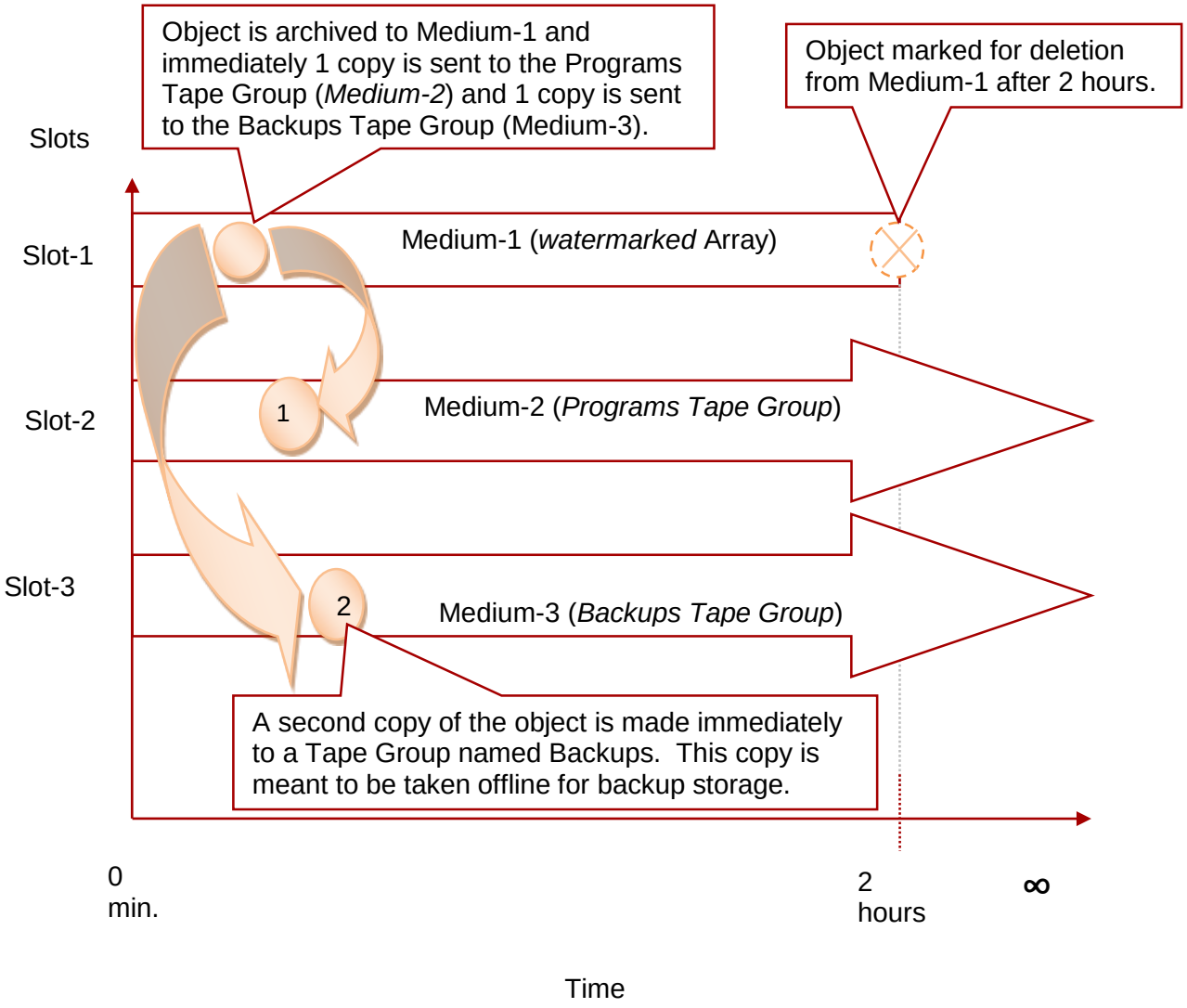
Figure 18: Additional SPM Configuration Example – 2



Note: Refer to Sections 1.7 and 2.5 for more information about Slot Types and Slot configuration.

The third diagram below is an example of a SPM configuration with one copy left in the online Tape Group (*Medium-2: Programs*) and a second backup copy made to a different Tape Group (*Medium-3: Backups*) that is to be taken offline. In this example, one copy is kept on the SPM watermarked Array (*with watermark set to **Y**es*) for 2 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 DIVArchive System and sent to an Iron Mountain type of facility for offline storage.

Figure 19: Additional SPM Configuration Example – 3



Note: Refer to Sections 1.7 and 2.5 for more information about Slot Types and Slot configuration.

2 Installation and Configuration

This section describes SPM installation and configuration. 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 DIVArchive Documentation library for full installation and configuration instructions for specific additional elements where necessary.

2.1 Installation Prerequisites

There are several prerequisites required before installing SPM:

- DIVArchive must be installed and running.
  - DIVArchive SPM Service must be installed and running.
  - Starting with DIVArchive version 7.1.2 the only supported Oracle version is 11.2 or greater and DIVArchive Oracle Package version 2.3.1 or greater.
  - The Database User Name and Password must be the same as the ones used for the DIVArchive installation. Typical values are:
    - **db_user**: **diva**
    - **db_password**: **lib5**

2.2 SPM Configuration File Parameters

SPM uses a plain text file to store its configuration parameters. The configuration file, **spm.conf.ini**, is supplied with SPM and is located in the **DIVA_HOME\Program\SPM\bin** folder. Rename the file to **SPM.conf** and change the parameter values to meet your requirements using a plain text editor (*for example Notepad*). An example of the SPM full configuration file is available in the A1 Default Configuration Appendix. Although all parameters are important, some are more important than others and all parameters are described in the following sections.

The following tables include the parameters that are used within the SPM Configuration File. Some parameters can also be set using the DIVArchive Configuration Utility. However there are two parameters whose settings in the SPM Configuration File will always override any settings created through the Configuration Utility for the parameters **METADATA_ARCHIVE_TRANSFORMED_OBJECT_NAME** and **ALLOW_OBJECT_DELETION**.

Table 6: DIVArchive SPM Service Name Parameters

Parameter	Definition
SERVICE_NAME	The DIVArchive SPM Service Name. This variable can be used to specify the name of the windows service. This is useful when multiple instances of SPM are running on the same server by giving different names for each SPM. • If this variable is used, the service name will be **DIVArchive Spm - <SERVICE_NAME>.** • If this variable is unset, the service name will be **DIVArchive Spm**.

Table 7: Parameters for DIVArchive Manager

Parameter	Definition
DIVA_MANAGER_ADDRESS	The IP Address or DNS Name of the DIVArchive Manager host. The default is **localhost**.
DIVA_MANAGER_PORT	The DIVArchive Manager Port. The default is 9000.

Table 8: Connection Parameters for the Oracle Database

Parameter	Definition
ORACLE_USER	The DIVArchive Database username. There is no default for this parameter but the typical username is **diva**.
ORACLE_PASSWORD	The DIVArchive Database user password. There is no default for this parameter but the typical password is **lib5**.
ORACLE_CONNECTION	The DIVArchive Database connection string. This is usually the SQL*Net address name declared in the local Oracle client **TNSNAMES.ORA** file for the DIVArchive Oracle instance. The typical connect string is **lib5**.
ORACLE_POOL_SIZE	Maximum number of database connections allowable by SPM. The default is 5. **Note: Do not change this value. Reducing the value can lead to database connection issues and cause SPM to freeze.**

Table 9: SPM Manager Monitor Parameters

Parameter	Definition
DIVA_MANAGER_MONITOR_MAX_REQUESTS	SPM will check with the DIVArchive Manager to find out how many requests are currently processing before it sends any requests. This parameter limits the number of requests (*processing + pending from all sources*) the SPM service will attempt to send to the DIVArchive Manager. If the number of requests already processing on the Manager 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 is 250.
DIVA_DELETE_MAX_REQUESTS	SPM will check with the DIVArchive Manager 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 check to see how many Delete 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** When SPM checks with Manager, if Manager is currently processing 75 requests total and only 25 of them are Delete Requests, SPM will send another 25 Delete Requests to Manager because both numbers are less that the parameter's value in the configuration file. The valid range is 1 to 300 and the default is 300. The value used here identifies how many of this type of request are simultaneously processed from the Action Queue at one time. To disable (*no limitations*) use 0 for the value. **This value cannot exceed the value of DIVA_MANAGER_MONITOR_MAX_REQUESTS.**

Parameter	Definition
DIVA_RESTORE_MAX_REQUESTS **DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS** **DIVA_METADATA_ARCHIVE_MAX_REQUESTS**	Restricts the number of Restore, Transcode Archived and Metadata Archive Requests SPM will submit to the Manager 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 used here identifies how many of this type of request are allowed on the Action Queue. Using the default value of -1 will disable 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** **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** SPM will fail to start using these configuration settings because the first One is enabled with a value of 3 while the next 2 are disabled. They must all be either enabled or disabled, not a combination of both.
DIVA_MANAGER_MONITOR_DELAY	The number of seconds between checks of the number of requests (*processing* + *pending*) on the DIVArchive Manager. The valid range is 1 to 600 and the default value is 30.

Parameter	Definition
DIVA_MANAGER_MONITOR_ACTION_DELAY	The number of seconds to delay an Action for execution if DIVArchive Manager 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 is 60.

Table 10: SPM Controller Parameters

Parameter	Definition
ACTION_STEP_WATCH_DELAY	The minimum delay in seconds before the next check on the request status. The valid range is 1 to 60 and the default is 10.
ACTION_QUEUE_SIZE	The maximum size of the Action Queue. • The number of actions that can be stored in the queue. • The valid range is 1 to 300.

Table 11: Rest Time Parameters for SPM Working Tasks

Parameter	Definition
SPM_EXECUTION_THREAD_REST	The resting time for the Execution Task in seconds. • The valid range is 0 to 600 and the default is 5. • Entering 0 will disable the task completely.
SPM_UPDATE_THREAD_REST	The resting time for the Update Task in seconds. • The valid range is 0 to 600 and the default is 20. • Entering 0 will disable the task completely.
SPM_LOAD_THREAD_REST	The resting time for the Load Task in seconds. • The valid range is 0 to 600 and the default is 20. • Entering 0 will disable the task completely. **Note: The Recovery Load Task will never rest; it stops upon completion of all work.**

Table 12: Sizes of Chunks of Rows SPM will Process at One Time

Parameter	Definition
RECOVERY_ACTIONS_CHUNK	The number of actions SPM will process (*from the database*) at one time during each call of the Recovery Task. The valid range is 1 to 65535 and the default is 500.
UPDATE_ACTIONS_CHUNK	The number of actions SPM will process at one time during each call of the Update Task. The valid range is 1 to 500 and the default is 50.
GET_ACTIONS_CHUNK	The number of actions SPM will process at one time during each call of the Load Task. The valid range is 1 to 100 and the default is 25.
UPDATE_ACTIONS_NEXT_UPDATE	The time (*in minutes*) added to the Action's date of next update after it has been updated by the Update task. The date of next update was actually the time of next examination of the Action in older DIVArchive versions (*before 5.10*). Current versions of DIVArchive SPM is of a more dynamic nature and constantly re-examines Actions, and the date of next update is used internally to sort the Actions to be re-examined. The valid range is 10 to 65535 and the default is 20160 (*two weeks*).
UPDATE_ACTIONS_RETRY_FAILED_DELAY	The delay in minutes before a *long* retry of a failed action. After the value entered here 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 as per the latest slot changes before executing the action. The valid range is 10 to 65535 and the default is 720 minutes.

Table 13: SPM Processing Options

Parameter	Definition
ALLOW_OBJECT_DELETION	**CAUTION! CAUSES COMPLETE OBJECT DELETION WHEN ENABLED!** When a Storage Plan has no slot at a given time, the object can be deleted. This feature must be managed carefully, since many objects can be lost according to the storage plan. • The valid settings are **true** or **false** and the default is **false**. • The user should carefully define the Storage Plans. To enable object deletion, set this field to **true**. • The Storage Plan definition in the database can either allow or deny deletion. If deletion is denied here, configuring a Storage Plan to allow deletion will have no effect. Note: The setting in the SPM Configuration File for this parameter overrides any settings created through the DIVArchive Configuration Utility. 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, objects will be deleted after 30 days since there are no slots during the second month (*day 31 to 60*).
DELETE_OBJECT_ONLY_LAST_INSTANCE	This parameter works in conjunction with the **ALLOW_OBJECT_DELETION** parameter. This parameter forces SPM to Delete Object at the end of the Storage Slot only if it is the last instance in the entire DIVArchive 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 do a Delete Object at the end of the Storage Slot, it will always only do a Delete Instance. Refer to Section 2.5.5 for more detailed information.

Parameter	Definition

METADATA_ARCHIVE_TRANSFORMED_OBJECT_NAME	Defines whether or not SPM will use the original or transformed object name as the target object name in a **METADATA_ARCHIVE** Action.
	• This setting overrides any configuration performed through the Configuration Utility.
	• The default is **true**.
	  ◦ **true**: use transformed object name.
	  ◦ **false**: use original object name.
	
	Note: The setting in the SPM Configuration File for this parameter overrides any settings created through the DIVArchive Configuration Utility.

Table 14: DSM Task Options

Parameter	Definition
DSM_DIW_REQUEST_PRIORITY	The priority of the requests produced by **DELETE INSTANCE** actions. Valid range is 1 to 100 and the default is 15.

Table 15: Rest Time Parameters for DSM Working Tasks

Parameter	Definition
DSM_SPACE_MONITOR_THREAD_REST	The resting time for the DSM Task in seconds. • The valid range is 0 to 600 and the default is 10 Entering 0 will disable the Task completely.

Table 16: Sizes of Chunks of Rows DSM will Process at One Time

Parameter	Definition
DSM_ACTIVATE_ACTIONS_CHUNK	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 is 50.

Table 17: Disk Arrays to be Monitored

Parameter	Definition
ARRAY[number]	Specifies watermark values for 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>** 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 path"] ...]** Where: • **ARRAY_NUMBER** is the name known to SPM for this array (*i.e.* ***ARRAY1***). • **"array name"** is the name of the Disk Array, as it is named in DIVArchive (*It is not the SPM "medium" name!*). • **"watermark low-level percent"** is a number from 0 to 100 (*represents the lower Array space setting usage percent*). DSM will try to reach this percent when cleaning. • **"watermark high-level percent"** is a number from 0 to 100 (*represents the higher Array space usage percent*). DSM will start the cleaning process when Array space usage is higher than this level. • **["mounted disk path"]…** is the mounting points of the Array disks. None, one, or many, separated by a comma. ◦ No spaces are allowed. ◦ If **one** or **many**, DSM will access arrays via the filesystem (*old*) interface. ◦ If **none**, DSM will access arrays via the DIVArchive API (*new*) interface (*recommended*).

Parameter	Definition
ARRAY[number] (*continued*)	• 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 CIFS Array, 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. 1. Click **Start** then **Run**. 2. Type **services.msc** in the **Run** text box and press **Enter**. 3. This opens the Windows Service Control Manager. 4. Select the SPM Service requiring the login change. 5. Right-click on the SPM Service and select **Properties**. 6. Select the **Log On** tab in the **Properties** window of the SPM Service. 7. Select the radio button next to **“This account”**. 8. Enter a valid windows user and password. 9. Click the **Apply** or **OK** button to commit the changes. **Examples:** 1. ARRAY1 = ShortClips;75;90 (*recommended syntax*) 2. ARRAY2 = Raid_001;60;85;X:\;Y:\;Z:\ (*legacy syntax*)

Table 18: Additional Parameters

Parameter	Definition
TRACE_LEVEL	The level of tracing to use for the logs. The valid range is 1 to 2 and the default is 2. • "1" will trace "entry" and "exit" points in all important functions. WARNING! This setting will generate large volumes of trace information and must be used for 'debug' and 'validation' purposes only! • "2" will produce a normal, *production* level of trace.

2.3 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 the settings to be ignored by SPM and the defined behavior will not be changed.

All, or none, of these parameters should be set 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 1 or 2 parameters are defined with other values, SPM will refuse to start because of the incorrect configuration.

Note: Whenever changing these parameter values, SPM must be restarted for the new values to take effect.

There are several rules to keep in mind for Request Type Distribution:

- SPM must not execute more Restore requests than **DIVA_RESTORE_MAX_REQUESTS** at the same time. 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** at the same time. 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 **DIVA_METADATA_ARCHIVE_MAX_REQUESTS** at the same time. If the **DIVA_METADATA_ARCHIVE_MAX_REQUESTS=0**, SPM will not execute any Metadata Archive Requests.
- SPM will do its best 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% 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 3 Restore Requests at the same time (*most of the time*).
- SPM will never execute more than 3 restore requests.
- Sometimes (*less than 50% 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.
  - 1000 copy actions are to be executed.
- There are 2 Restore slots with priorities 30 and 35.
  - 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 Object and Delete Instance requests will always have higher priority.

2.4 Storage Plan Definition

SPM requires configuration of Storage Plans before execution. Configure the SPM Storage Plans using the DIVArchive Configuration Utility.

There are two tabs in the DIVArchive Configuration Utility related to SPM configuration: **Storage Plans** and **Slots**.

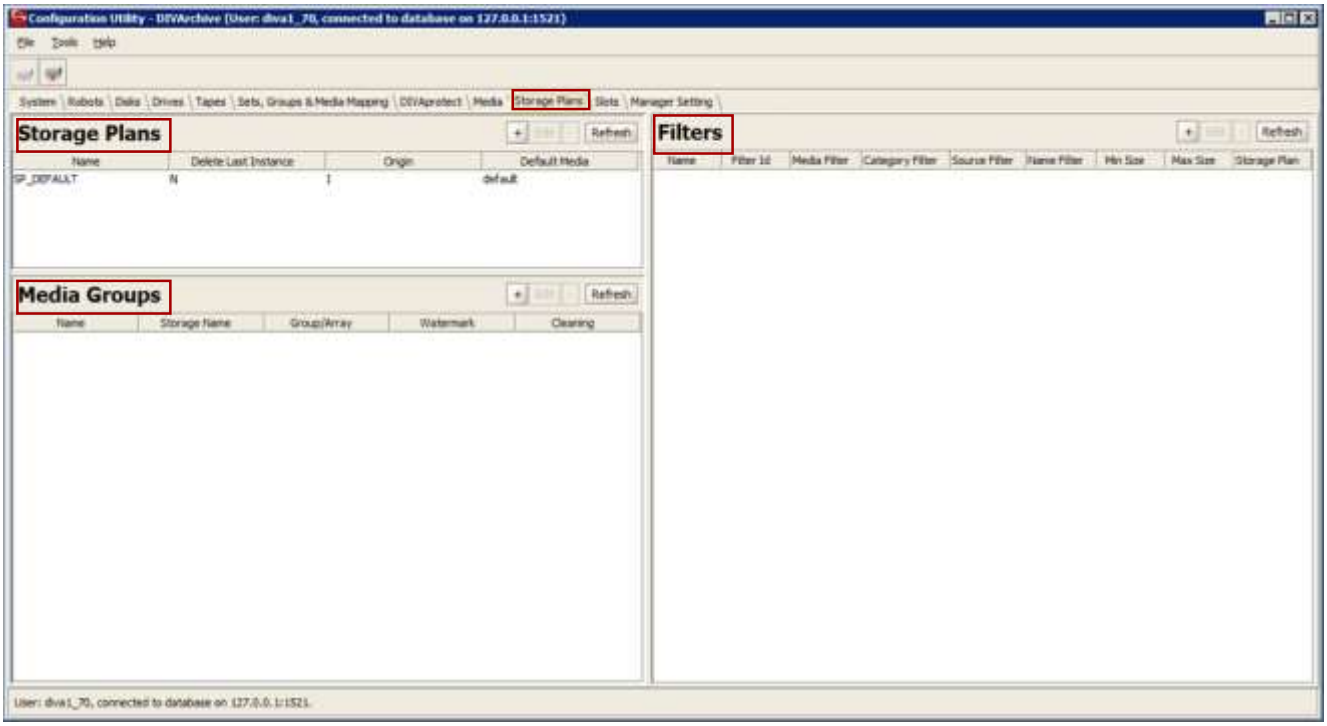
2.4.1 The DIVArchive Configuration Utility Storage Plans Tab

The **Storage Plans Tab** in the DIVArchive Configuration Utility has three sections:

- Storage Plans
- Media Groups
- Filters

Each of these sections is configured independently but associates with other sections to complete the configuration. The figure below shows the different sections.

Figure 20: DIVArchive Configuration Utility Storage Plans Tab



2.4.2 Recommended Practice

It is recommended to use the following order when configuring SPM through the DIVArchive Configuration Utility:

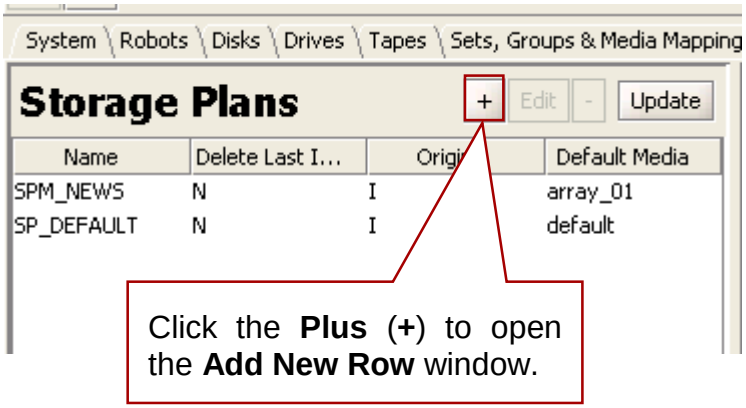
1. Create the Storage Plans.
2. Create the Media Groups.
3. Create the Filters.
4. Create the Slots (*in the **Slots Tab***).

2.4.3 Creating the new Storage Plan

The first step in configuring SPM is to create the Storage Plans. The steps below outline this procedure.

1. Click the **Plus** (+) in the **Storage Plans** section of the Configuration Utility.

Figure 21: Creating a New Storage Plan



2. When the **Add New Row** window opens begin configuring the new Storage Plan. Type in the new Storage Plan Name (***SPM_SPORTS** for this example*) and then use the pull-down boxes to make the remaining selections.

Figure 22: Add New Row Window for Storage Plans

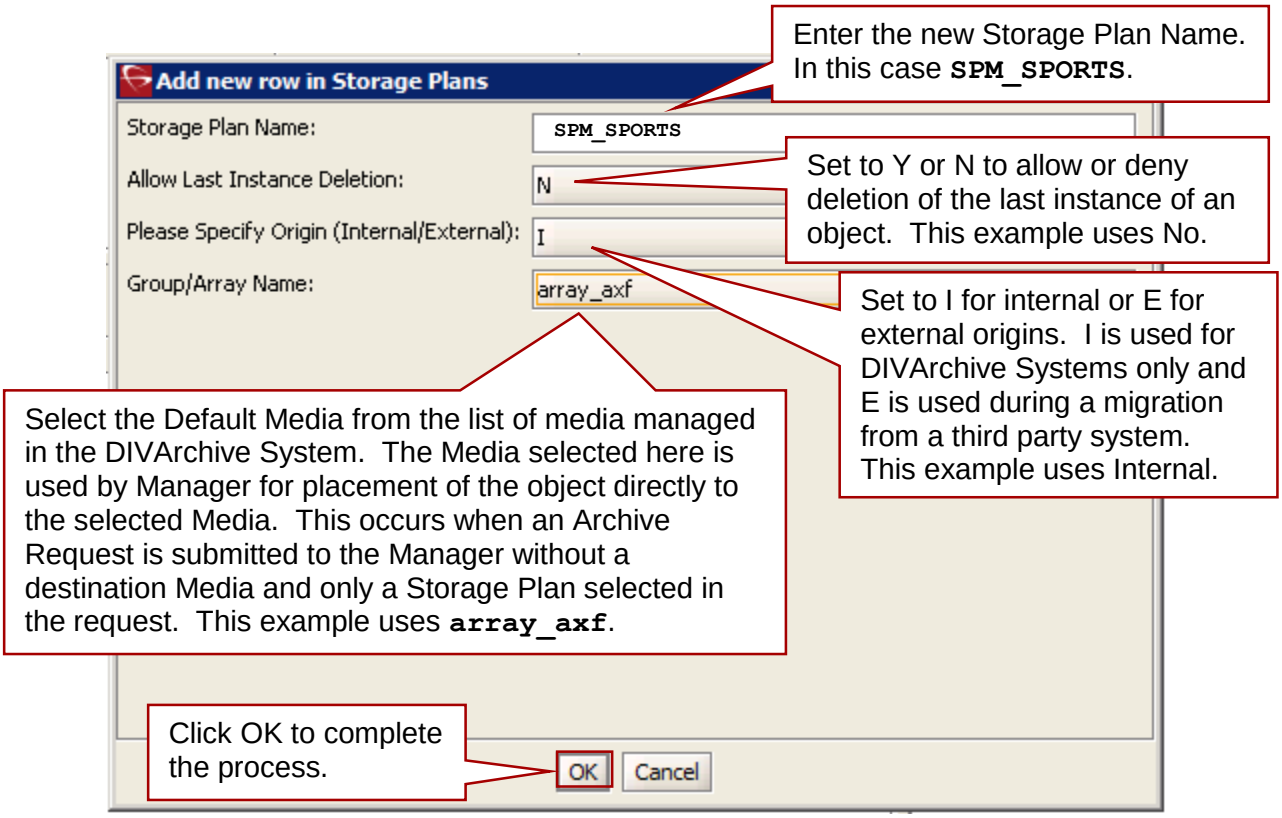
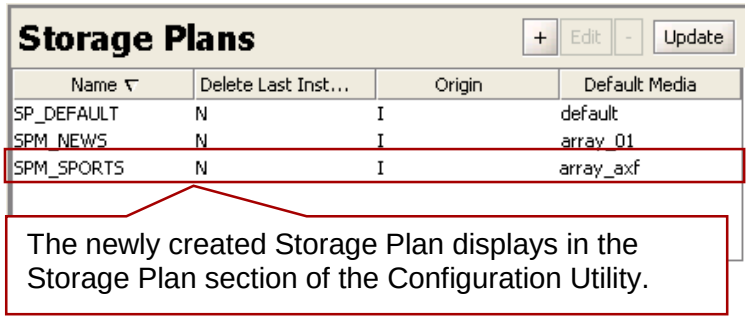


Table 19: DIVArchive Configuration GUI Storage Plan Parameters

Parameter	Definition
Storage Plan Name	The name of the new Storage Plan (***SPM_SPORTS*** *in the example*).
Allow Last Instance Deletion	Determines whether or not SPM is allowed to delete the last instance of an object when there is only 1 instance remaining on the DIVArchive System (***No*** *in the example*). • Only applies to a Storage Slot (*refer to Section 2.5*). • This parameter is overridden by the configuration of **ALLOW_OBJECT_DELETION** in the SPM Configuration File. USE THIS PARAMETER WITH CAUTION.
Please specify origin (Internal/External)	• Typically used for all SPM actions • Internal (*I*) is for objects contained within the DIVArchive System and should commonly be selected. • External (*E*) is used during data migration in combination with the SPM API and should only be used by Oracle in this version of SPM. ○ For example for use with Oracle DIVArchive Automatic Data Migration from Avalon to DIVArchive. • The example uses Internal.
Default Media	The Media selected here is used by the DIVArchive Manager for object placement when a new object is found and the Storage Plan in use has an active Storage Slot. This occurs when an Archive Request is submitted to the Manager without a destination Media and only a Storage Plan selected in the request. The example uses **array_axf**.

3. Click **OK** to complete the process. The new Storage Plan will be displayed in the **Storage Plans** section of the Configuration Utility as shown below.

Figure 23: New Storage Plan Showing in the Configuration Utility Storage Plan Section



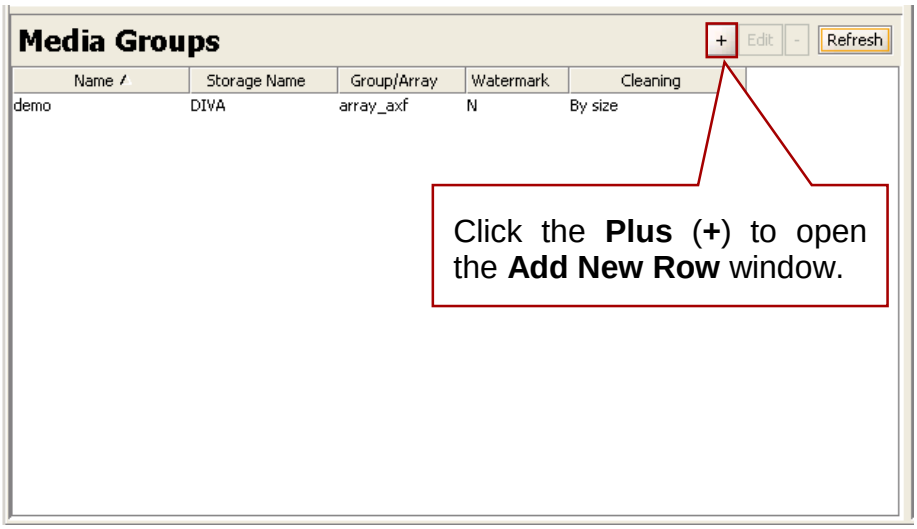
2.4.4 Creating Mediums

The Media Groups Section has two purposes:

- Any medium that is defined in the **Media Groups Section** of the **Configuration Utility Storage Plans Panel** is available for use by SPM. If a new 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.
- In order to select a medium as the target for actions when creating Slots, it must be defined in the **Media Groups** list.

1. Click the **Plus** (+) in the **Media Groups Section** of the Configuration Utility.

Figure 24: Creating a New Media Group



2. When the **Add New Row** window opens begin configuring the new Mediums. Type in the new Medium Name (*medium_a1 in the example*) and Storage Name (***DIVA*** *in the example*), and then use the pull-down boxes to make the remaining selections.

Figure 25: Add New Row Window for Media Groups

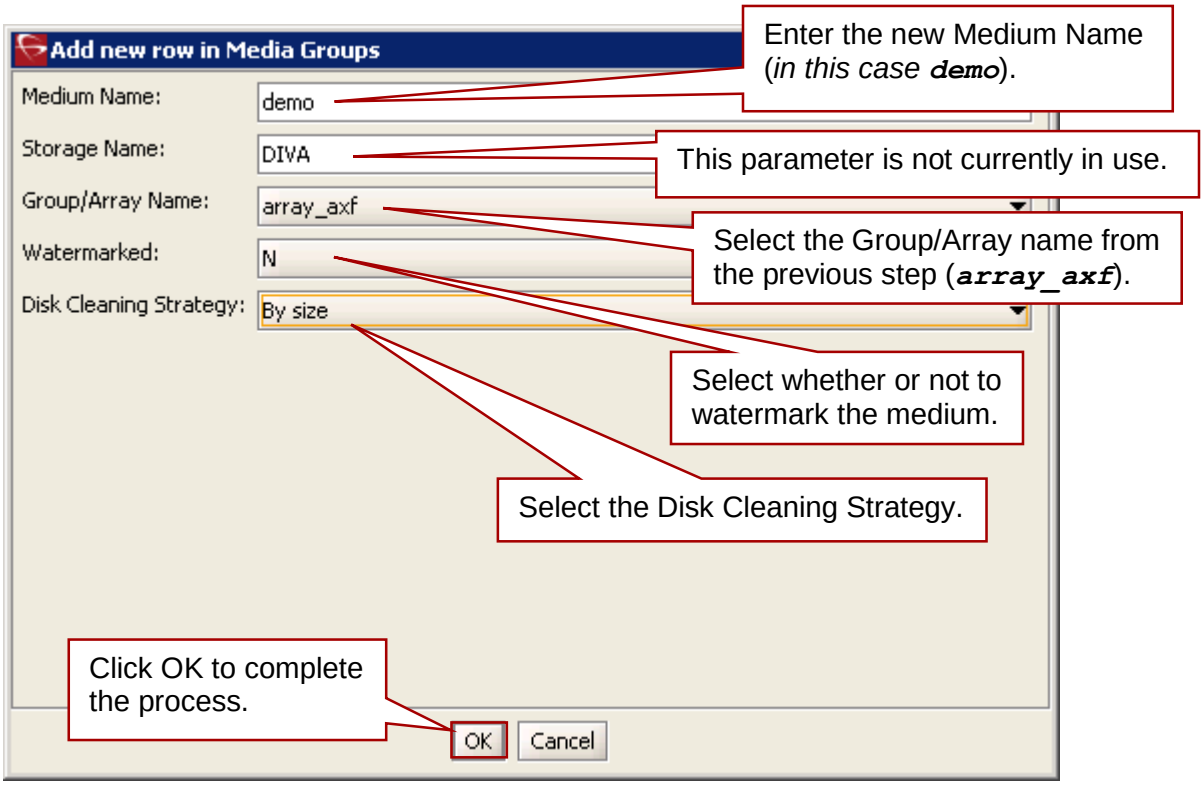


Table 20: DIVArchive Configuration GUI Mediums Parameters

Parameter	Definition
Medium Name	Enter the desired Medium Name (*in this case* ***demo***).
Storage Name	Enter the desired Storage Name. **DIVA** is the recommended Storage Name and is used in the example. This value is not currently in use.
Group/Array Name	Select the Tape Group or Disk Array to be associated with this Medium (*from the previous step –* ***array_axf***). The selections available in the pull-down box are determined by what media is defined in the DIVArchive System.

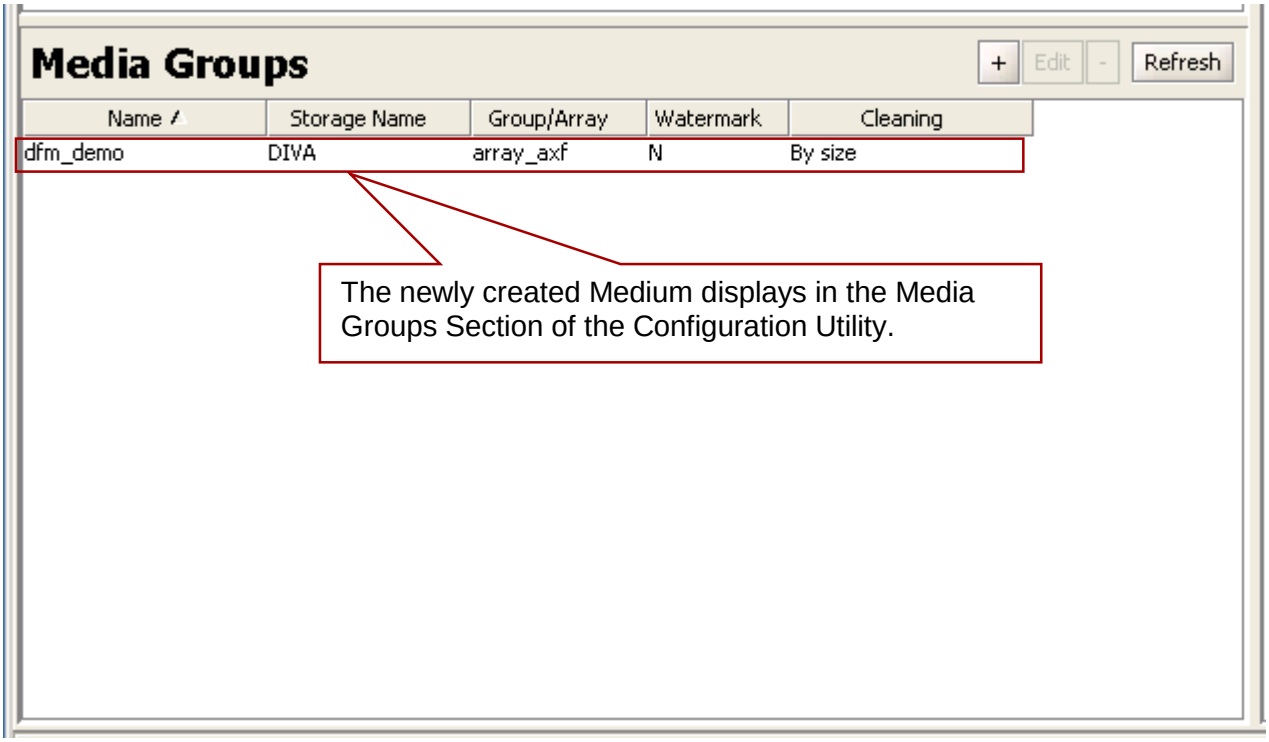
Parameter	Definition

Watermarked	Monitors the Medium's usage (*available space, used space, etc.*). Select **Y**, **N**, or **M** to specify whether or not to apply watermarking to this medium. Refer to Sections 1.7.1 and 2.6 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 DIVArchive 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.
	
	Watermark-Based Disk Cleaning Management
	• **Y** applies watermarking (*Yes*).
	• **N** does not apply watermarking (*No*).
	• **M** is a combination of watermarking (*Mixed*). The disk cleaning action taken is dependent upon what event happens first:
	  ◌ The Slot reaches its end time.
	  ◌ 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**.
	• Only for Disk Arrays configured as Medium.

Disk Cleaning Strategy	Defines the cleaning strategy to perform on objects marked for deletion on the Array when the high watermark is reached. Refer to Section 2.6 for detailed information on Disk Cleaning Management.
	• Two options
	  ○ By Last Access Time
	  iii. Older objects are cleaned up before recently accessed objects.
	  ◌ By Object Size
	  iv. The largest objects are cleaned up first until the low watermark is reached.

3. Click **OK** to complete the process. The new Media Group will be displayed in the **Media Groups Section** of the Configuration Utility as shown below.

Figure 26: New Medium Showing in the Configuration Utility Mediums Section



2.4.5 Creating Filters

Filters determine whether an action needs to be performed on a new object or if it should be ignored. Filters only functions for mediums defined in the Mediums list and monitored by SPM.

When a new object is archived into the DIVArchive System, and it matches a defined filter, a Storage Plan will be assigned to that object:

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

1. Click the **Plus** (+) in the **Filters** section of the Configuration Utility.

Figure 27: Creating a New Filter

2. When the **Add New Row** window opens begin configuring the new Filter. Enter the new Filter Name, and then the remaining information as required. Leaving the default * in **Object Category**, **Object Source**, and **Object Name** will cause no filtering to be identified for these parameters.
3.

Figure 28: Add New Row Window for Filters

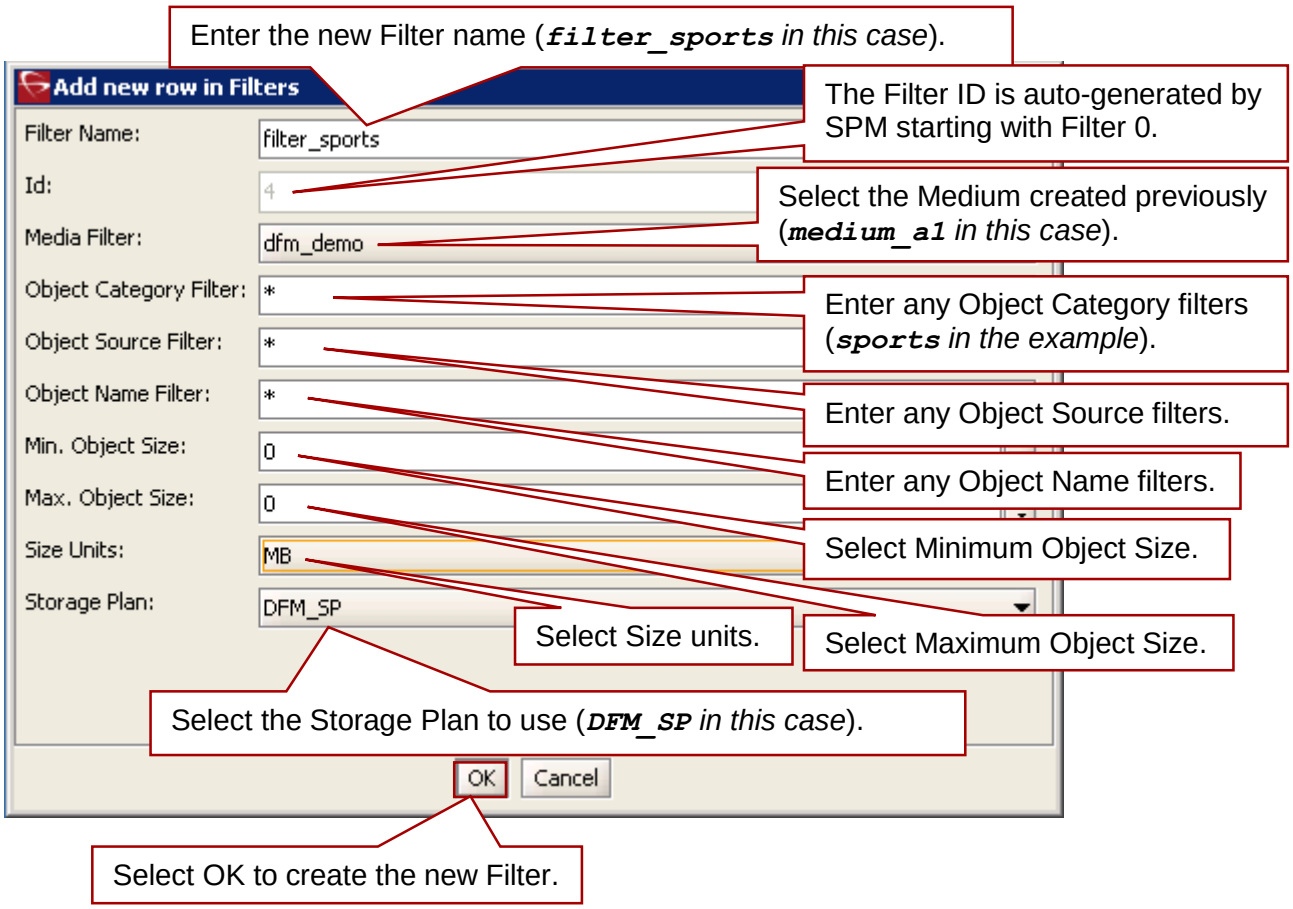


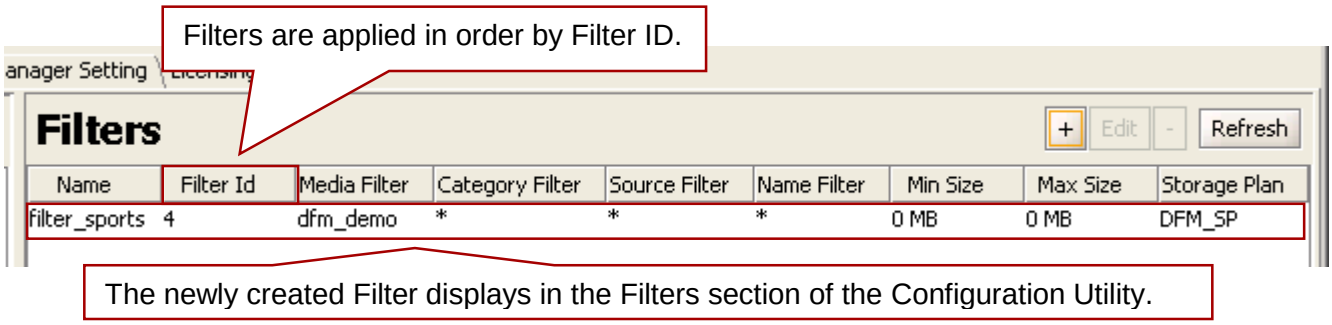
Table 21: DIVArchive Configuration GUI Filters Parameters

Parameter	Definition
Filter Name	Enter the desired Filter Name (*filter_sports in the example*).
Id	The Filter ID is auto-generated by SPM starting with Filter 0.
Media Filter	Use the pull-down box to select the Mediums created previously (*dfm_demo in this case*).
Object category filter	Enter any object category filters desired. Leaving the default ***** will cause no filtering for this parameter.
Object source filter	Enter any object source filters desired. Leaving the default ***** will cause no filtering for this parameter.

Parameter	Definition
Object name filter	Enter any object name filters desired. Leaving the default ***** will cause no filtering for this parameter.
Min. object size	Enter the minimum object size.
Max. object size	Enter the maximum object size.
Size units	Use the pull-down box to select the size units (*B, KB, MB, and GB - where the initials refer to **B**yte, **K**ilo**B**yte, **M**ega**B**yte, and **G**iga**B**yte respectively*).
Storage Plan	Use the pull-down box to select the Storage Plan to associate with this filter (***DFM_SP** in the example*).

3. Click **OK** to complete the process (*shown in the previous figure*). The new Filter will be displayed in the **Filters** section of the Configuration Utility as shown below.

Figure 29: New Filter Showing in the Configuration Utility Filters Section

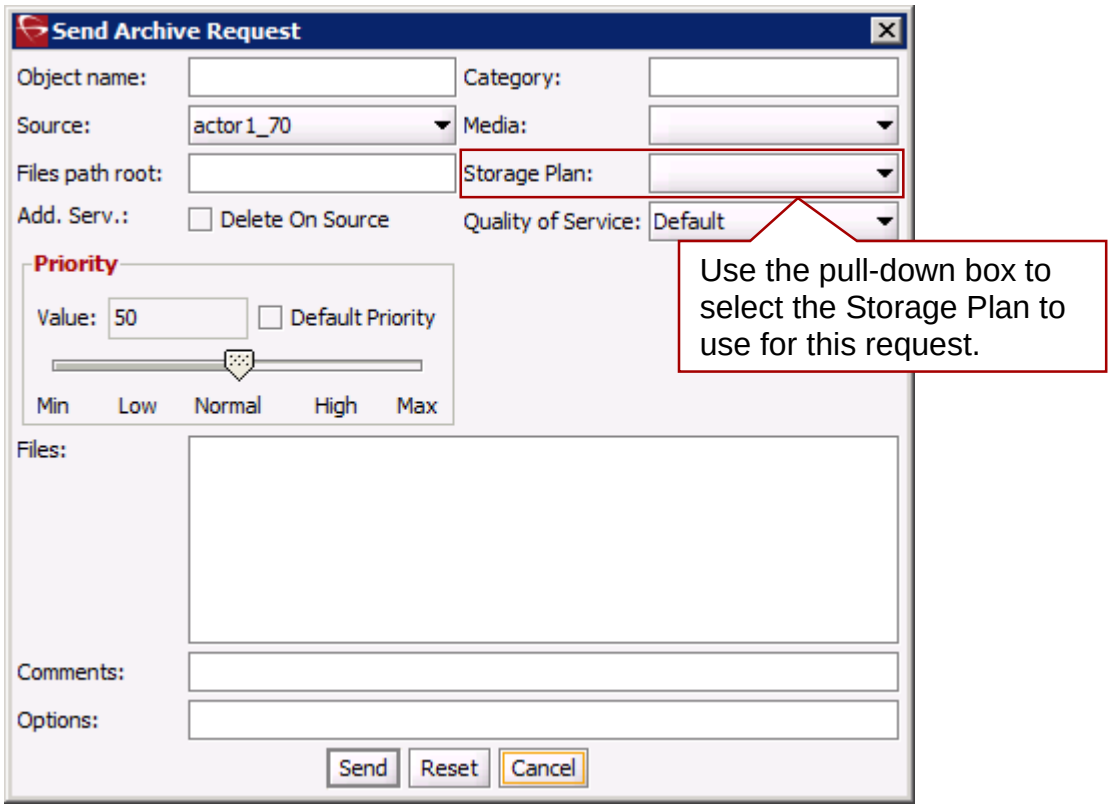


2.4.6 Alternate Methods of Assigning Storage Plans to an Object

There are several alternate methods of assigning a Storage Plan to an object including the procedures below:

- Assigning a New Storage Plan for Assigning a New Storage Plan to an Object via the DIVArchive GUI.
- DIVArchive API.
  - C++, Java, Web Services (*WS*)
  - Refer to the appropriate API documentation in the DIVArchive API Documentation library for more information.
- Mentioning a Storage Plan Name while submitting an archive request will forcefully apply a Storage Plan to the object avoiding filters.

Figure 30: Sending an Archive Request to Mapped Media

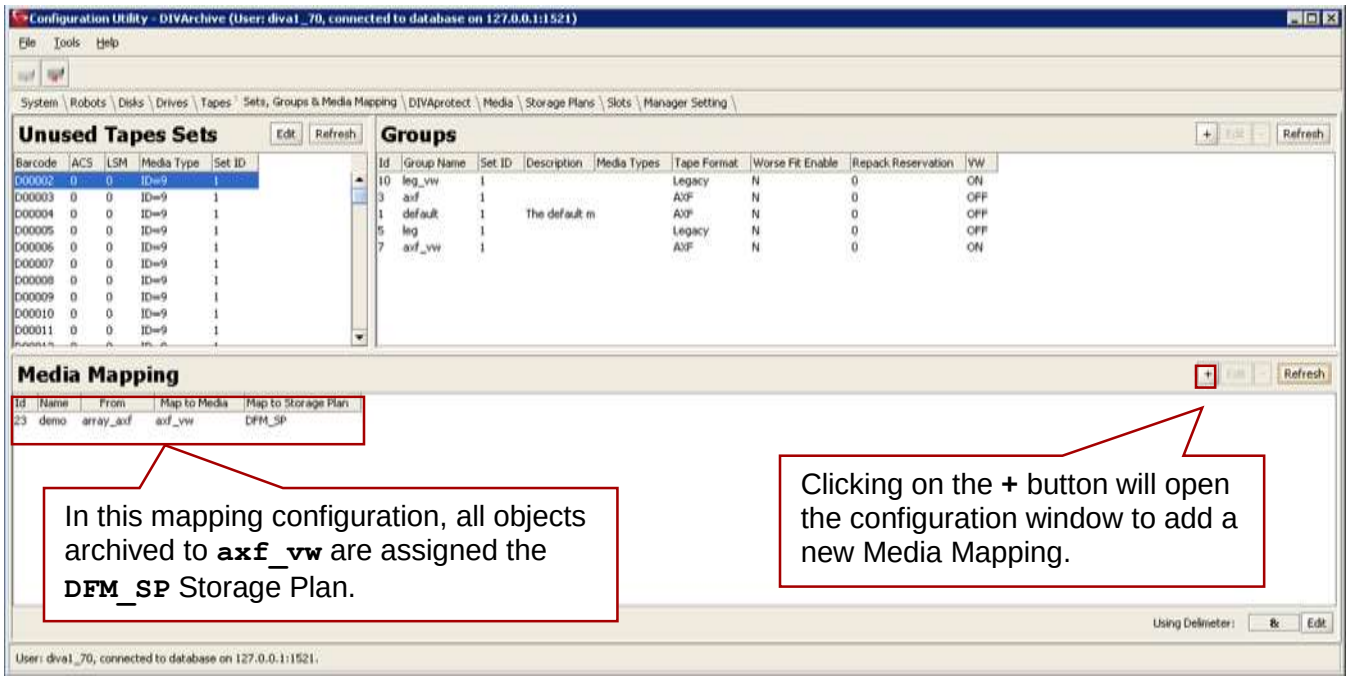


In the figure above, an Archive Request is sent to DIVArchive from the GUI. Mentioning a Storage Plan Name while submitting an Archive Request from the GUI will forcefully apply a Storage plan to the object avoiding filters.

- Media Mapping allows configuring alternate destination media and forcefully applying a Storage Plan, avoiding filters, when an object is archived to a specific Media (*Disk Array or Tape Group*). Media Mapping is configured using the **DIVArchive Configuration Utility** in the **Sets, Groups & Media Mapping Tab**.

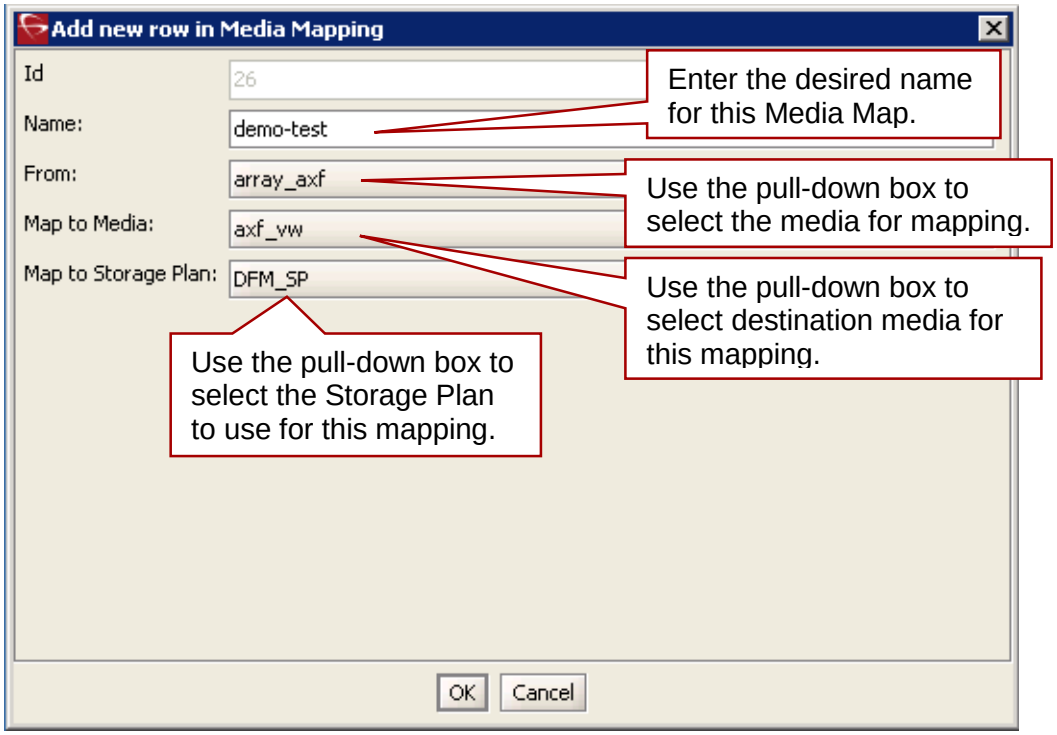
  In the first figure below, the Media Mapping is configured so that all objects archived to the media named **Array1** will be assigned the Storage Plan named **storagePlan1**.

Figure 31: Configuration Utility Showing Media Mapping



Clicking on the **+** button will open the window shown in the next figure to allow adding a new Media Mapping.

Figure 32: Adding a New Media Map Configuration



In the figure above, use the pull-down boxes to select the appropriate values:

- **Name:** Enter the desired name for this mapping.
- **From:** Select the media for mapping.
- **Map to Media:** Select the destination media for this mapping.
- **Map to Storage Plan:** Select the Storage Plan to use in this mapping.

Examples:

1. Media **Array1** is mapped to **Array2**.
    - In this scenario, objects archived to the media named **Array1** will be transferred to the media named **Array2** as its destination.
    - **Note: If there is a filter configured with destination media as Array2, the Storage Plan configured in that filter will be assigned to the archived object.**
2. Media **Array1** is mapped to the Storage Plan named **storagePlan1**.
    - In this scenario, objects archived to the media named **Array1** will be assigned the Storage Plan **storagePlan1,** avoiding filters.
3. Media **Array1** is mapped to **Array2**, and the Storage Plan named **storagePlan1.**
    - In this scenario, objects archived to the media named **Array1** will be transferred to the media named **Array2** as its destination, and assigned Storage Plan **storagePlan1**, avoiding filters.

2.5 Creating Slots

The following subsections describe the process for creating each different type of Slot.

2.5.1 Storage Slot Configuration

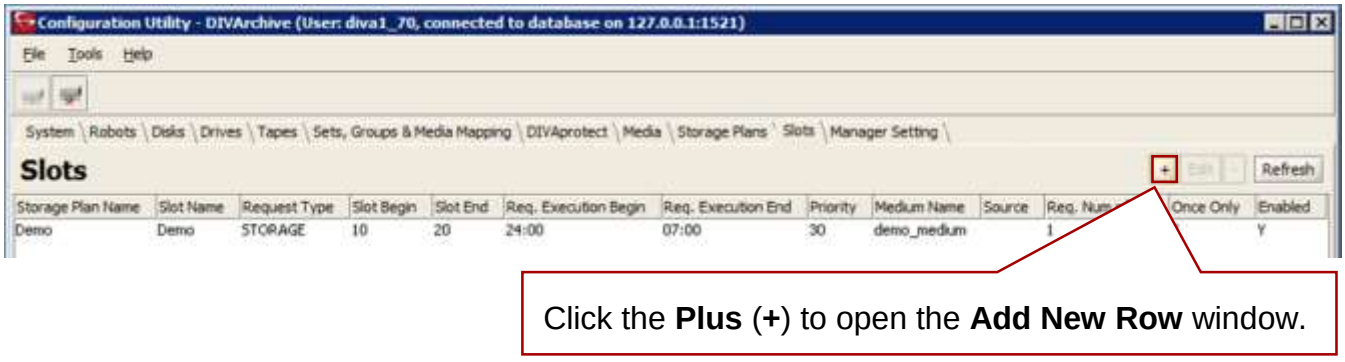
The Storage Slot is the only slot that has two Actions associated with it. It will create a copy of the object when it starts and delete it when the slot ends (*unless the medium used is watermarked – see Sections 1.8 and 2.6 for watermarking details*). The copy action will occur only if the slot destination 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*).

A **Storage Slot** is configured on the **Slots Tab** of the Configuration Utility.

To configure a Storage Slot:

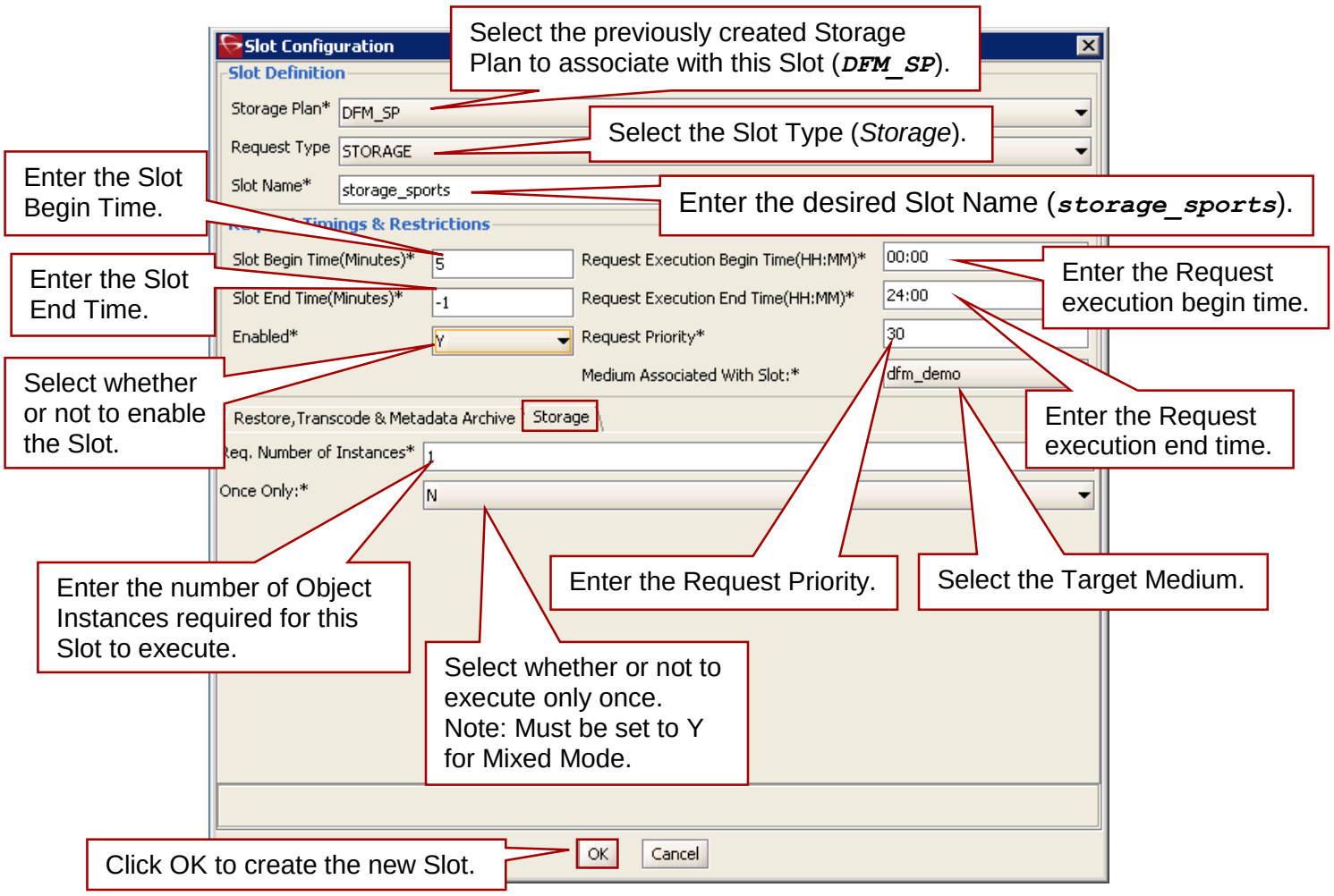
1. Click the **Plus** (+) in the **Slots** section of the Configuration Utility as shown in the figure below.

Figure 33: Click the Plus Sign to Begin Creating a New Slot



2. When the **Slot Configuration** window opens begin configuring the new Slot.

Figure 34: Slot Configuration Window for Storage Slots



There are a second set of tabs at the bottom of the Slot Configuration Window that contains the following tabs:

- Restore, Transcode & Metadata Archive
- Storage

The **Storage Slot <u>only</u>** uses the **Storage Tab**.

Table 22: DIVArchive Configuration Utility Parameters for Storage Slots

Parameter	Definition
Slot Name	Enter the desired Slot Name (***storage_sports*** *in the example*).

Parameter	Definition
Storage Plan	Select the Storage Plan to be associated with this slot (*previously created* ***DFM_SP***).
Medium associated with slot	Select the target medium using the pull-down box.
Slot request type	Select the type of request for this slot (*Storage, Transcode Archived, Metadata Archive, or Restore*). In this case select **STORAGE**.
Slot begin time (*min*)	This parameter identifies when the Action will occur. If set to zero (0) the Action will be initiated as soon as the object is archived. If set to another number, for example 10 (*minutes*), the Action will execute 10 minutes after the object is archived.
Slot end time (*min*)	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 object is archived. **Warning: If the Slot Begin Time and Slot End Time are both set to zero (0) then the behavior is unknown, except in the case of Storage Slots.** If a Storage Slot's Slot begin time and Slot end time are the same (*e.g. 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 instances are created by an external tool (*not by the SPM*), but use of the SPM's watermark-based clean-up mechanism is desired.

Parameter	Definition
Slot end time (*min*) (*continued*)	**Example** (*typical slot*): **Slot begin time:** 10 **Slot end time:** 20 **Workflow:** 1. The original object is archived onto the source medium. 2. 10 minutes later, the slot actions begin and it is copied to the target medium. 3. 10 minutes after that the copy is deleted from the target medium. **Example** (*idle slot*): **Slot begin time:** 0 **Slot end time:** 0 **Workflow:** 1. Object is archived on medium Array1 and our idle slot is defined for medium Array2. a. An Array2 **Copy** Action created and the status is **Scheduled**. b. An Array2 **DeleteInstance** Action created and the status is **Completed**. 2. An external tool creates an instance on Array2 using the **Copy** function. a. The Array2 **Copy Action** status is **Completed**. b. The Array2 **DeleteInstance Action** status is **Postponed**. 3. Array2 hits High Watermark. a. The Array2 **DeleteInstance Action** executed and the status is Completed. The Array2 **Copy Action** status remains as **Completed**.

Parameter	Definition
Request execution begin time (HH:MM)	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)	Slot execution window closing time in 24 hour time. If set to 24:00 the Slot closes at the end of 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 jobs*). If an object is archived to the source medium outside of the designated time frame for the Slot, even though the object meets the Filter and Storage Plan criteria, no actions from this particular Slot will occur.

If the original object was archived to the source medium while the Slot was inactive (*closed*) and has passed the **Slot End Time,** once 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: 10

Slot end time: 20

Request execution begin time (HH:MM): 08:00

Request execution end time (HH:MM): 14:00

Workflow:

1. A new object is placed onto the source medium at 07:00 (*while the Slot is closed*).
2. If the Slot had been open at 07:00 when the new object was created it would be copied to the target at 07:10 and then deleted from the target at 07:20.
3. Since the Slot was not open when the new object was created, and the **Slot end time** has passed when this Slot opens at 08:00, no actions will be taken on the object.
4. If a new object is created on the source medium at 14:15 today, then tomorrow when this Slot opens, no actions will be taken on the object because the **Slot end time** has passed.
5. If an object is created on the source medium at 07:59 (*before the slot opens*), at 08:09 (*10 minutes after the object existed on the source*) the 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 object existed on the source*) the object instance on the target will be deleted.

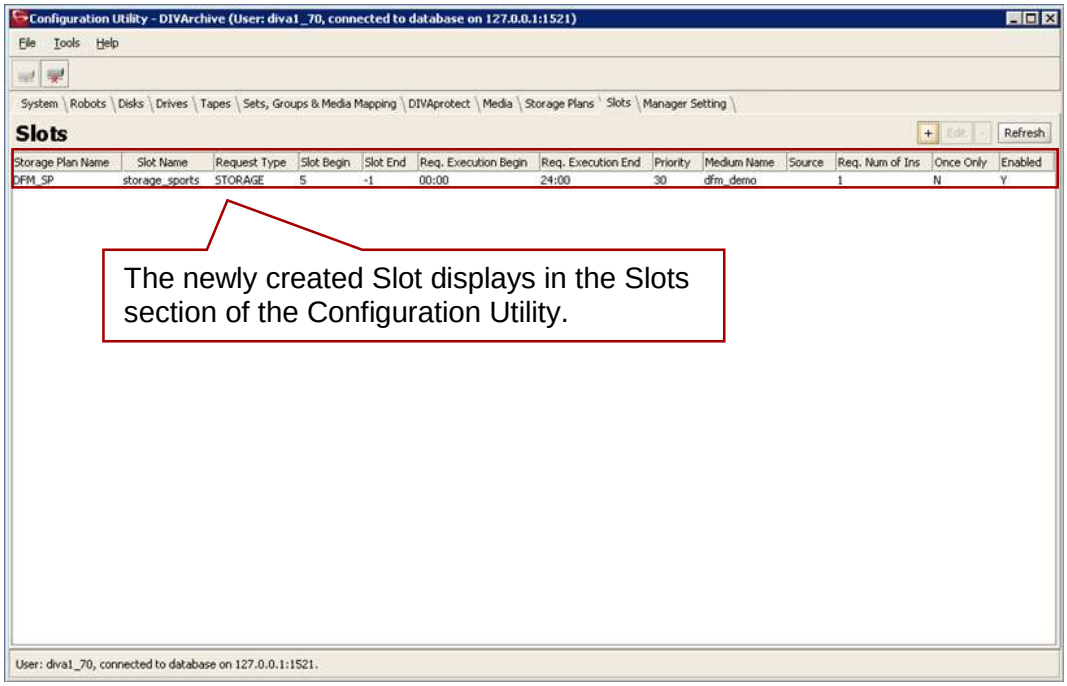
<table>
<tr><th>Parameter</th><th>Definition</th></tr>
<tr><td colspan="2">Example (Storage Slots):
Workflow:
1. A new object is placed onto the source medium while the Slot is closed.
a. A Copy Action is created and the status is Completed.
b. A DeleteInstance Action is created and the status is Scheduled.
c. No actions are executed because the slot is closed.
2. The Disk Instance expires after 20 minutes.
3. The slot opens the next day.
a. The DeleteInstance Action is executed and the status is Completed.
b. The Copy Action status remains as Completed.</td></tr>
<tr><td>Request priority</td><td>Identifies the priority of the request and order of execution when the Action is submitted to the DIVArchive Manager.</td></tr>
<tr><td>Once Only</td><td>Determines whether or not 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:
1. Object has been copied to the target medium but somebody manually deletes the copied instance.
2. This Slot will realize the object does not exist on the target medium and run again, resulting in another copy on the target medium being created.</td></tr>
</table>

Parameter	Definition
Once Only (*continued*)	**Example:** **Once Only:** Y **Workflow:** 1. Object has been copied to the target medium but somebody manually deletes the copied instance. 2. This Slot will not run again, resulting in no 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 object instance on the target medium as being deleted because it was manually deleted before the **Slot end time** was reached.
Enabled	Determines whether or not the Slot is active. **Y** indicates the Slot is enabled and **N** indicates it is disabled. When a slot is disabled (***N***) its Actions are not executed, however they are still created, and the status is updated. **Examples:** • A new object archived and assigned to a Storage Plan with a Tape Storage Slot will have the Tape Slot's **Copy Action** created, but not executed (*no Copy request is sent to the Manager*). • A **DeleteInstance Action** reaching its execution time will not send the **DeleteInstance** request to the Manager, but if somebody manually deletes that instance using the Control GUI, the Action will be updated to the **Completed** status. When the slot is enabled again, pending Actions will be executed immediately.

Parameter	Definition
Req. Number of Instances	The number of 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**: 1. One Object Instance already exists on the target medium. 2. Because there is only 1 instance existing, and this setting is 2, another copy will be made to another disk in the array or another tape in the group.

3. Click **OK** to complete the process (*shown in the previous figure*). The new Slot will be displayed in the **Slots Tab** of the Configuration Utility as shown below.

Figure 35: New Storage Slot Showing in the Configuration Utility Slots Section



2.5.1.1 Nearline Storage Slot Configuration

It is now possible to specify two new Quality of Service (*QOS*) parameters called **NEARLINE_ONLY** and **NEARLINE_AND_DIRECT** via the API and/or the Control GUI in Restore and Multiple Restore Requests. The default QOS for a Restore/Multiple Restore Request is **NEARLINE_AND_DIRECT**. The default QOS for all other requests remains **DIRECT_AND_CACHE**.

When the Manager receives either of the new QOS values, it will initiate a DIRECT restore from **any** available Disk instance, regardless of whether or not the disk instance is located on a disk with a DIVArchive Actor-Disk connection that is configured for Nearline. If no disk instance is found, the Manager will **not** abort the request but rather create a disk instance and proceed as follows:

1. Check for NEARLINE disks during disk selection.
2. If no disks are available, then for QOS:
    - **NEARLINE_ONLY**, the workflow will abort and an error will be 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.
    a. If the disk instance cannot be created, then for QOS:
        - **NEARLINE_ONLY**, the workflow will abort,
        - **NEARLINE_AND_DIRECT**, **DIRECT RESTORE** will be used.
    b. If the disk instance can be created, the workflow will proceed to steps 5-6.
5. Write the disk instance to the database upon success/failure of the Restore.
6. Save component checksums for the new disk instance.
7. Update storage capacity.
8. Update last access date of object instance if not already updated.

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

The Manager will abort a Restore Instance Request with a QOS of **NEARLINE_ONLY** or a QOS of **NEARLINE_AND_DIRECT** by sending an error message to the Control GUI 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 usage of a disk. The Nearline type is defined by specifying one of two usages under **Actor-Disk Connections** called **STORAGE_AND_NEARLINE** and **CACHE_AND_STORAGE_AND_NEARLINE**. Any disk defined with one of these usages, can be used for any of those operations.

A SPM Storage Slot has been updated to support this new work flow 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, the Storage Slot's Slot Begin Time and Slot End Time must set to zero (0) for the associated medium.

Two new options **NEARLINE_ONLY** and **NEARLINE AND DIRECT** have been added in the Restore Request dialog in the quality of service drop-down box.

SPM now supports purging of disk instances created by the Manager during Restore Requests in the following two work flows.

1. Near line Instances are stored on the same medium which is also used in 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 **Object Last Access Time**.
    - Once the Action is put into the **SCHEDULED** state, SPM will start processing and put it into the **POSTPONE** state until the disk reaches it high watermark.
    - Once the disk reaches its high watermark value all of the Delete Instance Actions in the **POSTPONE** state will be executed based on the Disk Cleaning Strategy configured for the medium; in this case it will be object last access time.
9. Separate medium is used for Near Line Storage.
    - In this work flow a Storage Slot is configured on the medium with the **Slot Begin Time** and **Slot End Time** set to zero (0).
    - The Storage Slot must have Once Only set to **N**.
    - Setting the **Slot Begin Time** and **Slot End Time** to zero (0) will cause 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.
    - Disk cleaning strategy for the medium must be based on **Object Last Access Time**.
    - Once the Action is put into the **SCHEDULED** state, SPM will start processing and put it in the **POSTPONE** state until the disk reaches it high watermark.
    - Once the disk reaches its high watermark value all of the Delete Instance Actions in the **POSTPONE** state will be executed based on the Disk Cleaning Strategy configured for the medium; in this case it will be **Object Last Access Time**.

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

2.5.2 Transcode Archived Slot Configuration

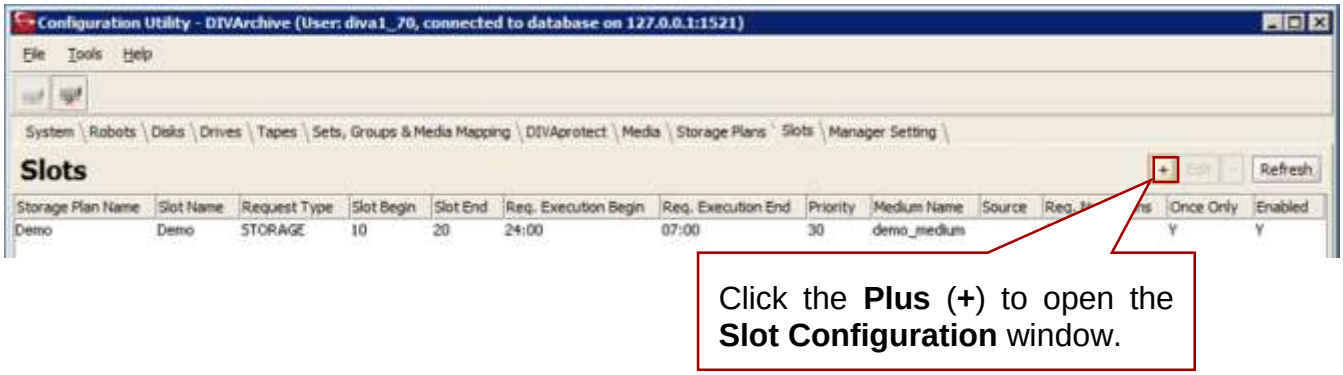
Transcode Archived Slots require configuration on the **Slots Tab** of the DIVArchive Configuration Utility and contain only a single Action. This Slot will take an existing clip in a DIVArchive Object, transcode it to another format, and archive the resulting clip as a new Object.

Because a **Transcode Archived** request creates a new object in the DIVArchive System, it requires directions as how to name the new object (*Transformation Rules*).

Note: Each Oracle DIVArchive Actor defined as a Transcoder must also be defined as a LOCAL type Source/Destination.

1. Click the **Plus** (+) in the **Slots** section of the Configuration Utility as shown in the figure below.

Figure 36: Creating a New Slot



2. When the **Slot Configuration** window opens begin configuring the new Slot by selecting **`TRANSCODE_ARCHIVED`** from the **Slot request type** pull-down box and then continue with the remaining configuration parameters.

*Figure 37: Select **`TRANSCODE_ARCHIVED`** from the Pull-down Box*

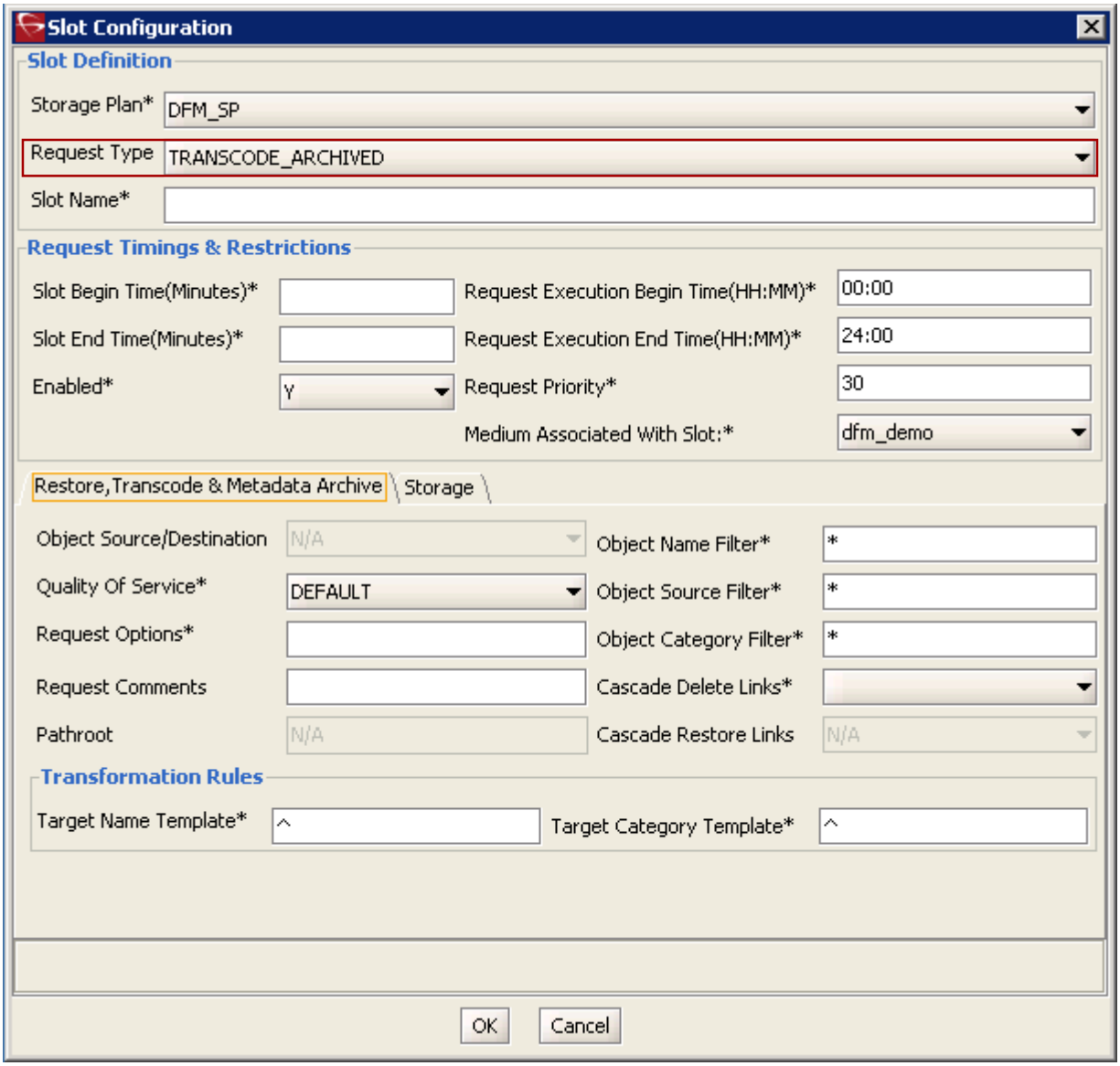


Figure 38: Add New Row Window for Transcode Archived Slots

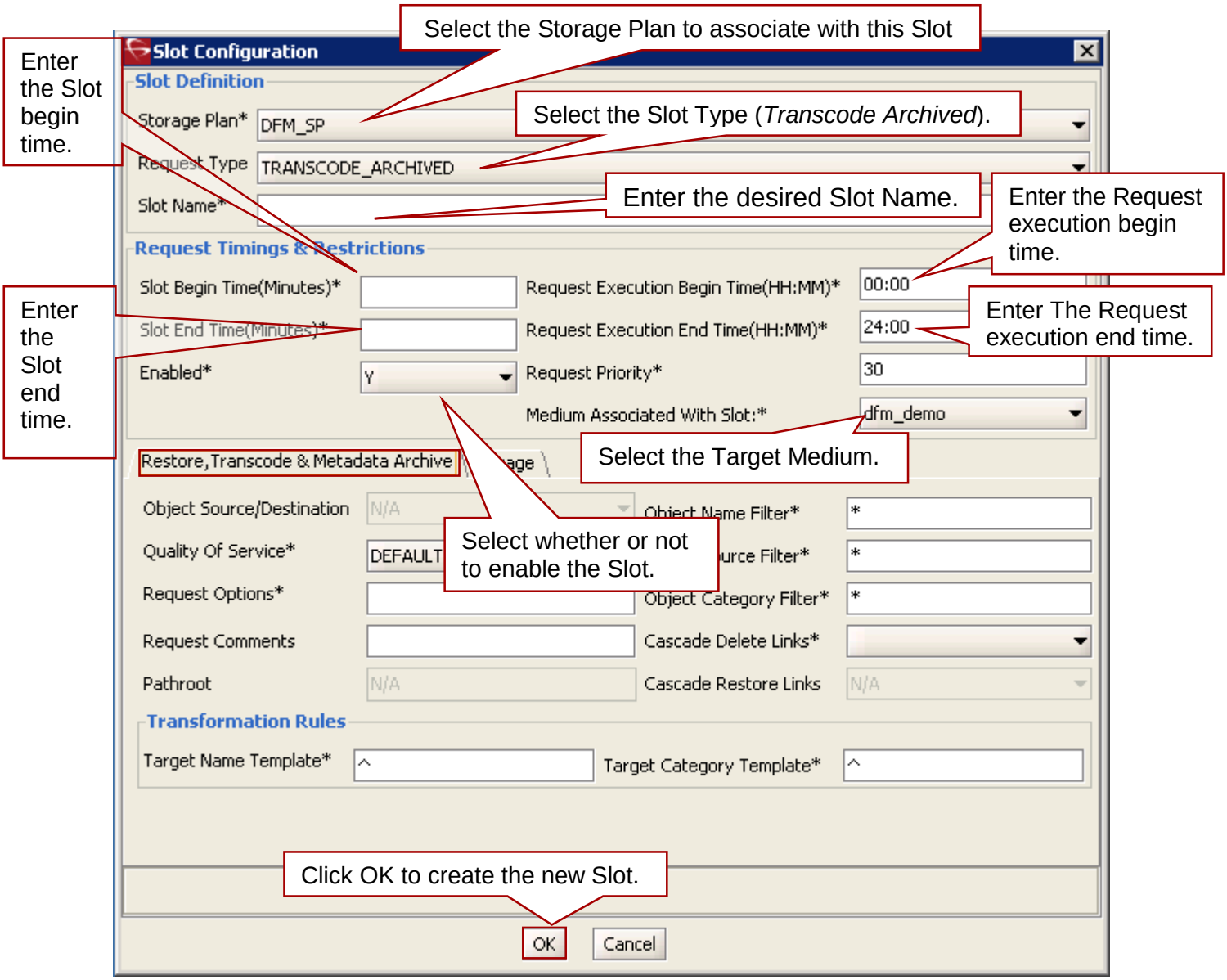


Table 23: DIVArchive Configuration Utility Parameters for Transcode Archived Slots

Parameter	Definition
Slot Name	Enter the desired Slot Name.
Storage Plan	Select the Storage Plan to be associated with this slot.
Medium associated with slot	Select the target Mediums using the pull-down box.
Slot request type	Select the type of request for this slot (*Storage, Transcode Archived, Metadata Archive, or Restore*). In this case select **TRANSCODE_ARCHIVED**.

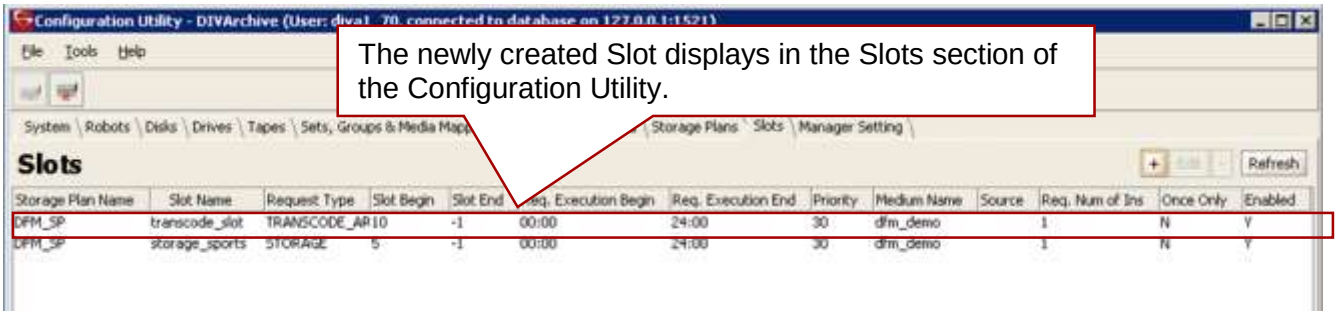
Parameter	Definition
Slot begin time (*min*)	This parameter identifies when the Action will occur. If set to zero (*0*) the Action will be initiated as soon as the object is archived. If set to another number, for example 10 (*minutes*), the Action will execute 10 minutes after the original object is archived.
Slot end time (*min*)	Setting this parameter to -1 indicates the slot will never end (*it is permanent*) and the **Transcode Archived** action will occur regardless of whether or not the slot is open or closed. **WARNING: If the Slot begin time and Slot end time are both set to zero (0) then the behavior is unknown.** **Example:** **Slot begin time**: 10 **Slot end time**: -1 **Workflow:** 1. The original object is archived onto the source medium. 2. 10 minutes later, the slot actions begin and it is transcoded onto the target medium. 3. 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 transcode will take place. If this was set to another number (*other than -1*), the normal rules will apply as described in the **Request execution end time** description below.
Request execution begin time (HH:MM)	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)	Slot execution window closing time in 24 hour time. If set to 24:00 the Slot closes at the end of 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 archiving to tape overnight*). If an object is archived to the source medium outside of the designated time frame for the Slot, even though the object meets the Filter and Storage Plan criteria, actions taken on the object are	

Parameter	Definition
determined but the setting of the **Slot end time** parameter. If the original object was archived to the source medium while the Slot was inactive (*closed*) and has passed the **Slot End Time** once the Slot finally becomes active (*open*), the action taken (*or not taken*) is determined by the **Slot end time** setting. **Example:** **Slot begin time:** 10 **Slot end time:** 20 **Request execution begin time (HH:MM):** 08:00 **Request execution end time (HH:MM):** 14:00 **Workflow:** 1. A new object is placed onto the source medium at 07:00 (*while the Slot is closed*). 2. If the Slot had been open at 07:00 when the new object was created it would have been transcoded to the target at 07:10. 3. Since the Slot was not open when the new object was created, and the **Slot end time** has passed when this Slot opens at 08:00, no action will be taken on the object. 4. If a new object is created on the source medium at 14:15 today, then tomorrow when this Slot opens, no action will be taken on the object because the **Slot end time** has passed. 5. If an object is created on the source medium at 07:59 (*before the Slot opens*), at 08:09 (*10 minutes after the object existed on the source*) the object will be transcoded to the target medium because the **Slot end time** has not yet passed. **Example:** **Slot begin time:** 10 **Slot end time:** -1 **Request execution begin time (HH:MM):** 08:00 **Request execution end time (HH:MM):** 14:00 **Workflow:** 1. A new object is placed onto the source medium at 07:00 (*while the Slot is closed*). 2. If the Slot had been open at 07:00 when the new object was created it would have been transcoded to the target at 07:10. 3. Since the Slot was not open when the new 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 object because the slot never ends. 4. If a new 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 object because the Slot never ends.

Parameter	Definition
Request priority	Identifies the priority of the request when the action is submitted to the DIVArchive Manager.
Once Only	Only applicable to **Storage Slots**.
Enabled	Determines whether or not the Slot is active. **Y** indicates the Slot is enabled and **N** indicates it is disabled. When the Slot is disabled (***N***), its Actions are not executed. However, they are still created when a new object is archived.
Req. Number of Instances	Only applicable to **Copy** Actions in a **Storage Slot**.

3. Click **OK** to complete the process (*shown in the previous figure*). The new Slot will be displayed in the **Slots** section of the Configuration Utility as shown below.

Figure 39: New Transcode Archived Slot Showing in the Configuration Utility Slots Section



There are a second set of tabs at the bottom of the **Slot Configuration Window** 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:

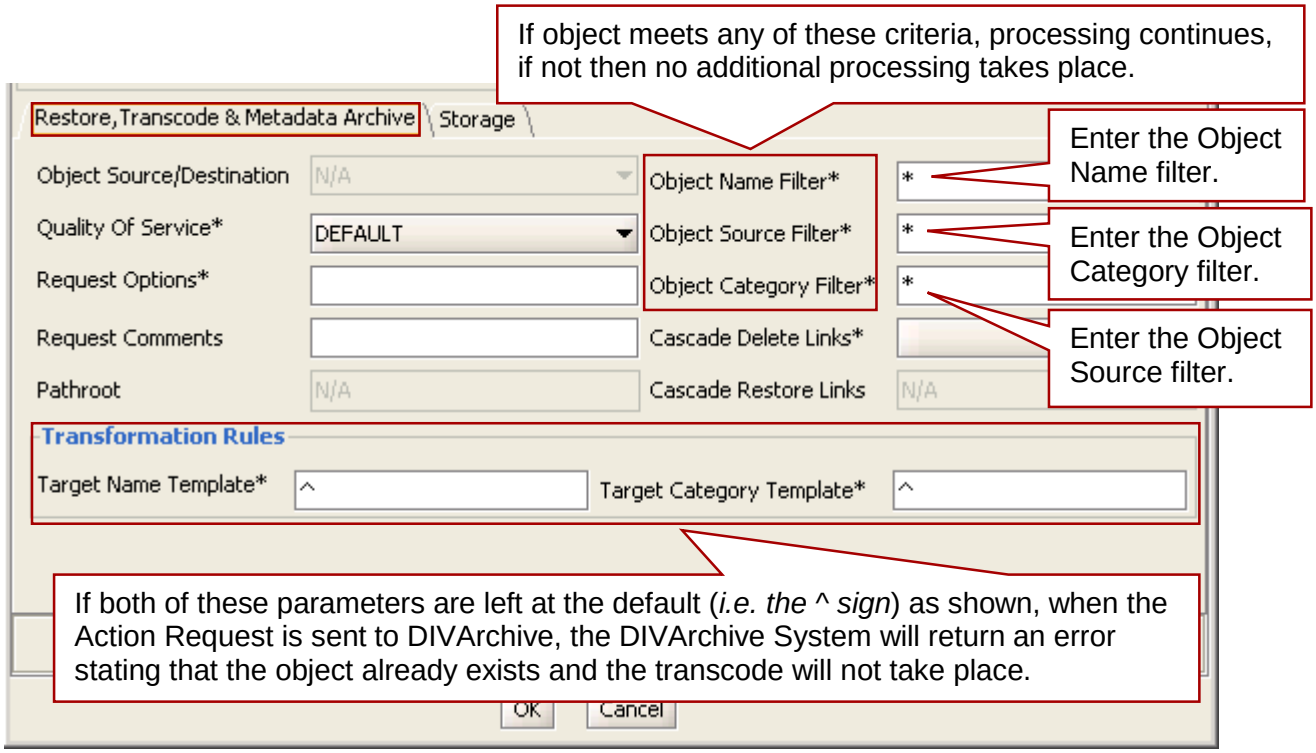
- QOS
- Request Options
- Request Comments
- Object Name Filter
- Object Source Filter
- Object Category Filter
- Cascade Delete Links
- Transformation Rules (*for renaming*).

Object Filters allow restricting the Transcode process to particular objects. They can be used if transcoding all objects belonging to a Storage Plan is not desired.

- **Transformation Rules**: Rules associated with the **Object Filters** to transform the name and category for Transcode Archived and Metadata Archive.

4. Using the **Restore, Transcode & Metadata Archive Tab** at the bottom of the Slot Configuration Window, configure the Object Filters and Transformation Rules as required:

Figure 40: Slot Configuration Window – Additional Configuration Bottom Tabs



If the 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 object for this slot.

Table 24: Object Filter Parameters for Transcode Archived Slots

Parameter	Definition
Object name filter	Enter any desired Object Name filtering criteria for this filter. ***** is a wildcard entry and the default.
Object category filter	Enter any desired Object Category filtering criteria for this filter. ***** is a wildcard entry and the default.

Parameter	Definition
Object source filter	Enter any desired Object Source filtering criteria for this filter. ***** is a wildcard entry and the default.
Target name template	Enter the name for the transcoded clip on the target medium. The caret (^) indicates the system should use the original name.
Target category template	Enter the category for the transcoded clip on the target. The caret (^) indicates the system should use the original category.

Transformation Rules transform the object's name and category during the request and the Transformation Name and Transformation Category 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** and/or **Target category template** will cause SPM to use the original clip name and/or the original clip category. The caret (^) may be used in conjunction with additional naming conventions as shown in the figure above (^_*wm9 and* ^_*cat*).

Using only the caret (^) with no additional characters will cause the DIVArchive System to initiate an error stating that an object with that name/category already exists. Changing just one of these parameters will avoid this error and create the new transcoded clip on the target.

5. Similar to the Transformation Rules, the **Request Templates** parameters are now located at the bottom of the **Slot Configuration Window** in the **Restore, Transcode & Metadata Archive Tab**; configure the additional parameters as required:

Figure 41: Slot Configuration Window – Additional Configuration Bottom Tabs

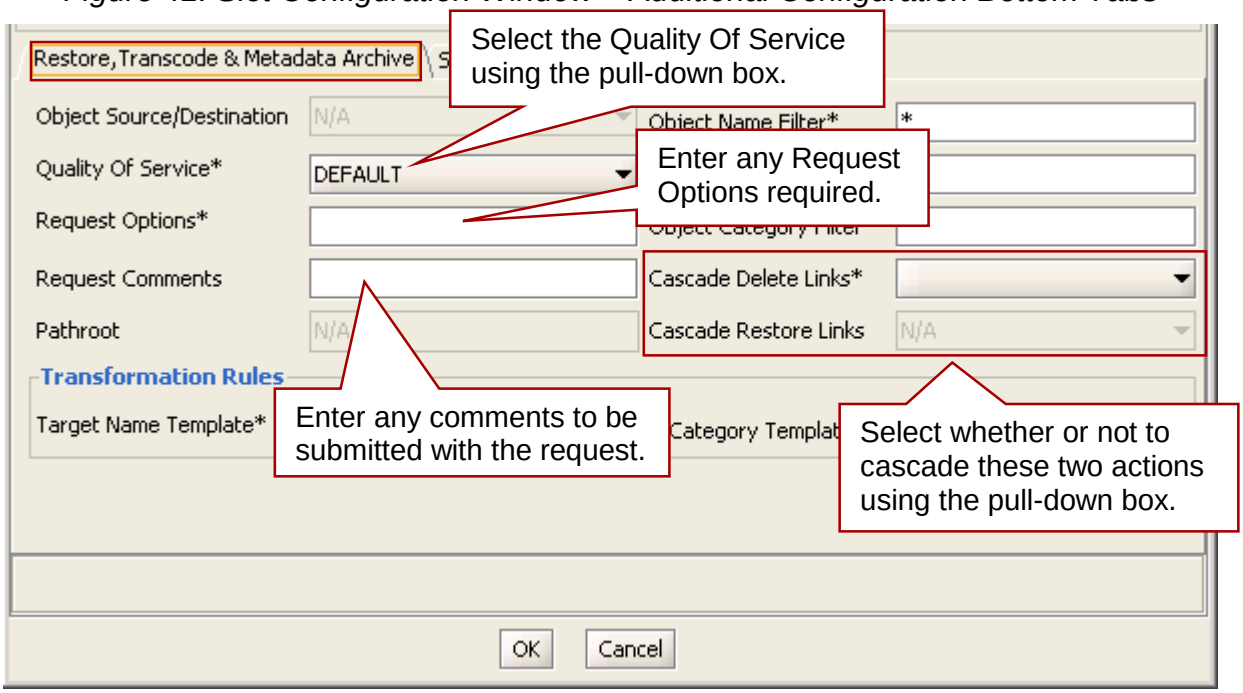


Table 25: Transcode Archived Request Template Parameters

Parameter	Definition
Object Source/Destination	This is not accessible because the newly created object is always archived from the Actor that performed the transcoding. For this reason, transcoding Actors must also be defined as **LOCAL** Source/Destinations.
Pathroot	This is not accessible because the **File Path Root** for the archive of the transcoded object is generated internally by the **Transcode Archived** request.

Request Options	Enter the transcoding options based on the transcoders defined in the Configuration Utility System Tab. Available options are:
	• **`-tr_names`**
	  ○ Specifies the transcoder to be used as defined in Configuration Utility Transcoders
	• **`-tr_archive_format`**
	  ○ Specifies the factory/format to be transcoded to for Archive Slots.
	• **`-tr_restore_format`**
	  ○ Specifies the factory/format to be transcoded to for Restore Slots.

Parameter	Definition
Request Comments	Enter any comments that are to be sent to the DIVArchive Manager with this action request.
Quality Of Service	Use the pull-down box to select the appropriate Quality Of Service for this request. The Quality Of Service parameter has 7 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**: Cache transfers from (*or to*) a Source/Destination to (*or from*) DIVArchive are preferred, but direct transfers will occur if it's not possible to process the request in cache mode. • **Direct&Cache**: Direct transfers from (*or to*) a Source/Destination to (*or from*) DIVArchive are preferred, but cache transfers will occur if it's 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. Alternatively, the data is first transferred entirely to Nearline storage on disk from tape, and then transferred to the destination. If no Nearline service is available, the request will be aborted. • **NEARLINE&DIRECT:** If Nearline transfer is not available (*e.g. no Actor with Nearline storage is available*), a direct transfer will be performed instead.
Cascade delete links	This parameter determines whether or not to perform a Delete Action on just the original object, or on both the original and the transcoded one. 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 or not to perform a Restore Action on just the original object, or on both the original and the transcoded one. 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.

2.5.3 Metadata Archive Slot Configuration

Metadata Archive Slots require configuration on the **Slots Tab** of the DIVArchive Configuration Utility and contain only a single Action. This Slot will archive a Metadata file after an Object is archived. The Metadata file's filename must be the Object Name with no extension, and reside on a location (*Source/Destination and Path*) that will be defined in the Slot Configuration Window. The Object Name and Category under which the Metadata file will be archived are controlled by the **Transformation Rule** entries.

1. Click the **Plus** (+) in the **Slots** section of the Configuration Utility.
2. When the **Slot Configuration** window opens begin configuring the new Slot by selecting **`METADATA_ARCHIVE`** from the **Slot Request Type** pull-down box and then continue with the remaining configuration parameters.

Figure 42: Add New Row Window for Metadata Archive Slots

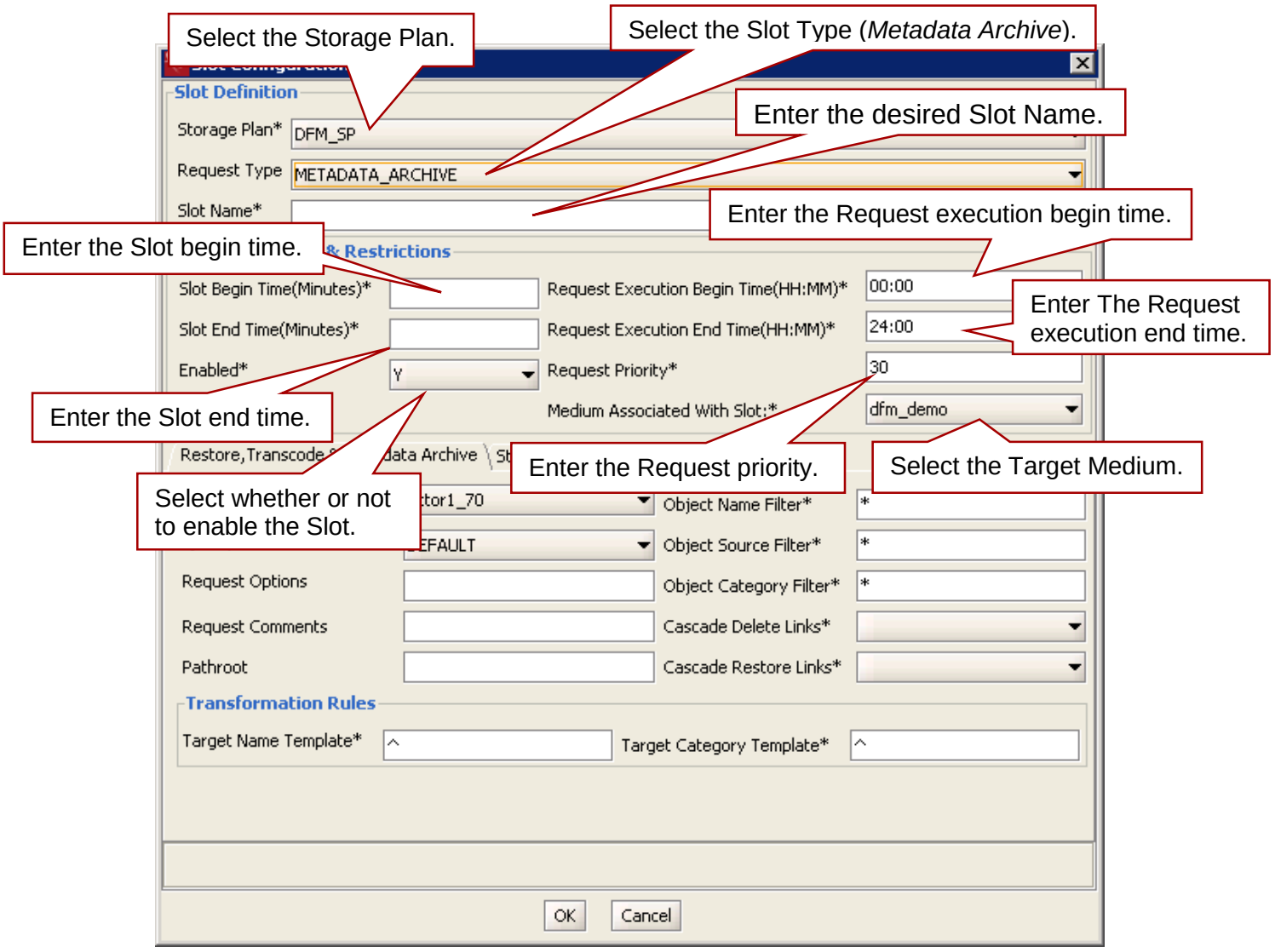


Table 26: DIVArchive Configuration Utility Parameters for Metadata Archive Slots

Parameter	Definition
Slot Name	Enter the desired Slot Name.
Storage Plan	Select the Storage Plan to be associated with this slot.
Medium associated with slot	Select the target Medium using the pull-down box.
Slot request type	Select the type of request for this slot (*Storage, Transcode Archived, Metadata Archive, or Restore*). In this case select **METADATA_ARCHIVE**.
Slot begin time (*min*)	This parameter identifies when the Action will occur. If set to zero (*0*) the Action will be initiated as soon as the object is archived. If set to another number, for example 10 (*minutes*), the Action will execute 10 minutes after the original object is archived.
Slot end time (min)	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 or not the slot is open or closed. **WARNING: If the Slot begin time and Slot end time are both set to zero (0) then the behavior is unknown.** **Example:** **Slot begin time:** 10 **Slot end time:** -1 **Workflow:** 1. The original object is archived onto the source medium. 2. 10 minutes later, the slot actions begin and the Metadata file is archived onto the target medium. 3. 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 in the **Request execution end time** description below.
Request execution begin time (HH:MM)	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*).

Parameter	Definition
Request execution end time (HH:MM)	Slot execution window closing time in 24 hour time. If set to 24:00 the Slot closes at the end of 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 archiving to tape overnight*). If an object is archived to the source medium outside of the designated time frame for the Slot, even though the object meets the Filter and Storage Plan criteria, actions taken on the object are determined but the setting of the **Slot end time** parameter.

If the original object was archived to the source medium while the Slot was inactive (*closed*) and has passed the **Slot End Time** once the Slot finally becomes active (*open*), the action taken (*or not taken*) is determined by the **Slot end time** setting.

Example:

Slot begin time: 10

Slot end time: 20

Request execution begin time (HH:MM): 08:00

Request execution end time (HH:MM): 14:00

Workflow:

1. A new object is placed onto the source medium at 07:00 (*while the Slot is closed*).
2. If the Slot had been open at 07:00 when the new object was created the Metadata file would have been archived to the target at 07:10.
3. Since the Slot was not open when the new object was created, and the **Slot end time** has passed when this Slot opens at 08:00, no action will be taken on the object.
4. If a new object is created on the source medium at 14:15 today, then tomorrow when this Slot opens, no action will be taken on the object because the **Slot end time** has passed.
5. If an object is created on the source medium at 07:59 (*before the Slot opens*), at 08:09 (*10 minutes after the 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: 10

Slot end time: -1

Request execution begin time (HH:MM): 08:00

Request execution end time (HH:MM): 14:00

Parameter	Definition
Workflow: 1. A new object is placed onto the source medium at 07:00 (*while the Slot is closed*). 2. If the Slot had been open at 07:00 when the new object was created the Metadata file would have been archived to the target at 07:10. 3. Since the Slot was not open when the new 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 object because the slot never ends. 4. If a new 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 object because the slot never ends.	
Request priority	Identifies the priority of the request when the action is submitted to the DIVArchive Manager.
Enabled	Determines whether or not the Slot is active. **Y** indicates the Slot is enabled and **N** indicates it is disabled. When the Slot is disabled (***N***), its Actions are not executed. However, they are still created when a new object is archived.

Similar to the Transformation Rules, the **Request Templates** parameters are now located at the bottom of the **Slot Configuration Window** in the **Restore, Transcode & Metadata Archive Tab**; configure the additional parameters as required:

- **Object Filters**: These filters allow restricting the Metadata Archive process to particular objects. They may be used if archiving Metadata files for all objects belonging to a Storage Plan is **not** desired.
- **Transformation Rules**: Rules associated with the **Object Filters** to transform the name and category during the request.

Figure 43: Slot Configuration Window – Additional Configuration Bottom Tabs

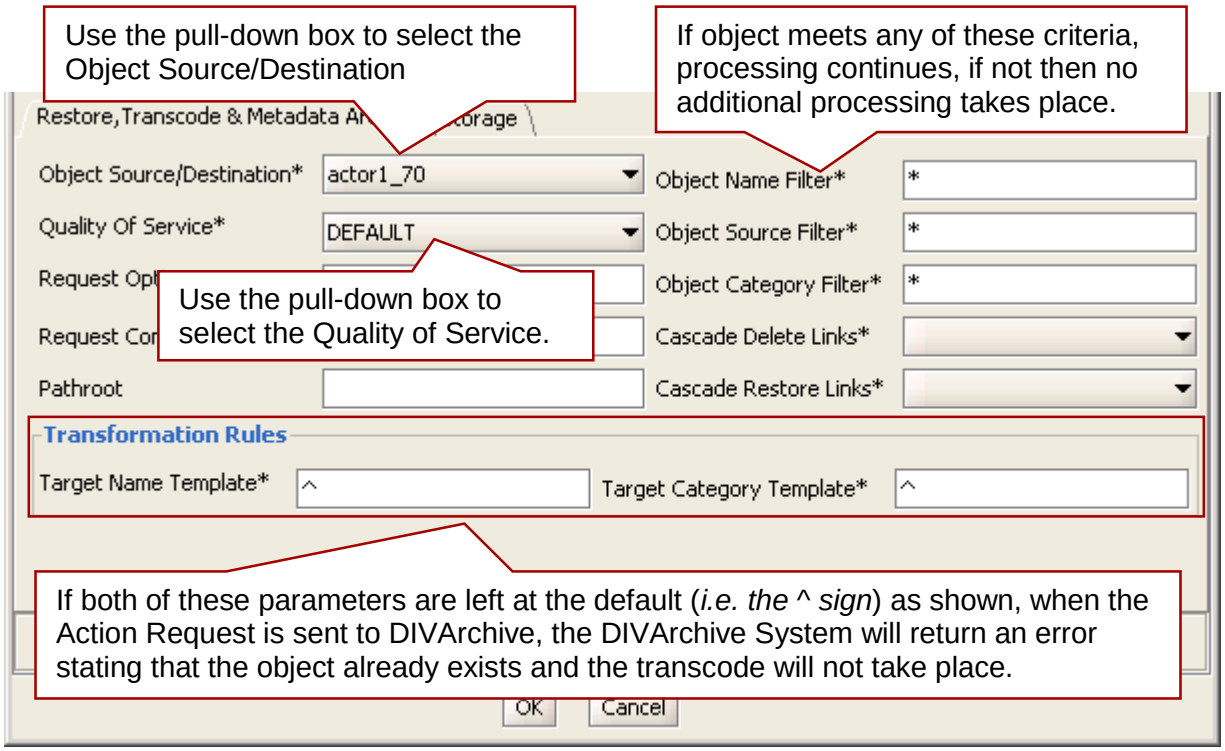


Table 27: Object Filter Parameters for Metadata Archive Slots

Parameter	Definition
Object Source/Destination	Use the pull-down box to select the Object Source/Destination.
Object name filter	Enter any desired Object Name filtering criteria for this filter. ***** is a wildcard entry and the default.
Object category filter	Enter any desired Object Category filtering criteria for this filter. ***** is a wildcard entry and the default.
Object source filter	Enter any desired Object Source filtering criteria for this filter. ***** is a wildcard entry and the default.

Parameter	Definition
Quality Of Service	Use the pull-down box to select the appropriate Quality Of Service for this request. The Quality Of Service parameter has 7 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**: Cache transfers from (*or to*) a Source/Destination to (*or from*) DIVArchive are preferred, but direct transfers will occur if it's not possible to process the request in cache mode. • **Direct&Cache**: Direct transfers from (*or to*) a Source/Destination to (*or from*) DIVArchive are preferred, but cache transfers will occur if it's 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. Alternatively, the data is first transferred entirely to Nearline storage on disk from tape, and then transferred to the destination. If no Nearline service is available, the request will be aborted. • **NEARLINE&DIRECT:** If Nearline transfer is not available (*e.g. no Actor with Nearline storage is available*), a direct transfer will be performed instead.
Target name template	For a Metadata Archive request, this parameter serves two purposes: • Notifies SPM to look for an object with this parameter value on the Source/Destination. • Notifies SPM to create a Metadata archive file on the target using this parameter value. The caret (^) indicates the system should use the original category.
Target category template	Enter the category for the transcoded clip on the target. The caret (^) indicates the system should use the original category.

If the object matches any of the filter criteria then the object 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 object.

Transformation Rules transform the object's name and category during the request and the Transformation Name and Transformation Category 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** and/or **Target category template** will cause SPM to use the original clip name and/or the original clip category. The caret (^) may be used in conjunction with additional naming conventions as shown in the figure above (^_*wm9 and* ^_*cat*).

Using only the caret (^) with no additional characters will cause the DIVArchive System to initiate an error stating that an object with that name/category already exists. Changing just one of these parameters will avoid this error and create the new transcoded clip on the target.

3. Similar to the Transformation Rules, the **Request Templates** parameters are now located at the bottom of the **Slot Configuration Window** in the **Restore, Transcode & Metadata Archive Tab**; configure the additional parameters as required:

Figure 44: Slot Configuration Window – Additional Configuration Bottom Tabs

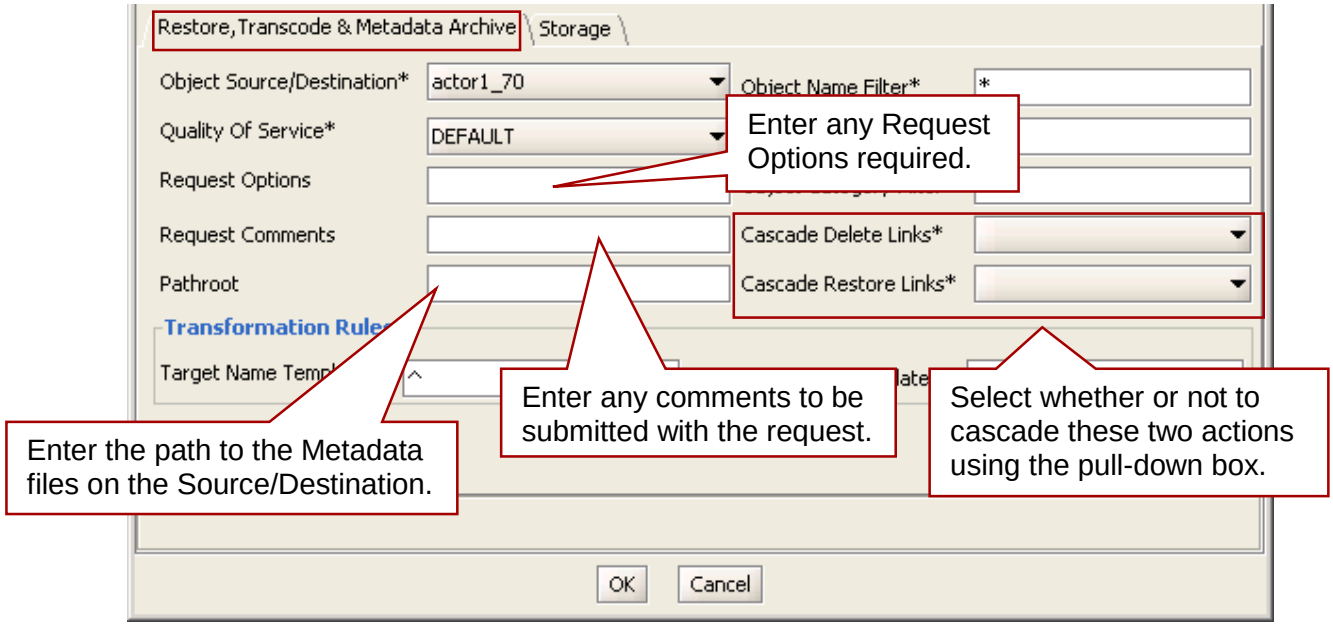


Table 28: Request Template Parameters for Metadata Archive Slots

Parameter	Definition
Request Options	This is only used if a username and password are required for the Source/Destination, otherwise leave this blank.
Request Comments	Enter any comments that are to be sent to the DIVArchive Manager with this action request.
Pathroot	Enter the path to the Metadata files on the Source/Destination.
Cascade delete links	This parameter determines whether or not to perform a **Delete Action** on just the original object or on both the original and the **Metadata Object**. If set to **Y**, when the original is deleted, the **Metadata Object** will also be deleted. When set to **N**, when the original is deleted, the **Metadata Object** remains untouched.
Cascade restore links	This parameter determines whether or not to perform a **Restore Action** on just the original object or on both the original and the **Metadata Object**. If set to **Y**, when the original is restored, the **Metadata Object** will also be restored. When set to **N**, when the original is restored, the **Metadata Object** remains untouched.

2.5.4 Restore Slot Configuration

Restore Slots require configuration on the **Slots Tab** of the DIVArchive Configuration Utility and contain only a single Action. This Slot will take an existing DIVArchive Object and then restore it onto the designated Source/Destination. Click the **Plus** (+) in the **Slots** section of the Configuration Utility.

1. When the **Slot Configuration** window opens begin configuring the new Slot by selecting **RESTORE** from the **Slot Request Type** pull-down box and then continue with the remaining configuration parameters.

Figure 45: Slot Configuration Window – Restore Slot

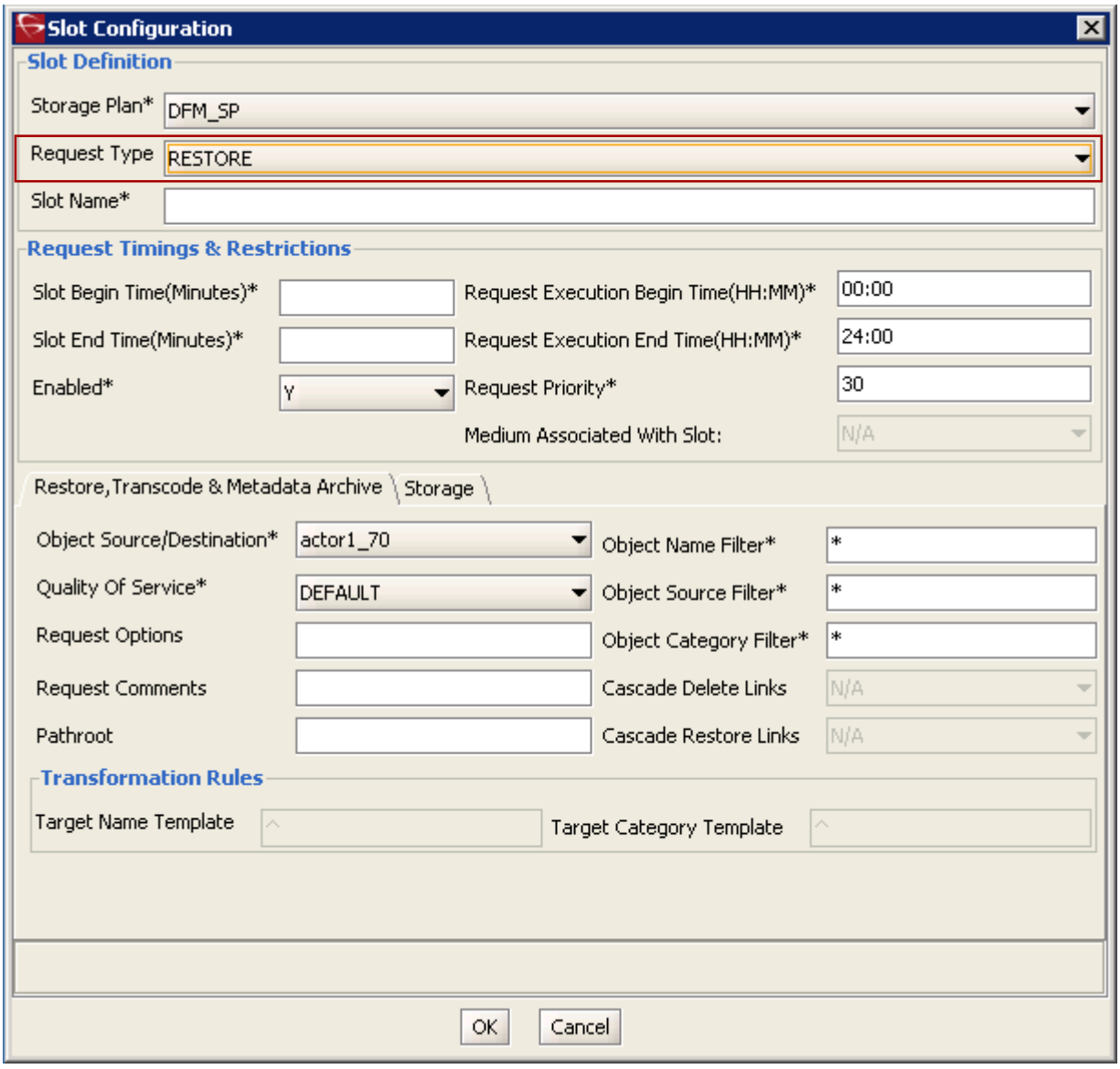


Figure 46: Slot Configuration Window for Restore Slots

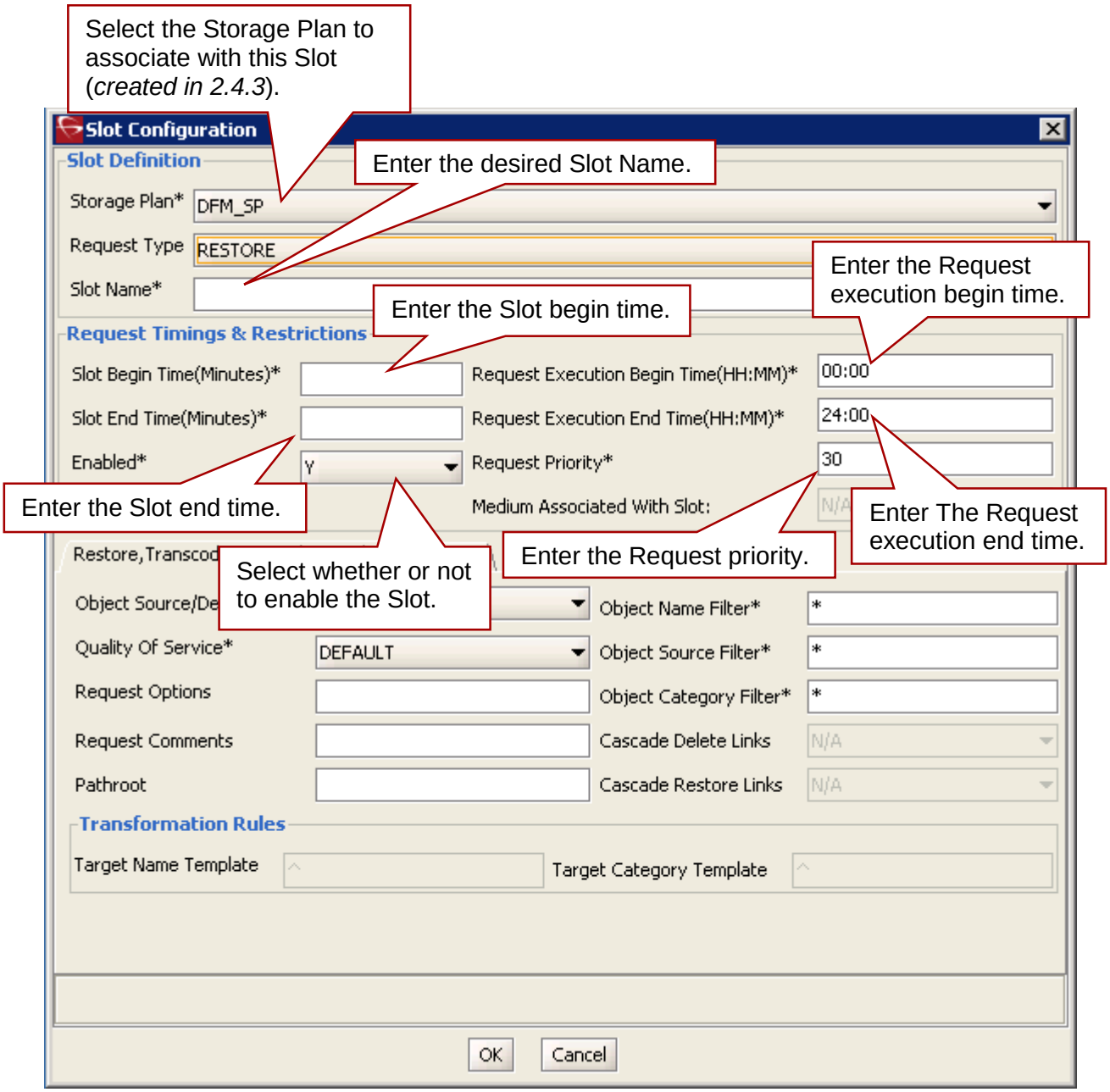


Table 29: DIVArchive Configuration Utility Parameters for Restore Slots

Parameter	Definition
Slot Name	Enter the desired Slot Name.
Storage Plan	Select the Storage Plan with which to associate this Slot.
Slot request type	Select the type of request for this Slot (*Storage, Transcode Archived, Metadata Archive, or Restore*). In this case select **RESTORE**.
Slot begin time (*min*)	This parameter identifies when the Action will occur. If set to zero (*0*) the Action will be initiated as soon as the object is archived. If set to another number, for example 10 (*minutes*), the Action will execute 10 minutes after the original object is archived.

Parameter	Definition
Slot end time (*min*)	Setting this parameter to -1 indicates the slot will never end (*it is permanent*) and the **Restore** action will occur regardless of whether or not the slot is open (*between the Slot start time and Slot end time*) or closed (*after the Slot end time*). **WARNING: If the Slot begin time and Slot end time are both set to zero (0) then the behavior is unknown.** **Example:** **Slot begin time:** 10 **Slot end time:** -1 **Workflow:** 1. The object is found on the Mediums monitored by SPM. 2. 10 minutes later, the slot actions begin and the object is restored to the target Source/Destination. 3. If the **Slot end time** was greater than 0, 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 in the **Request execution end time** description below.
Request execution begin time (HH:MM)	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)	Slot execution window closing time in 24 hour time. If set to 24:00 the Slot closes at the end of each day. If set to another time, for example 17:00, it will close at that time (*in this case 5:00 pm*).

Parameter	Definition

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 restoring from tape overnight*). If an object is archived while the Slot is *closed*, there will be no action taken on the object unless the **Slot end time** is set to -1 (*never ending*). In this case, the action will be taken the next time the Slot opens.

If the original object was archived to the medium while the Slot was inactive (*closed*) and has subsequently passed the **Slot End Time** once the Slot finally becomes active (*open*), the action will not be taken. Therefore, the action taken (*or not taken*) is determined by the **Slot end time** setting.

Example:

Slot begin time: 10

Slot end time: 20

Request execution begin time (HH:MM): 08:00

Request execution end time (HH:MM): 14:00

Workflows:

1. A new object is placed onto the medium at 07:00 (*while the Slot is closed*). Since the Slot is not open when the new object was created, and the **Slot end time** (*20 minutes after the object was placed on the medium i.e. 7:20*) has passed when this Slot opens at 08:00, no action will be taken on the object.

   Note: If the Slot had been open at 07:00 when the new object was created it would have been restored to the target at 07:10.

2. If a new object is placed onto the source medium at 14:15 today, then tomorrow when this Slot opens, no action will be taken on the object because the **Slot end time** has passed.

3. If an object is created on the source medium at 07:59 (*before the Slot opens*), at 08:09 (*10 minutes after the object existed on the source*) the object will be restored to the Source/Destination because the **Slot end time** has not yet passed.

Example:

Slot begin time: 10

Slot end time: -1

Request execution begin time (HH:MM): 08:00

Request execution end time (HH:MM): 14:00

Parameter	Definition
Workflows: 1. A new object is placed onto the source medium at 07:00 (*while the Slot is closed*). Since the Slot is not open when the new 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 object because the slot never ends. Note: If the Slot had been open at 07:00 when the new object was created it would have been restored to the Source/Destination at 07:10. 2. If a new 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 object because the Slot never ends.	
Request priority	Identifies the priority of the request when the action is submitted to the DIVArchive Manager.
Enabled	Determines whether or not the Slot is active. **Y** indicates the Slot is enabled and **N** indicates it is disabled. When the Slot is disabled, its Actions are not executed. However, they are still created when a new object is archived.

2. The **Object Filter** parameters are now located at the bottom of the **Slot Configuration Window** in the **Restore, Transcode & Metadata Archive Tab**; configure the additional parameters as required:
   - **Object Filters**: These filters allow restricting the Restore process to particular objects. They can be used if archiving Metadata files for all objects belonging to a Storage Plan is **not** desired.
   - **Transformation Rules**: Rules associated with the **Object Filters** to transform the name and category during the request. Transformation Rules are not used for Restore Slots, but a Transformation Rule is required to link the Object Filter to a Request Template for the Restore Slot. In DIVArchive 7.3 the Transformation Rule for Restore Slots remains at the default "**^**".

Figure 47: Slot Configuration Window – Additional Configuration Bottom Tabs

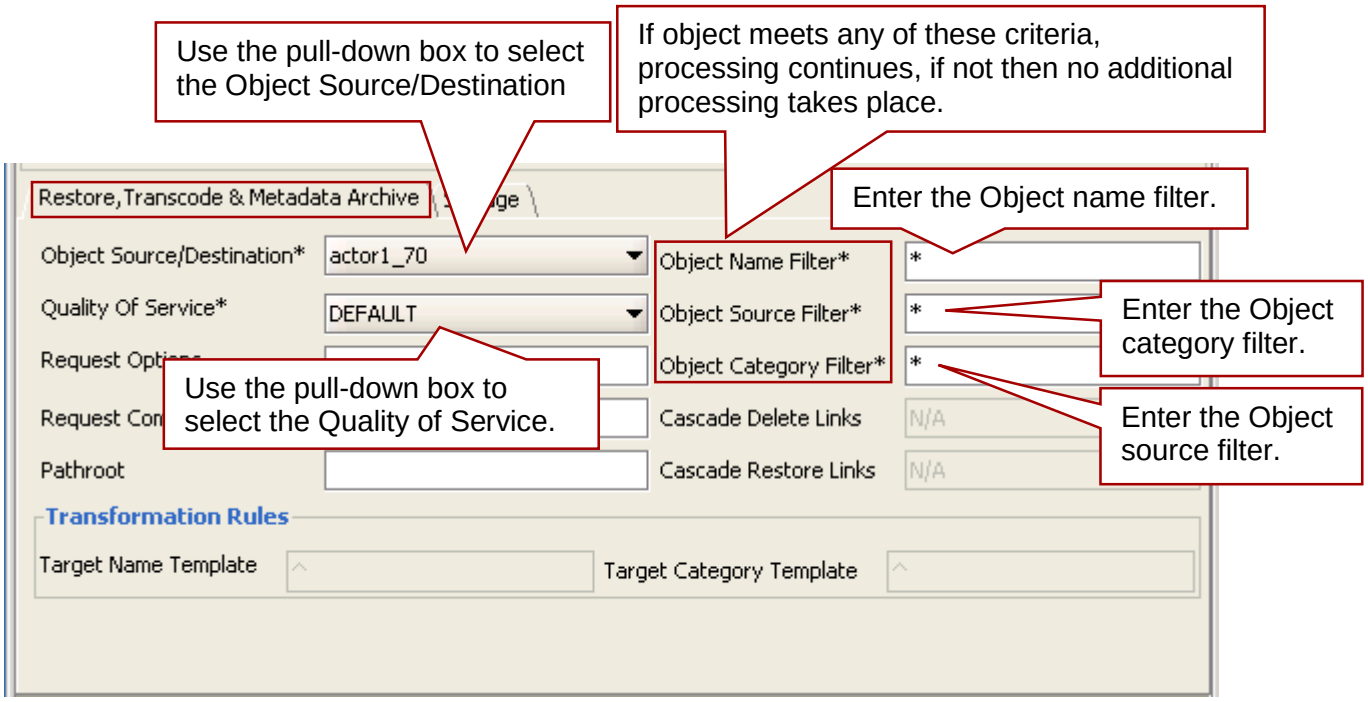


Table 30: Object Filter Parameters for Restore Slots

Parameter	Definition
Object Source/Destination	Use the pull-down box to select the Source/Destination
Object name filter	Enter any desired Object Name filtering criteria for this filter. ***** is a wildcard entry and the default.
Object category filter	Enter any desired Object Category filtering criteria for this filter. ***** is a wildcard entry and the default.
Object source filter	Enter any desired Object Source filtering criteria for this filter. ***** is a wildcard entry and the default.

Parameter	Definition
Quality Of Service	Use the pull-down box to select the appropriate Quality Of Service for this request. The Quality Of Service parameter has 7 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**: Cache transfers from (*or to*) a Source/Destination to (*or from*) DIVArchive are preferred, but direct transfers will occur if it's not possible to process the request in cache mode. • **Direct&Cache**: Direct transfers from (*or to*) a Source/Destination to (*or from*) DIVArchive are preferred, but cache transfers will occur if it's 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. Alternatively, the data is first transferred entirely to Nearline storage on disk from tape, and then transferred to the destination. If no Nearline service is available, the request will be aborted. • **NEARLINE&DIRECT:** If Nearline transfer is not available (*e.g. no Actor with Nearline storage is available*), a direct transfer will be performed instead.

If the object matches any of the filter criteria then the object will continue to be processed using the `Restore` Request Type. If all of the filter criteria are not matched then no further action is taken on the object.

3. The **Request Templates** parameters are now located at the bottom of the **Slot Configuration Window** in the **Restore, Transcode & Metadata Archive Tab**; configure the additional parameters as required:

Figure 48: Slot Configuration Window – Additional Configuration Bottom Tabs

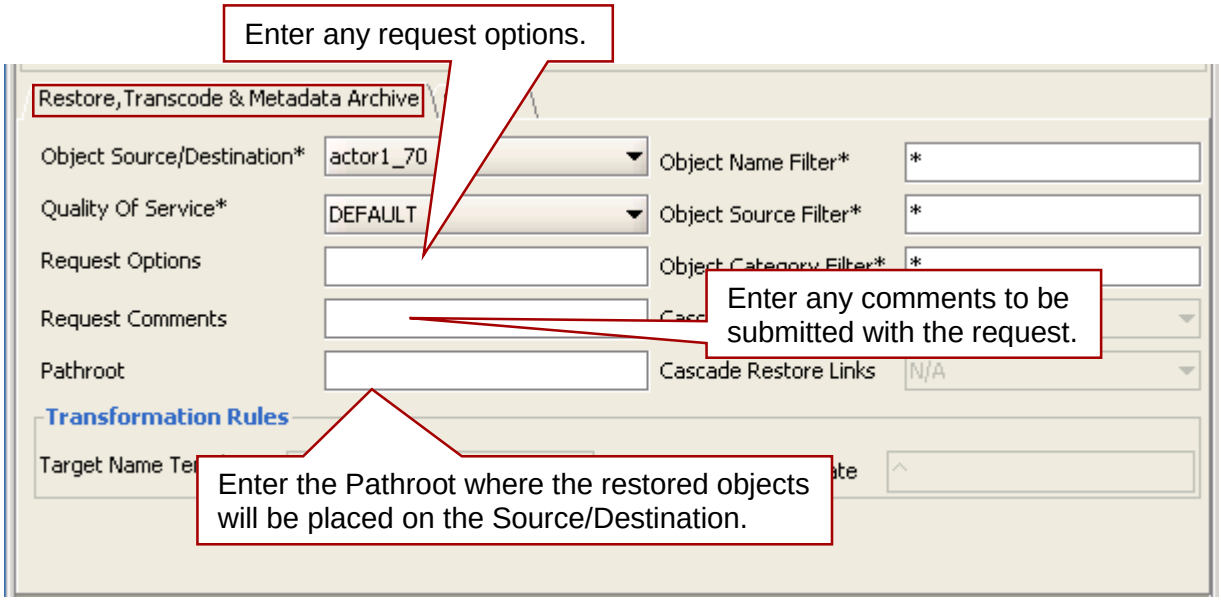


Table 31: Request Template Parameters for Restore Slots

Parameter	Definition
Request Options	Enter any options required that would be used for a normal Restore Request through DIVArchive, otherwise leave this blank. This field can be used for options such as transcoding: `-tr_names <transcoder name as configured on the System tab> -tr_restore_format <format/factory>`
Request Comments	Enter any comments that are to be sent to the DIVArchive Manager with this action request.
Pathroot	Enter the path to where the object should be restored on the Source/Destination. If this parameter is left blank SPM will use whatever Pathroot is associated with the Source definition in the Configuration Utility's System Tab Source/Destination listing. If there is no Pathroot in the Source definition, the original Pathroot used when the object was archived will be used. If a non-existent Pathroot is entered for this parameter, the request will create the new folder.

2.5.5 SPM Behavior with Delete Object

The way SPM reacts to a Delete Object request has changed in DIVArchive 7.3. **DELETE_OBJECT_ONLY_LAST_INSTANCE** is a new configuration parameter that works in conjunction with the existing **ALLOW_OBJECT_DELETION** parameter.

Warning: This feature must be managed carefully because you can lose objects.

2.5.5.1 *ALLOW_OBJECT_DELETION* Parameter

This parameter must be enabled for SPM to delete objects. Delete Object must be enabled or disabled for each Storage Plan separately during its configuration in Configuration Utility for additional security & flexibility. Delete 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 Object is executed only when the following conditions are satisfied:

- The **ALLOW_OBJECT_DELETION** parameter must be enabled.
- Delete object must be enabled for the storage-plan level through the Configuration Utility.
- All of the storage slot(s) in the storage plan associated with the object must have a slot end time.
- Delete object will be executed by the storage slot that has the highest end time in the storage plan.
  - An object can be associated with only one storage plan.
  - If multiple storage slots have a max end time in a storage-plan, the slot with the max slot id among the slots with max end time will do the delete object.
- Just enabling **ALLOW_OBJECT_DELETION** will not acknowledge a storage slot medium water mark and the delete object is execute 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**.

2.5.5.2 *DELETE_OBJECT_ONLY_LAST_INSTANCE* Parameter

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

Enabling this parameter forces SPM to delete an object at the end of a storage slot only if it is the last instance in the entire DIVArchive 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 do a delete object at the end of the storage slot, it will always only do a delete instance at the end of a storage slot. The following condition needs to be satisfied for this to work:

- This parameter requires **ALLOW_OBJECT_DELETION** to be enabled. Just enabling **DELETE_OBJECT_ONLY_LAST_INSTANCE** will have no effect and the SPM Delete Object feature will not be enabled.
- The parameter **DELETE_OBJECT_ONLY_LAST_INSTANCE** must be enabled.

- Delete Object must be enabled for the Storage-Plan level through Configuration Utility.
- All of the Storage Slots in the Storage Plan associated with the object must have a slot end time.
- Delete Object will be executed by the Storage Slot that has the highest end time in the Storage Plan.
  - An object can be associated with only one storage plan.
  - If multiple Storage Slots have a max end time in a Storage Plan, the slot with max slot id among the slots with max end time will do the Delete Object.
- At the end of the Storage Slot if more than two instances exist in the DIVArchive system or on the medium, only Delete Instance is executed.
- Delete Object is executed only if one instance exists on the entire DIVArchive 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 end time is water marked, the Delete Object is postponed until the watermark condition is satisfied.

2.6 Changing a Storage Plan

Storage Plans contain Slots that may be altered if required. New Slots may be added, while existing Slots may be edited, deleted, and/or disabled. These adjustments to Storage Plans must be performed carefully to ensure the results that are desired. Below is an example of adding a new Slot to a Storage Plan and changing and existing Slot within the same Storage Plan.

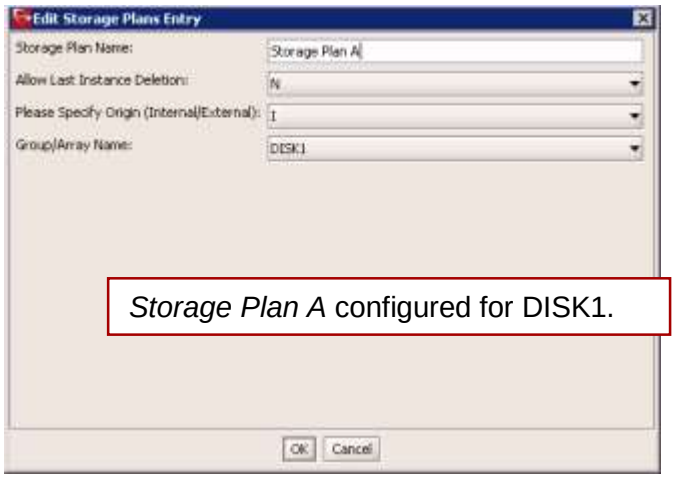
Example:

- DISK1 = Local DIVArchive 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 DIVArchive Managed Disk
  - Not an initial slot; added at Step 4 of the workflow below.
  - This Storage Slot is added later with a Slot Start Time=0, and Slot End Time= -1

The company has *Storage Plan A* which includes three total instances; 1 instance on disk and 2 instances on tape. Here is the workflow for *Storage Plan A*:

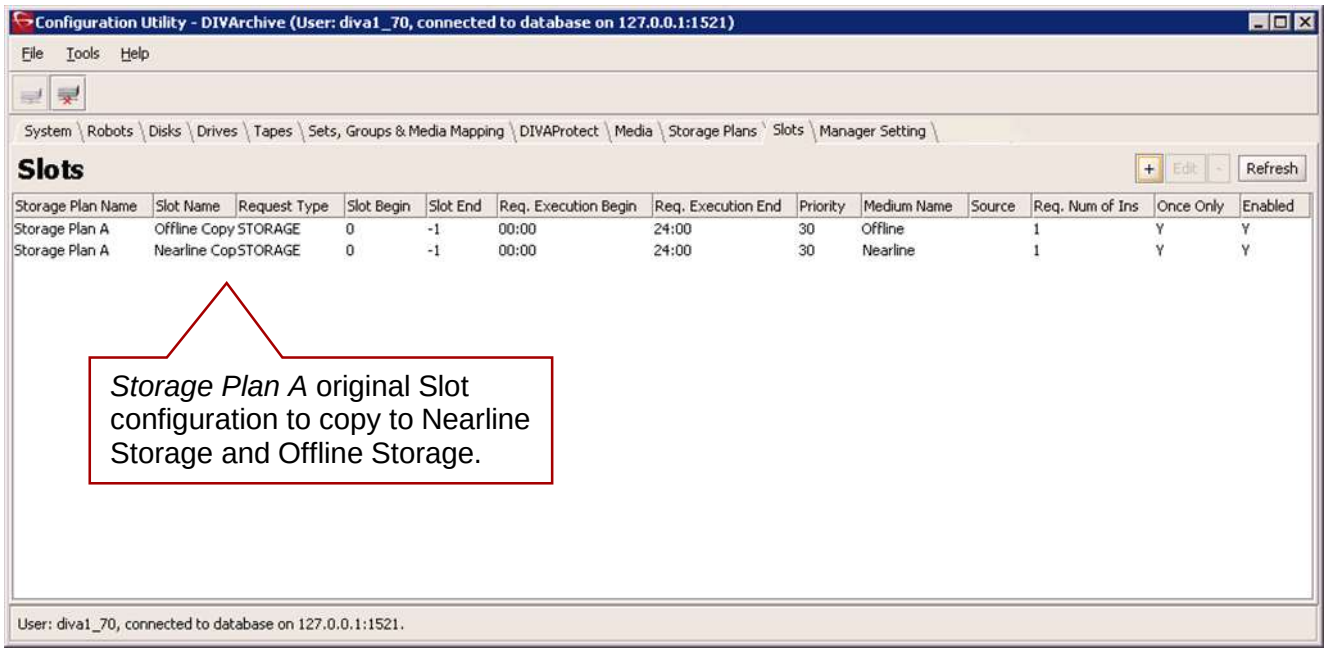
1. Object 1234 is archived to DISK1 under *Storage Plan A*.

Figure 49: Storage Plan A Configuration



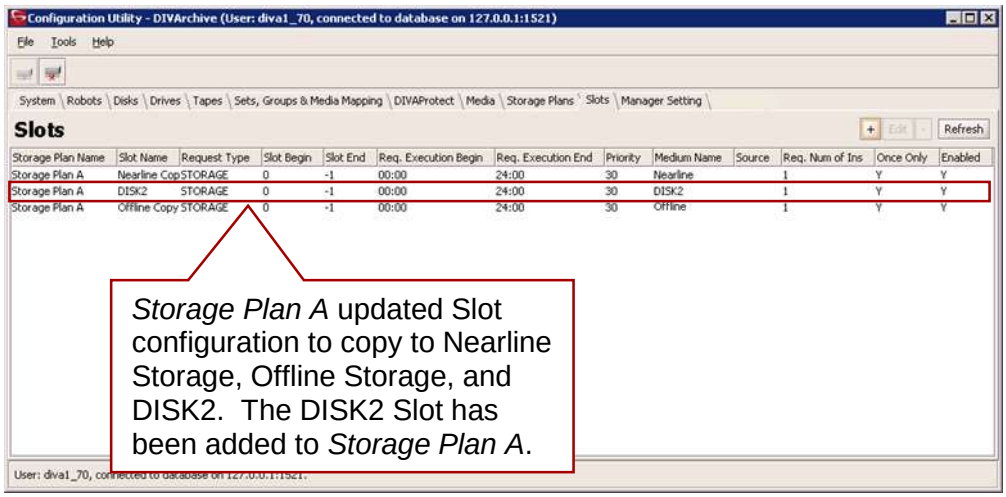
2. Storage Plan A has slots for the following operations and Object 1234 is copied to both Tape Groups:
    a. Copy to NEARLINE
    b. Copy to OFFLINE.

Figure 50: Initial Slot Configuration for Storage Plan A



3. At some point in the future, the company changes its internal policy for content with Storage Plan A to three total instances; 2 instances on disks (*DISK1 and DISK2*), and 1 instance on the Nearline Tape Group.

Figure 51: New Slot Configured for Storage Plan A

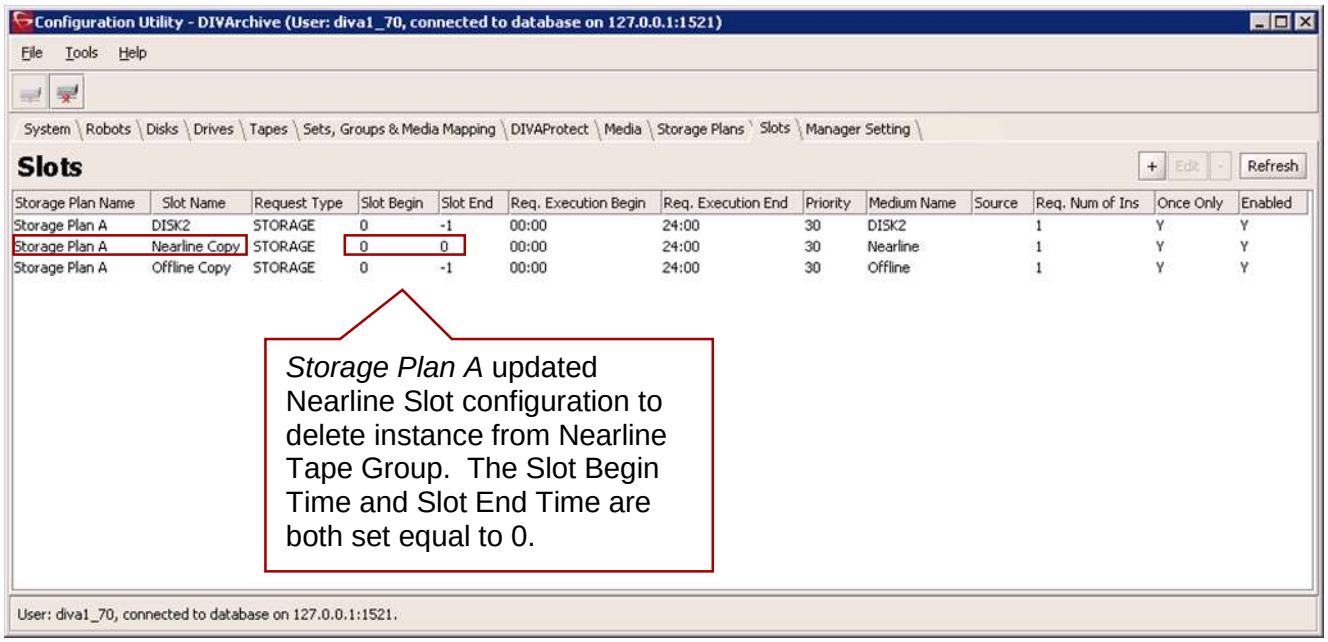


4. The new Slot change is added that would copy Object 1234 to DISK2 but it is necessary to also delete the object instance from the Nearline Tape Group.

    c. To delete the object instance from the Nearline Tape Group, the user would change the Slot End Time to 0 for the Nearline Storage Slot (*refer to Section 2.6.1 for more information*).

    d. A new Storage Slot on DISK2 is added with a Slot Start Time=0 and Slot End Time= -1 (*refer to Section 2.6.2 for more information*).

Figure 52: Updated Final Slot Configuration



2.6.1 Disabling and Deleting Slots

In general it is recommended that slots are **not** deleted, but disabled instead. Deleting a Slot does **not** remove the content form the DIVArchive 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 a 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.

2.6.2 Adding a Slot

The user has the ability to add a new Slot to an existing Storage Plan; however this must be done cautiously in order to achieve the desired results.

When adding a new Slot to existing SP, the user must make sure that the Slot Start Time and Slot End Time are both valid (*refer to Section 2.5.1 for valid parameters*). New Slots are always immediately applicable to new objects, but for existing objects the user must make sure that the Slot Start and Slot End Time are correct otherwise the action specified in the new Slot will not be executed upon existing objects.

Example:

A Storage Plan with numerous 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 objects because the Slot has already expired for all of the existing objects:

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

For existing objects to be affected by the new Storage Slot, the object must be copied immediately to the Storage Slot Medium, but never deleted. To do this, the Slot Start Time is set to 0 and the Slot End Time should be infinite (*equal to -1*). 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 (*e.g. restore*) Slots, configure the Slot Start Time=0 and the Slot End Time=100. In this case SPM will try restoring objects that have existed for less than the Slot End Time (*100 hours*); in other words the objects whose slot 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 object within 100 hours from the Archive Time, that object is never restored. If the Slot End Time= -1, SPM will try to restore all of the objects.

2.7 Watermark-Based Disk Cleaning Management

Disk Array cleaning can be managed by a fixed slot duration, in which case disk instances or objects are deleted when they expire, or using a watermark-based housekeeping mechanism, in which case disk instances or objects are only deleted when space needs to be recovered. This section discusses the details of the second option.

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

SPM allows explicitly specifying the order in which Array instances/objects of content are removed during DIVArchive's Array housekeeping activities. SPM includes an Array monitoring component to manage Array space usage.

Array housekeeping is based on a Watermark System monitored by the Disk Space Monitor component. When the Array's Storage Slot retention period for an object expires, the instance/object becomes eligible for deletion, but stays on the Array as long as there is enough space on the Array.

- **High Watermark**: When this threshold is reached DIVArchive initiates housekeeping activities and expired instances/objects are deleted according to the defined policies. If no instances/objects are eligible for deletion (*e.g. the retention period has not expired or a dependency is not met*), used Array space can exceed the watermark. It is the responsibility of the users to ensure that the Array filling rate matches the system configuration, and to monitor how Array usage evolves.
- **Low watermark**: At the low watermark DIVArchive stops deleting instances/objects. Supported policies and rules for housekeeping include deleting Array instances based upon Last Access Time or Largest Object Size.
- **Mixed Mode:** When an array is watermarked in **Mixed Mode**, both instances/objects marked for deletion and objects whose slots are still open are considered. Instances/objects in that Array will be deleted based upon whichever of the following comes first:

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

  - Array reached the **High Watermark**. Even if the object's slot has not ended, instances/objects will be deleted to bring the Array to the Low Watermark level.
  - Storage Slot for that object has ended. Even though the **High Watermark** is not reached, since the slot ended, the instance/object will be deleted.

  v. This does not mean the Array will be cleaned to the level of **Low Watermark**.

- Watermarking is only valid for **Storage Slots**.

Note: For SPM to delete an object belonging to a Storage Plan, the `ALLOW_OBJECT_DELETION` parameter must be set to 'Y' in both the SPM Configuration File and Configuration Utility.

2.8 External Storage Plans

External Storage Plan functionality allows management of the DIVArchive Object Lifecycle using 3rd party external systems. The SPM Database API can be utilized to develop integration of 3rd party components with DIVArchive and SPM.

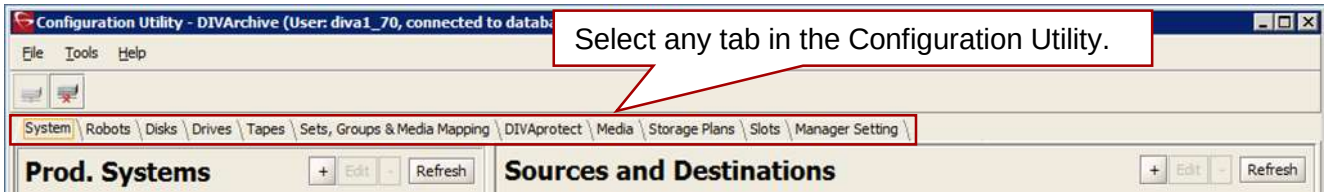
Note: In this version of SPM, External (*E*) should only be used by Oracle personnel.

2.9 Configuration Validation

The SPM configuration can be validated for correctness using the DIVArchive Configuration Utility. Use the following steps to validate the SPM configuration:

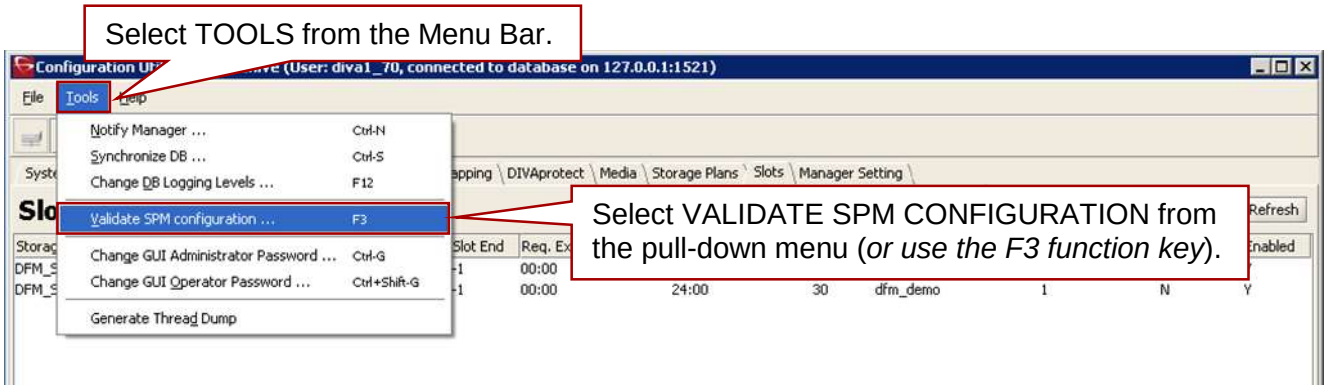
1. Start the DIVArchive Configuration Utility and select any tab across the top of the utility.

Figure 53: Open the Configuration Utility and Select Any Tab



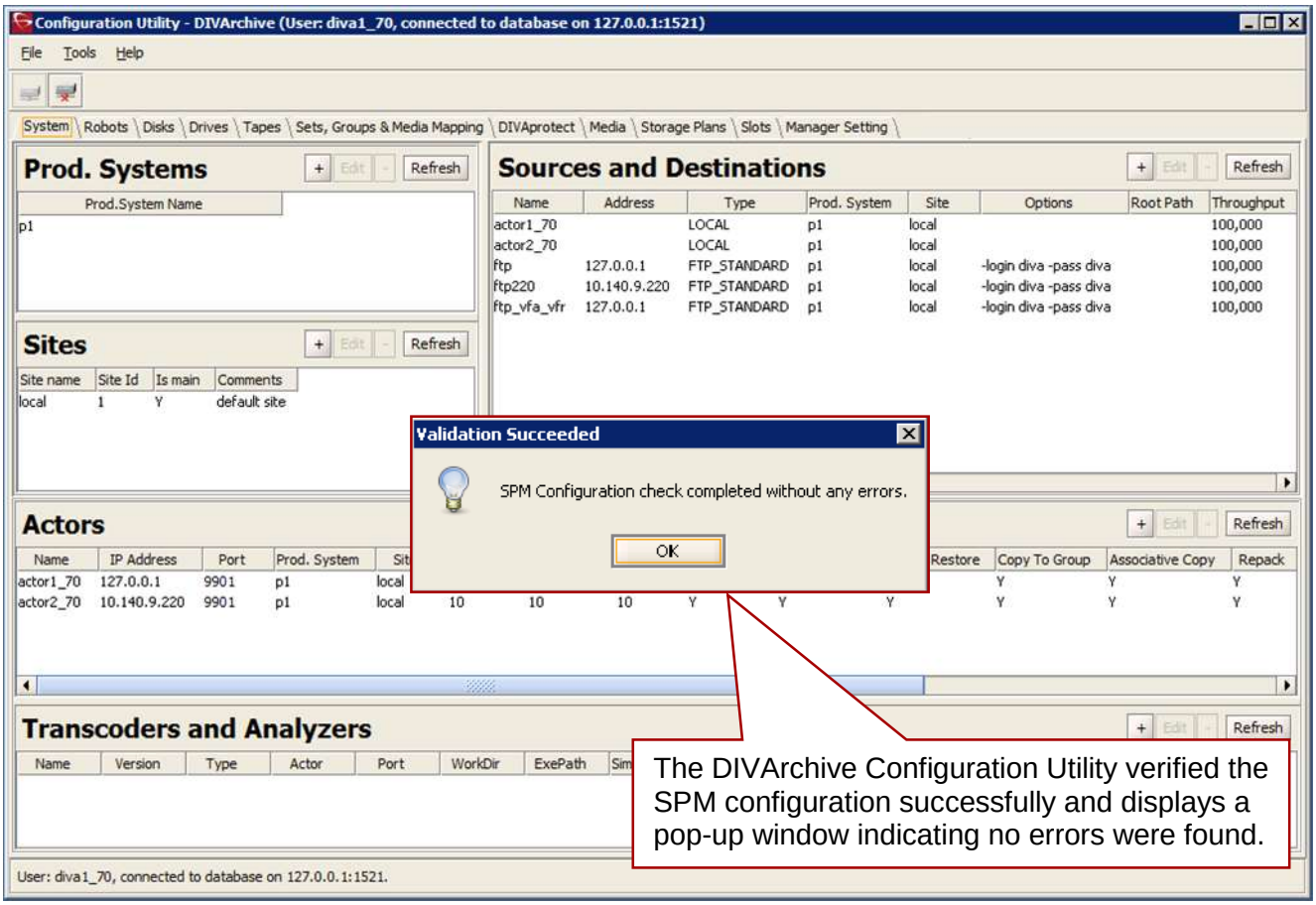
2. From the menu bar, select **TOOLS**, then **VALIDATE SPM CONFIGURATION** (*or use the **F3** function key*).

Figure 54: Validating the SPM Configuration using the DIVArchive Configuration Utility



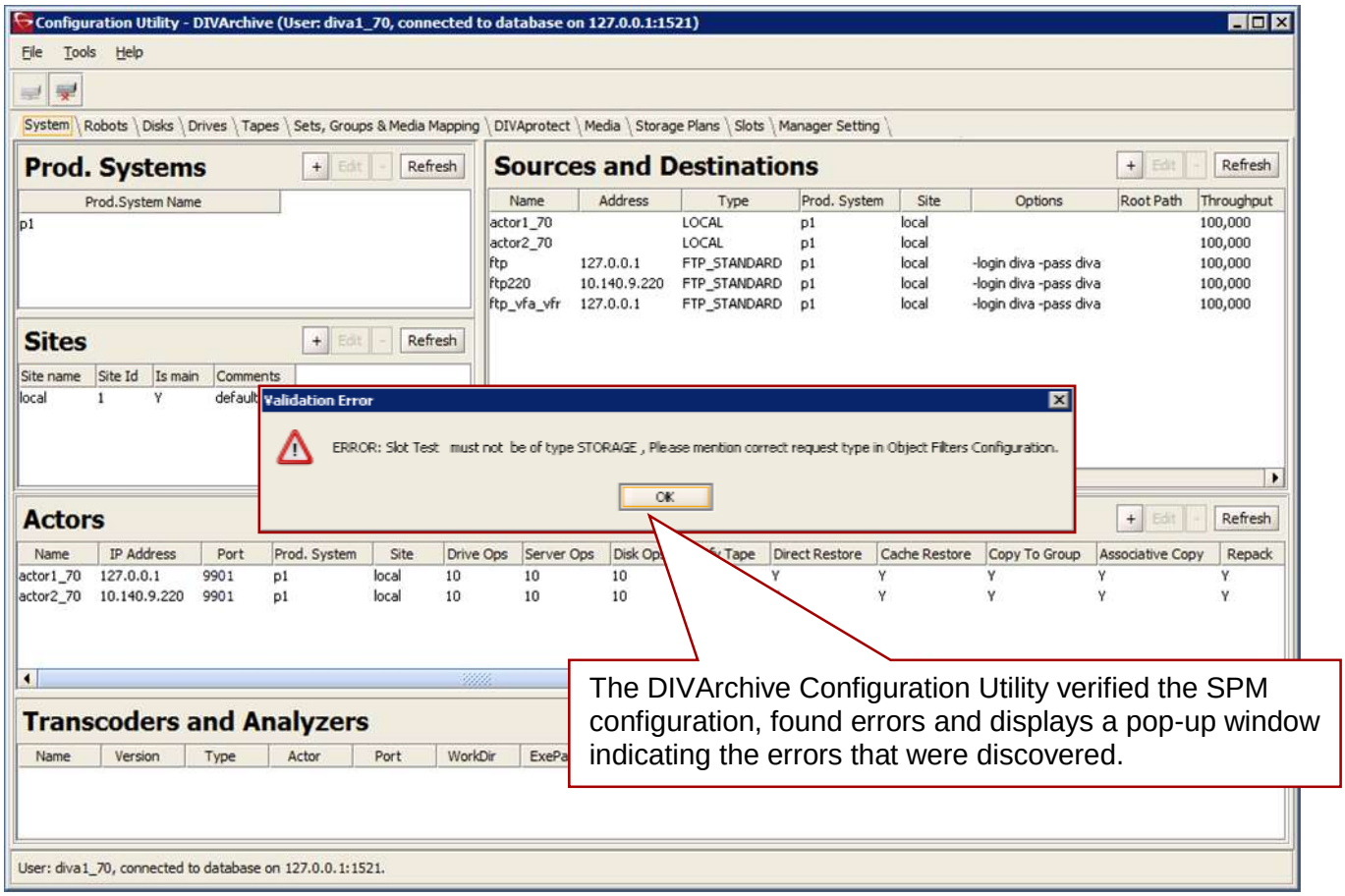
3. The DIVArchive Configuration Utility will verify the configuration of SPM and return a pop-up window with one of two results:
   - The configuration is valid (*as shown below*).

Figure 55: Successful SPM Validation Completed



- The configuration contained errors and is therefore invalid (*as shown below*).

Figure 56: SPM Validation Found an Error in the SPM Configuration



3 Operations

3.1 Service Management – SPM Command Line Interface

The SPM Service can be managed using the **Windows Command Line** interface (***START***, ***RUN***, ***CMD***) or through the **Windows Services Panel**. It is recommended that the command line interface be used.

Usage: `spmservice.exe command [options]`

Where **`command`** is one of the following:

Table 32: SPM Command Line Interface Commands

Command	Definition
`install` (*or* **`-i`**)	Installs SPM as a Windows System Service.
`uninstall` (*or* **`-u`**)	Removes the SPM Windows System Service.
`debug` (*or* **`-d`**)	Starts SPM in *console* mode for troubleshooting purposes.
`version` (*or* **`-v`**)	Displays SPM version information and then exits.
`help` (*or* **`-h`**)	Displays help information and then exits.

Where **`[options]`** includes:

Table 33: SPM Command Line Interface Options

Option	Definition
`-conf` (*or* **`-f`**)	Indicates to load the settings from a specific configuration file.

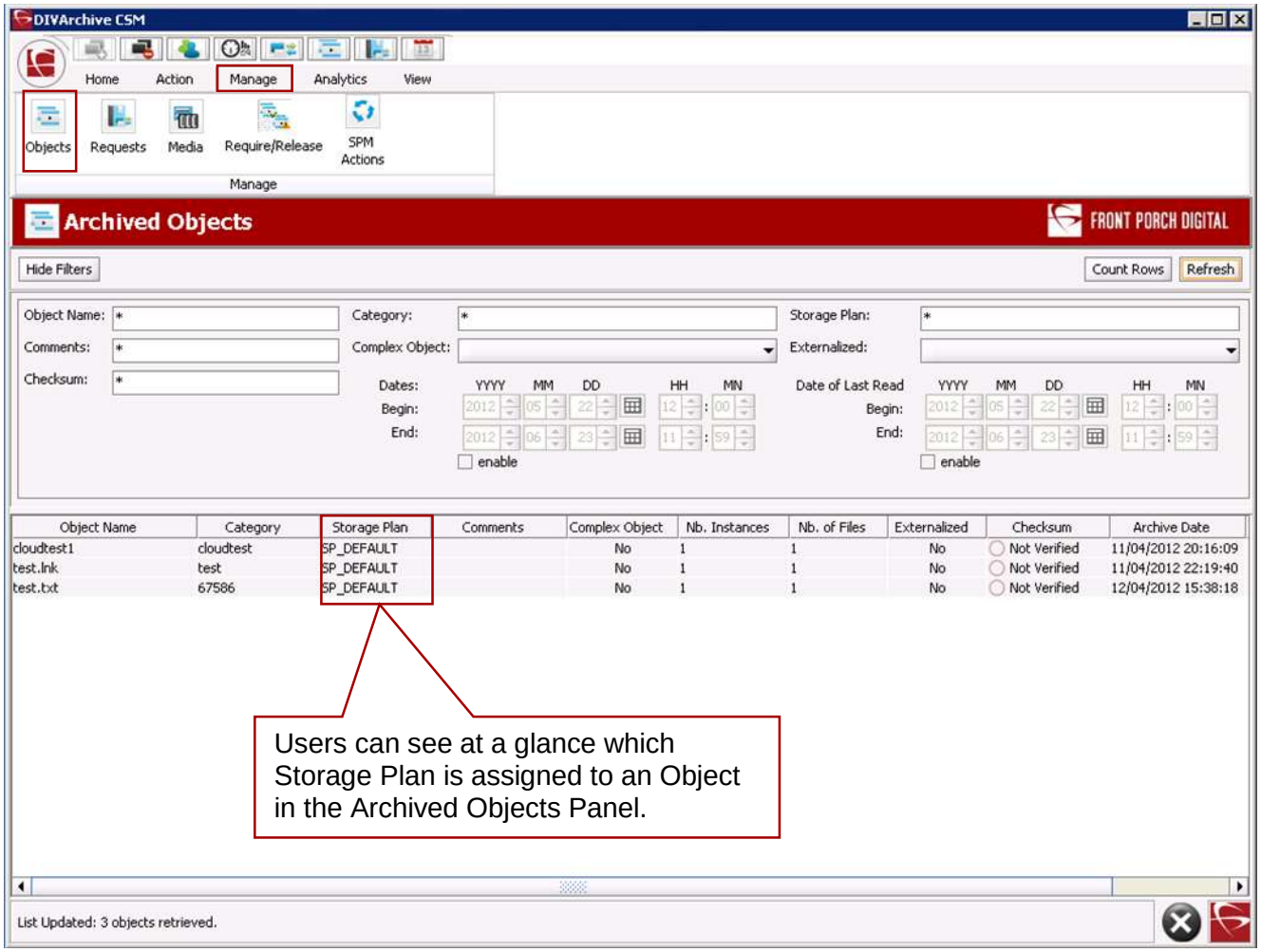
3.2 Accessing SPM Information through the DIVArchive Control GUI

SPM information for objects can be viewed and failed actions can be rescheduled using the DIVArchive Control GUI. This section describes how to use the Control GUI to access the information and what information is available through this interface.

Start the DIVArchive Control GUI and then follow through this section to discover where to view the SPM information required.

Users are able to see which Storage Plan is assigned to an object through the **Archived Objects** window as shown below:

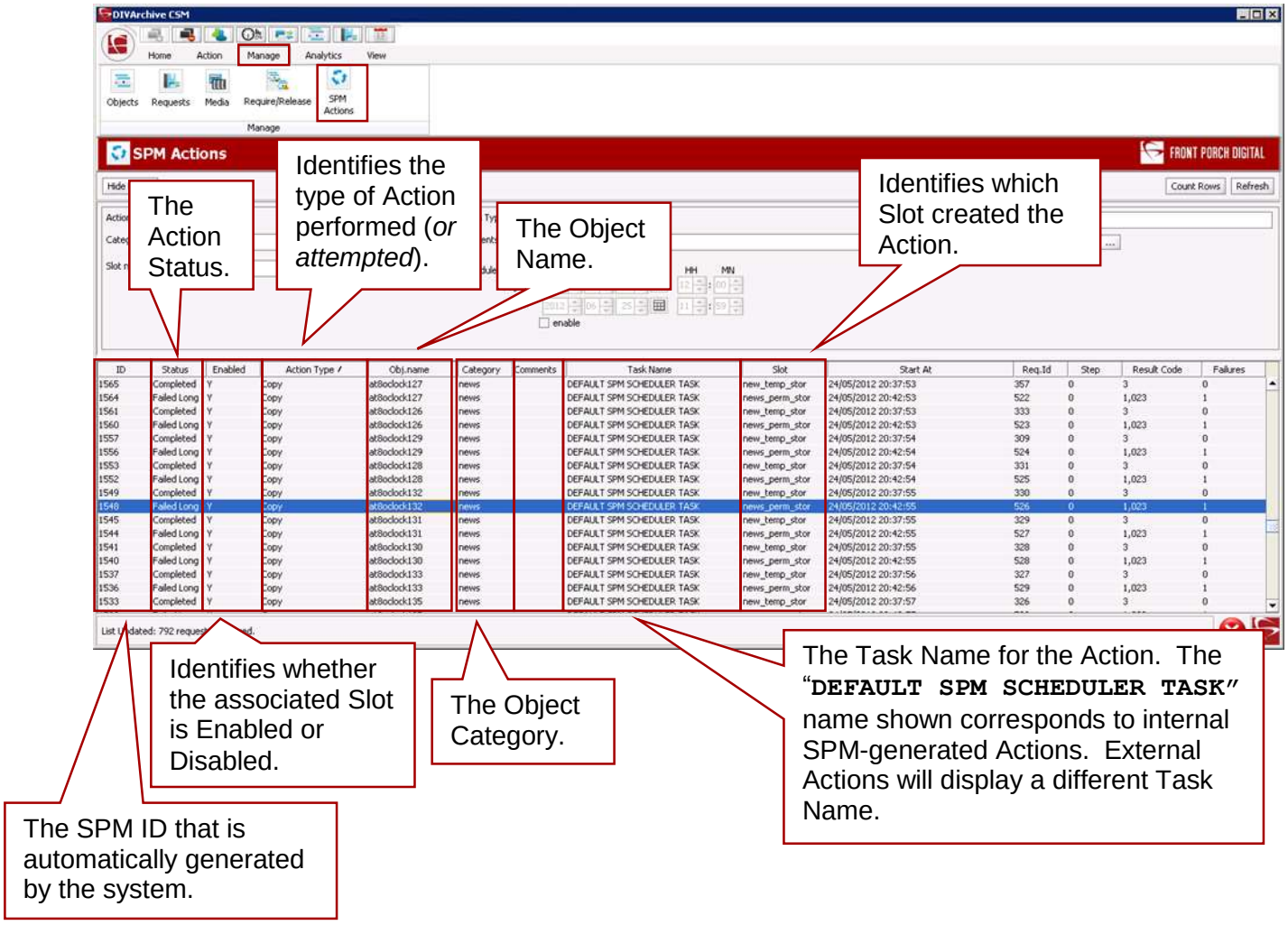
Figure 57: Control GUI Archive Objects Window Showing Object Storage Plans



If a newly archived 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.

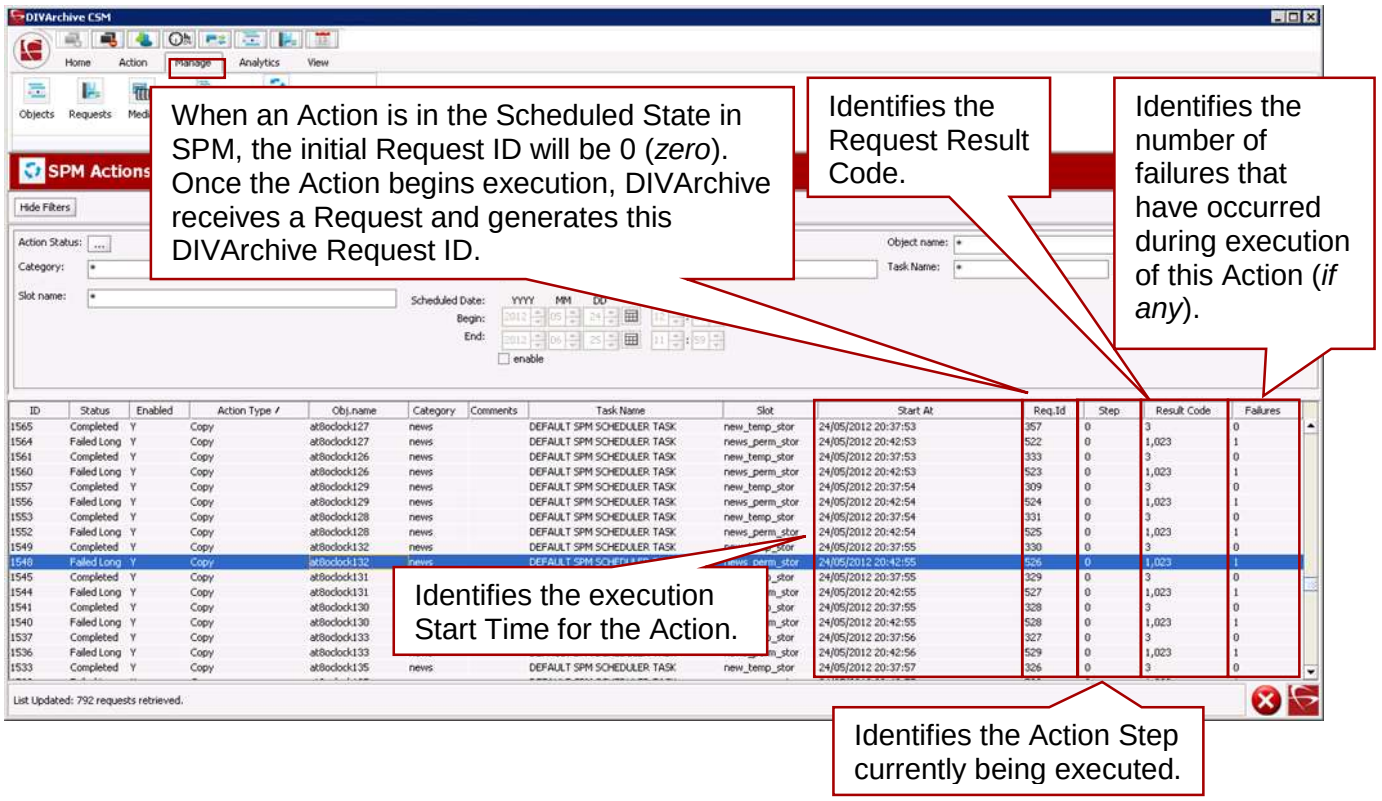
Information concerning all Actions established by SPM can be viewed immediately through the **SPM Actions Panel** in the Control GUI. The columns in the **SPM Actions Panel** are shown in two separate figures below for clarity. A description of each column is depicted in the figures.

Figure 58: DIVArchive Control GUI SPM Actions Panel – 1



ID	Status	Enabled	Action Type	Obj.name	Category	Comments	Task Name	Slot	Start At	Req.Id	Step	Result Code	Failures
1565	Completed	Y	Copy	at8oclock127	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:53	357	0	3	0
1564	Failed Long	Y	Copy	at8oclock127	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:53	522	0	1,023	1
1561	Completed	Y	Copy	at8oclock126	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:53	333	0	3	0
1560	Failed Long	Y	Copy	at8oclock126	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:53	523	0	1,023	1
1557	Completed	Y	Copy	at8oclock129	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:54	309	0	3	0
1556	Failed Long	Y	Copy	at8oclock129	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:54	524	0	1,023	1
1553	Completed	Y	Copy	at8oclock128	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:54	331	0	3	0
1552	Failed Long	Y	Copy	at8oclock128	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:54	525	0	1,023	1
1549	Completed	Y	Copy	at8oclock132	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:55	330	0	3	0
1548	Failed Long	Y	Copy	at8oclock132	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:55	526	0	1,023	1
1545	Completed	Y	Copy	at8oclock131	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:55	329	0	3	0
1544	Failed Long	Y	Copy	at8oclock131	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:55	527	0	1,023	1
1541	Completed	Y	Copy	at8oclock130	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:55	328	0	3	0
1540	Failed Long	Y	Copy	at8oclock130	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:55	528	0	1,023	1
1537	Completed	Y	Copy	at8oclock133	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:56	327	0	3	0
1536	Failed Long	Y	Copy	at8oclock133	news		DEFAULT SPM SCHEDULER TASK	news_perm_stor	24/05/2012 20:42:56	529	0	1,023	1
1533	Completed	Y	Copy	at8oclock135	news		DEFAULT SPM SCHEDULER TASK	new_temp_stor	24/05/2012 20:37:57	326	0	3	0

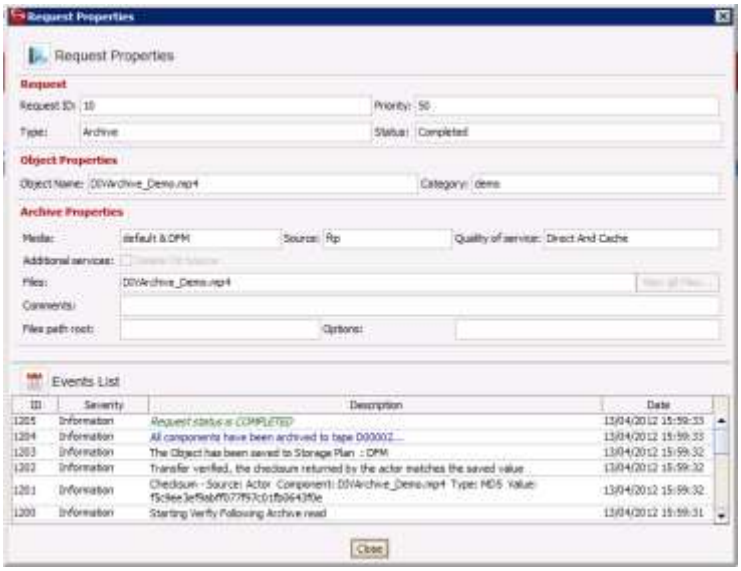
Figure 59: DIVArchive Control GUI SPM Actions Panel – 2



The **Request Properties** can be viewed by double-clicking on the Action or by right-clicking on the Action and selecting **Request Properties** from the resulting pop-up menu. If no Request Mapping exists for the object, the Request Properties window will not appear.

Note: If the Action is getting old, the DIVArchive request history may have cycled since and the request id may have been reused by a newer DIVArchive request. In this case, viewing the Request Properties will display a request other than the one expected.

Figure 60: Request Properties Pop-up Window

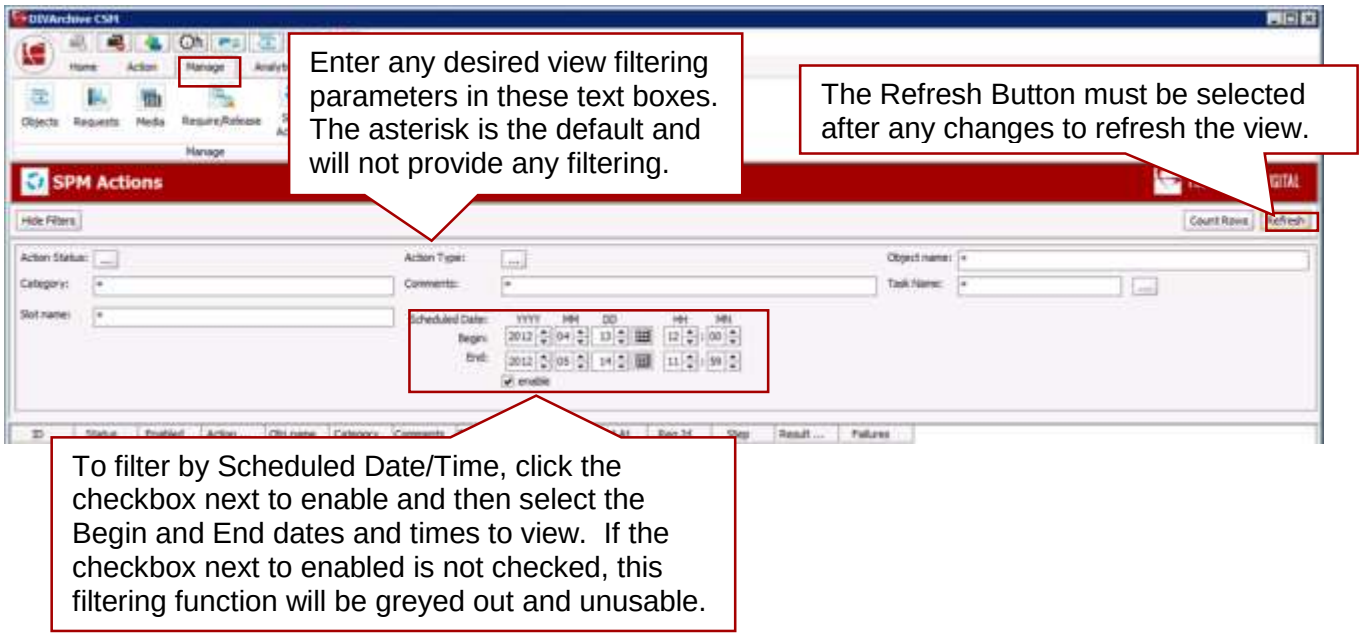


3.2.1 SPM Actions Panel View Filtering

The SPM Actions Panel allows for filtering of which Actions are to be viewed. This is accomplished using the pull-down menus and text boxes at the top of the SPM Actions Panel. The default asterisk (*) shown in the text boxes is a wildcard and signifies that no filtering restriction is being placed on the parameter where it is used (*i.e. Object name, Category, Task Name, Slot name*).

Note: After changing any of the viewing filters, the Refresh button must be selected to refresh the view.

Figure 61: SPM Actions Panel Filtering



Additional filtering is possible by **Action Status** and/or **Action Type**. Use the pull-down menus for either one and select the desired filters.

Figure 62: SPM Actions Panel Filtering by Action Status

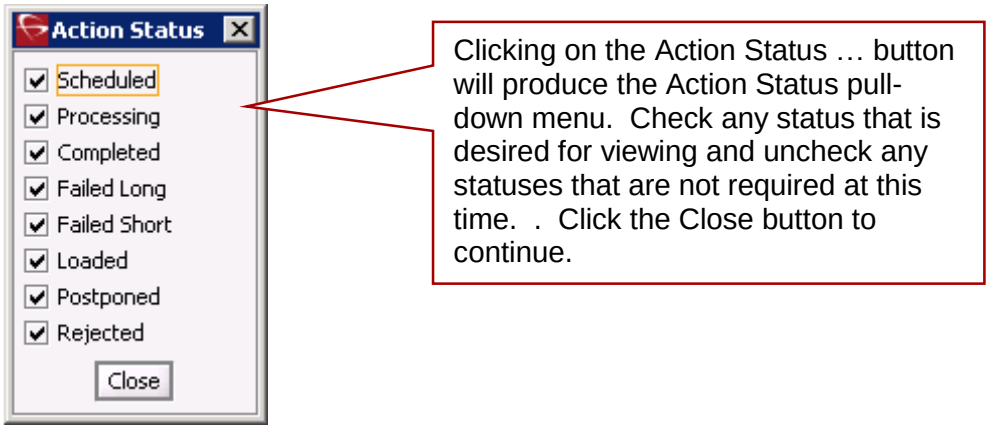
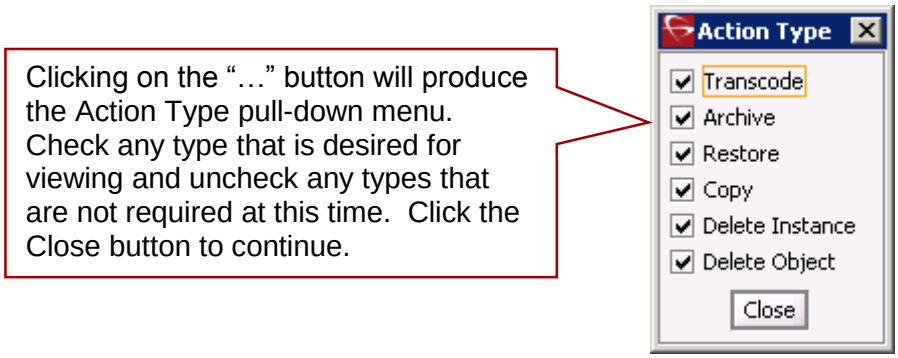


Figure 63: SPM Actions Panel Filtering by Action Type



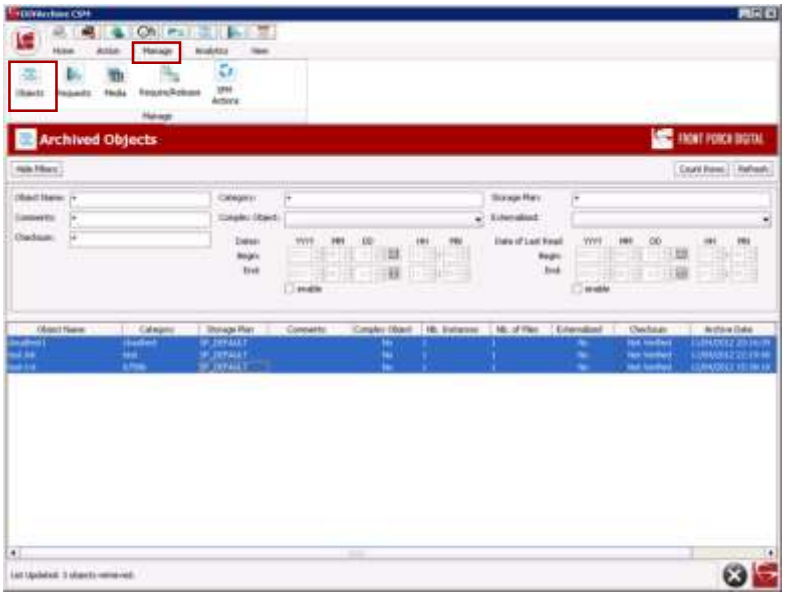
3.2.2 Assigning a New Storage Plan

Users have the capability to change Storage Plans only for an object that is assigned the default Storage Plan (***DEFAULT_SP***). When a new Storage Plan is assigned to an object, SPM sees that object as being a *new object* in the system because it has a new Storage Plan assigned.

Multiple objects may be selected in several different ways:

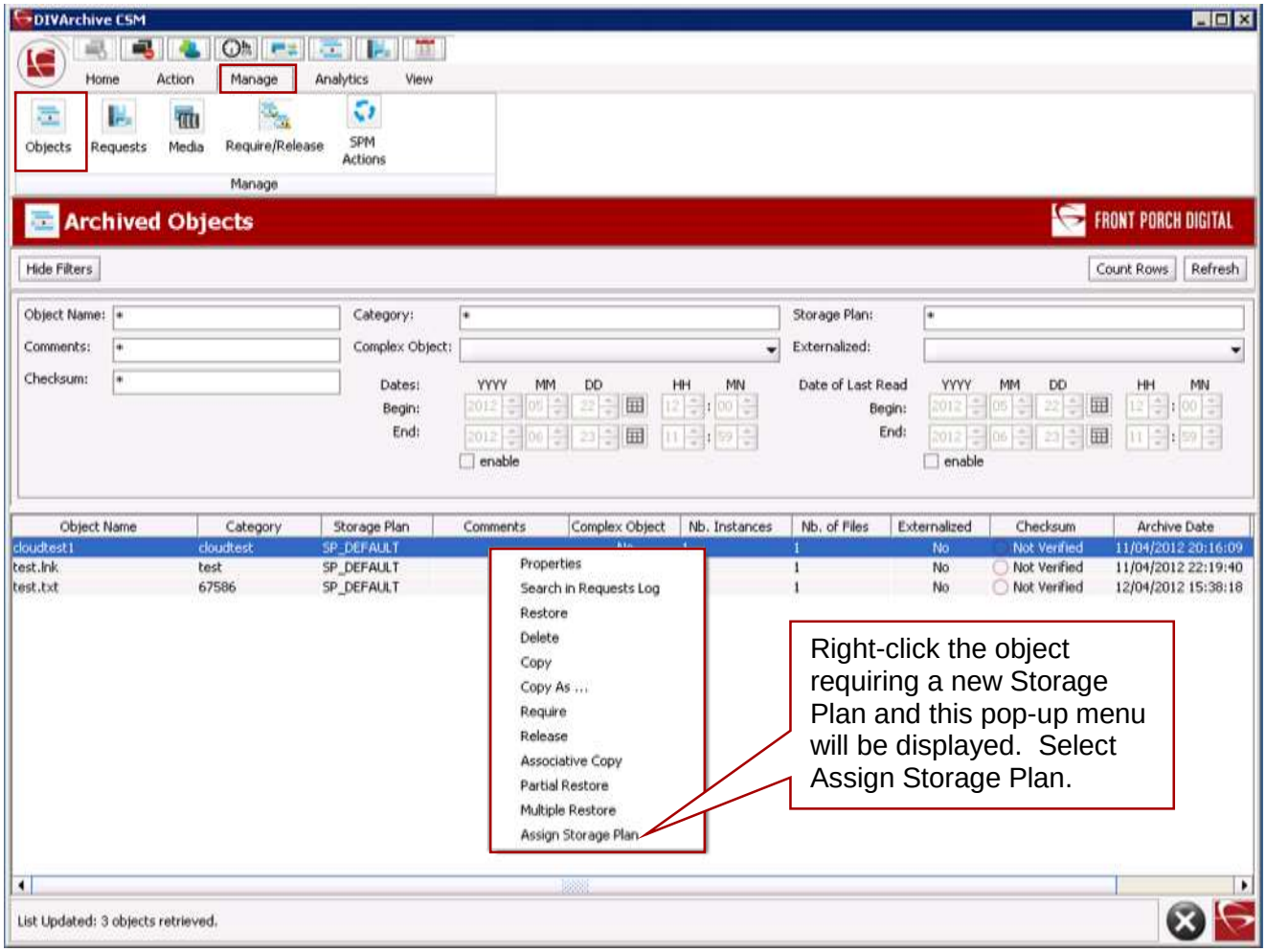
- Hold the **SHIFT** key, select the first object and then select the last object and release the **SHIFT** key.
  - Useful when more than one object needs to be selected in a row.
- Click and hold the mouse button on the first object then drag the mouse pointer to the last object and release.
  - Useful when more than one object needs to be selected in a row.
- Hold the **CONTROL** (*CTRL*) key and select each object individually, and then release the **CONTROL** key.
  - Useful when objects are not next to each other, but more than one object needs to be selected.

Figure 64: Selecting Multiple Objects for Assignment



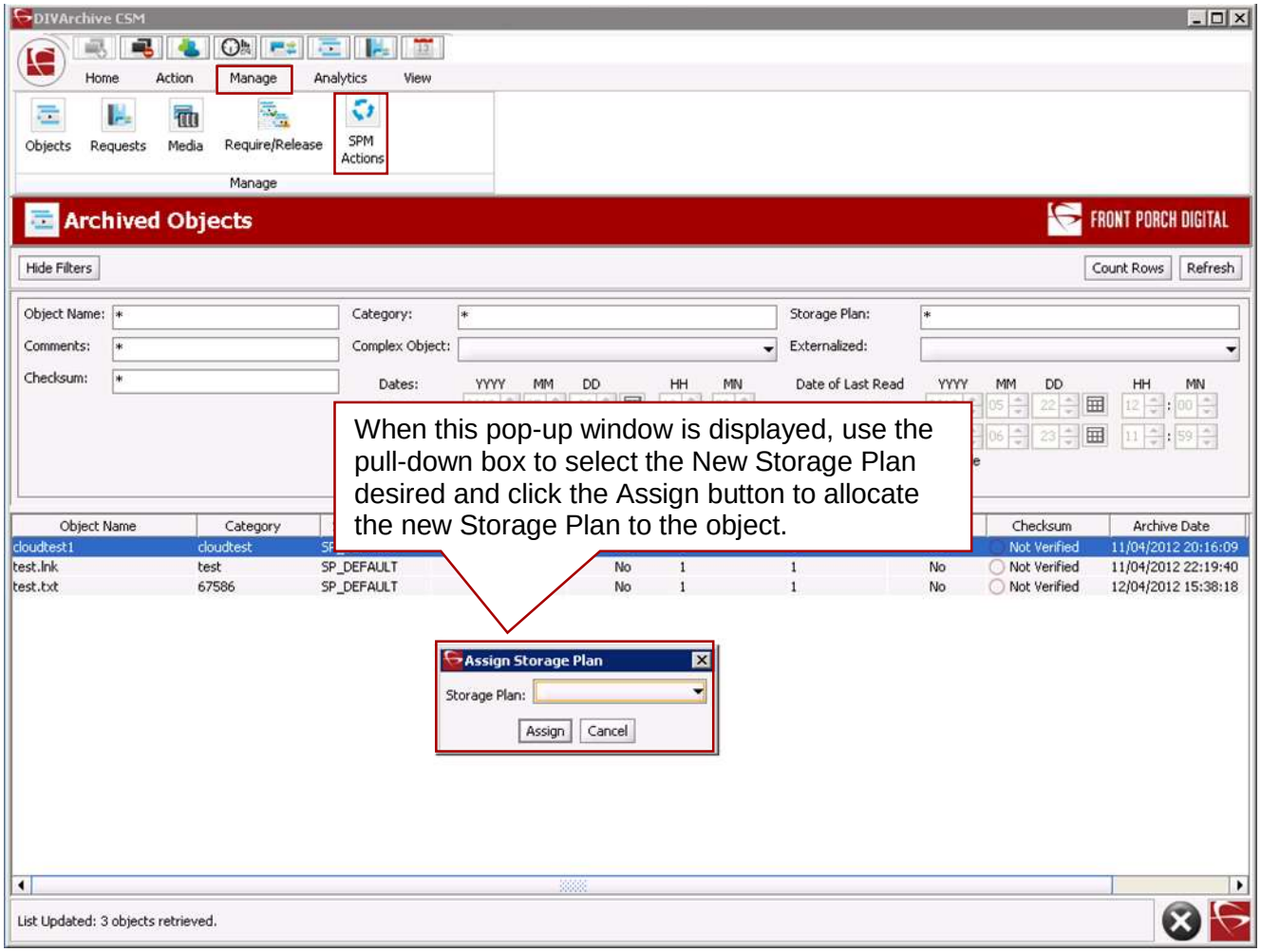
1. To assign a new Storage Plan to an object(s), go to the **Archived Objects Panel** in the Control GUI and right-click the object(s) requiring the new Storage Plan.
2. When the pop-up menu appears, select the **Assign Storage Plan…** option.

Figure 65: Assigning a New Storage Plan for an Object



3. A new pop-up window will appear allowing selection of the new Storage Plan. Using the pull-down box, select the new Storage Plan desired and click the **Assign** button. The selected Storage Plan will be assigned to the object.

Figure 66: Selecting the Object's New Storage Plan

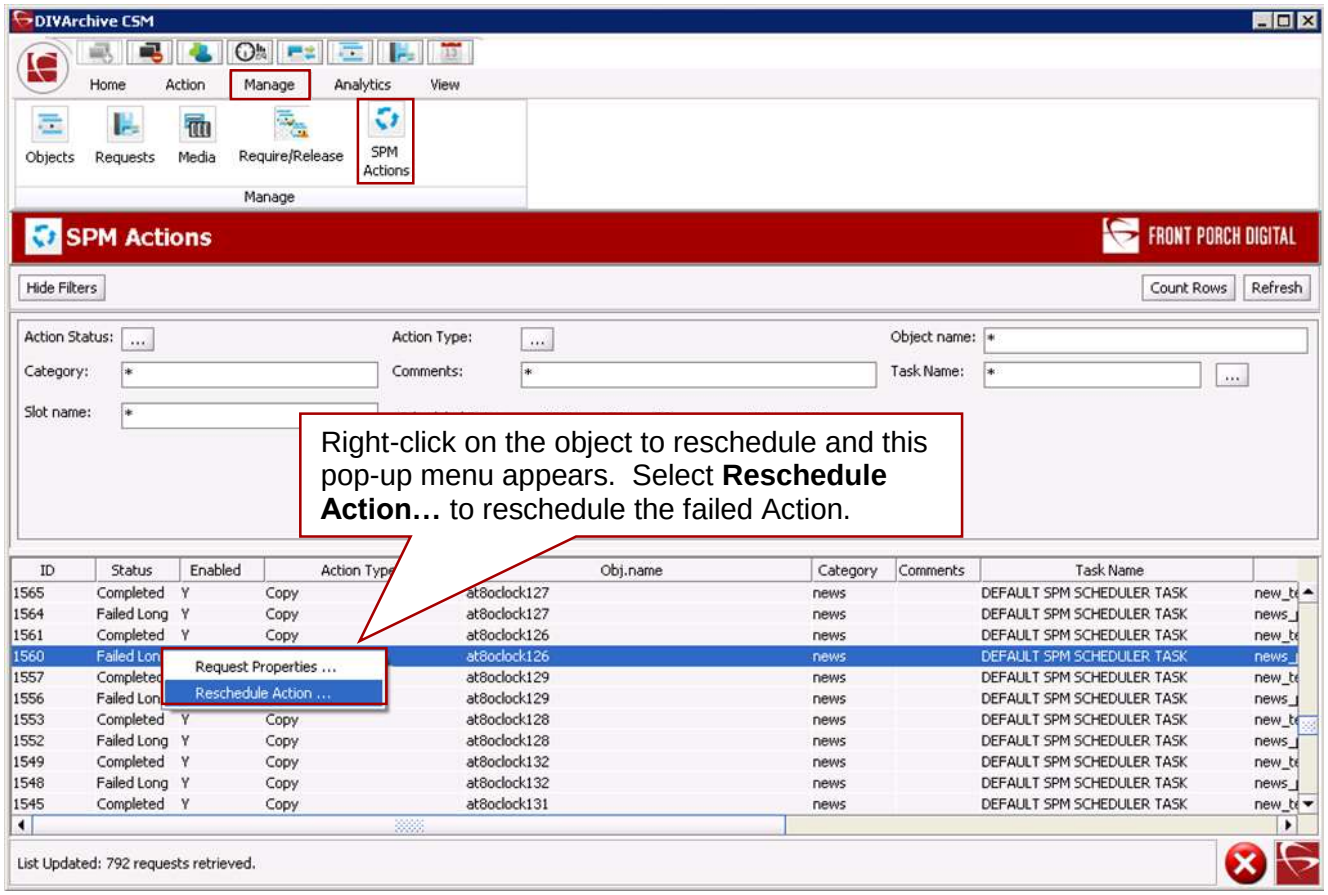


3.2.3 Rescheduling Failed Actions

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. The Action can also be manually rescheduled to run immediately using the procedure described below.

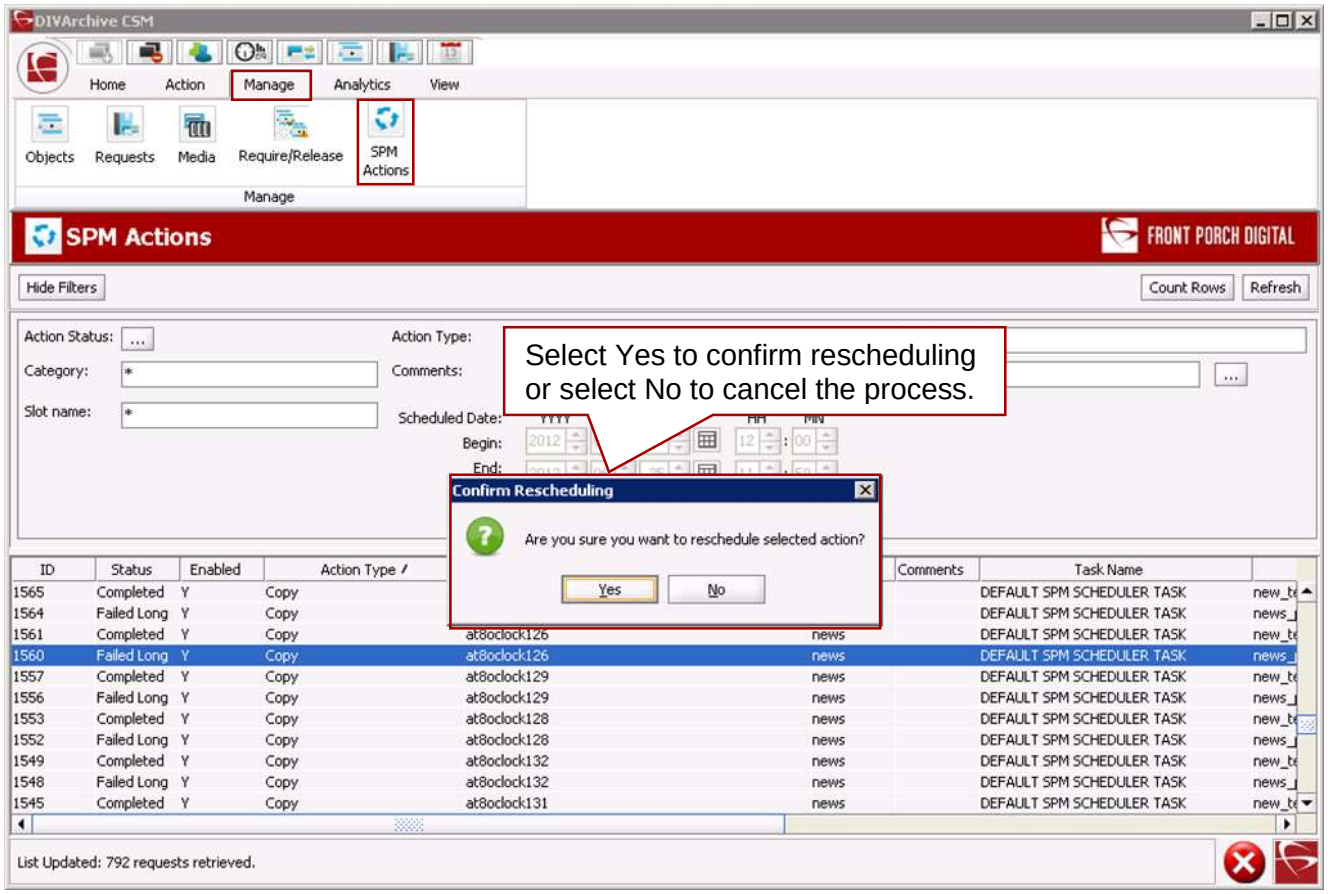
1. On the SPM Actions Panel right-click on the Action which requires rescheduling (*multiple Actions may be selected – refer to the previous section for multi-selection tips*). A pop-up menu will appear.

Figure 67: Reschedule Pop-up Menu



2. Once the **Reschedule Action…** is selected, a confirmation window will display. Select **Yes** to confirm rescheduling the Action to execute immediately, or select **No** to cancel the process.

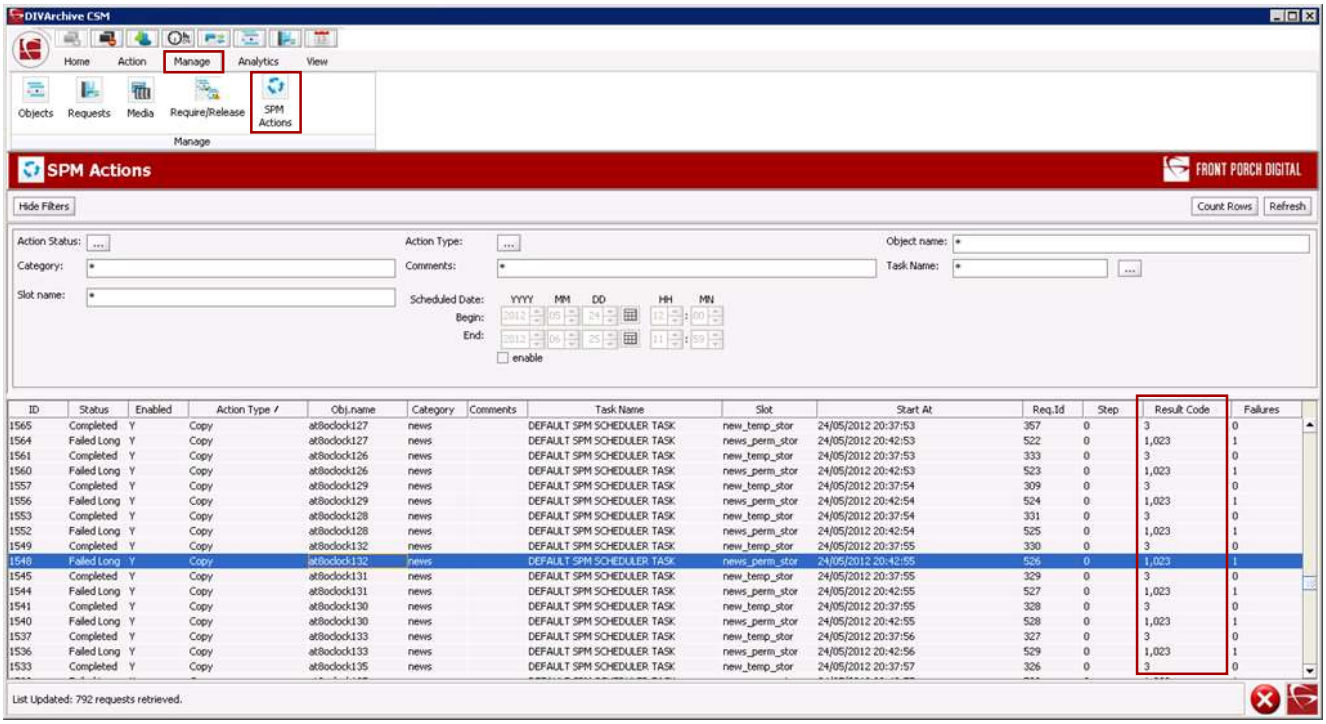
Figure 68: Action Rescheduling Confirmation Window



3.3 SPM Action Result Codes

Result codes are displayed for every processing and completed Action in the SPM Actions Panel of the Control GUI.

Figure 69: SPM Actions Tab Showing Action Result Codes



The **ACTION RESULT CODE** is a short string of text reflecting the latest available result of **ACTION** execution. The value is always a string similar to:

`XXX-NNNNN`

`XXX` = Prefix

`NNNNN`= Result Code

Table 34: SPM Action Result Codes

Prefix	Sample `RESULT_CODE`	Description	Details
`000`	`000-00000`	This will be always equal to `000-00000`.	Default (*Not always displayed on the GUI*).

Prefix	Sample RESULT_CODE	Description	Details
REQ	REQ-00003	DIVA_REQUEST_STATE code from the DIVArchive API.	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 DIVArchive 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 DIVArchive GET_REQUEST_INFO API call. Refer to the DIVArchive API documentation in the DIVArchive API Documentation library for more details.
API	API-01008	DIVA_STATUS of the DIVArchive API.	SPM uses the DIVArchive API to execute Actions and submit Request 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. Refer to the DIVArchive API documentation in the DIVArchive API Documentation library for more details.
SPM	SPM-00003	SPM business logic result code.	Refer to the below table.

Table 35: SPM Result Code Descriptions

SPM-00000	Name	Logged	Description
SPM-00001	SPM_LR_CONFIG_PREVENTS_DELETE	WARNING	ALLOW_OBJECT_DELETION parameter in the spm.conf file was set to false and prevented SPM from deleting the last instance of an object.
SPM-00002	SPM_LR_CANNOT_DELETE_LAST_INSTANCE	WARNING	SPM refused to delete the last instance on an object.
SPM-00003	SPM_LR_NO_MORE_INSTANCES_ON_SLOT	WARNING	SPM refused to create more instances on a slot than it was permitted.

SPM-00000	Name	Logged	Description
SPM-00007	SPM_LR_POSTPONED_AS_WATERMARKED	WARNING	SPM postponed a **DELETE INSTANCE** action because it was for a watermarked media.
SPM-00008	SPM_LR_NO_NEED_TO_LINK_OBJECTS	WARNING	The **Verify** stage of the **LINK OBJECTS** step. If the **METADATA ARCHIVE** slot was configured without Cascading Objects for neither delete nor restore (*both set to **N**o*), there will be no link between the objects, and this error will be produced.
SPM-00009	SPM_LR_CANNOT_SEND_MORE_REQUESTS	WARNING	Manager Monitor does not allow sending more requests to Manager.
SPM-00010	SPM_LR_ACTION_SLOT_EXPIRED	WARNING	An action will have this code for the following reasons 1) Object was deleted from DIVArchive but is still under SPM. 4) The Slot expired after loading the SPM Action into memory.

3.4 SPM Logging

The SPM Logging Configuration File assists with controlling the SPM log file size and switch; purging old SPM logs and identifying which specific module inside of SPM to enable or disable logging for. The description of global parameters is in the sample **`spm.trace.ini`** below.

```
#####################################################################
# 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.
```

Enabling and disabling SPM module logging is achieved as follows:

1. If the module name is prefixed with "**`!`**" it is enabled.
2. If it is prefixed with "**`#`**" it is disabled.

Example:

```
!SPMService
```

- Trace is **enabled** for the component **`SPMService`**.

```
#DbConnectionPool
```

- Trace is **disabled** for the component **`DbConnectionPool`**.

4 Frequently Asked Questions

4.1 What should be done if I see that no SPM Actions are being generated?

Answer:

- Check if SPM Service is running.
- 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.
- Check that slots are enabled.

4.2 What should be done if only STORAGE actions are being generated?

Answer:

- Check that the non-Storage type slots are enabled and are included in the correct Storage Plan.
- Check that in the SPM Slot Configuration on the Restore, Transcode & Metadata Archive Tab that the Object Name, Object Source, and Object Category Filters are set such that they do not filter out the SPM Actions you are expecting.
- Check the SPM Configuration File and confirm that **DIVA_RESTORE_MAX_REQUESTS**, **DIVA_TRANSCODE_ARCHIVE_MAX_REQUESTS**, or **DIVA_METADATA_ARCHIVE_MAX_REQUESTS** are not set to 0.

4.3 What should be done if SPM Actions are generated but are never executed (they stay in the Scheduled State)?

Answer:

- Check the Archive **Date** of the objects for which the actions were generated and compare it to the values configured in the **Slot Begin** and **End Times**, and **Request Execution Begin** and **End Times** in the Configuration Utility to verify if one of the following cases is true:
  - If the age of the 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 object has met or exceeded the number of minutes specified in the Slot Begin Time.
  - If the age of the 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 object has existed (e.g. 9:30 to the current time of 13:20) has not yet fallen within the Request Execution Begin and End Time range (e.g. 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 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 object is 20 minutes, the SPM Action for this object will not execute even if the clock time is within the **Slot Request Execution Begin** and **End Time** range because the object has *expired*.

This can occur by design where object's lifetime is expected to expire before the **Slot Request Execution Begin Time** is even reached. It can also occur if the 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 Manager to the point where the object eventually *expires* before its associated SPM Action is allowed to be submitted to the Manager.

4.4 What should be done if SPM Actions always Fail Long?

Answer:

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

4.5 Why are SPM Actions removed from the SPM Actions List?

Answer:

SPM Actions are removed from the SPM Actions List when the objects they are associated with have been deleted from DIVArchive.

4.6 Why does the Start At Time not update when an SPM Action is rescheduled?

Answer:

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

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

Answer:

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.

4.8 What are the different components of SPM?

Answer:

- **DSM Controller:** DSM stands for Disk Space Monitor. This module monitors the watermarked array and does the clean-up when the High Watermark is reached.
- **SPM Controller:** This module is responsible for Creating, Updating, and execution of actions based on the configuration.

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

Answer:

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

4.10 SPM appears to communicate only with the Oracle Database and the Manager (by way of DIVArchive API calls). Does SPM interact with any other processes?

Answer:

SPM only interacts with the Oracle Database and the Manager via the DIVArchive API. SPM may interact with the Access Gateway instead of the Manager if configured to do so. Otherwise, SPM does not interact with any other processes.

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

Answer:

Only the DSM Controller part of SPM is event driven as it waits for the watermarked array to reach the high water mark to trigger the clean-up of the array.

4.12 Is SPM stateful across restarts?

Answer:

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.

4.13 An archived Object can only belong to a single Storage Plan. When an Object is first archived, DIVArchive Manager 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?

Answer:

- Yes, an object can only belong to one Storage Plan.
- If a Storage Plan is specified in the request that object is always assigned to that Storage Plan.
- If the Storage Plan is specified and no destination media is specified in the request, the 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 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 aborted.

4.14 What are the meanings of the DATARCHIVED, DATELASTUPDATE, and DATENEXTUPDATE properties for objects under SPM?

Answer:

- DATEARCHIVED is the date when object was archived.
- DATELASTUPDATE is the date when actions were created or updated for this object the last time.
- DATENEXTUPDATE is the date when SPM will check for any changes in the slot that will affect any actions of this object and update them if any changes are identified.

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

Answer:

Yes, **DATELASTUPDATE** and **DATENEXTUPDATE** will change once actions are generated for that object.

APPENDIX

A1 Default Configuration (spm.conf.ini)

```
######################################################################
#     SPM4 Configuration File
#
#     Front Porch Digital, Inc. (C) 2005, 2006, 2007
#     All rights reserved.
#
#     $Source: spm.conf.ini $
#     $Date: 2010/09/09 14:59:37EDT $
#     $Revision: 1.28 $
#     $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 DIVArchive Manager
######################################################################


# 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_PASSWORD   = lib5

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 (processing+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 object
requests.This number is checked before sending new Delete Requests.
This value is internaly calculated (it's not received by Request to
Manager). Therefore it's approximation and sometimes we can
temporarily have Number of Delete requests greater 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 = 150


# The following parameters are used to restrict how many RESTORE,
TRANSCODE and METADATA requests SPM will submit to manager
simultaneously.

# 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
(processing + 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
#####################################################################

# 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 objects.
# This parameter must be enabled for SPM to delete objects.

#

# Additionally, delete object must be enabled or disabled for each
# storage plan separately during its configuration in configuration
# utility for additional security & flexibility.

# Delete 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 object is executed only in the following condition satisfy

# 1) This parameter ALLOW_OBJECT_DELETION must be enabled

# 2) Delete 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
#    object must have a slot end time.

# 4) Delete object will be executed by the storage slot that has the
#    highest end time in the storage plan.(Please note an 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
#    object.

# 5) Just enabling ALLOW_OBJECT_DELETION will not acknowledge Storage
#    slot medium water mark and Delete 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 Object feature.
# This parameter forces SPM to delete object at the end of a storage
# slot only it is the last instance in the entire DIVArchive 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 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 be 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 object feature will not be enabled.
# 2) This parameter DELETE_OBJECT_ONLY_LAST_INSTANCE must be enabled.
# 3) Delete 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
#    object must have a slot end time.
# 5) Delete object will be executed by the storage slot that has the
#    highest end time in the storage plan.(Please note an 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 slots with max end time will do the delete
#    object.
# 6) At the end of the storage slot if more than 2 instances exist in
#    the DIVArchive system or on the medium, only delete instance is
#    executed.
# 7) Delete object is executed only if one instance exist on the
#    entire DIVArchive 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 object is postponed until watermark
#    condition is satisfied.

DELETE_OBJECT_ONLY_LAST_INSTANCE = false
```

```
# Defines wether SPM will use original or transformed object name as
"target" object name in METADATA_ARCHIVE step (Default is "true):
#   - true:  use transformed object name
#   - false: use original 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 during each call
# of the 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
```

A2 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
```

A3 Configuration Parameter Summary

A3.1 Storage Plan Definitions

Parameter	Definition
Storage Plan Name	Desired Storage Plan name.
Allow Last Instance Deletion	• **Y** (Delete Object) • **N** (Just Delete Instance) • Can be overridden from the configuration file. • Disc cleaning strategy is not applicable with **Y.**

A3.2 Mediums Definitions

Parameter	Definition
Medium Name	Desired Mediums name (it is better if the name is the same as DIVArchive Group/Array Name).
Storage Name	Name of the Storage Manager. This value isn't used and it is recommended to keep the default value (***DIVA***).
Group/Array Name	DIVArchive Group/Array Name.

Parameter	Definition

Watermarked	Monitors the Medium's usage (*available space, used space, etc.*). Select **Y**, **N**, or **M** to specify whether or not to apply watermarking to this medium. Refer to Sections 1.7.1 and 2.6 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 DIVArchive System.
	
	In the case of a Storage Slot, setting the **Slot End Time** to 0 will force SPM to delete any content created by this slot and once all of the content is removed, the Storage Slot can be deleted or disabled. If retaining the content in a 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.
	
	Refer to Section 2.6 for detailed information on Watermarking.
	• **Y** applies watermarking (*Yes*).
	• **N** does not apply watermarking (*No*).
	• **M** is a combination of watermarking (*Mixed*). The disk cleaning action taken is dependent upon what event happens first:
	  ◦ The Slot reaches its end time.
	  ◦ The High Watermark is reached.
	• This is only valid for **Storage Slots**.
	• Only for Disk Arrays configured as Medium.

Disc Cleaning Strategy	• By last access time • By Size

A3.3 Object Slots Definitions

Parameter	Definition
Slot Name	Desired Slot name.
Storage Plan	Storage Plan name
Id	Automatically generated by SPM.
Medium Associated With Slot	Medium associated with the Slot.

Slot Request Type (*or Slot Type*)	The type of the request:
	• Storage
	  ○ Copy request initiated at the beginning of the Slot.
	  ○ Delete request initiated at the end of the Slot.
	• Transcode Archived
	  ○ Transcode Archived request initiated at the beginning of Slot.
	• Metadata Archive
	  ○ Metadata Archive request initiated at the beginning of Slot.
	• Restore
	  ○ Restore request initiated at the beginning of the Slot.

Slot Begin Time	This parameter identifies when the Action will occur. If set to zero (*0*) the Action will be initiated as soon as the object is archived. If set to another number, for example 10 (*minutes*), the Action will execute 10 minutes after the original object is archived.
Slot End Time	Time in minutes from the object creation- points to the Slot end.
Request Execution Begin Time	Time in HH:MM format (*00:00 to 24:00*) – points to the time of the day when execution can start.

Parameter	Definition
Request Execution End Time	Time in HH:MM format (*00:00 to 24:00*) – points to the time of the day after which execution cannot start. • If the Begin and End Time are 00:00 and 24:00 correspondingly, 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	Priority of operations for SPM and the DIVArchive Manager.

A3.4 Filter Definitions

Parameter	Definition
Filter Name	Desired Filter name.
Id	Automatically generated by SPM.
Medium Filter	Name of the media to monitor.
Object Category Filter	• blank = all ○ Any combination; example - **`abc*def`**
Object Source Filter	• blank = all ○ Any combination; example - **`abc*def`**
Object Name Filter	• blank = all ○ Any combination; example - **`abc*def`**
Min Object Size	Minimum Object Size: • 0 - no restrictions from Maximum Object Size.
Max Object Size	Maximum Object Size: • 0 = no restrictions from Minimum Object Size.
Size Units	Size units (*KB, MB, GB*).
Storage Plan	Name of the Storage Plan.

A3.5 SPM Actions Definitions (only used with Transcode Archived, Metadata Archive, and Restore Slots)

A3.5.1 Object Filters

Parameter	Definition
Filter name	Desired Filter name.
Storage Plan	Storage Plan name.
ID	Automatically generated by SPM.
Request Type	Used only with Transcode Archived, Metadata Archive, and Restore.
Object Name Filter	• blank = not usable ○ Any combination; example - **`abc*def`**
Object category Filter	• blank = not usable ○ Any combination; example - **`abc*def`**
Object Source Filter	• blank = not usable ○ Any combination; example - **`abc*def`**

A3.5.2 Transformation Rules (only used for Transcode Archived and Metadata Archive)

Request Type	Parameter	Definition

Request Type	Parameter	Definition
Transcode Archived	**Target Name**	Name of the transcoded object. • ^ = use the original name. ○ Can be added to as in: `^.txt`
	Target Category	Category of the transcoded object. • ^ = use the original category. ○ Can be added to as in: `^.txt`
Metadata Archive	**Target Name**	Name of the archived object. • ^ = use the original name. ○ Can be added to as in: `^.txt`

Request Type	Parameter	Definition
Metadata Archive *(continued)*	**Target Category**	Category of archived object • ^ = use the original category. ○ Can be added to as in: `^.txt`
Restore	**Target Name**	Always leave at the default: ^
	Target Category	Always leave at the default: ^

A3.5.3 Request Templates

Parameter	Definition
Request Type	Used only with Transcode Archived, Metadata Archive, and Restore requests.
Transformation Rule	Use an existing transformation rule.
Id	Automatically generated by SPM.
Object Source/Destination	For Metadata Archive this is the Source/Destination to archive. For Restore this is the Source/Destination to restore.
Pathroot	Pathroot used for Metadata Archive or Restore.
Request Options	Any additional options necessary for Transcode Archived, Metadata Archive, or Restore.
Request Comments	Comments for Transcode Archived, Metadata Archive, or Restore
QOS	DIVArchive Quality Of Service.
Cascade Delete Links	Used for Transcode Archived and Metadata Archive only.
Cascade Restore Links	Used for Transcode Archived and Metadata Archive only.